Numerical simulation of probabilistic computing to NP-complete number theory problems

Jie Zhu, Zhengxiang Xie, and Peter Bermel*

Purdue University, Elmore Family School of Electrical and Computer Engineering,
West Lafayette, Indiana, United States

Abstract. Probabilistic computing with $p$-bits is a powerful, unique paradigm alternative to classical computing and holds experimental advantages over certain forms of quantum computing. Stochastic nanodevices have been experimentally demonstrated to act as artificial neurons in solving certain problems through probabilistic computing. Still, many open questions about the breadth and size of soluble problems remain. We demonstrate the capability of probabilistic computing made of a stochastic nanodevice network in solving likely NP (non-deterministic polynomial time)-complete number theory problems associated with combinatorial optimization, which can be implemented using a network of optical parametric oscillators. These simulation results show robustness across all problems tested, with great potential to scale to solve substantially larger problems. © 2023 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: 10.1117/1.JPE.13.028501]

Keywords: spin electronics; stochastic nanomagnet; magnetic tunnel junction; probabilistic computing; optical parametric oscillator.

1 Introduction

Classical computing based on semiconductor devices meets obstacles when it comes to computationally complex problems that scale exponentially with the problem size. Quantum computing utilizing intrinsic parallelism of the quantum bit is able to speed up the solution finding process of some complex problems, relying on specific quantum algorithms. However, it often suffers decoherence and requires error correction. Also the low-temperature requirement for certain implementations, such as superconducting Josephson junctions, can make fabricating and measuring quantum computers very costly.

Probabilistic computing is an underexplored alternative to quantum computing, which brings much of the same power in a conceptually distinct manner that can hold advantages in experimental realization. Using a set of unstable nanodevices that flip rapidly with thermal fluctuation, the solution space can be sampled quickly and efficiently. In other words, thermal fluctuations can accelerate problem solving, rather than causing a loss in fidelity.

The Ising model is a canonical model for representing a wide range of problems with both point-to-point interaction and global interaction. In the original instance, it was used to represent magnetic dipole moments of spins that can be in one of two states $\pm 1$. The Ising Hamiltonian for a collection of spins is

$$H = -\sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i,$$

where each spin $s_i$ can take on one of two values, $\pm 1$. Based on Boltzmann law, different spin configurations $\{s_i\}$ have a probability proportional to $\exp(-H(\{s_i\})/kT)$, where $k$ is the Boltzmann constant and $T$ is the temperature. However, the Ising Hamiltonian can also be used to represent the energy of interaction for optical or other sorts of systems with interaction sites in a 1D, 2D, or 3D lattice.

*Address all correspondence to Peter Bermel, pbermel@purdue.edu
When the Ising model is discretized and updated step by step, it becomes the Hopfield model. Due to the mathematical isomorphism of the Ising model, the Hopfield model has been analyzed in great detail\(^4\) and has been applied to solve combinatorial optimization problems.\(^5\)–\(^8\) It can be proved that a Hopfield network with \(n\) units and asynchronous dynamics, which starts from any given network state, eventually reaches a stable state at a local minimum of the energy function.\(^9\)

A variety of physical systems have been developed to solve the Ising problem, including trapped atoms,\(^10\) superconducting circuits,\(^11\) coupled oscillators,\(^12\)–\(^14\) and nanomagnets.\(^15\) The optical approach has the key advantages of light speed and potential scalability. The injection-locked laser network can use either the two polarization modes\(^16\)–\(^18\) or the discrete phases\(^19\) of slave lasers to represent Ising spins. In the time-multiplexed optical parametric oscillator (OPO) approach,\(^20\)–\(^23\) by gradually tuning the parametric gain, one can effectively raise the optical system from an extremely “negative temperature” with no modes supported to a small but negative temperature with single-mode oscillation of the ground state spin configuration. It was later shown that the simplified dynamics described by the Gaussian state formalism is sufficient for attaining high success probabilities.\(^26\) In the spatial light modulation approach, Ising spin is encoded in the phase modulation of the field, and no electronic feedback is required.\(^27\) Four-body interaction can be realized with nonlinear optics,\(^28\) and the antiferromagnetic model can be implemented through the correlation between a distribution function and the measured optical intensity.\(^29\) The integrated photonics approach has also been demonstrated to efficiently solve the Ising problem.\(^30\),\(^31\)

The other possible computing model is the Boltzmann machine, which is a stochastic Hopfield network that overcomes the shortcoming of the Hopfield network—global optimality is not guaranteed.\(^9\),\(^15\) It can realize problems with large connectivity, both ferromagnetic and antiferromagnetic interaction, and arbitrary interaction strengths. One physical implementation of a Boltzmann machine is to represent each Ising spin \(s_i\) as a stochastic neuron with a certain magnetization.\(^32\) Stochastic nanomagnets with low-energy barriers \(\Delta\) have a small retention time \(\tau = \tau_c \exp(\Delta/kT),\)\(^33\) where \(\Delta\) is the energy difference between the up and down states. Even in the absence of input currents, thermal noise could rapidly flip such nanomagnets within the order of nanoseconds. The equilibrium state is then governed by Boltzmann statistics with thermal noise as an inherent characteristic of the system. On the other hand, spin currents could pin the stochastic nanomagnet to a given direction. Such stochastic neurons can also be arranged in multiple layers.\(^34\) Moreover, magnetic tunnel junction (MTJ)-based probabilistic bits, or \(p\)-bits, require less strict conditions and thus may be less costly than quantum bits made from superconducting, ion-trapped, or photonic units.\(^35\),\(^36\) Previous works have demonstrated invertible Boolean logic,\(^15\) optimization problems,\(^32\) and general neuromorphic computing.\(^37\),\(^38\) One research work demonstrated \(p\)-bits experimentally by compensating for modest levels of nonuniformity in MTJ units.\(^39\) With recent advances in CMOS fabrication technology,\(^40\) we envision that the challenge will be fully met with already-planned technology.

