

Efficient combinatorial optimization by quantum-inspired parallel annealing in analogue memristor crossbar

Mingrui Jiang, Keyi Shan, Chengping He and Can Li*

Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR

*Email: canl@hku.hk

Contents

1 Supplementary Figures	2
2 Supplementary Table	13
3 Supplementary Note	14
4 Supplementary References	15

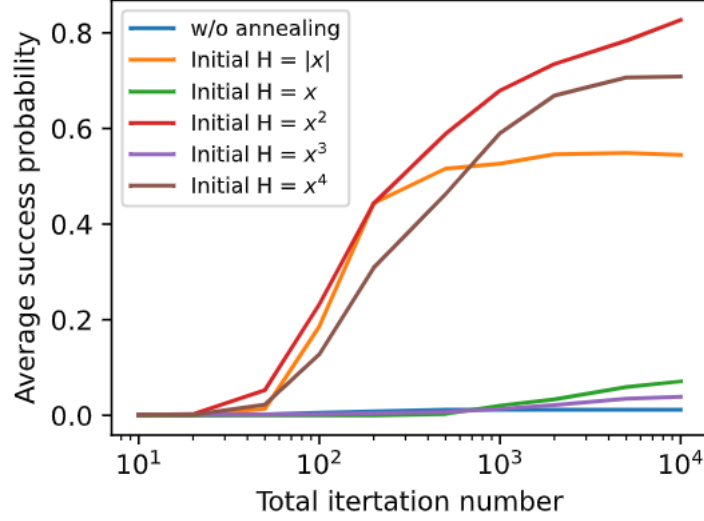


Figure S1. The impact of different initial Hamiltonians. This plot shows the average success probability of different $H_{initial}$ functions with different total iteration numbers. The curves with different color represent the case without $H_{initial}$, $H_{initial} = \sum_{i=1}^N |x_i|$, $H_{initial} = \sum_{i=1}^N x_i$, $H_{initial} = \sum_{i=1}^N x_i^2$, $H_{initial} = \sum_{i=1}^N x_i^3$ and $H_{initial} = \sum_{i=1}^N x_i^4$, respectively. For each data point, 20 60-node 50%-density unweighted Max-Cut problem instances are solved. For each problem instance, 100 trials from random initial states are conducted for calculating the success probability. The data in the plot shows the average success probability over 20 problem instances. From the plot, it can be observed that symmetric functions including $\sum_{i=1}^N |x_i|$, $\sum_{i=1}^N x_i^2$, $\sum_{i=1}^N x_i^4$, perform better than asymmetric ones. Since they all have an easily found ground state obtained when all $x_i = 0$. The success implementation of $\sum_{i=1}^N |x_i|$, $\sum_{i=1}^N x_i^2$ and $\sum_{i=1}^N x_i^4$ also implies that an arbitrary function can service as the initial Hamiltonian as long as its ground state can be easily solved. According to our empirical observation, among these three functions, $\sum_{i=1}^N x_i^2$ works the best, so it is chosen as the $H_{initial}$ throughout the paper. Nonetheless, it is possible that there are other well-designed functions that can enhance performance, which requires further exploration.

