An Analog Neuro-Optimizer with Adaptable Annealing Based on 64×64 0T1R Crossbar Circuit

M. R. Mahmoodi[1*], H. Kim[1*], Z. Fahimi[1], H. Nili[1], L. Sedov[2], V. Polishchuk[2], and D. B. Strukov[1*]
[1] UC Santa Barbara, Santa Barbara, CA 93106-9560, U.S.A. [2] Linkoping Univeristy, 60174 Norrkoping, Sweden
[*]email: {mrmahmoodi, hyungjin, strukov}@ece.ucsb.edu

Abstract---We demonstrate an analog neuro-optimization hardware, suitable for solving a large number of combinatorial optimization problems, based on a crossbar circuit with 4096 passively-integrated analog-grade memristors. The proposed hardware supports a variety of metaheuristic techniques for improving optimization performance, such as stochastic and chaotic simulated annealing, and novel "exponential" annealing. The hardware operation is successfully tested by experimentally solving weighted graph partitioning, maximum clique, vertex cover, and independent set problems, and observing good agreement with simulation results.

I. Introduction

Combinatorial optimization has broad applications in various fields of science and technology [1,2] (Fig. 1a). One promising approach for solving optimization problems is by finding a solution for an equivalent Ising model, which was originally introduced as a mathematical model of ferromagnetism. Recently, Ising solvers based on CMOS [2], superconductor [4], nanomagnet [5], and photonic [1] technology have been investigated. The best optimization performance is typically obtained with the aid of metaheuristic techniques, such as stochastic simulated annealing (SSA) [7], chaotic simulated annealing (CSA) [8], which helps avoiding getting stuck in a local minimum (Fig. 1c).

Another approach for solving combinatorial optimization problems, closely related to Ising models, is to use generalized Hopfield neural networks [3] (Fig. 1b). With properly selected synaptic weights, the network converges (in the ideal case) to its minimum energy state, which corresponds to the solution of the optimization problem. Similar to Ising solvers, annealing techniques are commonly used to improve performance, e.g. by probabilistically updating neuron states.

The goal of this work is to develop energy-efficient, fast, and versatile (i.e., supporting different annealing techniques) hardware for solving optimization problems using neural network approach. Achieving this goal requires finding efficient solutions for (1) computing dot-product operations with dense adjustable weights, the most common operation in Hopfield neural network, and (2) implementing stochastic neurons to support metaheuristic techniques. The first requirement can be addressed by utilizing analog or mixed-signal circuits based on adjustable resistive switching devices (such as metal-oxide memristor) [9, 10]. Stochastic operation, and even stochastic annealing in few-device networks, was experimentally demonstrated with magnetic devices [11-13]. Unfortunately, none of these works seemed to have an efficient solution to address both challenges. The main contribution of this work is the development of such efficient neuro-optimization hardware based on passive analog-grade metal-oxide memristor circuits with the largest (to the best of our knowledge) complexity. We also propose novel "exponential" annealing (EA), which is especially suitable for the proposed hardware.

II. ReRAM Crossbar Circuits

Passive ("0T1R") memristors are very promising analog nonvolatile memories for implementing synapses in analog neuromorphic circuits due to their excellent density prospects. On the other hand, using such technology in analog computing applications is much more challenging compared to active ("1T1R") memories, due to much stricter requirement for devices' *I-V* uniformity, needed for precise adjustment of conductance states.

In this work, we utilize passive crossbar circuits with 4K analog-grade memristors (Fig. 2), based on etch-down pattering process [14]. The new fabrication process allowed for very uniform device characteristics, with ~99% device yield and ~26% switching threshold voltage variations in the whole crossbar (Fig. 4), which is sufficient for programming crosspoint device conductances with <3.5% average tuning precision (> ~5 bits of precision) [14]. The typical *I-V* characteristics for as-fabricated and formed crossbar devices are shown in Fig. 3. Crossbar circuits were packaged and integrated in the experimental setup, which was used for all measurements (Fig. 24). In all experiments, the crossbar output currents with hw-injected noise were measured experimentally, while the neuron functionality was emulated in the software.