In this work, we demonstrate the capability of probabilistic computing to solve a variety of combinatorial optimization problems. The reminder of this paper is organized as follows. In Sec. 2, we show one possible physical implementation of a stochastic spin network using low barrier nanomagnets, give the simulation detail, and formulate some combinatorial optimization problems. The simulation results are then shown in Sec. 3. We discuss the results and possible scale-up solutions and photonic alternatives in Sec. 4. Finally, we summarize the work in Sec. 5.

2 Methodology

2.1 Physical Implementation of a Stochastic Spin Network

We briefly review the existing systems that implement the Ising model using artificial devices, including optics-based and stochastic neuron-based systems.

Using the optical oscillator as an example,\(^21\) an Ising machine represented by Eq. (1) is implemented using a network of degenerate OPOs because of their binary phase operation above threshold.\(^41\) A squeezed vacuum state is first initialized by parametric down conversion below the oscillation threshold. The couplings between the spins are realized by mutual injections of
the signal fields between the $i$'th and $j$'th OPOs. The $c$-number Langevin equation of the resulting system is

\[
\frac{d}{dt} c_j = \left[-1 + p - (c_j^2 + s_j^2)\right] c_j + \sum_{l=1, l\neq j}^{N} \xi_{jl} c_l + f_j',
\]

(2)

\[
\frac{d}{dt} s_j = \left[-1 + p - (c_j^2 + s_j^2)\right] s_j + \sum_{l=1, l\neq j}^{N} \xi_{jl} s_l + f_j',
\]

(3)

where $c_j$ and $s_j$ are the in-phase and quadrature components of the $j$'th OPO, respectively; $\xi_{jl}$ is the coupling coefficient between the $j$'th and $j'$th OPO; and $f_j'$ and $f_j'$ are the associated quantum noise. The operators are converted into complex numbers, and the pump field is assumed to be adiabatic. Approximating the in-phase component $c_j$ as the Ising spin as

\[
\sigma_j = \text{sgn}(c_j),
\]

(4)

it is proved that the global mode that achieves the minimum photon decay rate provides a ground state to the Ising problem under certain conditions. By gradually raising the gain by increasing the pump field of the OPOs, the network goes through the OPO phase transition and oscillates in one of the exact or approximate ground states, as shown in Fig. 1. In contrast to classical annealing and quantum annealing, the OPO network searches for the ground state in an upward direction and hence does not get stuck in the local minima. Time division multiplexing in a long ring cavity further facilitates scalability of the OPO network, in which less delay lines are required to realize coupling between oscillators, and thus the physical size of the system scales linearly. A programmable OPO network with all-to-all connections is later achieved. Alternately, the same Ising spin system can be implemented using stochastic nanomagnets. In this approach, the input spin current is taken as the weighted sum of magnetization moment of each nanomagnet as

\[
I_k(\{m_j\}) = I_0 \left( h_k + \sum_j^{N} 2J_{kj} m_j \right),
\]

(5)

where $I_0$ is a scaling constant related to elementary charge $q$, Planck constant $\hbar$, and damping factor $\alpha$. The spin current input has a constant bias $h_k$ together with a term proportional to the magnetization of the $j$'th nanomagnet $m_j$.

Figure 2(a) shows a possible implementation of each node in a stochastic spin network, which acts as a probabilistic bit ($p$-bit). A circular nanomagnet with in-plane magnetic anisotropy is used as a stochastic nanomagnet. In the absence of an input current, the
magnetization of the circular magnet rotates in the plane driven by thermal noise. Thanks to the circular shape, there is no preferred easy axis that would have arisen due to the shape anisotropy, effectively making its thermal stability $\Delta \ll kT$. The WRITE path is accomplished by the giant spin Hall effect (GSHE) of a heavy metal. A spin current is generated from a heavy metal by sending a charge current through the GSHE layer. This magnetization can be pinned by a spin current that is generated by flowing a charge current through the GSHE layer. The gain factor of GSHE, which relates to charge current to the spin current, is

$$
\beta = \frac{I_s}{I_c} = \theta_{SH} \frac{L_{FM}}{t} \left[ 1 - \text{sech} \left( \frac{t}{\lambda} \right) \right],
$$

(6)

where $\theta_{SH}$ is the Hall angle, $t$ is the thickness, and $\lambda$ is the spin-relaxation length of the heavy metal. The gain factor can be made $>1$ so that the small read current is sufficient to drive the next bit.

The free circular nanomagnet [Fig. 2(b)], together with a fixed layer, comprise an MTJ used to read out the magnetization state of the stochastic magnet. The conductance of the MTJ is governed by the relative angle between the magnetizations of the free layer and the fixed layer and is given as

$$
G_{MTJ} = G_0(1 + P^2 m_c),
$$

(7)

where $G_0$ is the average MTJ conductance and $P$ is the interface polarization. A small read current flows through the fixed layer. Two inverters are used to regularize the output signal. $V_R$ is the small READ voltage to be amplified by the inverters. Setting the bias resistance $R_0$ equal to $1/G_0$, the input voltage to the inverters $V_M$ becomes

$$
V_M = \frac{V_{DD}}{2} + \frac{V_R}{2 + m_c P^2},
$$

(8)

which has a mean value of $V_{DD}/2 + V_R/2$. To reduce nonlinearity, the interface polarization should be made small.

The design parameters used in the simulation are summarized in Table 1. The simulation is conducted in HSPICE. The dynamics of the nanomagnets, described by the stochastic Landau–Lifshitz–Gilbert equation, is solved using the modular framework. Circuit elements and CMOS inverter model naturally fit into HSPICE.
2.2 Formulation of Selected Combinatorial Optimization Problems

The analogy between Ising model and NP (nondeterministic polynomial time) hard problems, such as combinatorial optimization problems, have been widely studied and used in simulated annealing and, more recently, in quantum annealing. Here we select a few combinatorial optimization problems to demonstrate the ability of p-bit to solve NP hard problems stochastically. We present the formulation in this section and show the results in Sec. 3. The formulation follows another work. We briefly review it for self-consistency. Note that all diagonal terms are subtracted such that $J_{ii} = 0$ and that the resulting matrix is always symmetric.