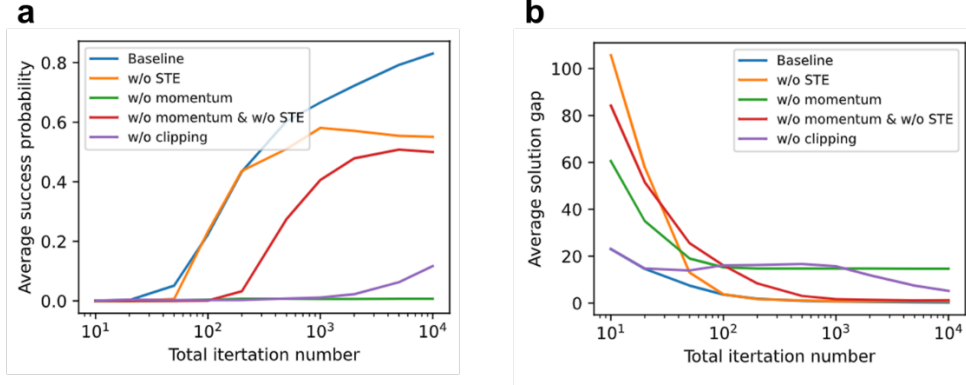


Figure S2. Comparison of different updating techniques in unweighted Max-Cut problem. a Average success probability vs. total iteration number. **b** Average solution gap (defined as the difference between the solved cut value and the maximum cut value) vs. total iteration number. Five different updating techniques are compared: the original updating algorithm used in the paper (denoted as Baseline in the figure), one without discretizing the proxy spin value when calculating the gradient (STE), one without adding a momentum term, one without momentum and without STE, and one without clipping the proxy spin value to be between -1 and 1. For each data point in the plot, 20 60-node 50%-density unweighted Max-Cut problem instances are solved. For each problem instance, 100 trials from random initial states are conducted. From the plot, it can be seen that all three techniques, STE, momentum, and clipping, are essential to the performance.

The performance of the case without STE works as well as with STE when the total iteration number is small. However, with larger total iterations, the performance decreases. This is because without STE, during the whole solving process, the real binary spin is not involved, and the analog spin proxy cannot accurately reflect the result of the binary spin, which causes a mismatch [1]. Further updating on the analog spin proxy lacks the ability to lower the energy calculated by the binary spin. For the case without momentum, the system works like a greedy gradient descent algorithm, and the annealing fails. For this case, a local-minimum solution can be easily obtained, however, the global minimum can hardly be found. The role of momentum in the quantum-inspired annealing is not clear yet, but this might be related to the property of STE. Since the performance of vanilla stochastic gradient descent is much poorer than those ones with momentum in the field of binary neural network (BiNN) training [2], similar with our observations. For the case without momentum and without STE, the annealing backs to work normally. The convergence speed is relatively slower than the case without momentum. For the case without momentum and without STE, the annealing works normally, but the convergence speed is relatively slower than the case without momentum. For the case without clipping, the analog spin proxy does not have any limitation, and it can grow infinitely towards one direction in the solving process. Then if it needs to go towards the other direction in the later solving process, it takes much longer time to accumulate the gradient until it can change its sign and thus largely decreases the solving speed. For STE, momentum, and clipping, they are all common techniques used in BiNN training. In fact, there are more advanced optimizers, such as Adam, and other sophisticated training methods that are widely applied in the field of neural network [3, 4]. These techniques have the potential to be applied to combinatorial optimization, but their applicability needs to be further researched due to the different nature of the target problem.

