

Quantum computation of Restricted Boltzmann Machines by Monte Carlo Methods

Yunkai Yao

School of Mathematics and Data Science, Minjiang University, Fujian, China

463244128@qq.com

Abstract. In recent years, the diversification of problems that require computers to solve has attracted attention to the construction of meta-heuristics that can be applied to a wide range of problems, and to specialized computers that implement these meta-heuristics in their devices. The representative meta-heuristics are Simulated Annealing (SA) and its extension to quantum computation, Quantum Annealing (QA), and its path-integral Monte Carlo method for classical simulation Crosson and Harrow showed that for certain problems where QA outperformed SA, SQA achieved performance close to that of QA, and SQA sometimes outperformed SA by an exponential time factor. On the other hand, it remains unclear whether SQA can work efficiently on a wide range of other problems. In this study, we experimentally compared SA and SQA on instances of the restricted Boltzmann machine RBM, known as a fundamental building block in deep learning, and 3SAT, a fundamental combinatorial optimization problem. The results show that SQA gives slightly better solutions than SA as the problem size increases for RBM in terms of both accuracy and computation time in our setting, but the opposite trend is observed for 3SAT, indicating that there is no significant difference between the two methods. From the viewpoint of artificial intelligence research, it is necessary to further examine whether deep learning can be made more efficient by applying QA and SQA to RBM.

Keywords: Restricted Boltzmann Machine, Simulated Annealing, Quantum Annealing, Deep Learning.

1. Introduction

Quantum computation is a computational paradigm that is expected to make possible calculations of a scale that is difficult to achieve with classical computation by using massively parallel computing based on the principles of quantum mechanics. Various models have been proposed for quantum computation, and besides the quantum circuit model, which is probably the best known, a wide variety of computational models exist. Among these, it is particularly promising as a framework for solving optimization problems at high speed, and has attracted considerable attention in recent years, as reported by Dwave, which has actually realized a small-scale quantum computer based on a quantum annealing framework. Quantum annealing is an extension of the (classical) annealing method (Simulated Annealing, SA) proposed by Kirkpatrick, Gelatt, and Vecchi, using quantum parallelism to solve a wide range of optimization problems, especially combinatorial optimization problems, at high speed. meta-heuristics. Quantum annealing method is faster than the classical annealing method, and it has been shown theoretically for several important cases. For example, if the energy gap of the Hamiltonian is with n as the input size of $\frac{1}{poly(n)}$, the In the case of quantum annealing, the quantum annealing method is more effective than the classical annealing method. It is known to compute the desired solution exponentially faster For certain types of objective functions, quantum annealing The superiority of the method has been demonstrated. In addition, the quantum annealing method is exponential time-consuming.

On the other hand, Crosson and Harrow demonstrated that meta-heuristics path-integral quantum Monte Carlo (in the framework of classical computation) achieves computational speed comparable to that of quantum annealing for a problem involving Hamming weights, where the quantum annealing method has been known to have an advantage. integral Quantum Monte Carlo. This method is one of the classical computational meta-heuristics called Simulated Quantum Annealing (SQA),

and is based on the Markov Chain Monte Carlo method, which uses the path integral process to approximate the distribution function of the Hamiltonian obtained by the Suzuki and Trotter expansion. The method of generating a random walk classically mimics the behavior of the quantum annealing method. In the following, the path integral quantum Monte Carlo method will be referred to as Simulated Quantum Annealing (SQA). As mentioned above, SQA is an algorithm that classically mimics the quantum process of computation. Therefore, the question naturally arises whether the superiority of quantum annealing over classical annealing can be resolved by a clever reduction of the quantum computational process to classical computation, as reproduced by SQA. While SQA can efficiently mimic the behavior of QA for certain objective functions (such as those corresponding to weights), it is not clear whether SQA can be as computationally efficient as QA for a wider range of functions, and thus whether SQA is truly efficient compared to SA. It is not yet known.

Therefore, in this study, we have developed a more broadly based experimental evaluation was conducted on a wide range of problems. The experiment was conducted using the problem was to find the basic belief network of the deep belief network in deep learning. A component of the Restricted Boltzmann Machine (RBM) and combinatorial optimization problems. The number of clauses in the satisfying logical formulas problem, which is a fundamental problem, is limited to 3. The 3SAT instance was used. In this setting, SQA has a clear advantage over SA. The results did not show any significant difference between the two. Specifically, the restricted Boltzmann machine, SQA reaches a slightly better solution. However, it was not a significant difference. In 3SAT, SA was faster and more frequent than the results show that the original problem was satisfied with a higher degree of success. However, in another of our experiments, we used the basic combinatorial optimization problem. In the case of the Max cut problem, which is known as SA is faster than SQA in finding good solutions. The results also show that the "M" in "M" is a good indicator of the quality of the product.