III. Neuro-Optimization Hardware

Fig. 5 shows the main idea of the proposed mixed-signal discrete-time/state stochastic Hopfield neural network circuits which support SSA, CSA, and EA. The stochastic dot-product computation with adjustable annealing schedule is implemented by controlling the signal-to-noise ratio (SNR) of the read-out current. Specifically, the output referred current noise on each differential line ($\overline{\imath_{n,out}^2}$) is the sum of current noises generated by the memory cells and that of peripheral circuits. For the practical operational frequencies (>100 MHz), such noise is predominantly thermal. Comparator circuit of a neuron implements the step function on a sampled current so that the probability of latching a digital '1' value is $0.5(1 + \mathrm{erf}\,\langle I\rangle/\sqrt{2}\sigma)$, where σ is the standard deviation of the output

current. The error function matches closely, with relative error always within 2% across the whole range of normalized currents, probabilistic neuron transfer function 1/(1+exp[-y/T]) commonly used in stochastic Hopfield neural networks, in which y is the pre-activation value and T is the temperature.

Because $\overline{\imath_{n,out}^2}$ is independent of the applied voltages and is contributed by all differential line memristors and the sensing circuit, the effective temperature is inversely proportional to the peak signal-to-noise ratio, i.e. $SNR_{max} = I_{max}/\sigma$. This is verified in Fig. 6, which shows experimentally measured SNR_{max} for a 20-input/single-ended dot-product computation with randomly initialized (between 0 and 1) weights. In turn, such design enables very compact SSA implementation (Fig. 7), in which the temperature can be controlled by altering V_{ON} (and hence modulating I_{max} and SNR_{max}), which is the amplitude of the applied 'on' voltage to the inputs of the crossbar circuit during the operation (Fig. 5).

To implement CSA, the neuron self-feedback weights are initially set to some large values (as compared to other weights in the network), which results in chaotic dynamics of the network. These weights are then exponentially decreased to zero during the operation so that the network slowly transitions from chaotic to periodic regime, eventually settling in a stable equilibrium. The weight adjustment is performed again by scaling the applied voltages. Because the voltages cannot be scaled for only diagonal devices, without affecting other devices on the same columns, CSA approach requires doubling the number of columns, and setting all but diagonal weights to zero in the additional crossbar array. The neuron switching activity (firing rate) for SSA, CSA, and baseline (i.e., with no annealing) approaches was studied using 10-node weighted graph partitioning problem with randomly initialized weights. The experimental results show that the neuron firing rates were initially above 5% for both SSA and CSA, corroborating the random switching of neurons, but eventually reduced to zero when the network stabilized.

Furthermore, we propose novel exponential annealing (EA), a deterministic approach in which the initial weights are modified to ensure funnel-shape energy landscape of the network and quick convergence to the global optimum. Specifically, the weights are defined as $W_{ij}(t) = T_{ij}(1-\exp[-t/\tau])$, where t is epoch (i.e., one neuron update) number, τ is the annealing factor, and T_{ij} is the predetermined weight matrix corresponding the problem in question. The network weights are then slowly modified to the baseline ones, with the goal of always keeping the network in the ground state during the transition. Such behavior of EA approach is experimentally confirmed on 10-node partitioning problem (Fig. 9) using the same (as in Fig. 8 experiment) weights.

Fig. 10 shows the final average energy of the network for all studies techniques on the same problem, in particular showing >2× higher convergence rate to top 5 solutions for EA over naive baseline approach. The experimental results also confirm that the weight scaling error due to I-V nonlinearity (which could lead to varying relative weights when changing V_{ON}) is negligible.

VI. Solving Combinatorial Optimization Problems

To further demonstrate the effectiveness of the proposed hardware, we solved experimentally four common combinatorial optimization problems.

For the first, 5-node **weighted maximum-clique problems** we used 10 random configurations of graph weights, corresponding to 10 kΩ to 150 kΩ range, programmed with <5% tuning error. The annealing parameter was exponentially scaled from 1 to 0.01 for SSA and from 20 to 0.1 for CSA, while $\tau = 20$ for EA. All three annealing techniques performed much better than the baseline with good agreement between simulations and experiment (Figs. 12-14). The simulation results for larger number (200) of random configurations clearly indicate better EA performance.

Figs. 16-18 show the results for the **weighted vertex cover problem**. The performance of annealing approaches, measured on various size graph problems, were always better compared to the baseline. The simulation results for various sizes problems with 200 random configurations further indicate superior EA performance.

The impact of annealing schedule was studied by solving 10-node **independent set problem**. The results are presented in Fig. 19 which shows a significant improvement for the average energy when using slower annealing.