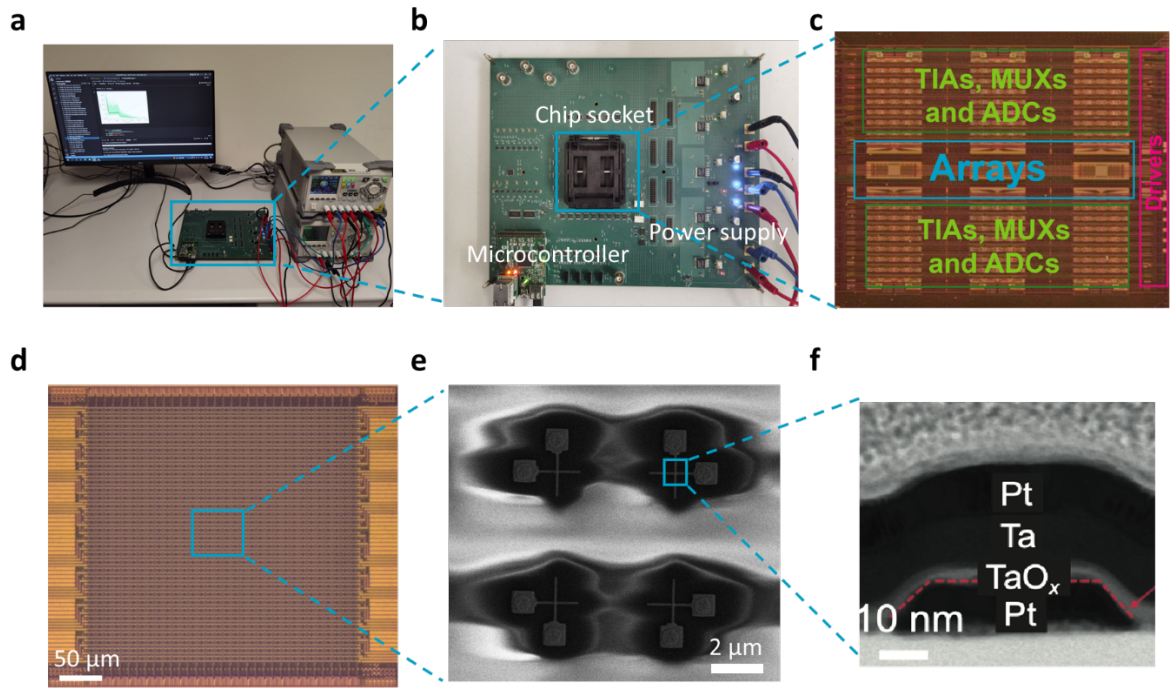


Figure S3. Experiment set-up and integrated memristor chip. **a** The overall experiment set-up. A PC is used to control the integrated memristor chip via microcontroller. **b** The printed circuit board (PCB) developed to measure the chip. **c** Optical image of the integrated memristor chip, which consists of three 64×64 1T1M memristor arrays and peripheral circuits including transimpedance amplifiers (TIAs), multiplexers (MUXs), analog-to-digital converters (ADCs) and drivers. CMOS peripheral circuits were taped out by TSMC's 180 nm technology, while the memristor devices were fabricated at backend of line with in-house cleanroom. **d** Optical image of the memristor crossbar array. **e** SEM image of the cross-point memristor device **f** Lateral TEM image of the $50 \text{ nm} \times 50 \text{ nm}$ Pt/Ta/TaO_x/Pt memristor device.

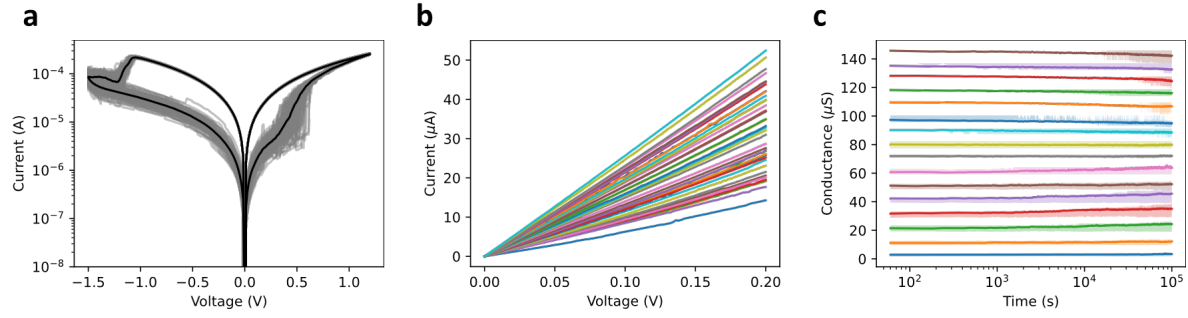


Figure S4. Performance of memristor device in our integrated hardware. **a** DC I-V measurement of 100 Set and Reset cycles demonstrating small cycle-to-cycle variations. The average value is shown by the dark line. **b** Linearity test of different conductance states, demonstrating excellent I-V linearity. **c** Retention test after dividing the entire 64×64 array into 16 subarrays, with each subarray programmed to target a conductance of 0 to 150 μ S with a step size of 10 μ S. The mean value is shown by the dark line, while the shaded area indicates the interquartile range.

