Surface programmable materials

Authors[1,*]
[1]XXX

I. INTRODUCTION

Definition: (Surface programmable material): a lattice (with translational invariance) model that can be programmed on its surfaces to perform universal computation.

[JGL: Please cover the following topics:

1. Computational models driven by the thermodynamics (cooling, or lowering the free energy), i.e. Brownian machine, such as DNA and Ising machine.
2. Potential advantages: energy saving, parallelism, no size limits (robust to quantum effects). Disadvantages: slow, error-prone.
3. Two approaches to lower the free energy: entropy driven, or energy based modeling. Embedding a computational model to energy models, the nondeterministic version: Circuit SAT, QUBO, SpinGlass and Ising machine.
4. Crutial issue of energy based model, hard to thermalize, hard to prepare.
5. Crystal based computing: Cellular automata as a computational model.
6. The method to thermalize, inspired from crystalization and heat zone.

]

II. MODEL AND METHOD

A. Elementary cellular automaton

An elementary cellular automaton is a 1-dimensional cellular automaton, where there are two possible states (labeled by 0 and 1). The rule to determine the state of the cell in next generation depends only on the current state of the cell and its two immediate neighbors.

There are $8 = 2^3$ possible configurations for a cell and its two immediate neighbors. Different elementary cellular automaton are only different from their translation rules. There are only $2^8 = 256$ different rules, so do the automatons.

If we put each possible current configurations in order: 111, 110, ..., 001, 000, and put the resulting state under them. We then get an integer in its binary representations. Then this integer is taken to be the rule number of the automaton. For example, rule 110.

Given that $110_d = 01101110_2$, so rule 110 is defined by the translation rule:

Current Pattern	111	110	101	100	011	010	001	000
New Center Cell	0	1	1	0	1	1	1	0

Rule 110 has been shown to be Turing Complete [1], and thus capable of universal computation.

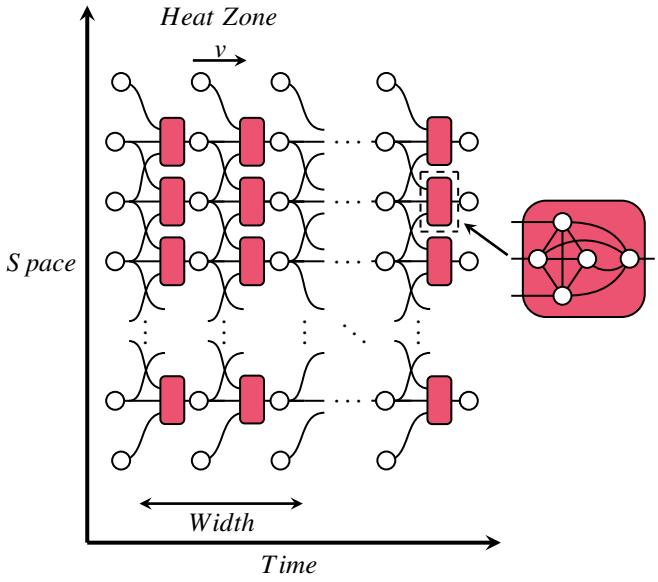
B. Energy model

Let us define a model Hamiltonian on a 2D lattice of size $S \times T$ as

$$H = \sum_{t=1}^{T-1} \sum_{i=1}^{S} (n_{i+1,t} \wedge \neg(n_{i,t} \wedge \neg n_{i-1,t})) \veebar n_{i,t+1}, \qquad (1)$$

where $n_{i,t} \in \{0, 1\}$ is the state of the cell at row i and column t, the boundary is periodic in the space direction, i.e. $n_{0,t} = n_{S,t}$. Its ground state with energy 0 encodes a 110 cellular automaton, where the direction of computation is from the left to right.

The gadget we constructed based on the Rule 110 cellular automaton naturally possesses Turing completeness. Therefore, it can be tiled in a two-dimensional plane to create a computational material with logical operation capabilities.



* xxx@xxx.com

[JGL: We need a main figure to show the tiling and cooling.]

One can easily verify that with this lattice-like structure, we can build infinitely large gadgets capable of universal computing in a surface. Thus we call it Surface Programmable Material.

C. Cooling computation with direction

Definitions: * in-surface/out-suface: The surface of a surface programmable material that associated with the input/output of the logic circuit.

The process of heating and annealing the entire material does not have computational capability, as the system lacks logical directionality.

Logical directionality needs to be introduced through an orderly annealing method. Similar to the localized heating-annealing approach that catalyzes the growth of ordered crystals, we introduced a temperature gradient annealing scheme.