2.2.1 Number partitioning problem

Number partitioning asks the following: given a set of $N$ positive numbers $S = \{n_1, \ldots, n_N\}$, how should this set be partitioned into two disjoint subsets such that the sum of the numbers in both sets is the same? This is written as an Ising model with the energy function

$$H = \left( \sum_{i=1}^{N} n_i s_i \right)^2 = \sum_{i,j} n_i n_j s_i s_j,$$

where $s_i = \pm 1$ is the Ising spin variable, representing the subset into which $n_i$ is partitioned. It is easy to see that the solution makes $H = 0$, whereas all other partitions lead to positive energy.

2.2.2 Exact cover

The exact cover problem asks the following: given a set $U = \{1, \ldots, n\}$, and $N$ subsets $V_i \subseteq U$, $i = 1, \ldots, N$, such that $U = \cup V_i$, how is a combination of the subsets $V_i$ found such that all of the subsets within this combination are disjoint sets and the union of them is $U$? The Hamiltonian for exact cover problem is formulated as

$$H = \sum_{a=1}^{n} \left( 1 - \sum_{i:a \in V_i} x_i \right)^2,$$

where $i$ runs between all numbers and $x_i$ is a binary variable that is 1 if set $i$ is included, and 0 if set $i$ is not included. The solution makes $H = 0$, whereas all other partitions lead to positive energy.

Table 1 Design parameters for nanomagnetic implementation of Ising Hamiltonian.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage $V_{DD}$</td>
<td>0.8 V</td>
</tr>
<tr>
<td>READ voltage $V_R$</td>
<td>0.5 V</td>
</tr>
<tr>
<td>Uniaxial-anisotropy field $H_k$</td>
<td>1 Oe</td>
</tr>
<tr>
<td>Saturation magnetization $M_s$</td>
<td>300 emu/cm$^3$</td>
</tr>
<tr>
<td>Magnet diameter and thickness</td>
<td>15 and 5 nm</td>
</tr>
<tr>
<td>MTJ polarization $P$</td>
<td>0.5</td>
</tr>
<tr>
<td>Damping coefficient $\alpha$</td>
<td>0.1</td>
</tr>
<tr>
<td>Spin Hall length $L$, width $W$, and thickness $t$</td>
<td>15, 15, and 3.15 nm</td>
</tr>
<tr>
<td>Hall angle $\theta$</td>
<td>0.5</td>
</tr>
<tr>
<td>Spin relaxation length $\lambda_{st}$</td>
<td>2.1 nm</td>
</tr>
<tr>
<td>Spin Hall resistivity $\rho$</td>
<td>200 $\mu\Omega$ cm</td>
</tr>
<tr>
<td>Temperature</td>
<td>300 K</td>
</tr>
</tbody>
</table>
energy. To come to the correct interconnecting matrix, we rewrite the Hamiltonian so that each binary variable can only take \( \pm 1 \) by defining the spin variable as \( x_i = (s_i + 1)/2 \) or \( s_i = 2x_i - 1 \).

### 2.2.3 Binary integer linear programming

Let \( x_1, \ldots, x_N \) be \( N \) binary variables. What is the largest value of \( c \cdot x \), given a constraint

\[
Sx = b,
\]

where \( S \) is an \( m \times N \) matrix and \( b \) is a vector with \( N \) components?

The Hamiltonian is written as

\[
H = H_A + H_B = A \sum_{j=1}^{m} \left[ b_j - \sum_{i=1}^{N} S_{ji}x_i \right]^2 - B \sum_{i=1}^{N} c_i x_i.
\]

The solution that satisfies the constraint makes \( H_A = 0 \) exactly. To ensure that the constraint is always satisfied while optimizing \( c \cdot x \), we require \( B \ll A \). A rule of thumb of the relative value of \( A \) and \( B \) is derived in Ref. 51.

### 2.2.4 Integer linear programming

The formulation of binary integer linear programming can be extended to any integer using integer coding. Here we use two’s complement. Each integer \( x \) can be coded by \( n \) bits \( s_1s_2\ldots s_n \), where \( x = -2^{n-1}s_1 + 2^{n-2}s_2 + \cdots + 2^1s_{n-1} + s_n \).

### 3 Results

In this section, we demonstrate the solution of some combinatorial optimization problems formulated in the previous section. The Hamiltonian and resulting interconnection matrices can be found in the Appendix.

We first show a 3-bit number partitioning in Fig. 3(a), where the constant bias \( h = 0 \). We then demonstrate a 3-bit binary integer linear programming problem in Fig. 3(b) and a 6-bit integer linear programming using two’s complement in Figs. 3(c) and 3(d) as examples of optimization with a constraint. To demonstrate the scale up potential, we then show five different exact cover problems with 3, 6, 9, 12, and 15 bits in Figs. 4 and 5.

In the 3-bit number partitioning problem, 3-bit binary integer linear programming problem, 6-bit integer linear programming problem, and 6- and 9-bit exact cover problems, the total simulated time is 0.1 \( \mu s \) with 0.05 ps time step. In the 12- and 15-bit exact cover problems, the total simulated time is 10 \( \mu s \) with 0.1 ps time step because more samples are needed to sample the larger solution space. In addition, the results of 10 simulations with different seeds are averaged to smooth out the histogram for 6-bit integer linear programming problems and 12- and 15-bit exact cover problems. No significant difference is observed between different seeds as shown in Fig. 5(b).

#### 3.1 Number Partitioning Problem

If \( n_1 = 1, n_2 = 2, n_3 = 3 \), apparently the lowest energy state will be 001 and 110. The energy for each state is given in Table 2.

The distribution of solutions for the 3-bit problem is provided in Fig. 3(a). Overall, our numerical model shows a strong agreement with the expectations of the Boltzmann law. The differences can be explained as sampling errors and generally go down systematically as more data points are collected by the system.
3.2 Binary Integer Linear Programming

As an example, we choose $S = (1,1,1)$ and $b = (1)$ to be the constraint and $c \cdot x$ to be the objective function to maximize under the constraint, where $c = (-3,-2,-1)$.

If we consider the solution to the given constraint only, there are three binary solutions that satisfy the constraint, namely $x = (1,0,0)^T$, $(0,1,0)^T$, or $(0,0,1)^T$. Among them, the optimal solution that maximizes the objective function is $(0,0,1)^T$. As shown in Fig. 3(b), the solution clearly outweighs the other states.