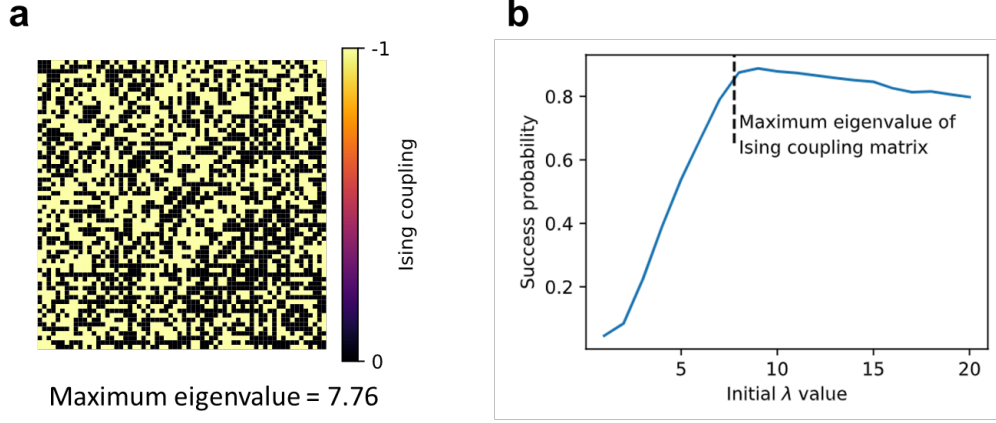


Figure S5. The impact of initial λ value and its relationship to maximum eigenvalue of Ising coupling matrix. **a** Ising coupling matrix. The maximum eigenvalue is 7.76. **b** Success probability of the problem instance in (a) vs. initial λ value. The black dash line shows the position of initial λ value equal to the maximum eigenvalue of Ising coupling matrix.

In our QPA, the coefficient of the initial Hamiltonian should start with a large enough that the system can be initiated with a function that the ground state can be easily found, i.e. usually a convex function. Since in our updating scheme, the gradient calculated by the binary spin σ_i is used as an estimator to update the analog proxy spin x_i . By replacing binary spin configuration σ in the Ising Hamiltonian with analog spin configuration $\mathbf{x}$, we have:

$$H_{system} = \frac{1}{2}\lambda \sum_i x_i^2 - \frac{1}{2} \sum_{i,j} J_{ij} x_i x_j = \frac{1}{2} (\lambda \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{J} \mathbf{x}) = \frac{1}{2} \mathbf{x}^T (\lambda \mathbf{I} - \mathbf{J}) \mathbf{x}$$

In this case, if $\lambda \mathbf{I} - \mathbf{J} \geq \mathbf{0}$, the H_{system} becomes convex and has a ground state when all $x_i = 0$. Therefore, to guarantee a convex function, λ should start with a value that is larger than the maximum eigenvalue of Ising coupling matrix $\mathbf{J}$. However, it cannot be too large, as with further larger initial λ , more iterations will be wasted and thus affect the performance. From the above plot, it can be seen that the best performance point locates where initial λ value is a bit larger than the maximum eigenvalue of Ising coupling matrix, which further prove our theory.