More specifically, the computation, with computing direction from in-surface to out-surface, contains the following steps: (1). Initialize the in-surface configuration. By removing some atoms on the in-surface. (2). Connect the in-surface/out-surface to external heat sources at temperature $T_1 < T_2$, respectively. We also require that the energy gap between the ground state and the first excited state of the Hamiltonian to be $\Delta E < T_1$. (3). Lower the temperature of the heat sources "slowly" to cool the system to the ground state of the Hamiltonian. The temperature of the heat sources at time t is $T_{1/2}(t) = T_{1/2}(0)\lambda^{-ct}$, where $T_{1/2}(0)$ is the initial temperature of the heat sources, and α is a constant.

The computing direction could be reversed by swaping the temperature of the heat sources connected in-surface and out-surface. However, solving the ground state from the non-deterministic direction is at least as hard as solving the circuit satisfiability problem, which is NP-complete [2].

D. Computing by cooling

We test the hypothesis: Cooling is easy if the process is from the deterministic direction, hard if the process is from the non-deterministic direction.

In our gadget, cooling from deterministic direction is from input to output, non-deterministic direction is from output to input. The latter one must be non-deterministic because this gadget is Turing-Complete.

The goal is to drive the the state to the one with lowest energy as fast as possible, while some bits set to be 1 or 0 in the state.

1. Estimation of the computing time along computation direction

We fixed the width of each layer and denote m as the number of layers in this material.

Let the temperature of the k-th layer at time t be $T(t,k) = T\lambda^{ct-k}$, where T is the initial temperature, c is a constant, and $\lambda < 1$ is a constant. At any given time t, we denote the subset of atoms at depth $ct - \frac{W}{2} \le k \le ct + \frac{W}{2}$ as the active zone, where W is the width of the sliding window such that $e^{-\Delta E_{max}/\lambda^{W/2}} = \epsilon \ll 1$. The active zone is the region where non-trivial computation occurs. The atoms outside the active zone are either frozen or completely randomized.

Clearly, W asymptotically scales as $\frac{\log(-\frac{1}{\log\epsilon})}{\log(\lambda)}$. The situations we care about are those $\lambda \lessapprox 1$, so one can rewrite W as $W \approx (1-\lambda)^{-1}\log(-\frac{1}{\log\epsilon})$.

We consider thermalizing the system in units of W. ϵ is the error probability of each unit of time, which should scale as $\epsilon \sim \left(\frac{m}{W}\right)^{-1}$.

The probability transition matrix of the active zone at any given time t (except the starting and ending time) is the same, so we denote it as $P = P(t)$. The error tolerance requires the zone to be thermalized to certain extent, i.e. $\left(\frac{\lambda_2(P)}{\lambda_1(P)}\right)^{t_{\text{th}}} < \epsilon$, where $\lambda_1(P) \ge \lambda_2(P)$ are the two largest eigenvalues of P. We have $t_{\text{th}} \sim \left(1 - \frac{\lambda_2(P)}{\lambda_1(P)}\right)^{-1}\log(\epsilon^{-1})$.

Under the assumption that $\left(1 - \frac{\lambda_2(P)}{\lambda_1(P)}\right) \sim e^{-W}$, we have $t_{\text{th}} \sim e^{(1-\lambda)^{-1}}\log^2(\epsilon^{-1})$. The total time for the computation should be $t_{\text{total}} \sim \left(\frac{m}{W}\right)\log^2(\frac{m}{W})e^{(1-\lambda)^{-1}}$.

2. Estimation of the computing time along input memory direction

We fixed the number of layers and denote n as the width of each layer in this material.

However, we fail to derive an explicitly expression for the time complexity in this direction. Numerical result shows that computing complexity is linear to n. Thus a total time for the computation should scale as $nm\log^2(m)e^{(1-\lambda)^{-1}}$

3. Testing model through equilibrilium SA

We used simulated annealing to mimic the heating-annealing process. However, classical simulated annealing can only handle the situation where the temperature is space-invariant. Thus, a scheme change is needed.

One can simply make the following modification to the transition probability in the markov process and return to the situation, called energy-gradient scheme, where the temperature is space-invariant.

$$e^{-\frac{\Delta E_k}{T\lambda^{ct+k}}} = e^{-\frac{\Delta E_k(\frac{1}{\lambda})^k}{T\lambda^{ct}}} = e^{-\frac{\Delta E_k \Lambda^k}{T(t)}} \tag{2}$$

Where we denote $\Lambda = \frac{1}{\lambda}$ and the time complexity in Λ form is apparently $T_{total} \sim \frac{m}{W}\log^2(\frac{m}{W})e^{(\Lambda-1)^{-1}}$

E. Algorithm beyond temperature gradient

Applying a huge energy gradient or temperature gradient to this material seems not a good idea when the system become larger.