These results suggest that SQA is more capable than SA for certain problems, but not for general problems.

2. Experiment Summary

2.1 SA and SQA Overview

Both SA and SQA are algorithms that use a random walk to find the globally optimal solution with respect to an arbitrary objective function f . Here we present an overview of both algorithms as a problem of minimizing f . It should also be noted that both algorithms have their own parameters, and the performance of the algorithms depends greatly on the tuning of these parameters.

Simulated Annealing (SA)

Parameters: T_0 , T_f

1. Select a solution S , the current solution $S_C = S$, and the temperature parameter Set the tab $T = T_0$
2. Find a neighbor S_N of the current solution S
3. The probability $\min(1, \exp(\frac{f(S_C) - f(S_N)}{T}))$ for the current solution Change S_C to S_N
4. Reduce the temperature according to the determined schedule T
5. Return to 2 if temperature is not $T < T_f$

Simulated Quantum Annealing (SQA)

Parameters: G_0 , G_f , P , T

1. Select one solution S and set $G = G_0$
2. S Replicate and slice $S_i (i \in \{1, \dots, P\})$ is created. The following table shows the results of the process. Each slice is then used as the current solution $S_{iC} (i \in \{1, \dots, P\})$
3. Find the S_{iN} in the vicinity of each S_{iC} slice.
4. Probability $\min(1, \exp(\frac{f(S_{iC}) - f(S_{iN}) + J_P}{PT}))$, the current Change the current solution S_{iC} to S_{iN} (J_P is the boundary condition and value determined by G , called G)
5. Find the S_{iN} in the vicinity of each S_{iC}
6. Probability, and for all i , the change from S_{iC} to S_{iN} to be
7. Decrease G according to a set schedule make
8. If $G < G_f$ then go back to 2

T and G are parameters specific to the annealing method. If these parameters are large, transitions to solutions with bad values can occur frequently. On the other hand, if the parameters are small, most of the transitions are to solutions with good values. By setting these parameters appropriately, each algorithm can reach the global optimum without being stuck in the local optimum. In SA, the probability of a transition is determined by the difference between the current solution and the candidate solutions, while in SQA, in addition to the difference between the solutions, a constant called the boundary condition J_P . See 2.2 for details.

2.2 Experimental Technique

The Ising model plays an important role in this experiment. The following is a summary of the results of the study. The Ising model was originally used in statistical mechanics to study ferromagnetism. It was introduced to model the computer science and discrete mathematics. The number of people who have been affected by the earthquake has been increasing. One of the reasons for this is that the various combinations of The fact that the problem of combinatorial optimization is expressed in the form of an Ising model QA is essentially a basis for the Ising model. state (i.e., the solution with the minimum energy state). QA is an algorithm for combinatorial optimization problems. The quantum computer that solves each problem is called an Ising The calculation is performed after transforming the data into the form of a This shows that As can be seen, the Ising model is also a major The importance of this concept is clear from the fact that it is not only a matter of "how" but also "when. In addition, the fundamental The restricted Boltzmann machine, which is a building block, and the combinatorial The most computationally difficult problem in the optimization problem, called NP-Hard, is the NP-Hard problem. Max cut, which is the most basic of these problems, is a simple the ground state on the Ising model by performing a simple transformation. It is known that this is equivalent to the problem of finding the Our In our experiments, we focused on restricted Boltzmann machines. performance of SA and SQA for various combinatorial optimization problems. To evaluate each problem in a unified manner, the Ising model SA and SQA after transforming it into the problem of finding the ground state The performance evaluation is conducted by applying the In addition, the implementation of SA and SQA The code is based on the code by Hazard with some modifications. The following table shows the results of the analysis.

Next, a brief explanation of the Ising model is given. Izzy A nesting model is one in which the points and branches are assigned real-valued weights. It is an undirected graph with n points. The weights of h_i and $J_{\{i,j\}}$ are assigned to point i and branch $\{i, j\}$, respectively. The following is an

example of a case in which a user is assigned a Spin on the Ising model A "span" is a state consisting of binary values assigned to each point. and $\sigma_i \in \{-1, 1\}$ is assigned to point i. The given For a given spin, the energy corresponding to that spin The energy E is given by In the Ising model minimization problem is the problem of minimizing the energy The problem is to find a spin.

$$E = \sum_{i,j}^n J_{ij} \sigma_i \sigma_j + \sum_{i=1}^n h_i \sigma_i$$

The boundary condition at the kth slice can be expressed as follows.

$$J_p = -PT \log \left(\tanh \left(\frac{G}{PT} \right) \right) \sigma_{i,k} (\sigma_{i,k-1} + \sigma_{i,k+1})$$

We denote the spin of point i in the kth slice by $\sigma_{i,k}$ and define $\sigma_{i,0} = \sigma_{i,P+1}$. It should be noted that the values of the boundary conditions are derived from the approximation of the distribution function of the Ising model. The algorithm uses a 1-bit bit inversion to find the nearest neighbor solution, and $A = \frac{1}{1+s} A (A = T \text{ or } G)$, where s is some constant, is used as the schedule for decreasing T and G.