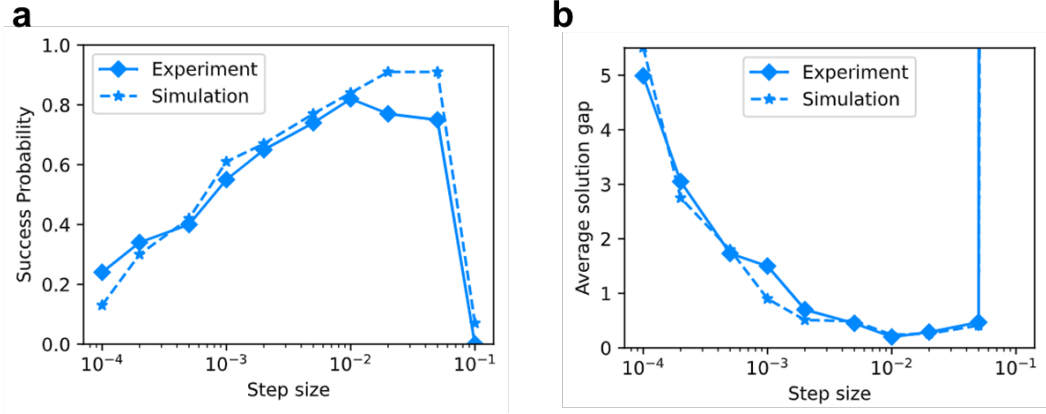


Figure S6. The impact of step size. **a** Success probability and **b** average solution gap vs. step size. For this plot, the target problem of Fig. 2c was solved with 100 trials. The dash line shows the simulation result with our experimentally validated memristor model, which complies well with the experimental data. From the plot it can be seen that, when the step size is relatively small, increasing it can lead to improved system performance due to each iteration progressing further in the direction of gradient. However, as the step size becomes sufficiently large, the system will diverge. This aligns with the experience in neural network training. In fact, more advanced adaptive changing of step sizes is commonly implemented in the training of neural networks [3, 5, 6]. The selection of optimal hyperparameters including step size, the coefficient of momentum term, is a vast and intricate problem that requires further research in the future.

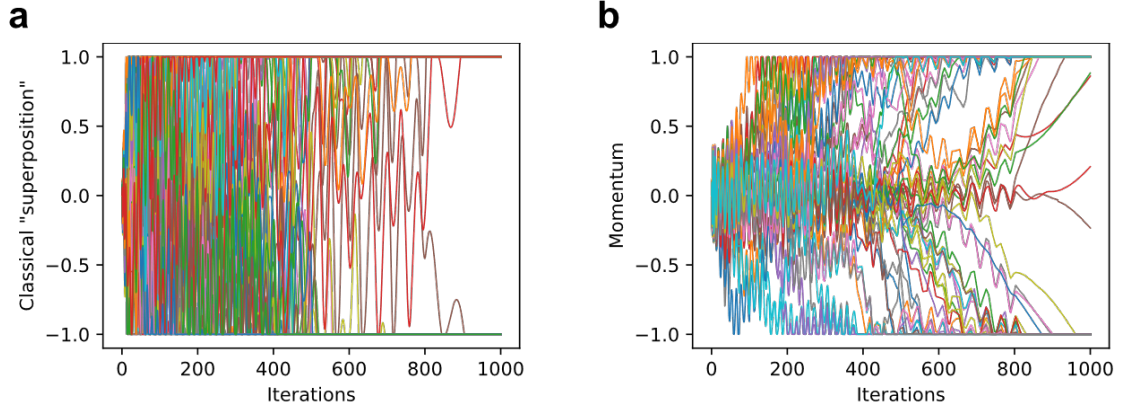


Figure S7. The evolution of intermediate values in experimental solving process of weighted Max-Cut problem. (a) The evolving process of classical “superposition” (analog spin proxy). (b) The evolving process of momentum. During the early stage of the solving process, the classical “superposition” rapidly oscillates between -1 and 1 due to higher impact of initial Hamiltonian, and eventually converges to -1 or 1 to match the binary spin configuration.

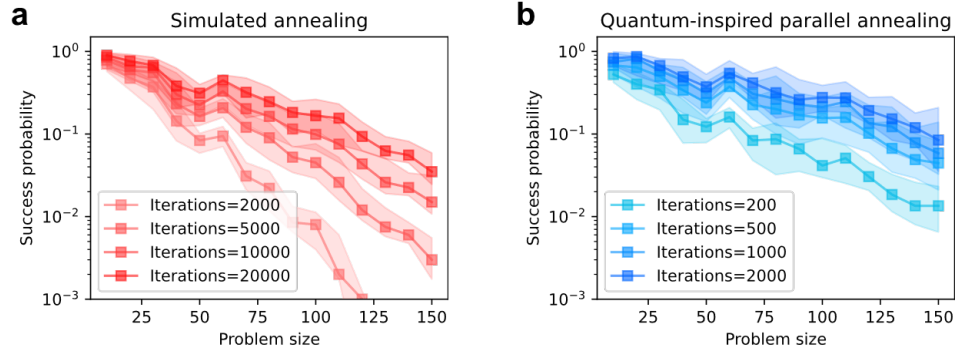


Figure S8. Success probability plot used for calculating TTS of weighted Max-Cut problem. a Success probability vs. problem size for simulated annealing **b** Success probability vs. problem size for quantum-inspired annealing.