Inspired by the way clothes are ironed, we conceived an idea called wave-packet scheme, that slowly scanning a wave-shaped temperature curve from one end to the other. This method is widely used in studying the heating-recrystallization properties of real materials [3].

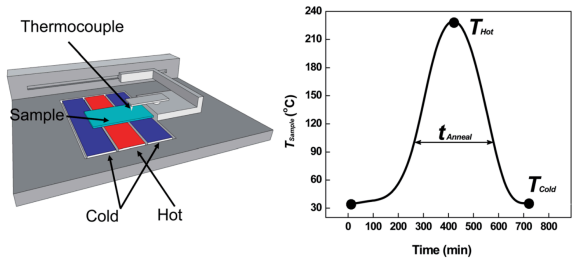


FIG. 1. Method to scan the wave packet

There are various specific choices for the wave packet, such as power-law, exponential, and others. In this context, we primarily focus on a exponential wave packet, which can be directly linked to the concept of energy gradient.

A expoential wave packet with central position $middle(t)$, amplitude A and base Λ, can be express as following.

$$T(k,t) = A\Lambda^{|k-middle(t)|}$$

So slowly moving the wave packet means slowly changing $middle(t)$

Let's review the concept active zone defined earlier. When the temperature of the system is defined as $T_{tg}(k,t) = T\lambda^{ct-k}$, atoms in the active zone are those whose layer k belongs to $k < ct - \frac{W}{2}$, where W are given by $e^{-\frac{\Delta E_{max}}{\lambda^{\frac{W}{2}}}} \leq \epsilon$. For those $k < ct - \frac{W}{2}$, the probability that SA would flip an atom's state is no more than $e^{-\frac{\Delta E_{max}}{T\lambda^{ct}\lambda^{-ct+\frac{W}{2}}}} = e^{-\frac{\Delta E_{max}}{T\lambda^{\frac{W}{2}}}} \leq \epsilon^{\frac{1}{T}}$, which means these previous layer would stay "frozen". For those layer much deeper than $k = ct$, SA would just randomly flip them. These layer won't affect layers in active zone.

Now turn our sight into the wave packet model, we can construct a similar concept heat zone. Layers in the heat zone are those $A\lambda^{|k-middle(t)|} \geq \epsilon \rightarrow middle(t) + \frac{\log(\epsilon)}{\log(\Lambda)} < k < middle(t) - \frac{\log(\epsilon)}{\log(\Lambda)}$.

The layers that shallower than $middle(t) + \frac{\log(\epsilon)}{\log(\Lambda)}$ correspond to layers shallower than $ct - \frac{W}{2}$ in temperature gradient model, since they are both stay "frozen"; layers between $middle(t) + \frac{\log(\epsilon)}{\log(\Lambda)}$ and $middle(t)$ are effectively cooled down, which correspond to the active zone; layers deeper than $middle(t)$ are either randomly fliped or stay "frozen", but we don't care their state.

Therefore, the wave packet model are just the same as the temperature gradient model, and is more likely to be realized physically.

1. Some details about non-equilibrium simulated annealing

The classical simulated annealing can only deal with the situation that the temperature is space-invariant. Although there are many ways to simulated a real-world system with temperature gradient, such as Molecular Dynamics [4, 5], but these methods mainly depend on dynamically adjusting the atomic velocities to achieve a temperature gradient, which can't be directly used in our model.

There are also Monte Carlo-based methods to simulate a moving temperature wave packet that facilitates material growth [6, 7]. However, these methods fail to establish a direct connection between the simulated temperature and experimental temperature [8]. Therefore, based on HeatBath acceptance rule, we made a slight modification to the classical SA model.

When we flip one atom's state, there would be some energy difference ΔE_i in field i with temperature T_i. The probability to accept this flip would be

$$P = \frac{1}{1 + e^{\sum_i \frac{\Delta E_i}{T_i}}} \tag{3}$$

It is easily to observe that when the system returns to heat-equilibrium, the probability reverts to the classical HeatBath acceptance probability.

III. NUMERICAL RESULT

To determine large-scale effect, we mainly focus on the simple toy model in Equation (1).

We conducted numerical experiments both for energy-gradient and wave-packet scheme, under the condition $\Lambda = 1.3, n = 15$, and fitted each sweep time against $m\log^2(m)$. Where m represents the number of the single layers, n represents the width per single layer.

Another numerical experiment are conducted to determine the time complexity of the computation along the input memory direction.