SQA has two phases: one to perform transitions for the entire slice and another to perform transitions within the slice. Therefore, in order to fairly synchronize the number of times SA and SQA search and update the solution in the neighborhood, SA shall be as follows.

2.3 Data used in the experiment

In this experiment, we compared SA and SQA in the Restricted Boltzmann Machine. as described in 2.2, we transformed instances of the Restricted Boltzmann Machine and 3SAT into the Ising model and compared them to the The performance of SA and SQA is evaluated by applying SA and SQA to the The results are shown below.

The restricted Boltzmann machine (RBM) is a point- and branch-weighted energy minimization defined on an undirected bipartite graph with problem. First, for a point i, θ_i , and for a branch $\{i, j\}$ The weights $\omega_{\{i,j\}}$ are given respectively. Furthermore, the Ising model for point i, as well as the spins assigned to the points of type The state $s_i \in \{0, 1\}$ is given. Given a state The energy is given as follows.

$$E = - \sum_{i,j}^n \omega_{ij} s_i s_j - \sum_i^n \theta_i s_i$$

The transformation from a restricted Boltzmann machine to an instance of the Ising model is done as follows First, halve the weights of all branches in the graph. Then, add a point v to the graph, add a branch between each $k \times k$ point in the original graph and the newly added point v, and set the weight of the branch $\{v, i\}$ to $\frac{\theta_i}{2}$. Finally, the weights of the points are all set to 0. If the resulting graph is used as an Ising model with $2k + 1$ points, the energy of the ground state is found to be equal to the optimal value of the restricted Boltzmann machine plus the sum of the weights of the points and branches of the restricted Boltzmann machine.

The instances of the restricted Boltzmann machine used in the experiments were generated as follows. First, $k \times k$ bipartite graphs were generated for $k = 50, 100, 200, 300, 400$, with $\frac{k^2}{4}$, $\frac{k^2}{2}$ and $\frac{3k^2}{4}$ as the number of branches. Assign a probability of -1 or 1 for the branch weights and -1 or 1 for the point weights, each with a probability of 12. For each instance of the restricted Boltzmann machine generated in this way, the upper bound of its optimal solution has been obtained by Yasuda using SDPA (Semidefinite Programming Algorithms).

We will also discuss the transformations for 3SAT. The conversion is based on a method for converting 3SAT to Max cut is described. However, to simplify matters after the conversion, some of this conversion was changed uses a problem called NAE SAT, which requires not only one or more literals in a clause to be true but also one or more to be false in order for a clause to be satisfiable.

This transforms $3SAT \rightarrow NAE4SAT \rightarrow NAE3SAT \rightarrow \text{Max cut}$, except for NAE 3SAT to Max cut, which uses the method used in, so here we use.

Only the conversion from NAE 3SAT to Max cut is explained. First, for each logical variable x in NAE 3SAT, let x and $\neg x$ be points in the graph. Then, let S be the set of logical variables for which the point is true, and assign branch weights so that the cuts in S correspond to the number of satisfied nodes in NAE 3SAT. Specifically, since vertices x and $\neg x$ can never be true at the same time, these two vertices are given edges and weights M . In each clause, the number of positive literals and the number of negative literals must be 2 on one side and 1 on the other in order to be satisfied. Then, consider a triangle connecting three literals in a clause by an edge of weight 1. In this case, when the cut is satisfied, the weight of the cut is 2, but when it is not satisfied, the weight is always 0. Using these, if the number of logical variables in NAE 3SAT is k and the number of clauses is c , then the maximum cut is $kM + 2c$. Then, the original NAE 3SAT is satisfied when there exists a cut that satisfies it.

On the other hand, we tried to avoid having the number of vertices be twice the number of logical variables in NAE 3SAT. Specifically, using after creating the graph, the edges between vertices x and $\neg x$ corresponding to the logical variable x were contracted, and the x and $\neg x$ vertices were combined into a single vertex. In doing so, the sign of the edge connected to the vertex $\neg x$ is reversed. With such an operation, a clause with both positive and negative logical variables as literals has a cut weight of 0 when satisfied and a cut weight of -2 when unsatisfied. On the other hand, clauses that have all positive or negative logical variables as literals have a cut of weight 2 when satisfied and a cut of weight 0 when not satisfied.