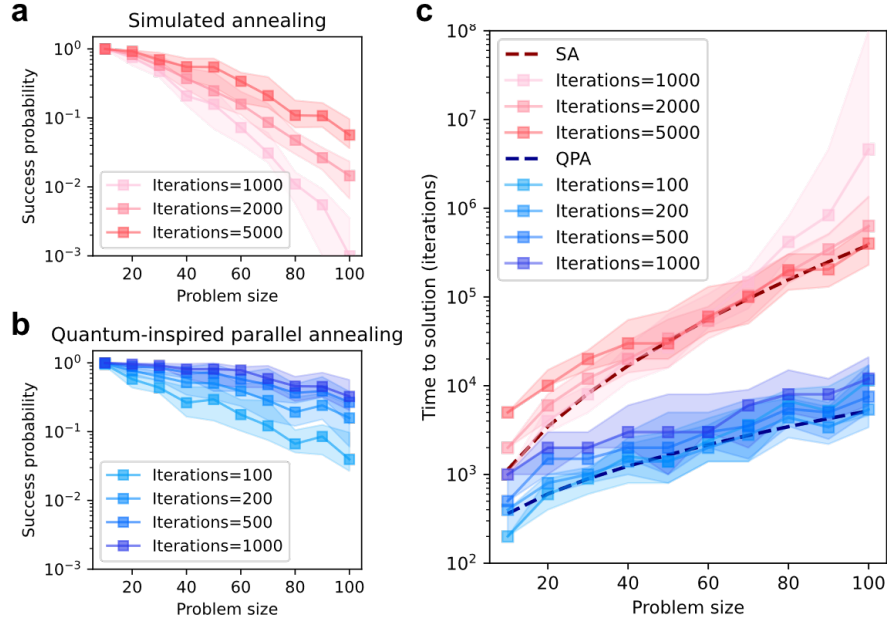


Figure S9. Scaling trend for unweighted Max-Cut problem. **a** Success probability vs. problem size for simulated annealing **b** Success probability vs. problem size for quantum-inspired annealing. **c** Time to solution vs. problem size. For each problem size, 20 randomly generated problem instances solved 1000 times with random initial spin configuration. The shaded area in the plot shows interquartile range. The square marker indicates the median.

