2020 International Conference on Rebooting Computing (ICRC) |978-1-6654-1975-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICRC2020.2020.00016

Classical Adiabatic Annealing in Memristor Hopfield Neural Networks for Combinatorial Optimization

Suhas Kumar, Thomas Van Vaerenbergh, John Paul Strachan
Hewlett Packard Labs, Palo Alto, CA, USA
Emails: (S.K.) suhas.kumar@hpe.com, (T.V.V.) thomas.van-vaerenbergh@hpe.com, (J.P.S.) john-paul.strachan@hpe.com

***Abstract*— There is an intense search for supplements to digital computer processors to solve computationally hard problems, such as gene sequencing. Quantum computing has gained popularity in this search, which exploits quantum tunneling to achieve adiabatic annealing. However, quantum annealing requires very low temperatures and precise control, which lead to unreasonably high costs. Here we show via simulations, alongside experimental instantiations, that computational advantages qualitatively similar to those gained by quantum annealing can be achieved at room temperature in classical systems by using a memristor Hopfield neural network to solve computationally hard problems.**

Keywords—memristor, Hopfield network, optimization, NP-hard, adiabatic annealing

I. Introduction

The race to invent new ways of computing to address the imminent end of Moore's law has led to intense research into electronic accelerators to supplement high-performance digital processors. [1] Among the multitude of workloads addressed by special-purpose accelerators, combinatorial optimization remains the most challenging but is also not heavily explored. Combinatorial optimization appears in many enterprise-level applications including airline scheduling, supply chain optimization, real-time bandwidth management, gene sequencing, to name a few. [2]

Quantum computing attempts to address such problems by leveraging quantum tunneling within the complex and non-convex cost function of the problem to find a globally minimum cost or the optimal solution, while classical electronic schemes have utilized simulated thermal annealing to achieve similar results. [3-6]

Here we simulate and experimentally implement analog memristor crossbar Hopfield networks [7] to realize adiabatic annealing by gradually changing the problem instance programmed onto the hardware, thereby allowing a new problem to be solved by using the solution of previous problems of comparable size and difficulty as the initial state. By combining simulated annealing, that harnesses intrinsic circuit noise, with adiabatic annealing, we demonstrate a clear increase in success probabilities (probability of reaching an optimal solution) by using adiabatic annealing over pure simulated annealing. Thus, adiabatic annealing is a viable approach not only in quantum computing, but also in classical computing, while also benefitting from the much higher maturity and energy efficiency, such as avoiding cryogenic cooling, in today's electronic technologies (Figs. 1a-1c). [3-6]

II. Simulation and Experimental Setup

The Hopfield neural network implements recurrent vector-matrix multiplications to iterate on candidate solutions. The update mechanism can be represented by:

$$u_i = \sum_{j \neq i} W_{ij} v_j, \quad v_i = \begin{cases} +1 \text{ if } (u_i + \eta_i) \geq \theta_i \\ -1 \text{ if } (u_i + \eta_i) < \theta_i \end{cases}, \qquad (1)$$

where v is the state of the neuron that holds the candidate solutions, u is the weighted feedback provided to the next time step of the recurrent vector matrix multiplication, W is the zero-diagonal symmetric weights matrix representing the problem, θ is a binary threshold implementing the nonlinear feedback to the recurrent network, and η represents perturbations, both intrinsic and injected.

The Hopfield network described above implements minimization via gradient descent in exploring the cost function or 'energy' (E) associated with the problem, where the global minimum of the energy corresponds to the optimal solution to the problem.

$$E = -\frac{1}{2}\sum_{i,j}^{N} W_{ij} v_j v_i + \sum_{i}^{N} \theta_i v_i \qquad (2)$$

Although the Hopfield network implements gradient descent, most weight matrices, especially those corresponding to practical problems, yield a highly non-convex energy landscape, finding the global minimum (optimal solution) of which is a computationally hard problem on its own. This drawback has prevented Hopfield networks from commercial adoption. However, recent work has shown that multiple sources of perturbations, including intrinsic circuit noise and external chaotic dynamics, can be tunably injected to dislodge the system from local minima (bad solutions), a process related to simulated annealing. Such systems have been projected to outperform both prevalent general-purpose computing processors and application-specific chips by several orders of magnitude. [7]