From this, it is equivalent that there are m true clauses in NAE 3SAT and that the size of the cut is $2(m - (\text{the number of clauses including both positive and negative literals}))$. m being the number of clauses in all of NAE 3SAT, the transformation to the maximum cut problem is possible.

For instances of 3SAT that can be converted to a maximum cut problem by setting m to be the number of clauses in all NAE 3SATs, we used those in. All satisfiable items were used. Therefore, all optimal solutions are known. We used one uniformly random and one uniformly random with what is called a backbone. A backbone is a set of literals that satisfies the condition that if all elements of the set are true, then the SAT is necessarily satisfied. It is also known from that for a random 3SAT, $\#\{\text{clauses}\} = 4.26 \times \#\{\text{variables}\}$ is the most difficult. In this case, the idea was to perform comparisons on difficult instances. So, the instances handled are all variables with ratios and number of clauses that are close to this equation. The following table shows the results of the analysis. The number of logical variables in an instance is 20, 50, 10 The number of items handled were 150, 200, 250, 600, 1000, and 2000. Also The instances with backbones are the instances of logical variables 100 and a backbone of 10, 30, 50, 70, or 90. The following table shows the results.

2.4 Experimental Evaluation Methods and Results

The number of steps of 3 to 6 in SQA and 2 to 4 in SA are The speed of SA and SQA is called one step. The evaluation of the degree of performance is based on a comparison of the number of steps. For the parameters of the SQA, we take $P = 1, 40, 80$, and T is set so that $P T = 1$ for SA. The parameter T is matched to G in SQA. This is the case. The experimental results obtained under these settings are shown in Table 1.

3. Summary

In this study, we compare SA and SQA, meta-heuristics methods in classical computation, for the problems of restricted Boltzmann machines and 3SAT. Although some differences were observed in the accuracy of the solutions obtained, these differences were not considered to be significant in nature. Furthermore, for 3SAT, SA was able to reach the optimal solution for more problems than SQA, and found a solution that satisfied the original problem, although the difference as a solution of the Ising model was slight. In addition to the restricted Boltzmann machine, experimental

performance evaluations of SA and SQA for various other combinatorial optimization problems were also conducted, which were not included in this paper. According to the results, it was experimentally confirmed that SA can obtain more accurate solutions to the problem of finding the max cut on a graph, which is known to be equivalent to the Ising model and the Boltzmann machine, assuming the planarity of the graph, and that SA can also solve the problem for which fast approximate computation methods exist, although exact computation is computationally difficult. The SQA is more efficient than the SA for vertex coverings.

It is important to note that the performance of SQA and SA may depend on the nature of the graph on which the problem is defined. Based on the results of this experiment, further experiments on the behavior of SA and SQA for problems on bipartite graphs, such as restricted Boltzmann machines, with parameters from actual deep learning are warranted. Calculations on graphs that satisfy planarity. The possibility that SQA is faster for The input instances in which SQA and SA are good at each are shown in Table 1. Theoretical analysis based on the results of this experiment is an issue for future study.

References

- [1] R. Martoňák, G. E. Santoro and E. Tosatti, "Quantum annealing by the path-integral Monte Carlo method: The two-dimensional random Ising model", *Phys. Rev. B*, vol. 66, pp. 094203, Sep 2002.
- [2] E. Farhi, J. Goldstone and S. Gutmann, "Quantum adiabatic evolution algorithms versus simulated annealing", 2002.
- [3] Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." *Proceedings of the 24th international conference on Machine learning*. 2007.
- [4] Huang, Li, and Lei Wang. "Accelerated Monte Carlo simulations with restricted Boltzmann machines." *Physical Review B* 95.3 (2017): 035105.
- [5] Lu, Na, et al. "A deep learning scheme for motor imagery classification based on restricted Boltzmann machines." *IEEE transactions on neural systems and rehabilitation engineering* 25.6 (2016): 566-576.
- [6] Kirkpatrick, Scott, C. Daniel Gelatt Jr, and Mario P. Vecchi. "Optimization by simulated annealing." *science* 220.4598 (1983): 671-680.
- [7] Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: an experimental evaluation. part i, graph partitioning. *Oper. Res.* 37(6), 865–892 (1989)
- [8] Berne, Bruce J., and D. Thirumalai. "On the simulation of quantum systems: path integral methods." *Annual Review of Physical Chemistry* 37.1 (1986): 401-424.
- [9] D. Steurer: Reduction from 3SAT to max cut, lecture notes, Introduction to Analysis of Algorithms CS4820 Cornell University, march 2014.
- [10] Kostuk, Mark, et al. "Dynamical estimation of neuron and network properties II: path integral Monte Carlo methods." *Biological cybernetics* 106.3 (2012): 155-167.