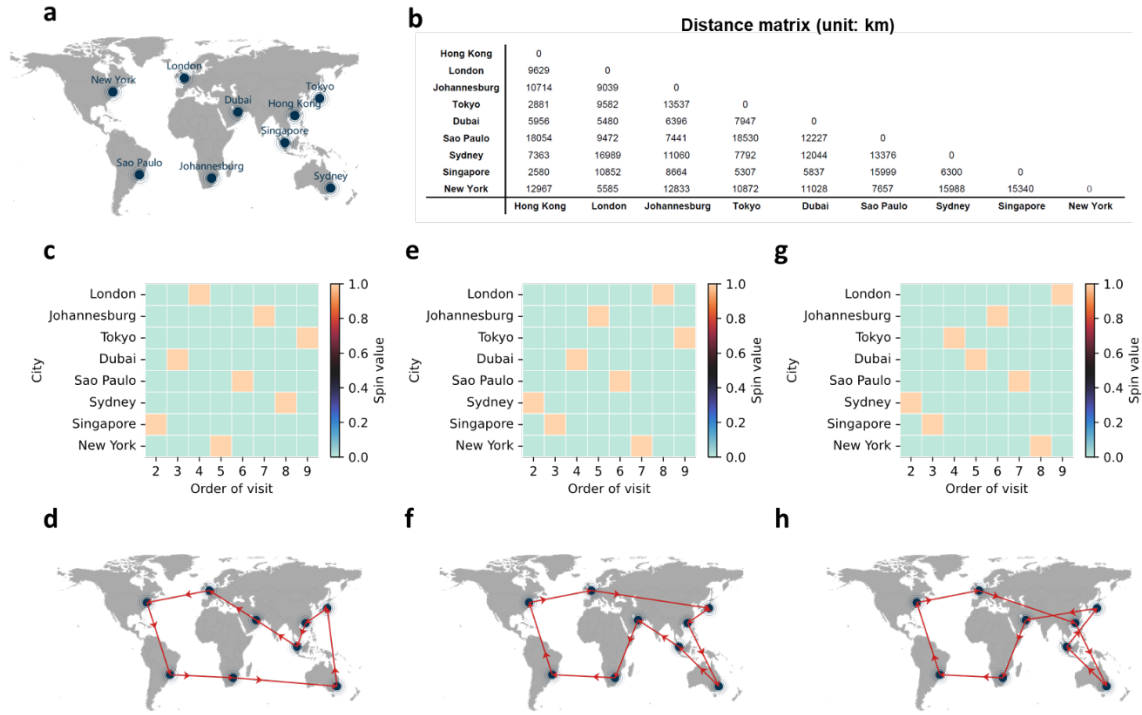


Figure S10. TSP instance and traveling path comparison. **a** The target problem instance solved in this paper. **b** The corresponding distance matrix between each city with the unit of km. The distance between each two cities is the shortest distance along the surface of earth and calculated by the coordinates of the city center. **c** The spin value matrix representation of the minimum-distance traveling path found by QPA. **d** Its corresponding traveling path in map. **e** The spin value matrix representation of the minimum-distance traveling path found by SA. **f** Its corresponding traveling path in map. **g** The spin value matrix representation of the minimum-distance traveling path found by DHNN. **h** its corresponding traveling path in map.

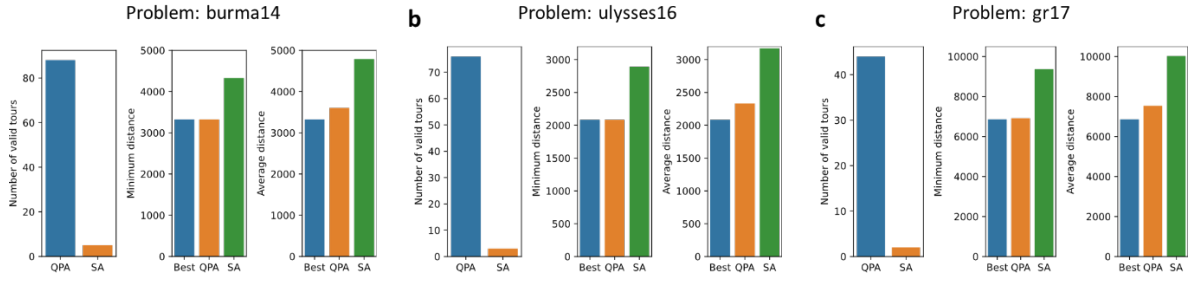


Figure S11. Simulation result of solving larger problems in TSPLIB [7]. **a** Result of solving burma14 problem **b** Result of solving ulysses16 problem **c** Result of solving gr17 problem. The Best in the plot represents the best-known solution provided by the library [7]. For each problem, 5000 iterations are conducted for both QPA and SA with 100 trials from random initial states.

Table S1. Energy and area estimation break down.

Module	Area/ μm^2	Latency/ns	Energy/pJ
Array	137.21	<0.3	1.54
Drivers	275.56	0.1	0.34
S&H	3.16	-	0.02
MUX	101.80	0.008	0.20
ADC	14250	-	2.50
Sum	14767.73	-	4.60

Table S1 shows the estimated area and energy breakdown of one iteration in our memristor-based Ising solver. To ensure a fair comparison with mem-HNN [8], our estimations are based on 16 nm technology node. For the analog-to-digital converter (ADC), an 8-bit state-of-the-art successive-approximation-register (SAR) ADC design is used for estimation [9]. For the implementation of QPA, the same working frequency with mem-HNN of 500 Mhz (2 ns for each iteration) is used for benchmarking the time to solution. The power dissipation of this system is estimated by energy consumption for each iteration dividing the time needed for each iteration (4.60 pJ / 2 ns) and the result is 2.3 mW.