Hopfield networks can encode any problem with an Ising representation; in other words, problems that can be represented using a weight matrix. Ising representations of most constrained optimization problems convert constraints to objectives, which together with the original objective are subject to gradient descent on the energy landscape. Here we chose the maximum cut graph partitioning problem, which is a generic version of problems such as viral quasispecies reconstruction, genome contig orientation, VLSI routing optimization, etc. The problem has a simple Ising formulation:

$$\text{Max cut} = \max \frac{1}{2}\sum_{i<j} a_{ij}\left(1 - x_i x_j\right), \quad x_i = \begin{cases} +1 \; i \in S \\ -1 \; i \in \bar{S} \end{cases} \qquad (3)$$

978-1-6654-1975-8/20/$31.00 ©2020 IEEE
DOI 10.1109/ICRC2020.2020.00016

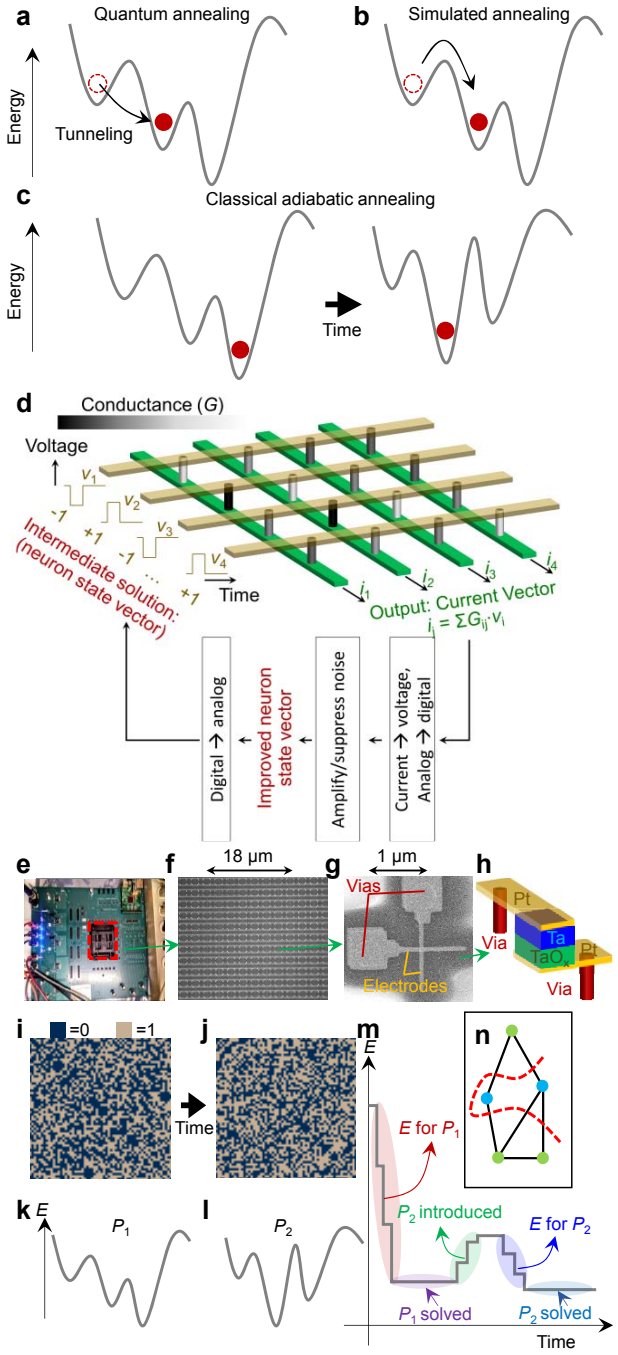

Figure 1: Illustration of classical adiabatic annealing. Schematic illustration of (a) quantum annealing, (b) simulated annealing and (c) classical adiabatic annealing. (d) Illustration of the memristor Hopfield network. (e-h) Optical and electron micrographs and illustration of the experimental setup, the chip and the memristor device. (i-j) Two 60x60 weight matrices used in the simulations. (k-l) Illustrative energy landscapes corresponding to any two problems being solved by adiabatic annealing. (m) Illustration of energy minimization during adiabatic annealing. (n) A graph problem and its maximum cut solution (dashed curve, and two subsets of nodes in different colors).

where a is the weight matrix representing the edges of the graph and x is the solution vector that cuts the graph into two subsets S and its complement. From (3), it can be deduced that w=-a, meaning that transforming the graph definition to a weight matrix is fairly trivial for the maximum cut problem, thereby allowing any random zero-diagonal symmetric matrix to represent a valid maximum cut problem. This versatility has led to many neuromorphic hardware systems to utilize graph partitioning problems to benchmark their performance. Details on the foregoing discussions can be found elsewhere. [7-8] Fig. 1d illustrated the memristor Hopfield network, while Figs. 1e-1h depict the experimental system.