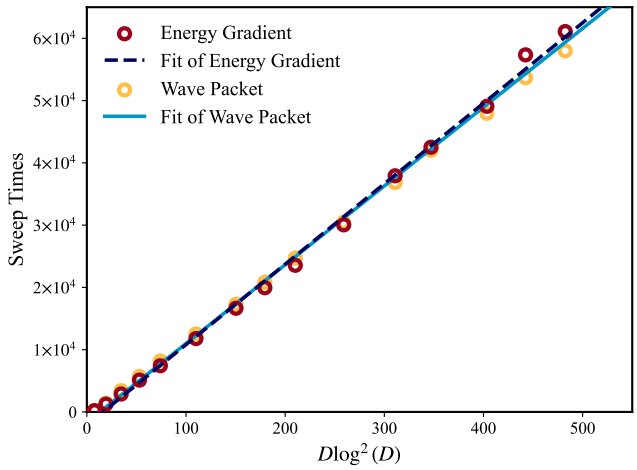


FIG. 2. Time v.s. Depth of the material

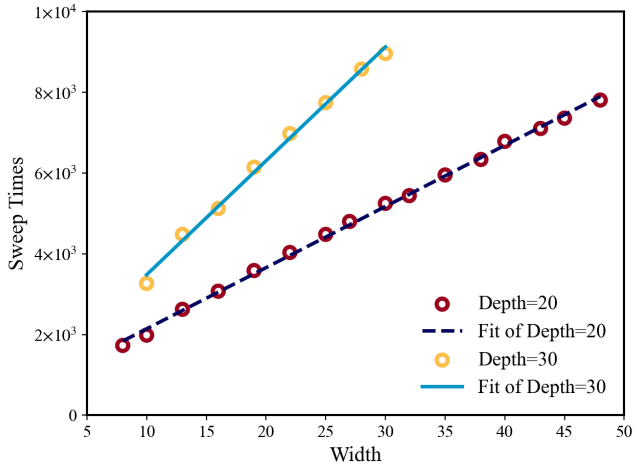


FIG. 3. Time v.s. width of the material

This indeed informs that the time scale along the memory direction is linear.

IV. PHYSICAL IMPLEMENTATION ON ISING MACHINE

To examine whether this model could achieve a better performance when using adiabatic methods, we need to generalize this energy function to some good hamiltonian. There has already been work on mapping combinatorial optimization problems to a spin-glass model (known as the Ising Machine) and leveraging the properties of digital devices that simulate this Ising Machine to find an appropriate ground state, which encodes computational information [9, 10].

A spin-glass hamiltonian is described as follows.

$$H = \sum_{u,v\in E} J_{u,v} s_u s_v + \sum_{i\in V} h_i s_i \tag{4}$$

where h_i is onsite energy, $J_{u,v}$ is interaction energy and $s_i \in \{-1, 1\}$ is local spin. The mapped spin-glass model should possess propertys that it has exactly 8 ground states, each correspond to one input-output relation in Cellular Automata.

A. Ising machine gadget design

To map our toy model in Equation (1) to spin-glass model, we used Linear Programming algorithm. Linear problems are problems that can be express in standard form as

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n} c_i x_i \\ &\text{s.t. } l_j \le \sum_{i=1}^{n} a_{i,j} x_i \le u_j,\ j = 1,\dots,m \\ &p_i \le x_i \le q_i,\ i = 1,\dots,n \end{aligned} \tag{5}$$

Here with a spin-glass consists N atoms, we choose n to be $\frac{N(N-1)}{2} + N$, which means each variable x_i represents a correspond interaction energy $J_{u,v}$ or a correspond onsite energy h_u, we assigned aliases to these x_i as $x_{u,v}$ or x_u.

Let the configuration space of the system be $S = \mathbf{s}_i \mid i = 1,\dots,2^N$, each associated with a energy $H(\mathbf{s}_i)$. We denote the target states with minimum energy as $S_{\min} \subset S$. We can express the linear programming problem as

$$\begin{aligned} &\min_{J\in\mathbb{R}^{N(N-1)/2}, h\in\mathbb{R}^N} 0 \\ &H(\mathbf{s}_i) < H(\mathbf{s}_j), \forall \mathbf{s}_i \in S_{\min}, \mathbf{s}_j \in S \setminus S_{\min} \\ &H(\mathbf{s}_i) = H(\mathbf{s}_j), \forall \mathbf{s}_i, \mathbf{s}_j \in S_{\min} \end{aligned} \tag{6}$$

Note that H is a linear function of J and h, so the constraints are linear. The less constraints and equality constraints can be easily transformed into inequality constraints by adding ancilla variables.