### Table 2  Relative energy of states for the number partitioning problem – the predicted and observed frequencies are provided in Fig. 3(a).

<table>
<thead>
<tr>
<th>State</th>
<th>000</th>
<th>001</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative energy</td>
<td>9</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 3 Left axis: distribution of solutions from modeling compared to the expectations from the Boltzmann law. Right axis: the corresponding energy in unit of $kT$. The probability and energy of a particular state is related by $P_i = \frac{e^{-E_i/kT}}{\sum_j e^{-E_j/kT}}$. Results for four different problems: (a) a 3-bit number partitioning problem; (b) a 3-bit binary integer linear programming problem; and a 6-bit integer linear programming problem, (c) without an objective function and (d) with an objective function. Overall, a strong agreement between theory and modeling is observed. Therefore, the lowest energy states consistently show the highest probabilities and can be interpreted as the correct results. Note that the energy is not linear in this figure. Because $P = 0$ corresponds to $E = \infty$, we cut off energy at a finite value when the probability is sufficiently close to zero. The normalization constant (canonical partition function) in the denominator sets the ranges and absolute values of energy, and thus the range of energy is different in each graph. The negative values of energy come from removing the quadratic and constant terms in the Hamiltonian when constructing the interconnection matrices. It can be justified by the fact that the reference value for each energy is arbitrary, so we can add any constant value without changing the results.
As an example, we demonstrate a 2-integer problem with the constraint $S = (1, 3)$, $b = 0$, and objective function $c \cdot x$, where $c = (1, -1)$.

We use three bits two’s complement to represent each integer and restrict $x, y$ to be integers in $[-4, 3]$. Within this range, the solutions to the constraints are $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}$, or in binary form $\begin{pmatrix} 101 \\ 001 \end{pmatrix}, \begin{pmatrix} 000 \\ 000 \end{pmatrix}, \begin{pmatrix} 011 \\ 111 \end{pmatrix}$, which are equivalent to 41, 0, 31 in decimal. Among them, the optimal solution is $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$, which is state 31. As shown in Figs. 3(c) and

Fig. 4 Simulated results compared to expected values of exact cover problems. (a) 3 bits, (b) 6 bits, (c) 9 bits, and (d) 12 bits. Due to the large solution space, only the top 10 solutions are shown in (c) and (d).

Fig. 5 (a) Simulated results of the 15-bit exact cover problem after averaging the histograms from 10 simulations with different pseudorandom seeds. (b) The probabilities of the four most probable states obtained from different seeds are shown. Consistent results are observed. The order of the seeds do not matter.

3.3 Integer Linear Programming

As an example, we demonstrate a 2-integer problem with the constraint $\vec{S} = (1, 3)$, $b = 0$, and objective function $c \cdot x$, where $c = (1, -1)$.

We use three bits two’s complement to represent each integer and restrict $x, y$ to be integers in $[-4, 3]$. Within this range, the solutions to the constraints are $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}$, or in binary form $\begin{pmatrix} 101 \\ 001 \end{pmatrix}, \begin{pmatrix} 000 \\ 000 \end{pmatrix}, \begin{pmatrix} 011 \\ 111 \end{pmatrix}$, which are equivalent to 41, 0, 31 in decimal. Among them, the optimal solution is $\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$, which is state 31. As shown in Figs. 3(c) and
3(d), the three candidate solutions correspond to the three most pronounced peaks, and the probability of the best solution is enhanced after putting in the objective function.

### 3.4 Exact Cover

#### 3.4.1 3 p-bits

Given a set of four integer numbers \( U = \{1, 2, 3, 4\} \) and three subsets \( V_1 = \{1, 3\}, V_2 = \{2, 3\}, \) and \( V_3 = \{2, 4\}, \) the most probable state is \((101)_2 = (5)_{10}\). The simulation result is shown in Fig. 4(a). Good agreement with Boltzmann distribution is observed.

#### 3.4.2 6 p-bits

Given a set of seven integer numbers \( U = \{1, 2, 3, 4, 5, 6, 7\} \) and six subsets \( V_1 = \{1, 4\}, V_3 = \{4, 5, 7\}, V_4 = \{3, 5, 6\}, V_5 = \{2, 3, 6, 7\}, \) and \( V_6 = \{2, 7\}, \) the most probable state is \((010101)_2 = (21)_{10}\), covering sets 2, 4, and 6. The simulation result is shown in Fig. 4(b).

#### 3.4.3 9 p-bits

Given a set of nine integer numbers \( U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \) and nine subsets \( V_1 = \{4, 8, 9\}, V_2 = \{1, 7, 8\}, V_3 = \{1, 2, 8\}, V_4 = \{1, 6, 7\}, V_5 = \{7, 8, 9\}, V_6 = \{4, 5, 6\}, V_7 = \{2, 5, 8\}, V_8 = \{1, 2, 3\}, \) and \( V_9 = \{3, 4, 6\}, \) the most probable state is \((11100000000)_2 = (3840)_{10}\), covering sets 5, 6, and 8. The simulation result is shown in Fig. 4(c). Due to the large solution space, we only show the histogram of the ten most probable states from the simulation results.

#### 3.4.4 12 p-bits

Given a set of 10 integer numbers \( U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \) and 12 subsets \( V_1 = \{1, 2, 5, 7\}, V_2 = \{3\}, V_3 = \{4, 6, 9\}, V_4 = \{8, 10\}, V_5 = \{1, 8, 9\}, V_6 = \{3, 4, 7\}, V_7 = \{1, 3, 5, 6\}, V_8 = \{2, 5, 8\}, V_9 = \{1, 2, 9\}, V_{10} = \{4, 5, 6, 10\}, V_{11} = \{1, 3, 9\}, \) and \( V_{12} = \{5, 6, 8\}, \) the most probable state is \((11100000000000000)_2 = (1530)_{10}\), covering sets 1 to 4.