Quantum computers, especially those that utilize quantum adiabatic annealing, also specifically target certain types of optimization problems, such as prime factorization. The key ingredient that enables quantum adiabatic annealing is quantum tunneling. This process usually starts with a relatively simple problem and gradually transforms it into a hard problem, thereby introducing multiple local minima. The idea behind quantum annealing is to let the system tunnel into the lower energy minima, thereby effectively staying in the global minimum during the entire transformation period. The main issues with this process are: (a) very low temperatures are required to maintain quantum coherence and enable tunneling and (b) the degree of tunneling is not tunable, so the physical process which achieves the solution cannot be easily tuned. [3-5]

It has been unclear if the advantages of such an adiabatic annealing process can be realized in a classical system. Since Hopfield networks naturally implement gradient descent, they are model systems to study the feasibility of room-temperature classical adiabatic annealing. Here we perform calculations and perform circuit simulations of memristor Hopfield networks by implementing adiabatic annealing via a gradual changing of the problem definition. Every time the problem is changed, the solution corresponding to the previous problem is used as the initial condition for the new problem. The eventual aim is to evaluate if solving two (or more) problems via adiabatic annealing provides a higher probability of solving them directly (starting from a randomized initial condition). In other words, if classical adiabatic annealing is advantageous, it would obviate the need to start with random initial conditions for every problem. An illustration of this process is provided in Figs. 1i-1m. After P_1 is solved via gradient descent, P_2 is gradually introduced over time while P_1 is gradually erased, which will lead to an increase in energy in typical cases since the solution of P_1 does not correspond to the solution of P_2. Following this process, the system solves P_2 by gradient descent. Introduction of P_2 follows:

$$P = \frac{1}{t_{ann}-t_0}[P_1(t_{ann}-t_k) + P_2(t_k-t_0)]$$

$$\text{for } t_0 < t_k < t_{ann} \qquad (4)$$

where P is the effective problem at the k^{th} time step t_k, t_0 is the time step at the start of annealing, and t_{ann} is the annealing end time.

Alongside adiabatic annealing, we also included simulated annealing to maximize the probability of finding optimal

