CHINESE OPTICS LETTERS

Vol. xx, No. x | August 2024

Smoothed analysis based noise manipulation for spatial photonic Ising machines: supplemental document

Xin Ye (叶 新)[1], Wenjia Zhang (张文甲)[1], and Zuyuan He (何祖源)[1]

State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai, China 200240*Corresponding author: wenjia.zhang@sjtu.edu.cn

Received Month X, XXXX | Accepted Month X, XXXX | Posted Online Month X, XXXX

Photonic Ising machine, a promising non-von Neumann computational paradigm, offers a feasible way to address combinatorial optimization problems. We develop a digital noise injection method for spatial photonic Ising machines based on smoothed analysis where noise level acts as a parameter quantifying the smoothness degree. Through experiments on 20736-node Max-Cut problems, we establish stable performance within a smoothness degree of 0.04 to 0.07. Digital noise injection results in a 24% performance enhancement, showing a 73% improvement over heuristic SG algorithms. Furthermore, to address noise-induced instability concerns, we propose an optoelectronic co-optimization method for a more streamlined smoothing method with strong stability.

Keywords: photonic Ising machine, smoothed analysis, optoelectronic co-optimization.
DOI: xxxxxxx/COLxxxxxxx.

1. The FLIP algorithm

The FLIP algorithm is a natural algorithm for solving the Max-Cut problems and requires exponential time before reaching a local maximum in the worst-case scenario[1,2]. Nevertheless, based on the framework of smoothed analysis, the smoothed complexity of the FLIP algorithm for local Max-Cut is quasi-polynomial, where the degree of smoothing is determined by the magnitude of perturbations[3,4].

The FLIP algorithm starts with an initial partition and subsequently moves nodes individually to the opposite side if the flip enhances the weight of the cut. This process persists until no further local improvements can be achieved. Mapping to the Ising model, the Hamiltonian $H = -\sum_{<l,k>} w_{l,k} x_l x_k$ quantity replaces the cut value $W = \frac{1}{2}\sum_{<l,k>} w_{l,k}(1-x_l x_k)$ as cost function, and the movement of the vertex corresponds to the flip of the spin.

Algorithm 1 The FLIP Algorithm

Input: A complete graph $G(V,E)$ with edge weights ω_{lk}, $\forall l,k \in V$, $l \neq k$

Output: A cut $S_1, S_2 : S_1 \cup S_2 = V$, $S_1 \cap S_2 = \emptyset$ with the max cut value $cut(S_1, S_2)$

```
Initial spin state (node patition) x ← x_0 ∈ {−1,1}^N, flip number n
for i = 1 : n do
    Randomly flip one spin to generate a new spin vector x^new;
    H_new = −∑_{<l,k>} w_{l,k} x_l^new x_k^new;
    if H_new < H then
        x ← x^new; H ← H_new
    end if
end for
Return x;
for j = 1 : N do
    if x_j = 1 then
        add j* to S_1
    else
        add j* to S_2
    end if
end for
W = 1/2 ∑_{<l,k>} w_{l,k}(1 − x_l x_k)
Return cut S_1, S_2; cut value W
```

2. Implementation of noise-injected SPIM

2.1. Experimental setup

The experimental configuration for the noise-injected SPIM is illustrated in Fig. 1 (a). A coherent beam at 633*nm* originating from a stabilized red HeNe laser is injected

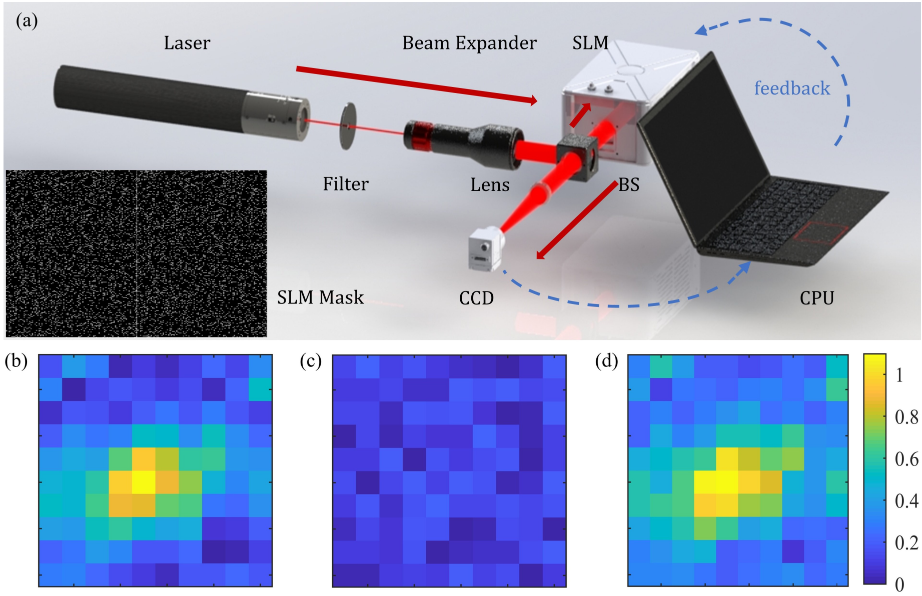

Fig. 1. The experiment system of noise-injected SIPM. (a) The architecture of SIPM with phase-amplitude modulation based on Euler' s formula. (b) Image captured with CCD. (c) Image characterized by random noise at the noise level of 0.2. (d) Image containing artificially injected noise. .