Note S1. Details of the estimation of benchmark comparison and comments on other techniques

In this paper, the performance of solving 100-node 50%-density unweighted Max-Cut problems was compared between our QPA implemented on memristor crossbar array and other six Ising machines based on different technologies. For our memristor-based QPA, the annealing time was set at 200 iterations (400 ns), which results in a success probability of 16%. It takes 27 runs to reach a 99% success probability, resulting in a time to solution of 10.8 μ s. The energy to solution is calculated by multiplying the energy needed for a single iteration (4.60 pJ) with the overall iterations needed to reach 99% success probability (5400), resulting in an energy consumption of 24.8 nJ. For memristor-based Hopfield networks (mem-HNN), a hybrid updating scheme is utilized, which updates 10 nodes at a single iteration. To reach optimal time to solution, the annealing time is chosen as 50 updating cycles, where one updating cycle is equivalent to $N/10$ iterations (N is the problem size), resulting in the annealing time of 1 μ s. In addition, this hybrid updating method breaks the guaranteed convergence condition of simulated annealing and might lead to divergence, especially on denser problems with larger connectivity between nodes. More research is needed for the general adaptivity of this hybrid method. For phase-transition nano-oscillators (PTNO) based system, the fully analog property enables the low energy consumption. However, the experimental demonstration is limited to a problem size of $N=8$, which raises doubts about the effectiveness of the system on larger problems, especially considering of larger parasitic resistance and capacitance for larger problems. For ring oscillator (ROSC) based system, though the annealing time is ultra-fast compared to other techniques, its lack of annealing procedure results in a poor success probability and thus a relatively longer time-to-solution. As success probability of ROSC is not reported, an upper bound value of 1% is used for estimation, according to our previous experience on local-search algorithms. This results in the need of 459 runs to reach 99% success probability and the corresponding time to solution of 23 μ s without considering the overhead brought by comparing such number of solutions. In addition, as ROSC system only supports sparse connection, to solve a dense problem, more spins are needed for embedding techniques which might further degrade the performance. For simulate bifurcation machine (SBM), the reported time to solution is for solving an all-to-all connected problem with both positive and negative couplings. However, since the difficulty of the problem does not change fundamentally, it is used for estimated comparison purposes here. Additionally, its energy consumption is estimated based on the power dissipation of field programmable gate array (FPGA), which is taken as 200W in this estimation. For coherent Ising machine (CIM), as the energy consumption is not reported, we have estimated a lower bound by considering only the energy consumption of the FPGA part (The power dissipation is assumed to be 200 W), without taking into account of the energy consumption of the optical part. For D-Wave 2000Q quantum systems, the energy consumption is limited by cryogenic cooling [8, 10], and the speed of solving dense problems is limited by the sparse connection [11].

Reference

1. Leleu, T., et al., *Destabilization of Local Minima in Analog Spin Systems by Correction of Amplitude Heterogeneity*. Physical Review Letters, 2019. **122**(4).
2. Alizadeh, M., et al. *An empirical study of binary neural networks' optimisation*. in *International conference on learning representations*. 2018.
3. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
4. Le, H., et al. *AdaSTE: An Adaptive Straight-Through Estimator to Train Binary Neural Networks*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.
5. Duchi, J., E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*. Journal of machine learning research, 2011. **12**(7).
6. Zeiler, M.D., *Adadelta: an adaptive learning rate method*. arXiv preprint arXiv:1212.5701, 2012.
7. Reinelt, G., *TSPLIB—A traveling salesman problem library*. ORSA journal on computing, 1991. **3**(4): p. 376-384.
8. Cai, F., et al., *Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks*. Nature Electronics, 2020. **3**(7): p. 409-418.
9. Liu, J., M. Hassanpourghadi, and M.S.-W. Chen. *A 10GS/s 8b 25fJ/c-s 2850um² Two-Step Time-Domain ADC Using Delay-Tracking Pipelined-SAR TDC with 500fs Time Step in 14nm CMOS Technology*. in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*. 2022. IEEE.
10. Dutta, S., et al., *An Ising Hamiltonian solver based on coupled stochastic phase-transition nano-oscillators*. Nature Electronics, 2021. **4**(7): p. 502-512.
11. Hamerly, R., et al., *Experimental investigation of performance differences between coherent Ising machines and a quantum annealer*. Science Advances, 2019. **5**(5): p. eaau0823.