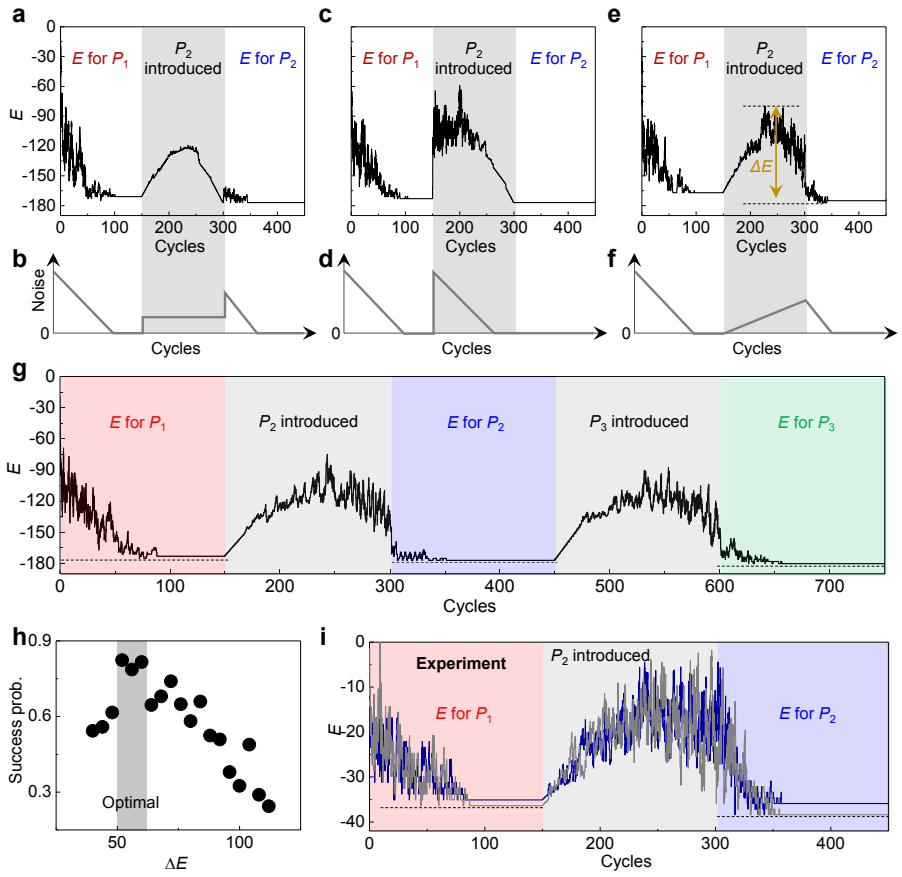

Figure 2: Adiabatic annealing schemes and examples. (a) Simulation of energy minimization during one of the possible adiabatic annealing schemes with the corresponding noise profile illustrated in (b). (c-f) Two other annealing schemes similar to (a-b). (g) Similar to (e), but solving three problems instead of two. (h) Average success probability of solving P_1 and P_2 plotted against ΔE. (i) Experimental result on two 30x30 matrices, using annealing scheme similar to (e-f).

solutions. In order to implement simulated annealing, we utilized a hysteretic threshold function that can either generate or suppress perturbations to a system. The scheme for simulated annealing (degree of perturbation, number of cycles of solution, etc.) was optimized on a set of calibration problems similar to those that are being solved (in size, density and quantity) and utilized throughout the study.

III. Results and Discussions

We introduce three classical adiabatic annealing schemes (Figs. 2a-2f), which involve different schemes of incorporation of simulated annealing.

In the first scheme (Figs. 2a-2b), as P_2 is introduced, we hold a small magnitude of constant perturbation, and upon the completion of the introduction of P_2, we introduce a slightly higher magnitude of perturbation, which is then annealed. In the second scheme (Figs. 2c-2d), upon the beginning of the introduction of P_2, we introduce a large magnitude of perturbations, which is then annealed roughly within the same time it takes to complete the introduction of P_2. In the third scheme (Figs. 2e-2f), during the introduction of P_2, we increase the magnitude of perturbations, and following the completion of the introduction of P_2, we anneal the system by reducing the perturbations over time.

The abrupt increase in the magnitude of perturbations in the first two schemes (Figs. 2a-2d) led to severe dislodging of the system form its low energy state, which adversely affected the system's ability to achieve an optimal solution within a given number of cycles. Thus, the gradual increase in the magnitude of perturbations in the third scheme (Figs. 2e-2f) ensured that the system maintained a higher quality of solution (supporting detailed data is not displayed here). In Fig. 2g, we display an example of adiabatic annealing similar to that in Fig. 2e, but with three problems instead of two.

The survey of different schemes above was meant to illustrate the different ways of performing adiabatic annealing in memristor Hopfield networks, with the different tradeoffs that need to be considered. While it was not meant to be a comprehensive investigation of different schemes and their merits, we are able to utilize one of them to show the feasibility of classical adiabatic annealing. For benchmarking of performance, we compare energy minimization corresponding to solving the two problems separately (starting from randomized initial states), along with solving both of them using the third scheme of adiabatic annealing. Since keeping the overall energy as low as possible likely enables higher quality solutions in subsequently introduced problems, we study the average success probability of solving both the problems as a function of the increase in energy during annealing, ΔE (Figs. 2e, 2h). While it is apparent that a very large ΔE leads to worse solutions, a very low ΔE also deteriorates the solution quality; because forcing ΔE to be minimum prevents the system from exploring its global energy landscape. Here it is apparent that the lowest energy is not maintained throughout the process, which is one of the differences between classical and quantum adiabatic annealing. Fig. 2i displays experimental data of solving two problems represented by 30x30 matrices using the scheme of Figs. 2e-2f. The system landing in the global minima (dashed horizontal lines in Fig. 2i) shows the experimental feasibility of the approach. While full experimental benchmarking is deferred to a future work, here we perform simulations of performance benchmarking with initial experimental feasibility established by this result.

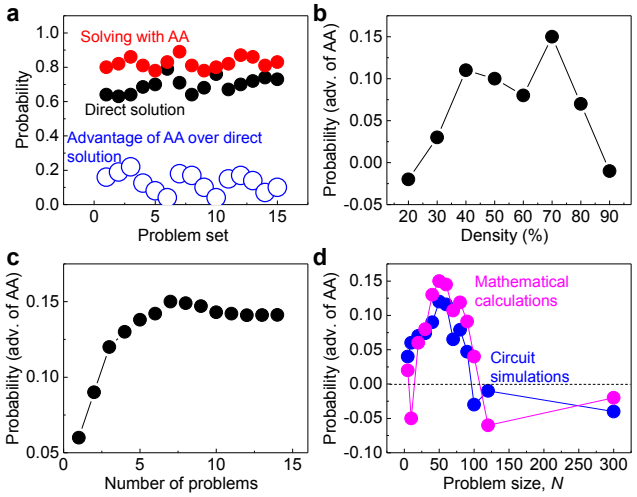

Figure 3: Performance benchmarking of adiabatic annealing. (a) Success probabilities of solving 30 sets of randomly generated three problems using adiabatic annealing or AA (solid red circles) and direct solving (solid black circles), along with their difference indicating the relative advantage of AA. Relative advantage of AA plotted against (b) density of weight matrices, (c) number of problems in the sequence of AA, and (d) problem size N (of NxN weight matrix). (d) also contains circuit simulations.

