2022 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2022, Virtual, December 12-15, 2022

Evaluation of Temperature-Control-Free Replica-Replica-Interactive Simulated Annealing using 100 Max-Cut Problems

Akio Yoshizawa†

† National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan
Email: yoshizawa-akio@aist.go.jp

Abstract– The path-integral quantum Monte Carlo (PIQMC) method is widely used as a classical simulation algorithm for quantum annealing. In this method, replicas represent different points in imaginary time. We propose and demonstrate a temperature-control-free simulated annealing algorithm inspired by the PIQMC method. Replicas help each other in a swarm-like manner to search for the ground state. We solve 100 max-cut problems for algorithm evaluation, each of which corresponds to a graph of 100 vertices. We also solve the same problems using a conventional method based on the Metropolis algorithm for comparison.

1. Introduction

In physics-inspired computation, metaheuristic algorithms reflecting the laws and principles of physics are developed. The earliest, and still well-known, algorithm is the so-called Metropolis algorithm [1]. The annealing process in material science based on classical thermo-dynamics is simulated. Quantum annealing (also known as adiabatic quantum computation) is its quantum version [2], which was demonstrated using superconducting flux qubits [3]. The path-integral quantum Monte Carlo (PIQMC) method is a classical algorithm for simulating quantum annealing. Its superiority over conventional methods has been reported for many optimization problems [4-5]. We anticipate that more insight into the PIQMC method will lead to the creation of novel unique quantum-inspired algorithms.

We propose and demonstrate PIQMC-based simulated annealing, in which replicas representing different points in imaginary time act cooperatively like a swarm for ground-state searching. Their interactions are local, simple and fluctuate to a certain degree. Such fluctuations are necessary to escape local minima. However, we do not need explicit control of the fluctuations, making our annealing temperature-control-free.

The rest of the paper proceeds as follows. Section 2 provides an explanation of the PIQMC method and our temperature-control-free algorithm. The main results and a discussion are given in Sections 3 and 4, respectively. We solve 100 max-cut problems for algorithm evaluation, each of which corresponds to a graph of 100 vertices.

2. Proposed simulated annealing

2.1. PIQMC method

The PIQMC Hamiltonian with K replicas is denoted by [4]

$$H = \sum_{k=1}^{K} H_k + W(t) H_{\text{int}}, \tag{1}$$

where

$$H_{\text{int}} = -\sum_{k=1}^{K} \sum_{i=1}^{N} \sigma_i^k \sigma_i^{k+1}. \tag{2}$$

For a spin at the i site of the k-th replica, $\sigma_i^k = \pm 1$ and $\sigma_i^1 = \sigma_i^{K+1}$. Replicas have the same objective function given by the following N-spin Ising model:

$$H_k = -\sum_{i,j=1}^{N} J_{ij} \sigma_i^k \sigma_j^k. \tag{3}$$

Our purpose is to find the ground state. Eq. (2) is the interaction energy expressing replica-replica interactions. Since $W(t) \geq 0$ is a time-dependent monotonically increasing function, Eq. (2) induces replica states to become identical or nearly identical as annealing proceeds, although their interactions are limited to nearest neighbors. This is analogous to local interactions leading individuals (replicas in our case) to form a swarm and to move in the same direction as their neighbors without a leader's instruction.

To make our annealing temperature-control-free, we do not use $W(t)$ in our annealing process. Thus, we cannot precisely mimic the PIQMC method. However, we can still use the essential features of Eq. (2) to implement replica-replica interactions. In the Metropolis algorithm, each replica separately and independently receives thermal noise in the form of classical artificial fluctuations in order to escape local minima; thus, the searching ability of the algorithm depends on how suitable the temperature control is. This case is therefore as if one replica plays a leading role, determining the temperature control and sending instructions to the other replicas. It is interesting to compare annealing methods with and without swarm-like replica-replica interactions.

ORCID iD Akio Yoshizawa: 0000-0001-5153-801X

2.2. Two-party rule

We assume that Alice and Bob possess σ_i^k and σ_i^{k+1}, respectively. When their spins are parallel, the product $-\sigma_i^k\sigma_i^{k+1}$ in Eq. (2) becomes negative. This negative value contributes to lowering the interaction energy; thereby, Alice and Bob leave their states unchanged. On the other hand, when their spins are antiparallel, Bob tosses a coin and randomly but evenly chooses his state from $\sigma_i^{k+1} = \pm 1$. Coin tossing creates fluctuations, which work as a driving force to help replicas escape local minima. Table 1 summarizes our two-party rule. Since the chance of encountering antiparallel spins becomes small if annealing goes well and considering that such antiparallel spins determine the frequency of Bob's coin tossing, fluctuations gradually and automatically diminish as annealing proceeds. In this case, we do not need to explicitly control the fluctuations.