There is no solution with 4-atoms spin-glass that can satisfy the constraints. The minimum number of atoms required to satisfy the constraints is 5. We set the target states to

input 1	input 2	input 3	output	ancilla
↓	↓	↓	↓	?
↓	↓	↑	↑	?
↓	↑	↓	↑	?
↓	↑	↑	↑	?
↑	↓	↓	↓	?
↑	↓	↑	↑	?
↑	↑	↓	↑	?
↑	↑	↑	↓	?

TABLE I. The eight degenerate ground states of the spin-glass model to implement the 110 rule.

where the ancilla bit in each row can be either ↑ or ↓ (256 total possibilities). One of the solutions satisfying the constraints is

$$J = \begin{pmatrix} \cdot & 1 & 1 & 2 & 3 \\ \cdot & \cdot & 2 & 2 & 5 \\ \cdot & \cdot & \cdot & 2 & 5 \\ \cdot & \cdot & \cdot & \cdot & 6 \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}, h = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 5 \end{pmatrix} \quad (7)$$

1. Numerical result

V. DISCUSSION AND OUTLOOK

Connect real-world thermalization time and the simulated annealing time to explore the connection between time and energy cost per computation. The emergence of wisdom?

ACKNOWLEDGMENTS

We thank Lei Wang, Madelyn Cain and XXX for helpful discussions on the simulation methods.

[1] M. Cook, A concrete view of rule 110 computation, arXiv preprint arXiv:0906.3248 (2009).
[2] C. Moore and S. Mertens, The nature of computation (Oxford University Press, 2011).
[3] X. Zhang, K. G. Yager, J. F. Douglas, and A. Karim, Suppression of target patterns in domain aligned cold-zone annealed block copolymer films with immobilized film-spanning nanoparticles, Soft matter 10, 3656 (2014).
[4] X.-M. Bai, Y. Zhang, and M. R. Tonks, Testing thermal gradient driving force for grain boundary migration using molecular dynamics simulations, Acta materialia 85, 95 (2015).
[5] D. Deng and H. Murakawa, Numerical simulation of temperature field and residual stress in multi-pass welds in stainless steel pipe and comparison with experimental measurements, Computational materials science 37, 269 (2006).
[6] A. Godfrey and J. Martin, Some monte carlo studies of grain growth in a temperature gradient, Philosophical Magazine A 72, 737 (1995).
[7] Y. Tan, A. Maniatty, C. Zheng, and J. Wen, Monte carlo grain growth modeling with local temperature gradients, Modelling and Simulation in Materials Science and Engineering 25, 065003 (2017).
[8] D. Zöllner, A new point of view to determine the simulation temperature for the potts model simulation of grain growth, Computational materials science 86, 99 (2014).
[9] N. A. Aadit, A. Grimaldi, M. Carpentieri, L. Theogarajan, J. M. Martinis, G. Finocchio, and K. Y. Camsari, Massively parallel probabilistic computing with sparse ising machines, Nature Electronics 5, 460 (2022).
[10] C. Bybee, D. Kleyko, D. E. Nikonov, A. Khosrowshahi, B. A. Olshausen, and F. T. Sommer, Efficient optimization with higher-order ising machines, Nature Communications 14, 6033 (2023).
[11] M.-T. Nguyen, J.-G. Liu, J. Wurtz, M. D. Lukin, S.-T. Wang, and H. Pichler, Quantum optimization with arbitrary connectivity using rydberg atom arrays, PRX Quantum 4, 010316 (2023).
[12] R. P. Feynman, Feynman lectures on computation (CRC Press, 2018).
[13] D. Reeb and M. M. Wolf, An improved landauer principle with finite-size corrections, New Journal of Physics 16, 103011 (2014).
[14] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, Quantum optimization for maximum independent set using rydberg atom arrays, arXiv preprint arXiv:1808.10816 (2018).
[15] L. Wang, T. F. Rønnow, S. Boixo, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Comment on:" classical signature of quantum annealing", arXiv preprint arXiv:1305.5837 (2013).

Appendix A: The implmentation in Rydberg atoms array

Statement 1: The classical part of Rydberg Hamiltonian encodes an independent set problem.

The Rydberg Hamiltonian [11] is defined as

$$H_{\mathrm{Ryd}} = \sum_v \frac{\Omega_v}{2}\sigma_v^x - \sum_v \Delta_v n_v + \sum_{v<w} V_{\mathrm{Ryd}}(|\vec{\mathbf{r}}_v - \vec{\mathbf{r}}_w|) n_v n_w. \quad \text{(A1)}$$

where Ω_v is the Rabi frequency, Δ_v is the detuning, $n_v = \frac{1}{2}(1-\sigma_v^z)$ is the number operator, and $V_{\mathrm{Ryd}}(|\vec{\mathbf{r}}_v - \vec{\mathbf{r}}_w|) = C_6/|\vec{\mathbf{r}}_v - \vec{\mathbf{r}}_w|^6$ is the Rydberg interaction potential.