In order to compare the performance of adiabatic annealing over directly solving the problems, we repeated the exercise illustrated in Figs. 2e-2f over 30 randomly generated sets of three problems each (with size N=60, density of ~50%, solved over 750 cycles).

The average success probabilities of solving all three problems within each set was clearly higher with adiabatic annealing (Fig. 3a), with the advantage being up to ~20%. We also studied the advantage of adiabatic annealing upon varying the density of the problems (percentage of edges in the graph, with 100% being an all-to-all connectivity) (Fig. 3b), which was performed by repeating the exercise in Fig. 3a over problem sets of different densities.

It is apparent that adiabatic annealing exhibits an advantage at intermediate densities, whereas at very low densities the advantage is lost, likely owing to the problems being sufficiently easy to be solved to high degrees of success by direct solutions alone. Similarly, with very dense problems, there are often multiple optimal solutions equal in energy, thereby enabling direct solutions to be highly efficient and obviating the need for adiabatic annealing. Further, we studied the effect of adding more problems on the average success probability of solving all of them (repeated on 15 different sets of problems). The advantage of adiabatic annealing is clearly revealed to settle down at about 14% compared to directly solving the problems. This quantity could be different depending on the nature of the problems, optimization of the simulated annealing scheme, etc. However, this data is clear evidence of the feasibility of classical adiabatic annealing. Further, the advantage of simulated annealing studied for problems of different sizes (by using the same number of cycles for all sizes) revealed a diminished advantage for very small problems (Fig. 3d). This effect is intuitively apparent because very small problems are relatively trivial and can be solved very efficiently by direct means, especially when direct solution is allowed to operate over a large number of cycles. Also, very large problems too exhibited a diminished advantage of adiabatic annealing. It is likely that the diminishing of the advantage of adiabatic annealing at small and large problem sizes is due to the non-optimal choice of annealing schemes (number of cycles, perturbation processes, etc.), since there is nothing fundamentally different between problems of different sizes if the solution scheme is optimized or scaled for the relevant situations. All of the simulated data presented so far were based on mathematical calculations (by ignoring the effects of circuit non-idealities). This was done intentionally to eliminate any possible circuit issues altering the demonstration of the advantage of adiabatic annealing. We were able to reproduce all the observations so far using circuit simulations as well. To provide an illustration of the degree of discrepancy between calculations and circuit simulations, we plot the results form circuit simulations in Fig. 3d, which exhibits reasonable agreement with calculations.

IV. Conclusions

We show clear feasibility of classical adiabatic annealing, qualitatively similar in function to quantum adiabatic annealing, which is a key compute scheme in quantum computing. Specifically, classical adiabatic annealing provides the advantage of not needing expensive cooling infrastructure that quantum technologies require, and more importantly, classical adiabatic annealing provides control over the degree of annealing. Although our results show modest improvements corresponding to the use of classical adiabatic annealing, it highlights that the memristor Hopfield network is a flexible platform to implement a variety of physics-inspired meta-heuristics to solve different types of problems.

[1] W. Zhang et al., "Neuro-inspired computing chips," Nature Electronics 3, 371 (2020)

[2] B. Korte, J. Vygen, Combinatorial Optimization, Springer (2012).

[3] G.E. Santoro, E. Tosatti, "Optimization using quantum mechanics: quantum annealing through adiabatic evolution," Journal of Physics A: Mathematical and General, 39, 393 (2006)

[4] A. Das, K. Bikas, Quantum annealing and related optimization methods. Springer Science & Business Media (2005)

[5] C.C. McGeoch, "Adiabatic quantum computation and quantum annealing: Theory and practice." Synthesis Lectures on Quantum Computing, 5.2, 1 (2014)

[6] M. R. Mahmoodi et al., "Versatile stochastic dot product circuits based on nonvolatile memories for high performance neurocomputing and neurooptimization," Nature Communications, 10, 5113 (2019)

[7] F. Cai et al., "Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks," Nature Electronics, 3, 409 (2020)

[8] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics, 12, 2 (2014)