Table 1 Two-party rule

Input		Output
Alice (σ_i^k)	Bob (σ_i^{k+1})	Bob (σ_i^{k+1})
± 1	± 1	± 1
± 1	∓ 1	Coin toss

2.3. Annealing algorithm

Algorithm 1: Two-party rule

1: **for** each run **do**
2: initialize to random initial state
3: **for** each renewal **do** (n_1 times)
4: **for** each replica **do**
5: **for** each spin **do**
6: propose a flip
7: calculate energy increase ΔE after the flip
8: **if** $\Delta E \leq 0$, **then** accept spin flip
9: **end for**
10: **end for**
11: **for** each replica **do**
12: **for** each spin **do**
13: **if** $\sigma_i^k \neq \sigma_i^{k+1}$ **then** $\sigma_i^{k+1} = \pm 1$ randomly & evenly
14: **end for**
15: **end for**
16: **end for**
17: **for** each renewal **do** (n_2 times)
18: **for** each replica **do**
19: **for** each spin **do**
20: propose a flip
21: calculate energy increase ΔE after the flip
22: if $\Delta E \leq 0$, accept spin flip
23: **end for**
24: **end for**
25: **end for**
26: **end for**

To make our annealing temperature-control-free, we do not explicitly use $W(t)$ in our algorithm (Algorithm 1), where each replica is cooled separately and independently using the Metropolis method with zero temperature (see lines 3-10). Replica-replica interactions based on Table 1 are implemented in line 11 to line 15. We repeat cooling and replica-replica interactions alternately n_1 times (line 3) and continue cooling alone another n_2 times (line 17). We compare our algorithm with the temperature-controlled Metropolis method (see Algorithm 2) and examining their success probabilities. Here, the success probability is defined as follows.

$$\text{Success probability} = \frac{\text{Number of replicas reaching the ground state}}{\text{Total number of replicas: } K} \quad (4)$$

Algorithm 2: Metropolis method

1: **for** each run **do**
2: initialize to random initial state
3. set an initial temperature $T(t=0)$
4: **for** each renewal **do** (n_3 times)
5: **for** each replica **do**
6: **for** each spin **do**
7: propose a flip
8: calculate energy increase ΔE after the flip
9: **if** $\Delta E \leq 0$, accept spin flip
10: **else if** $\exp(-\Delta E/T(t)) >$ random [0,1]
11: **then** accept spin flip
12: **end for**
13: **end for**
14: $t = t + 1$
15: **end for**
16: **end for**

For each replica, we renew the spin states n_3 times (line 4). In line 8, we calculate the energy difference ΔE before and after spin flipping. In line 10, random [0,1] generates a uniform random number between zero and unity, yielding thermal-noise-based fluctuations. Since its searching ability depends on the suitability of the temperature control, we need an optimized temperature function $T(t)$.

3. Numerical demonstration

3.1. Max-cut problem

Let V and E be the sets of vertices and edges, respectively, that form an undirected graph. Max-cut problems determine a subset S which maximizes the sum of the weights w_{ij} (= w_{ji}) of the edges crossing S to its compliment $V \setminus S$. Setting $w_{ij} = 0$ for $(i,j) \notin E$, the objective function to be maximized is given by

$$\sum_{i\in S}\sum_{j\in V\setminus S} w_{ij} = \frac{1}{4}\sum_{i,j\in V} w_{ij} - \frac{1}{4}\sum_{i,j\in V} w_{ij}\sigma_i\sigma_j \,. \qquad (5)$$

Here, $\sigma_i = +1$ for $i \in S$ and $\sigma_i = -1$ for $i \in V \setminus S$. Since the first summation on the right-hand side of Eq. (5) is constant, minimizing the second summation is crucial; therefore, our goal is to find the ground state of Eq. (3) at $J_{ij} = -w_{ij}$. We prepare 100 such problems by randomly but evenly choosing $w_{ij} = 0,1$ for $i \neq j$ for algorithm evaluation. We set the problem size as N = 100, which corresponds to a graph with 100 vertices.