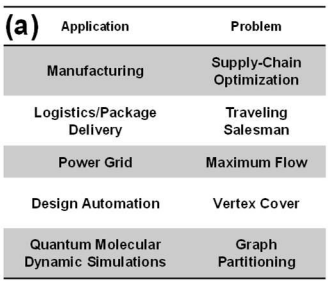
Finally, we considered solving **graph partitioning optimization problem** for a fully-connected weighted 6-node graph (Fig. 20a), with the corresponding baseline synaptic conductances in the hardware implementation shown in Fig. 20b. As discussed earlier, the synaptic weights are first increased for EA approach, by voltage scaling (Fig. 21). Fig. 22 shows the final experimental results and its comparison to simulations, further validating the proposed hardware functionality, and >20% improvements in the success rate over the baseline approach.

V. Summary

In summary, we first proposed and experimentally verified an efficient implementation of stochastic dot-product circuits, with adjustable temperature, based on 64×64 crossbar circuits with passively integrated analog-grade metal-oxide memristors. We then used such circuits to implement stochastic and chaotic simulated annealing, and a novel exponential annealing. The performance of all techniques was experimentally verified by solving four common combinatorial optimization problems. The experimental results showed improved performance for all annealing techniques over the baseline approach, and superior performance of exponential annealing compared to stochastic and chaotic annealing.

References

[1] T. Inagaki et al. *Science* **354** 6312 (2016). [2] T. Takemoto et al. *ISSCC* 2.6A (2019). [3] J. Hopfield et al. *PNAS* **81** 10 (1984). [4] S. Boixo et al. *Nat. Phys.* **10** 3 (2014). [5] P. Debashis et al. *IEDM* 34 (2016). [6] M. Yamaoka et al. *JSSC* **51** 1 (2015). [7] S. Kirkpatrick et al. *Science* **220** 4598 (1983). [8] L. Chen et al. *Neural Networks* **8** 6 (1995). [9] M. Prezioso et al. *Nat. Commun.* **9** 1 (2018). [10] S. Kumar et al. *Nature* **548** 7667 (2017). [11] J. H. Shin et al. *IEDM* 3 (2018). [12] B. Sutton et al. *Sci. Rep.* 7 (2017). [13] V. Ostwal et al. *Sci. Rep.* **8** 1 (2018). [14] H. Kim et al. arXiv (2019).



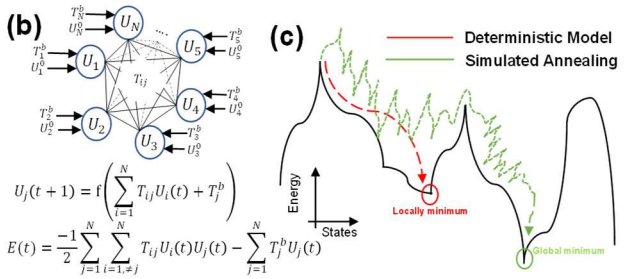


Fig. 1: Neuro-optimization with annealing: (a) Applications of combinatorial optimization; (b) Energy-based Hopfield model for solving optimization problems; (c) the conceptual diagram showing how simulated annealing avoids getting stuck at local minima of the network's energy function.

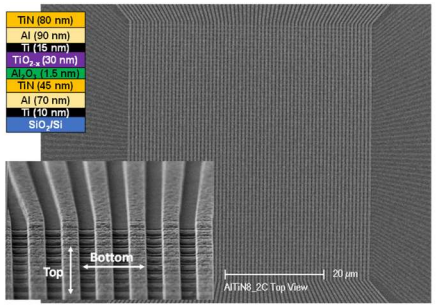


Fig. 2: SEM image of the fabricated 4096 integrated 0T1R memristor array. Th top inset shows the device cross-section and the bottom inset is the zoom-in on the portion of the crossbar.

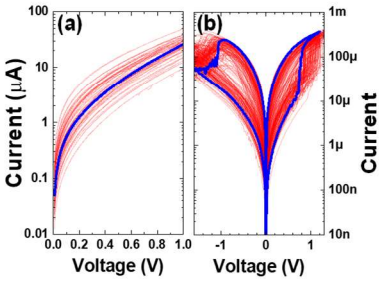


Fig. 3: The I-V characteristics of (a) as-fabricated devices and (b) after forming, measured with quasi-DC sweep.