to a variable neutral density filter. A 40X beam expansion setup is positioned behind the filter to generate a large spot, ensuring sufficient coverage of the employed reflective phase-only SLM (HOLOEYE LETO-3-CFS-127, 1920×1080 pixels). The implementation of amplitude and phase modulation is crucial for the mapping of combinatorial optimization problems. In the experiment, we adopt the phase-amplitude modulation based on Euler' s formula [5]. This modulation scheme simplifies system design by using only one SLM, thus effectively addressing the inherent challenge of pixel alignment in spatial light systems. In order to reduce position deviation, each spin is encoded by a macropixel with 10-by-10 pixels on SLM with a pixel pitch of 6.4 µm. A beam splitter is placed opposite the SLM in order to separate the reflected light. Following the Fourier transform of the lens (focal length $f = 150mm$), the light is concentrated at the focal point, and its central light intensity shows an expression akin to the Hamiltonian. We measure the intensity distribution on charge-coupled device (CCD), and the obtained images (see Fig. 1 (b)) are subsequently transmitted to the CPU for the purpose of artificial noise injection. The 10 × 10 pixels covering the complete center spot are selected as the detection area with a pixel pitch of 4.54 µm.

2.2. Noise-injected method

Adjusting the exposure time to indirectly control the intrinsic noise of the system is considered an easy-to-operate approach for noise injection [6]. Nonetheless, quantifying the noise level and extending this method to other analog hardware poses challenges. To achieve a controllable noise source, we generate a normal distribution with a mean value of 0 based on the size of the acquired image, shown in Fig. 1 (c) . The variance of this random matrix corresponds to the level of noise. The acquired image is then combined with additive white noise to produce the noise-injected image, shown in Fig. 1 (d).

Algorithm 2 The SG Algorithm

Input: A complete graph $G(V,E)$ with edge weights $\omega_{ij},\ \forall i,j \in V,\ i \neq j$

Output: A cut $S_1, S_2 : S_1 \cup S_2 = V,\ S_1 \cap S_2 = \emptyset$ with the max cut value $cut(S_1,S_2)$

$V' = V$
Pick the maximum weighted edge (x,y)
$cut(S_1,S_2) = \omega_{xy}$
$V' = V' \backslash \{x,y\}$
$S_1 = \{x\};\ S_2 = \{y\}$
for $j = 1 : n-2$ **do**
  for $i \in V'$ **do**
    $score(i) = |\omega(i,S_1) - \omega(i,S_2)|$
  end for
  Choose the vertex i^* with the maximum score
  if $\omega(i,S_1) > \omega(i,S_2)$ **then**
    add i^* to S_2
  else
    add i^* to S_1
  end if
  $V' = V' \backslash \{i\}$
  $cut(S_1,S_2) = cut(S_1,S_2) + max\{\omega(i^*,S_1) - \omega(i^*,S_2)\}$
end for