3.2. Success probability

Setting $n_1 = 1900$ and $n_2 = 100$ for Algorithm 1 and $n_3 = 2000$ for Algorithm 2, we compare their success probabilities under the condition that $n_1 + n_2 = n_3$. For Algorithm 2, we use the temperature function

$$T(t+1) = \gamma T(t). \qquad (6)$$

Here, we set the parameter γ and the initial temperature $T(0)$ as 0.9999 and 10, respectively. We solve 100 max-cut problems with the number of replicas set to K = 100. Fig. 1(a) shows the mean success probabilities (100 problems = 100 results), each of which is obtained by solving the same problem 10 times. Fig. 1(b) shows the raw success probabilities (100 problems × 10 times = 1000 results).

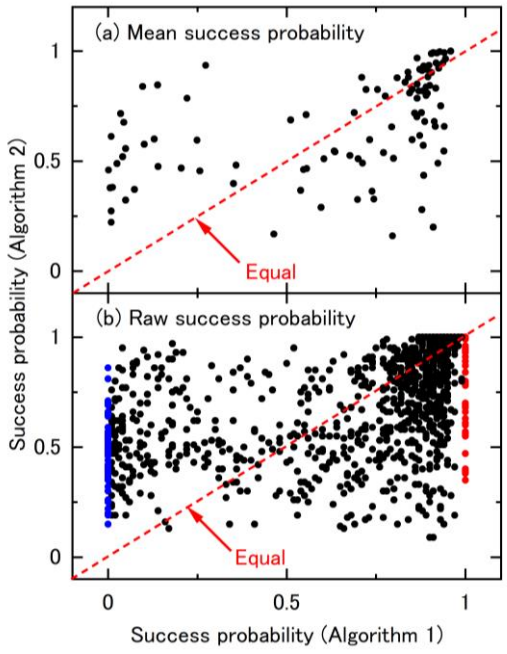

Fig. 1 Results of 100 max-cut problems.
(a) Mean success probability (100 results)
(b) Raw success probability (1000 results)

In Fig. 1(a), the chance that Algorithm 1 presents a higher or equal success probability is 49%. In Fig. 1(b), it becomes 51.5%; thus, both algorithms have similar searching abilities. However, a better choice of $T(t)$ may improve the performance of Algorithm 2, although finding such a $T(t)$ is a time-consuming task. It is natural to consider that the difference in success probabilities obtained for the same problem originates from the different search routes given by the different methods, which yield different fluctuations. In Fig. 1(b), one- or zero-value results were obtained from Algorithm 1 (see red and blue solid circles). This is a sign that replica-replica interactions based on Table 1 induce replicas to form a swarm. If all replicas reach the ground state, the success probability becomes unity. On the other hand, if they are trapped at nearby states, it becomes zero. This swarm-like movement is emphasized by decreasing the number of replicas K. For example, Fig. 2 was obtained from Algorithm 1 when K = 25, 50 and 100. Here, we again solved 100 max-cut problems 10 times for each case. Although results are not shown, we always observed a single swarm when K = 10. On the other hand, no such grouping occurred when using Algorithm 2. We will discuss the relation between the number of replicas K and swarm formation in the next section.

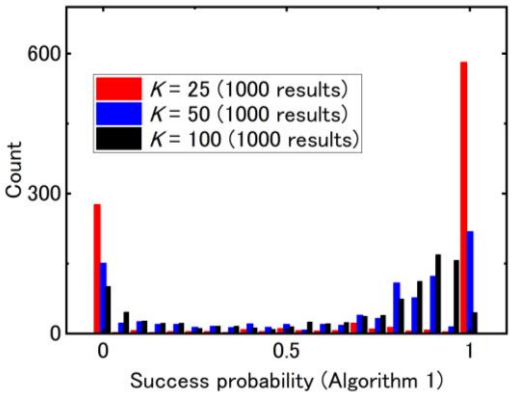

Fig. 2 Success probability (Algorithm 1).
K = 25 (red), 50 (blue) and 100 (black)

4. Discussion

4.1. Number of replicas K

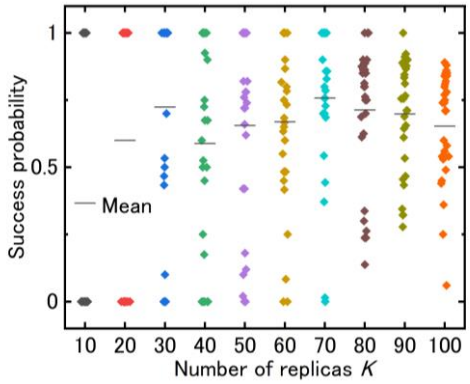

Fig. 3 K-dependent success probability.