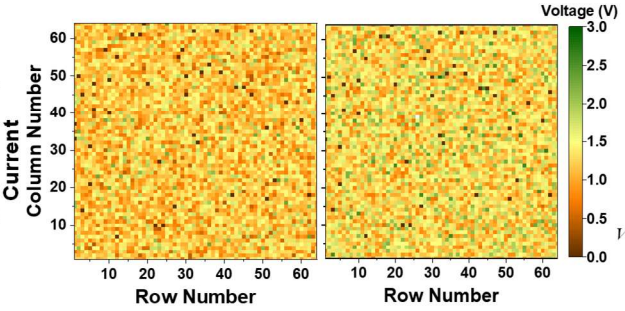


Fig. 4: Measured absolute (a) set and (b) reset switching threshold voltage (smallest applied voltage to alter the measured conductance at 0.25 V by 20%) map of a 64×64 0T1R memristor crossbar.

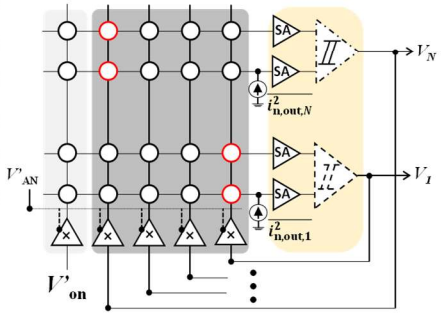


Fig. 5: Circuit diagram of the versatile neuro-optimizer. Two sense amplifiers and a comparator represent a neuron.

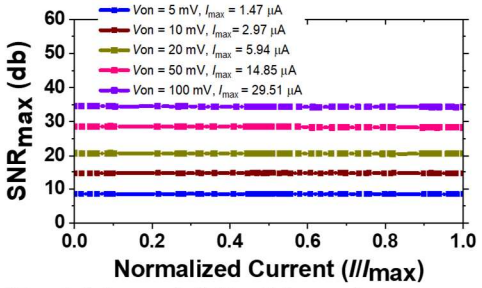


Fig. 6: Measured SNR of the readout current for various values of V_{on}.

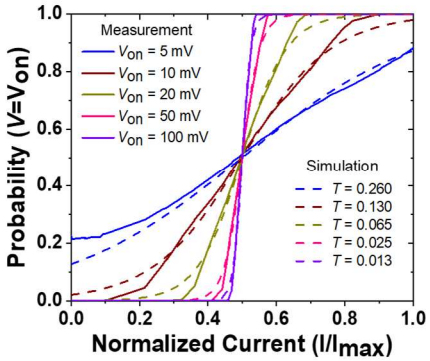


Fig. 7: Measured probabilistic sigmoid function using the intrinsic noise of the circuit versus simulation of ideal sigmoid.

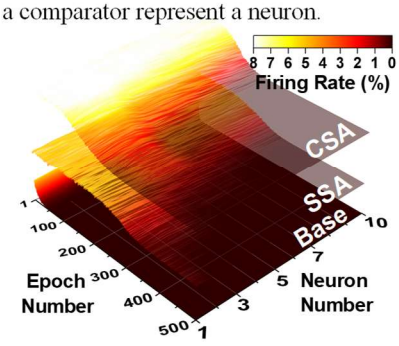


Fig. 8: The firing rate of a 10-node graph partitioning problem (with uniformly distributed weights in the range of 1-20) for different techniques.

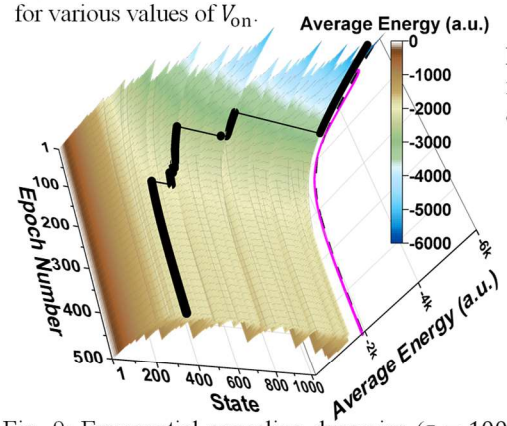


Fig. 9: Exponential annealing dynamics ($\tau = 100$) in solving the 10-node weighted graph partitioning problem. The network finds the ground state at earlier epochs and tracks it while transitioning to the baseline. Black: the transitory ground state of the system, magenta: average energy over 50k cases.

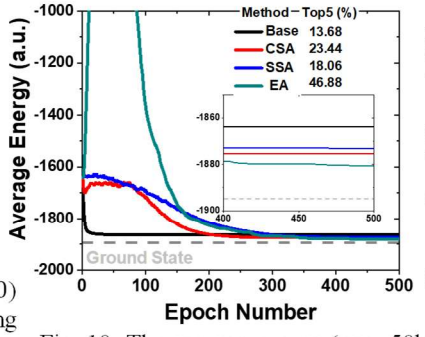


Fig. 10: The average energy (over 50k cases) and top-5 successrate of the neuro-optimizer. Inset shows the zoomed-in view of the last 100 epochs.