The second most probable states are 73, 329, and 2816, where \((73)_{10} = (000010010001001)_2\) covers sets 1, 2, 3, 4, 5, 6, 7, 8, 9, \((329)_{10} = (0001010001000101)_2\) covers sets 1, 2, 3, 4, 5, 6, 7, 8, (twice), 9, 10, \((2816)_{10} = (1011000000000000)_2\) covers sets 1, 2, 4, 5, 6, 7, 8, 9, 10. The simulation result is shown in Fig. 4(d). It can be seen that not only the best solution but also the second best solutions can be clearly distinguished.

#### 3.4.5 15 p-bits

Given a set of 12 integer numbers \( U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\} \), and 15 subsets \( V_1 = \{1, 2, 5, 7\}, V_2 = \{3\}, V_3 = \{4, 6, 9, 12\}, V_4 = \{8, 10, 11\}, V_5 = \{1, 8, 9\}, V_6 = \{3, 4, 7, 11\}, V_7 = \{1, 3, 5, 6\}, V_8 = \{2, 5, 8\}, V_9 = \{1, 2, 9\}, V_{10} = \{4, 5, 6, 10\}, V_{11} = \{1, 3, 9\}, V_{12} = \{5, 6, 8, 12\}, V_{13} = \{2, 10, 11\}, V_{14} = \{1, 3, 7, 8\}, \) and \( V_{15} = \{2, 6, 11, 12\}, \) the most probable state is \((30720)_{10} = (1111010000000000)_2\), covering sets 1 to 4.

The second most probable states are \((584)_{10} = (000011010010001001)_2\), covering sets 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, \((4102)_{10} = (001000000000110110)_2\) covering sets 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, \((22528)_{10} = (1011000000000000)_2\) covering sets 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12. The result is shown in Fig. 5(a). We also compare the results from 10 different uncorrelated random seeds, and consistent results are observed, as shown in Fig. 5(b).

### 4 Discussion

As demonstrated in the previous section, p-bit computing shows great scaling potential. From the 3-bit to the 15-bit exact problem, the result is not degraded by the increasing number of bits and the corresponding increasing solution space.
When the best solution consists of multiple solutions with the same energy levels [Fig. 3(c)], the dynamics of the magnet may be affected by the seed of the random number generator and the initial condition. In such conditions, one has to run the simulation with more seeds and take the average of the probability from each seed. However, when there is degeneracy in the second (or higher) best solution [Figs. 4(d) and 5(a)], they generally all have experienced sufficient sampling to satisfy the Boltzmann law. Overall, the agreement is good in that the lowest energy states consistently show up most frequently in our model sampling, and the consistency of the distribution with the Boltzmann law improves as the number of samples increases.

However, in this implementation, the simulation time generally scales exponentially with the number of bits to reach an exact convergence to the Boltzmann distribution. When the solution to a larger problem is needed, it requires huge computing resources to perform such a simulation on a serial computer. However, we note that the solution strategy of probabilistic computing is not to look for an exact solution, if it exists. One may run the simulation for a reasonable amount of time, identify the most probable states, and use a classical computer to check the solutions in polynomial time. In addition, it has been reported that downscaling of the nanodevices leads to a lower magnetization stability of the free magnet and thus increases the speed of the $p$-bit network exponentially, in contrast to MRAM technology.\textsuperscript{38,52} Finally, the OPO approach is able to sidestep this exponential increase by reaching the ground state through a virtually negative temperature which allows one to directly reach one or more ground states. This approach is in principle even more efficient than simulated annealing with quantum tunneling.\textsuperscript{21}

Another path to speed-up of these problems involves parallelization. Parallel algorithms have been developed to solve combinatorial optimization problems distributively on a classical computer.\textsuperscript{53-55} However, these algorithms are available to classical computers but may not be directly applied to probabilistic computer. Here we propose a heuristic using linear programming as an example. Assuming that the constraint matrix is block-diagonal dominant, we approach the problem in an iterative procedure. We first distributively solve the block-diagonal terms on a parallel computer. The residue of the solution caused by off-diagonal terms is used to construct the new block-diagonal terms. The new problem with the new block-diagonal matrix is again solved distributively, and the residue is calculated. We repeat the process until the residue is small. It will potentially reduce the computational complexity to $O(N \log N)$. To implement such algorithms, the interconnection matrices should be designed on the fly, and a programmable analog photonic network is needed.

Fabrication of low-energy barrier nanomagnets and MTJs presents great challenges,\textsuperscript{56} and the system is not completely immune to defects and device variations.\textsuperscript{57} Recent advances in fabrication techniques can enable the physical realization of stochastic magnet networks for probabilistic computing.\textsuperscript{39,58,59} Also a successful $p$-bit network relies on the fast response of the interconnection. Fast switching and signal transmission is of paramount importance in such applications.

The proposed system with nanomagnets can be readily mapped to an OPO network. A time-division multiplexing scheme is used in an OPO network so that a single-ring resonator accommodates many Ising spins simultaneously, whereas in a nanomagnet network, each Ising spin is represented by an MTJ. Delay lines are used for mutual coupling in an OPO network, and spin current is used to connect different spins in a nanomagnet network. Both systems can operate under room temperature with manageable noise. Though an OPO network utilizes existing optical components including mirrors and beam splitters and periodically poled lithium niobate crystals, a femtosecond laser can be costly. Still, as mentioned above, the OPO network solution mechanism in experiments can be highly efficient and may be the best approach for large problems that cannot be easily parallelized.