The classical part of which can be written as

$$H_{\mathrm{MWIS}} = -\sum_{v\in V}\delta_v n_v + \sum_{(u,v)\in E} U_{uv} n_u n_v. \tag{A2}$$

The ground state of which encodes the maximum weight independent set (MWIS) problem.

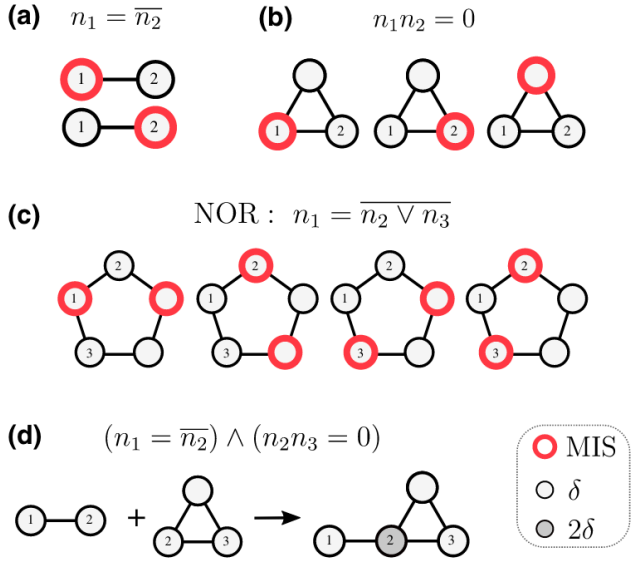
> Wikipedia: In graph theory, a maximal independent set (MIS) or maximal stable set is an independent set that is not a subset of any other independent set. In other words, there is no vertex outside the independent set that may join it because it is maximal with respect to the independent set property.

Statement 2: Finding the ground state of the classical part of the Rydberg Hamiltonian is equivalent to finding the maximum weight independent set.

a. Energy based universal computation with Rydberg atoms array

Statement 3: The classical Rydberg Hamiltonian is universal for classical computation.

The NOR gate can be implemented using the Rydberg Hamiltonian (subfigure c below). The NOR gate is a universal gate for classical computation.



The conjunction of gates can be implemented by "gluing" the Rydberg atoms together (subfigure d below). The weights are added together.

For more logic gates, please check the GitHub repository UnitDiskMapping.jl.

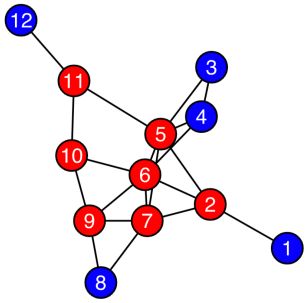
b. Cooling the Rydberg Hamiltonian

Statement 4: The Rydberg Hamiltonian, if cooled successfully with some vertices fixed to certain configuration, can be used to solve the circuit satisfiability problem, which is NP-complete [2].

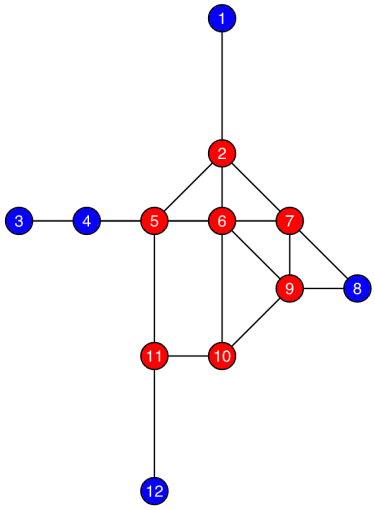
$P \neq NP$: Cooling is generally hard, especially when from the non-deterministic direction.

c. The Rule 110 Gadget

We can encode the Rule 110 cellular automaton into a Weighted Maximum Independent Set Problem, with blue vertices assigned a weight of 1 and red vertices assigned a weight of 2, as follows.



This graph can be embedded into a grid graph, where two vertices are connected if and only if their Euclidean distance is no more than 2.