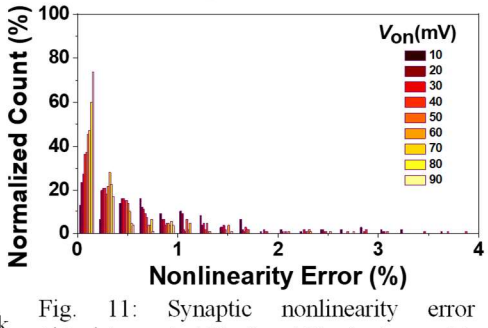


Fig. 11: Synaptic nonlinearity error ($|\Delta w|/w_{\max}$)×100 for 110 devices with random weights during SSA. Owing to the linear I-V characteristics, the nonlinearity becomes less of a concern.

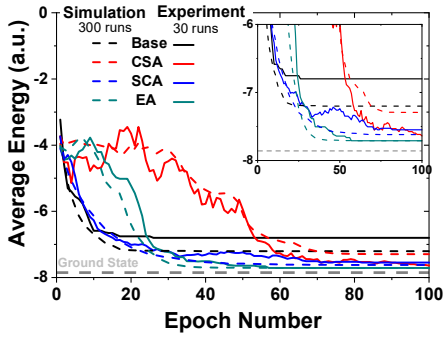


Fig. 12: The measured evolution of the average energy for a 5-node maximum-weighted clique problem. The inset shows the zoomed-in view of the figure.

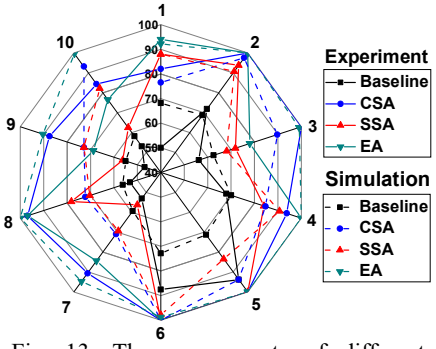


Fig. 13: The success rate of different annealing techniques on 10 5-node weighted maximum clique problems.

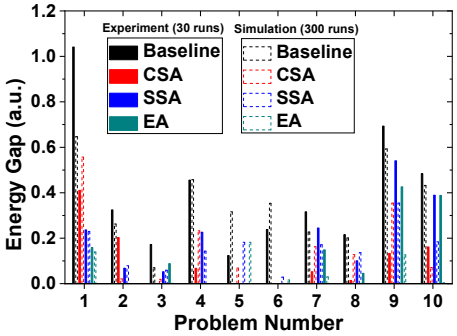


Fig. 14: The energy gap (the average energy of solution minus the ground state) for all 5-node weighted maximum clique problems.

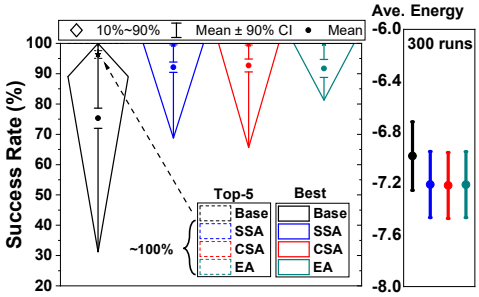


Fig. 15: Simulation results of 200 randomly chosen weighted maximum clique problems.

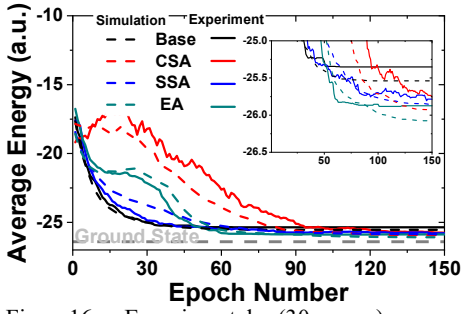


Fig. 16: Experimental (30 runs) versus simulation (300 runs) results of solving a 12-node minimum-weighted vertex cover. Inset is the zoomed-in view.