5 Summary
We demonstrate the capability of probabilistic computing in solving combinatorial optimization enabled by stochastic nanomagnet networks. The system is compatible with current fabrication techniques and thus is physically realizable. The simulation results of four different combinatorial optimization problems show robust results from 3 to 15 bits. Further, scaling in simulation capability can be achieved by parallel algorithms or transition to OPO networks that can more efficiently solve larger problems.
6 Appendix: Hamiltonian and Interconnection Matrices

6.1 Number Partitioning

We write down the constraint part of the Hamiltonian as integer Linear Programming

\[ H = (s_1 + 2s_2 + 3s_3)^2 = s_1^2 + 4s_2^2 + 9s_3^2 + 4s_1s_2 + 6s_1s_3 + 12s_2s_3, \]  
\[ J = -\begin{pmatrix} 0 & 4 & 6 \\ 4 & 0 & 12 \\ 6 & 12 & 0 \end{pmatrix}. \]

6.2 Binary Integer Linear Programming

Let \( A = 4, B = 1 \) and replace \( x_i \) by spin variable \( x_i = (s_i + 1)/2 \); the Hamiltonian then becomes

\[ H = 4(1 - x_1 - x_2 - x_3) + (3x_1 + 2x_2 + x_3) \]
\[ = (2s_1^2 + 2s_2^2 + 2s_3^2 + 4s_1s_2 + 4s_1s_3 + 4s_2s_3) + (7s_1 + 6s_2 + 5s_3) + 8, \]

where we scale the Hamiltonian by a constant in the last step. From the above Hamiltonian, we construct the interconnection matrix as

\[ J = -\begin{pmatrix} 0 & 4 & 4 \\ 4 & 0 & 4 \\ 4 & 4 & 0 \end{pmatrix}, \quad h^T = -(7 \ 6 \ 5). \]

6.3 integer Linear Programming

We write down the constraint part of the Hamiltonian as

\[ H_A = (x + 3y)^2. \]

\( x, y \) is represented by

\[ x = -4x_1 + 2x_2 + x_3, \]
\[ y = -4x_4 + 2x_5 + x_6, \]
\[ x_i = (s_i + 1)/2, \]

where \( x_i \) takes 0 or 1 and \( s_i \) takes \( \pm 1 \). Then

\[ H_A = 4 + 8s_1 - 4s_2 - 2s_3 + 24s_4 - 12s_5 - 6s_6 + 4s_1^2 - 4s_1s_2 - 2s_1s_3 + 24s_1s_4 - 12s_1s_5 
- 6s_1s_6 + s_2^2 + s_2s_3 - 12s_2s_4 + 6s_2s_5 + 3s_2s_6 + 1/4s_3^2 - 6s_3s_4 + 3s_3s_5 + 3/2s_3s_6 
+ 36s_4^2 - 36s_4s_5 - 18s_4s_6 + 9s_5^2 + 9s_5s_6 + 9/4(s_6^2). \]

From the above Hamiltonian, we construct the interconnection matrix as

\[ J = \begin{pmatrix} 0 & 4 & 2 & -24 & 12 & 6 \\ 4 & 0 & -1 & 12 & -6 & -3 \\ 2 & -1 & 0 & 6 & -3 & -1.5 \\ -24 & 12 & 6 & 0 & 36 & 18 \\ 12 & -6 & -3 & 36 & 0 & -9 \\ 6 & -3 & -1.5 & 18 & -9 & 0 \end{pmatrix}. \]

\[ h^T = (-8 \ 4 \ 2 \ -24 \ 12 \ 6). \]
We then write down the objective part of the Hamiltonian as
\[
H_B = -c \cdot x \\
= -x + y \\
= 2s_1 - s_2 - 0.5s_3 - 2s_4 + s_5 + 0.5s_6,
\]
which changes the bias \( h \) only. We scale \( H_B \) by \( 1/5 \), and then the overall biasing matrix should be modified as
\[
h^T = (-8.4 \quad 4.2 \quad 2.1 \quad -23.6 \quad 11.8 \quad 5.9). \tag{20}
\]

6.4 Exact Cover

6.4.1 3 p-bits

The Hamiltonian is
\[
H = A\left[(1 - x_1)^2 + (1 - x_2 - x_3)^2 + (1 - x_4 - x_5)^2 \right] \\
= A/2\left[(s_1^2 + s_2^2 + s_3^2 + s_4s_2 + s_2s_3) - (s_1 + s_3) + 1\right], \tag{21}
\]
where the spin variables \( s_i = 2x_i - 1 \) takes values of \( \pm 1 \) and the coefficients are absorbed in \( A \). Thus
\[
J = -\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad h^T = (1 \quad 0 \quad 1).
\]

6.4.2 6 p-bits

The Hamiltonian is then
\[
H = A\left[(1 - x_1 - x_2)^2 + (1 - x_3 - x_6)^2 + (1 - x_4 - x_5)^2 + (1 - x_1 - x_2 - x_3)^2 \right] \\
+ (1 - x_3 - x_4)^2 + (1 - x_4 - x_5)^2 + (1 - x_1 - x_2 - x_3)^2 \right] \\
= A/4\left[3s_1^2 + 2s_2 + 3s_3^2 + 3s_4^2 + 4s_5^2 + 2s_6^2 + 4s_1s_2 + 4s_1s_3 + 2s_1s_5 + 2s_1s_6 + 2s_2s_3 + 2s_3s_4 + 2s_3s_5 + 2s_4s_5 + 2s_4s_6 + 2s_5s_6 + 6s_1 + 2s_2 + 6s_3 + 4s_5 + 4s_6 + 5\right], \tag{22}
\]
where the coefficients are again absorbed in \( A \). Thus
\[
J = -\begin{pmatrix} 0 & 4 & 4 & 0 & 2 & 2 \\ 4 & 0 & 2 & 0 & 0 & 0 \\ 4 & 2 & 0 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 & 4 & 0 \\ 2 & 0 & 2 & 4 & 0 & 4 \\ 2 & 0 & 2 & 0 & 4 & 0 \end{pmatrix}, \quad h^T = -\begin{pmatrix} 6 & 2 & 6 & 0 & 4 & 4 \end{pmatrix}.
\]
6.4.3 9 p-bits