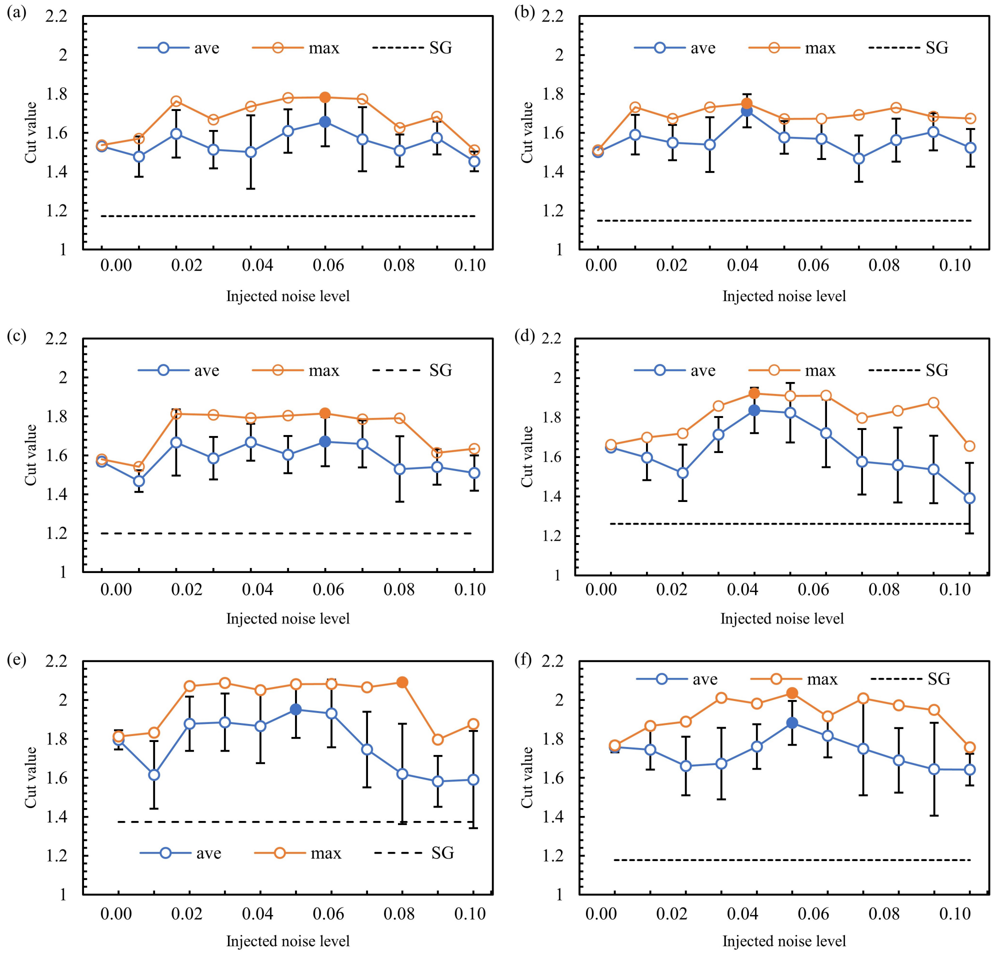

Fig. 2. Experimental results the Max-Cut problems with different density. (a)-(e) Experimental results for solving the Max-Cut problem with density= 0.5, 0.6, 0.7,0.8 0.9, 1.0 using noise-injected SPIM, compared with the SG algorithm. .

It should be mentioned that the variance of the noise needs to be normalized within the range of $[0, 1]$, otherwise the noise will overwhelm the original image. The noise-injected image intensity will replace the original image intensity as the cost function of the FLIP algorithm in the electrical threshold calculation, realizing a semi-random input process in the smoothed analysis theory. Moreover, to avoid trapping in local optima, we incorporate the concept of simulated annealing. During the experiment, inferior solutions are accepted with a modest probability, providing the possibility of jumping out of the local optimum to explore diverse regions of the solution space. The outcome of the soft judgment will determine the flip of the spins in the next iteration. And the noise vector is updated independently in each iteration.

3. The SG algorithm for solving the Max-Cut Problems

Algorithm 2 shows the SG algorithm for solving the Max-Cut Problems[7]. For obtaining the maximum cut value, the SG algorithm starts from the first vertex and divides all the vertices into two subsets one by one according to a certain order and rule. The division of vertices is based on the calculation of the score of each point, and there may be more than one point with the maximum score. A fully connected graph certainly increases the likelihood of multiple choices, and the SG algorithm will not continue to determine which of these edges or points satisfy the same condition better or worse but will simply choose the first vertex that satisfies the condition.

4. the optimal noise levels and the threshold for noise tolerance for various graph densities

We expand our experimental scope to address the weighted Max-Cut problems across a range of graph densities of [0.5,1.0]. Fig. 2 illustrates the effect of the injected noise on different Max-Cut Instances in experiments, respectively. And the optimal noise levels and the threshold for noise tolerance for different graph densities are summarized in Table 1. And in Fig. 3, we plot the performance gain of the

SPIM in solving Max-Cut problems with varying graph densities relative to the SG algorithm. The outcomes of the SG algorithm serve as standards by which to evaluate the effectiveness of SPIMs.

Table 1. **Optimal noise levels and tolerance thresholds of the SPIM for graph densities of $[0.5, 1.0]$. .**

Density	Optimal noise level	Noise tolerance threshold
0.5	0.06	0.04-0.07
0.6	0.04	<0.06
0.7	0.06	0.02-0.07
0.8	0.04	0.03-0.06
0.9	0.05	0.02-0.06
1.0	0.05	0.04-0.07