The correspondence between the Maximum Weighted Independent Set (MWIS) Solution and Rule 110 is as follows:

The states of vertex 1, vertex 3, and vertex 8 represent the states of the middle, left, and right cells of the automaton's input, respectively. If the input value of a cell is 1, then the corresponding vertex must be in the MWIS solution; otherwise, it is not. Vertex 12 corresponds to the automaton's output. If the automaton output is 1,

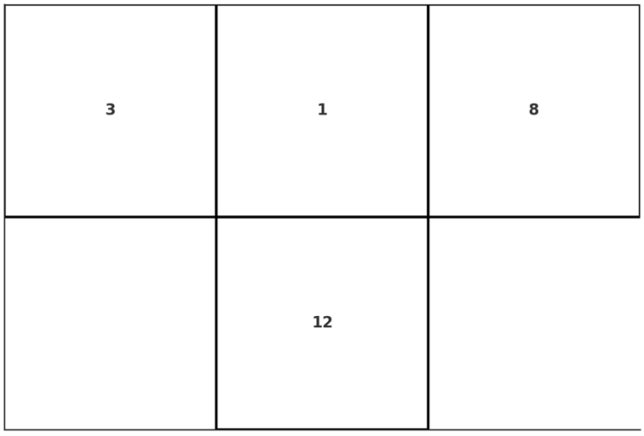
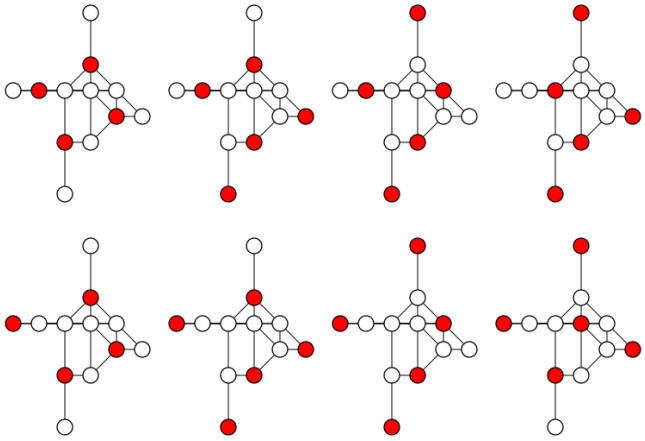


FIG. 4. Alt text

then vertex 12 is in the MWIS solution; otherwise, it is not.

In the automaton diagram, the above gadget is equivalent to:

There are exactly 8 different MWIS solutions in this graph (the weighted size of each MWIS solution is 7), each corresponding to one of the 8 possible outputs of the automaton. We list them as follows.



Utilizing copy gadget and cross gadget [11], we construct a Surface Programmable Material with open boundary conditions as follows.

The above gadget depicts a two-layer cellular automaton. The vertices in blue, red, green and black have weights of 1, 2, 3 and 4, respectively. In the automaton diagram, the above gadget is equivalent to:

1. Speed and work

The trade-off between the energy consumption and the speed of computation [12]. To avoid confusion, we emphasize the "energy consumption" is defined as the work done in a computational process, which is the same as the amount of heat dissipated to the environment. This

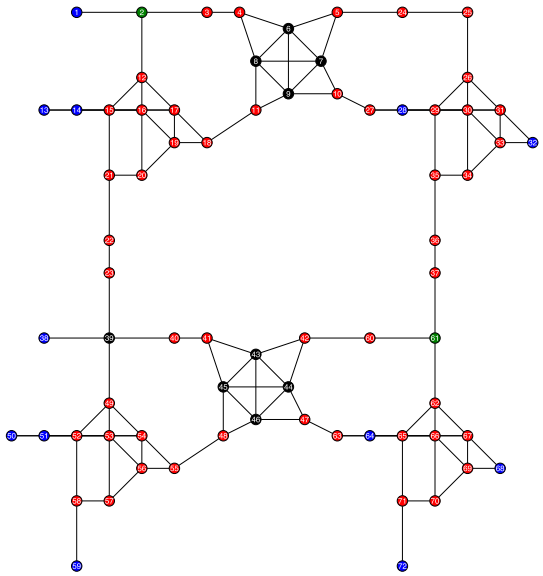


FIG. 5. Alt text

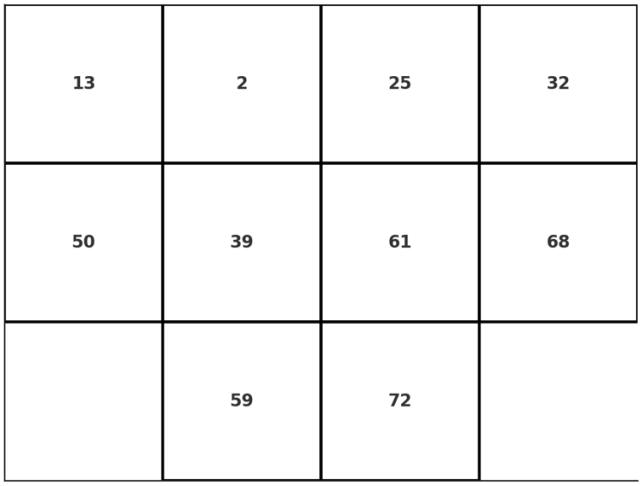


FIG. 6. Alt text

quantity has a lower bound given by the Landauer principle, which states that the work done in a computation is at least $kT\ln 2$ per bit erased[13].