The Hamiltonian is then

\[
H = A[(1 - x_2 - x_3 - x_4 - x_9)^2 + (1 - x_3 - x_7 - x_8)^2 + (1 - x_8 - x_9)^2 \\
+ (1 - x_1 - x_6 - x_9)^2 + (1 - x_9 - x_{10})^2 + (1 - x_3 - x_7 - x_{10})^2 + (1 - x_2 - x_4 - x_5)^2 \\
+ (1 - x_1 - x_2 - x_3 - x_5 - x_7)^2 + (1 - x_1 - x_5)^2] \\
= A/4[3s_1^2 + 3s_2^2 + 3s_3^2 + 3s_4^2 + 3s_5^2 + 3s_6^2 + 3s_7^2 + 3s_8^2 + 3s_9^2 + 2s_1s_2 + 2s_1s_3 + 4s_1s_5 \\
+ 2s_1s_6 + 2s_1s_7 + 2s_1s_8 + 4s_2s_3 + 4s_2s_4 + 4s_2s_5 + 2s_2s_7 + 2s_2s_8 + 2s_3s_4 + 2s_3s_5 \\
+ 4s_3s_7 + 4s_3s_8 + 2s_4s_5 + 2s_4s_6 + 2s_4s_8 + 2s_4s_9 + 2s_5s_7 + 2s_5s_8 + 2s_5s_9 + 2s_7s_8 \\
+ 2s_8s_9 + 8s_{1+12} + 12s_{2+3} + 8s_{4+5} + 8s_{5+7} + 6s_8 + 4s_9 + 17],
\]

where the coefficients are absorbed in \( A \). Thus

\[
J = \begin{pmatrix}
0 & 2 & 2 & 0 & 4 & 2 & 2 & 0 & 2 & 0 \\
2 & 0 & 4 & 4 & 4 & 0 & 2 & 2 & 0 & 0 \\
2 & 4 & 0 & 2 & 0 & 4 & 4 & 0 & 0 & 0 \\
0 & 4 & 2 & 0 & 2 & 2 & 0 & 2 & 2 & 2 \\
4 & 4 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 2 \\
2 & 2 & 4 & 0 & 2 & 2 & 0 & 2 & 0 & 0 \\
0 & 2 & 4 & 2 & 0 & 2 & 0 & 2 & 0 & 0 \\
2 & 0 & 0 & 2 & 0 & 4 & 0 & 2 & 0 & 0
\end{pmatrix},
\]

\[
H^T = -(8 \quad 12 \quad 12 \quad 8 \quad 8 \quad 4 \quad 8 \quad 6 \quad 4).
\]

6.4.4 12 p-bits

The Hamiltonian is then

\[
H = A[(1 - x_1 - x_5 - x_7 - x_9 - x_{11})^2 + (1 - x_1 - x_8 - x_9)^2 + (1 - x_2 - x_6 - x_7 - x_{11})^2 \\
+ (1 - x_3 - x_6 - x_9)^2 + (1 - x_1 - x_7 - x_8 - x_{10})^2 + (1 - x_3 - x_7 - x_{10} - x_{12})^2 \\
+ (1 - x_1 - x_6)^2 + (1 - x_4 - x_5 - x_8 - x_{12})^2 + (1 - x_3 - x_5 - x_9 - x_{11})^2 \\
+ (1 - x_4 - x_{10})^2] \\
= A[4s_1^2 + 2s_1s_5 + 2s_1s_6 + 2s_1s_7 + 4s_1s_8 + 4s_1s_9 + 2s_1s_10 + 2s_1s_{11} + 2s_1s_{12} \\
+ s_2^2 + 2s_2s_5 + 2s_2s_6 + 2s_2s_7 + 2s_2s_{11} + 3s_3^2 + 2s_3s_5 + 2s_3s_6 + 2s_3s_7 + 2s_3s_9 + 4s_3s_{10} \\
+ 2s_3s_{11} + 2s_3s_{12} + 2s_4^2 + 2s_4s_5 + 2s_4s_6 + 2s_4s_7 + 2s_4s_{10} + 2s_4s_{12} + 3s_5^2 + 2s_5s_7 + 2s_5s_8 \\
+ 4s_5s_9 + 4s_5s_{11} + 2s_5s_{12} + 3s_6^2 + 2s_6s_7 + 2s_6s_9 + 2s_6s_{10} + 2s_6s_{11} + 4s_7^2 + 2s_7s_8 + 2s_7s_9 \\
+ 4s_7s_{10} + 4s_7s_{11} + 4s_7s_{12} + 3s_8^2 + 2s_8s_9 + 2s_8s_{10} + 4s_8s_{12} + 3s_9^2 + 4s_9s_{11} + 4s_9s_{12} \\
+ 4s_{10}^2 + 3s_{11}^2 + 3s_{12}^2 + 14s_1 + 4s_2 + 10s_3 + 4s_4 + 14s_5 + 6s_6 + 20s_7 + 12s_8 \\
+ 12s_9 + 12s_{10} + 14s_{11} + 14s_{12} + 36],
\]

where the coefficients are absorbed in \( A \).
Thus

\[
J = \begin{pmatrix}
0 & 0 & 0 & 0 & 2 & 2 & 4 & 4 & 4 & 2 & 2 & 2 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 4 & 2 & 2 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 2 & 0 & 2 \\
2 & 0 & 2 & 2 & 0 & 0 & 2 & 2 & 4 & 0 & 4 & 2 \\
2 & 2 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 2 \\
4 & 2 & 2 & 0 & 2 & 2 & 4 & 4 & 4 & 0 & 0 & 0 \\
4 & 0 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 0 & 4 \\
4 & 0 & 2 & 0 & 4 & 0 & 2 & 2 & 0 & 0 & 4 & 0 \\
2 & 0 & 4 & 2 & 0 & 2 & 4 & 2 & 0 & 0 & 0 & 4 \\
2 & 2 & 2 & 0 & 4 & 2 & 4 & 0 & 4 & 0 & 0 & 0 \\
2 & 0 & 2 & 2 & 2 & 0 & 4 & 4 & 0 & 0 & 0 & 0 
\end{pmatrix}
\]

\[
\]