Fig. 3 gives an example of the K-dependent success probability. We solve the same problem 30 times; thereby, the number of results plotted for each K is 30 and each bar represents their mean value. At K = 10 and 20, we always observed one- or zero-value results. For instance, a mean success probability of 60% obtained at K = 20 means that all 20 replicas form a single swarm that settles into the ground state 18 out of 30 times. In this case, Table 1 surely works as swarm-like replica-replica interactions to induce replica states to become identical as annealing proceeds. For $K < 50$, a single swarm was formed with high probability whereas multiple swarms were observed for $K > 50$, which suggests that replica-replica interactions based on Table 1 are not strong enough to form such a large swarm. In this case, some replicas collectively reach the ground state, while others stay in states near the ground state (the second or third lowest state in most cases). Of course, the swarm size strongly depends on Bob's coin tossing; therefore, we have a nonzero chance that all replicas form a single swarm even when K exceeds 50.

4.2. Limitations

We confirmed that swarm-like replica-replica interactions based on Table 1 induce replicas to collectively and cooperatively search for the ground state as annealing proceeds. However, Algorithm 1 forces Bob to stop tossing a coin once all replicas fall into the same state, no matter what this state is. If all replicas are trapped and frozen at some nearby state before ever reaching the ground state, there is no driving force to escape this local minimum since no further fluctuations are created. In this case, we have to restart the algorithm with new random initial states. The timing of freezing strongly depends on Bob's coin-tossing results. However, if a relatively large number of replicas is chosen (K = 100, for example), the chance that all replicas have the same state becomes small, and we can avoid this type of forced stop.

Meanwhile, swarm-like formation never occurs in Algorithm 2 since each replica acts separately and independently while receiving thermal-noise-induced fluctuations. The success probability in this case depends on the suitability of the temperature control. Of course, we also have a small but nonzero chance that all replicas reach the ground state by chance.

5. Conclusions

We proposed and demonstrated a PIQMC-based simulated annealing method for temperature-control-free annealing. Replicas help each other to search for the ground state of an Ising model in a manner similar to a swarm (or multiple swarms in the case of large K values) acting cooperatively. Their interactions are local, simple and fluctuate to a certain degree. Our two-party rule solved 100 max-cut problems, each of which corresponds to a graph of 100 vertices (a 100-spin Ising model). The same problems were also solved using the Metropolis algorithm, whose searching ability depends on the suitability of the temperature control. It turns out that the two algorithms tend to have similar search abilities. However, different success probabilities were obtained for the same problem; therefore, the algorithms may complement each other in improving the accuracy of ground-state searching.

For cooling replicas, our simulated annealing used the Metropolis method with zero temperature. Of course, this cooling method can be replaced with other (possibly more efficient) cooling algorithms (for example, the r-fractional update algorithm that was initially developed for the parallel simulation of the Hopfield neural network model [6]). Although replica-replica interactions were quite simple, and interactions were limited to between nearest neighbors, it may be interesting to modify such interactions if we can increase the success probability.

In the present work, we solved only 100 max-cut problems. The number of vertices (spins) were limited to 100 due to the performance limit of our PC. We will upgrade our PC performance and investigate a variety of max-cut problems to more precisely evaluate our method.

Acknowledgments

This work was supported in part by the Quantum Leap Flagship Program (MEXT Q-LEAP).

References

[1] S. Kirkpatrick, C. D. Gelatt, Jr., M. P. Vecchi, "Optimization by simulated annealing," *Science* vol.220, pp.671-680, 1983.

[2] T. Kadowaki, H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev.* E58, pp.5355-5363, 1988.

[3] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, G. Rose, "Quantum annealing with manufactured spins," *Nature* vol.473, pp.194-198, 2011.

[4] R. Martoňák, G. E. Santoro, E. Tosatti, "Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model," *Phy. Rev.* B66, 094203, 2002.

[5] T. Albash, D. A. Lidar, "Demonstration of a scaling advantage for a quantum annealer over simulated annealing," *Phy. Rev.* X8, 031016, 2018.

[6] D. H. Ballard, P. C. Gardner, M. A. Srinivas, "Graph problems and connectionist architectures," *Comp. Sci. Dept., University of Rochester,* TR 167, pp.1-24, 1987.