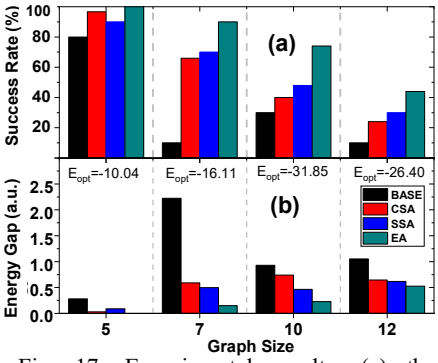


Fig. 17: Experimental results: (a) the success rate and (b) the energy gap on various sizes of minimum-weighted vertex cover problem.

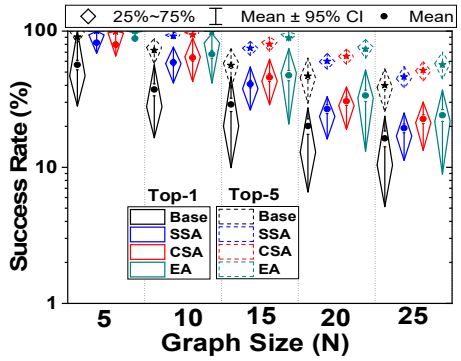


Fig. 18: Statistics of the neuro-optimizer solving minimum-weight vertex cover problem of different sizes. For each size, 200 random weighted graphs are considered.

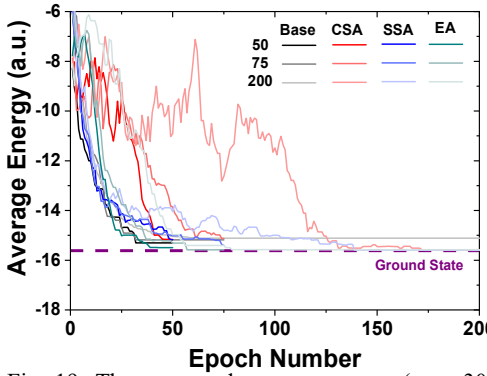


Fig. 19: The measured average energy (over 30 runs) when solving a 10-node maximum-weight independent set problem with three different annealing schedules.

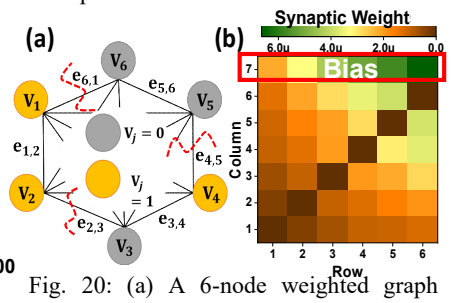


Fig. 20: (a) A 6-node weighted graph partitioning problem and (b) its corresponding synaptic weights mapped to the neuro-optimizer to implement EA.

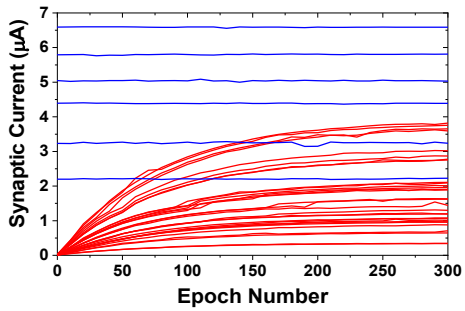


Fig. 21: Synaptic weight evolution during the exponential annealing.

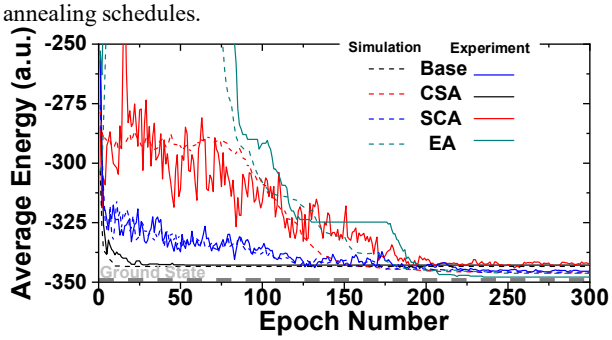


Fig. 22: The average energy (300 and 30 cases for simulation and experiment, respectively) of the neuro-optimizer when solving the 10-node graph partitioning problem.

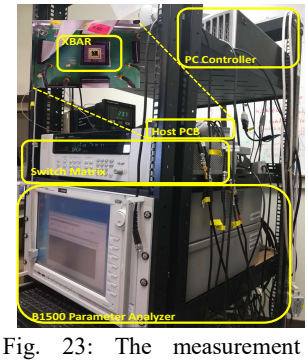


Fig. 23: The measurement setup of the RRAM-based neuro-optimizer.