6.4.5 15 p-bits

The Hamiltonian is then

\[
H = A [(1 - x_1 - x_5 - x_7 - x_9 - x_{11} - x_{13} - x_{14})^2 + (1 - x_1 - x_3 - x_7 - x_{11} - x_{14})^2] \\
+ (1 - x_2 - x_6 - x_7 - x_{11} - x_{14})^2 + (1 - x_3 - x_6 - x_{10})^2 \\
+ (1 - x_1 - x_7 - x_{8} - x_{10} - x_{12})^2 + (1 - x_3 - x_7 - x_{10} - x_{12} - x_{15})^2 \\
+ (1 - x_1 - x_6 - x_{14})^2 + (1 - x_4 - x_5 - x_8 - x_{12} - x_{14})^2 \\
+ (1 - x_3 - x_5 - x_9 - x_{11})^2 + (1 - x_4 - x_{10} - x_{13})^2 \\
+ (1 - x_4 - x_{6} - x_{13} - x_{18})^2 + (1 - x_3 - x_{12} - x_{15})^2 \\
= A/4 [4s_1^2 + 2s_1s_5 + 2s_1s_6 + 4s_1s_7 + 4s_1s_8 + 4s_1s_9 + 2s_1s_{10} + 2s_1s_{11} + 2s_1s_{12} + 2s_1s_{13} + 4s_1s_{14} + 2s_1s_{15} + s_5^2 + 2s_2s_6 + 2s_2s_7 + 2s_2s_{11} + 2s_2s_{14} + 4s_2^2 + 2s_3s_5 + 2s_3s_6 \]
\+ 2s_3s_7 + 2s_3s_9 + 4s_3s_{10} + 2s_3s_{11} + 4s_3s_{12} + 4s_3s_{15} + 3s_4^2 + 2s_4s_8 + 2s_4s_9 + 2s_4s_{10} + 2s_4s_{12} + 4s_4s_{13} + 2s_4s_{14} + 2s_4s_{15} + 3s_5^2 + 2s_5s_7 + 2s_5s_8 + 2s_5s_9 + 4s_5s_{11} + 2s_5s_{12} + 4s_5s_{14} + 4s_6^2 + 2s_6s_7 + 2s_6s_{10} + 2s_6s_{11} + 2s_6s_{13} + 4s_6s_{14} + 2s_6s_{15} + 4s_7^2 + 2s_7s_8 + 2s_7s_9 + 4s_7s_{10} + 4s_7s_{11} + 4s_7s_{12} + 4s_7s_{13} + 2s_7s_{15} + 3s_8^2 + 2s_8s_9 + 2s_8s_{10} + 4s_8s_{12} + 2s_8s_{15} + 3s_9^2 + 2s_9s_{11} + 2s_9s_{13} + 2s_9s_{14} + 2s_9s_{15} + 4s_{10}^2 + 4s_{10}s_{12} + 2s_{10}s_{13} + 2s_{10}s_{15} + 3s_{11}^2 + 4s_{11}s_{14} + 4s_{12}^2 + 2s_{12}s_{14} + 4s_{12}s_{15} + 3s_{13}^2 + 4s_{13}s_{15} + 4s_{14}^2 + 4s_{15}^2 + 2s_{11} + 6s_5 + 4s_{11} + 12s_4 + 18s_5 + 14s_6 + 26s_7 + 18s_8 + 18s_9 + 16s_{10} + 18s_{11} + 22s_1 + 6s_5 + 14s_3 + 12s_4 + 18s_5 + 14s_6 + 26s_7 + 18s_8 + 18s_9 + 16s_{10} + 18s_{11} + 20s_{12} + 12s_{13} + 22s_{14} + 18s_{15} + 73],
\]

where the coefficients are absorbed in $A$. Thus
\[
J = \begin{bmatrix}
0 & 0 & 0 & 2 & 2 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 4 & 2 \\
0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 2 & 2 \\
0 & 0 & 0 & 2 & 2 & 2 & 2 & 2 & 4 & 4 & 0 & 0 & 4 & 0 \\
0 & 2 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 4 & 2 & 2 & 0 & 2 \\
0 & 0 & 2 & 2 & 4 & 0 & 4 & 2 & 0 & 4 & 0 & 2 & 2 & 2 \\
0 & 2 & 0 & 0 & 2 & 2 & 0 & 2 & 4 & 2 & 0 & 2 & 4 & 2 \\
0 & 0 & 0 & 2 & 4 & 4 & 4 & 4 & 0 & 4 & 2 & 2 & 2 & 2 \\
0 & 0 & 0 & 2 & 4 & 0 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
0 & 0 & 4 & 0 & 2 & 2 & 2 & 2 & 0 & 2 & 0 & 2 & 0 & 2 \\
0 & 0 & 4 & 2 & 0 & 2 & 0 & 2 & 0 & 4 & 0 & 2 & 2 & 2 \\
0 & 0 & 0 & 4 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
0 & 0 & 4 & 2 & 0 & 2 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
0 & 0 & 0 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
0 & 0 & 0 & 4 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2
\end{bmatrix},
\]

\[
h^T = -\begin{bmatrix}
22 \\
6 \\
14 \\
12 \\
18 \\
14 \\
26 \\
18 \\
18 \\
16 \\
18 \\
20 \\
12 \\
22 \\
18
\end{bmatrix}.
\]

Note that the lower triangular part of $J$ is not shown here for readability.

**Acknowledgments**

We would like to thank Dr. Kerem Camsari for valuable discussions. We would also like to acknowledge funding support from the National Science Foundation (Grants Nos. EEC1454315-CAREER, EEC-1227110, and CBET-1855882); the Office of Naval Research (Grant No. N00014-19-S-B001); and the Purdue University Big Idea Competition 2.0. Z.X. would like to acknowledge support from Purdue OUR Scholarship.

**References**

Zhu, Xie, and Bermel: Numerical simulation of probabilistic computing to NP-complete number theory...


**Jie Zhu** received his bachelor's degree in electronic engineering from Fudan University, Shanghai, China, in 2018. He is currently a PhD candidate at Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, United States. His research interests include computational electromagnetics, quantum optics, and probabilistic computing.

**Zhengxiang Xie** received his bachelor's degree in electrical engineering from Purdue University, West Lafayette, Indiana, United States, in 2022. He is now pursuing his master's degree at the same university.

**Peter Bermel** leads the Energy and Nanophotonics Research Group at Purdue University that leverages nanophotonics to improve the performance of photovoltaics, thermophotovoltaics, and microelectronics. He received an NSF CAREER award, a Winston Churchill Foundation Scholarship, an NSF Graduate Research Fellowship, an MIT Compton Fellowship, and the Elmore Named Associate Professor Position. He is widely published in scientific peer-reviewed journals, and has been cited more than 9300 times, for an h-index of 38.