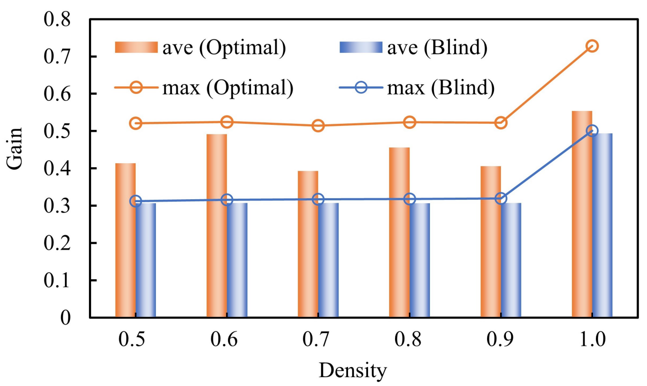

Fig. 3. **Performance gain of the SPIM compared to the SG algorithm under blind and optimal noise levels for graph densities of $[0.5, 1.0]$. .**

5. the two-stage optoelectronic co-optimization method

The smoothing effect alters the energy landscape of the original Ising model, and this effect exhibits a global orientation rather than being confined to specific regions. The smoothness level within the system can be regulated by adjusting the magnitude of the injected noise, which is beneficial during the initial stages of the search as it helps in avoiding premature convergence to local optima. However, the proportion of globally optimal solutions in the majority of combinatorial optimization problems is significantly low. Consequently, once the globally optimal solution is smoothed out by noise (perhaps even the spontaneous noise of a photonic Ising machine), it inevitably leads to a failure in the ground state search. This makes us realize that any noise is not expected in the later stages of the convergence. Therefore, we would like to add the exact search in the electric domain to correct the smoothing effect caused by spontaneous noise of the photonic Ising machine.

Flow 3 The two-stage optoelectronic co-optimization method

Input: An N-dimension Ising model with the interaction coefficient $J_{l,k}$, annealing coefficient λ

Output: Ground state x, Ground Hamiltonian H

procedure ONE(Fast annealing) ▷ SPIM
Random spin vector $x \leftarrow x_0 \in \{-1,1\}^N$, Hamiltonian $H \leftarrow H_0$, annealing temperature $T \leftarrow T_0$
  while $T > T_{end}$ **do**
    Randomly flip several spins;
    Update SLM;
    Detect the intensity I_{new} by CCD;
    Calculate H_{new};
    if $\|I_T - I_{new}\|_2 < \|I_T - I\|_2$ **then**
      $x \leftarrow x_{new}$; $H \leftarrow H_{new}$
    else
      if $rand(0,1) < e^{-\frac{H-H_{new}}{T}}$ **then**
        $x \leftarrow x_{new}$, $H \leftarrow H_{new}$
      end if
    end if $T \leftarrow \lambda * T$
  end while
Return x_{opto},H_{opto}
end procedure
procedure TWO(Exact search) ▷ CPU
$x \leftarrow x_{opto}$, $H \leftarrow H_{opto}$
  while $T > T_{end}$ **do**
    Randomly flip one spin;
    Calculate H_{new};
    if $H_{new} < H$ **then**
      $x \leftarrow x_{new}$; $H \leftarrow H_{new}$
    else
      if $rand(0,1) < e^{-\frac{H-H_{new}}{T}}$ **then**
        $x \leftarrow x_{new}$, $H \leftarrow H_{new}$
      end if
    end if $T \leftarrow \lambda * T$
  end while
Return x_{elec},H_{elec}
end procedure

The algorithm 3 shows the flow of the two-stage optoelectronic co-optimization method for solving combinatorial optimization problems. The two processes consist of the fast annealing stage and the exact search stage, done on the SPIM and CPU, respectively. A modified FLIP algorithm is applied in the SPIM, where we flip multiple spins in each iteration to speed up convergence[5]. Conversely, the exact search conducted within the CPU employs a single spin-flip strategy to identify superior solutions in the vicinity of the solution X_{opto} provided by the SPIM during the initial stage.

References

1. A. A. Schäffer, "Simple local search problems that are hard to solve," SIAM journal on Comput. **20**, 1 (1991).

2. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis, “How easy is local search?” J. computer system sciences **37**, 1 (1988).
3. X. Chen, C. Guo, E. V. Vlatakis-Gkaragkounis, Yannakakis, M., and Zhang, X., “Smoothed complexity of local max-cut and binary max-csp,” Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, (2020).
4. A. Bibak, C. Carlson, and K. Chandrasekaran, “Improving the smoothed complexity of flip for max cut problems,” ACM Transactions on Algorithms (TALG) **17**, 3 (2021).
5. X. Ye, W. Zhang, S. Wang, X. Yang, and Z. He, “20736-node weighted max-cut problem solving by quadrature photonic spatial ising machine,” SCIENCE CHINA Inf. Sci. **66**, 229301 (2023).
6. D. Pierangeli, G. Marcucci, D. Brunner, and C. Conti, “Noise-enhanced spatial-photonic Ising machine,” Nanophotonics **9**, 13 (2020).
7. S. Kahruman, E. Kolotoglu, S. Butenko, and I. V. Hicks, “On greedy construction heuristics for the max-cut problem,” Int. J. Comput. Sci. Eng. **3**, 3 (2007).