Information erasure in the surface programmable material is proportional to the volume of the material, which is $O(tS)$, where t is the time of computation, and S is memory (proportional to the surface area) of the material.

From the chemical reaction perspective, the speed of computation is determined by the parameter λ.

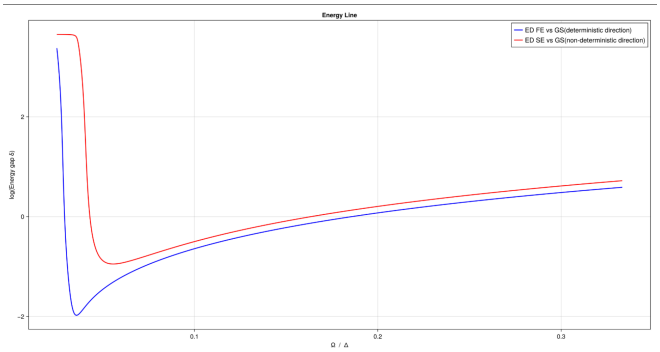


FIG. 7. Alt text

a. Quantum adiabatic annealing energy gap

One possible way is to use quantum adiabatic annealing: start from a simple hamiltonian $H(0)$ and its simple ground state $|\psi(0)\rangle$, then gradually change the parameters until reaching the desire hamiltonian $H(t)$.

More specifically, set $\Delta(t = 0) < 0$ and $\Omega(t = 0) = 0$ initially, then first turning on $\Omega(t)$ to a non-zero value, sweeping $\Delta(t)$ to final value, and finally turning off $\Omega(t)$.

$$H_{QAA}(t) = \sum_{v\in V}(-\Delta(t)w_v\hat{n}_v + \Omega(t)\sigma_v^x) + \sum_{(u,w)\in E} U\hat{n}_u\hat{n}_w \quad \text{(A3)}$$

If the time evolution is sufficiently slow, then by the adiabatic theorem, the system follows the instantaneous ground state, ending up in the solution to the MWIS problem [14]. Then we only need to evalute the minimum energy gap Δ_{QAA} between the ground and first-excited states of instantaneous hamiltonian.

We set $\Omega = 1\times 2\pi$ and sweep the Δ from $3\times 2\pi$ to $40\times 2\pi$ with 1*1 gadget. For deterministic direction, we simply set the weight of the input vertices to 50; as for non-deterministic direction, we set the weight of the output vertice to 50.

Result listed as follows. However, we didn't see cooling from deterministic direction would give a smaller energy gap than the other direction. We think that's because the size of this gadget is too small.

Appendix B: Classical Dynamics

We firstly tried classical adiabatic annealing with the classical-spin mapping method [15]

$$\frac{\partial\vec{M}_i}{\partial t} = \vec{M}_i \times \vec{H}_i(t) \quad \text{(B1)}$$

where

$$\vec{H}_i(t) = -\frac{\partial H(t)}{\partial\vec{M}_i} \quad \text{(B2)}$$

Here the hamiltonian of the system is

$$H(t) = \frac{t}{T}(\sum_{u,v} J_{u,v}M_{u,z}M_{v,z} + \sum_u h_u M_{u,z}) + (1-\frac{t}{T})(I\sum_u M_{u,x}) \quad \text{(B3)}$$

The former term is the generally increasing target hamiltonian, the latter term is a generally decreasing known Ising-transverse field. We can explicitly write out the effective magnetic field.

$$\vec{H}_i(t) = -\frac{t}{T}(\sum_v J_{i,v}M_{v,z} + h_i)\hat{e}_z - (1-\frac{t}{T})I\hat{e}_x$$

Integrate the ordinary differential equation then we get the classical dynamics of this spin-glass model. However, result shows that it is extremely hard to find the solution when the number of layers exceed 4 (input layer is pinned).

The reason maybe that the mapped spin-glass model lies in the hard region in [15]. Because every interaction energy is positive, which gives rise to strong frustration. Results in this reference also shows that even in random spin-glass model, there exist hard instance that can't be solve.