Microcanonical Mean Field Annealing: A New Algorithm for Increasing the Convergence Speed of Mean Field Annealing.

Hyuk Jae Lee and Ahmed Louri

Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, 85721

Abstract

In this paper, we consider the convergence speed of mean field annealing (MFA). We combine MFA with microcanonical simulation(MCS) method and propose a new algorithm called microcanonical mean field annealing(MCMFA). In the proposed algorithm, cooling speed is controlled by current temperature so that computation in the MFA can be reduced without degradation of performance. In addition, the solution quality of MCMFA is not affected by the initial temperature. The properties of MCMFA are analyzed with a simple example and simulated with Hopfield neural networks(HNN). In order to compare MCMFA with MFA, both algorithms are applied to graph bipartitioning problems. Simulation results show that MCMFA produces better solution than MFA.

1 Introduction

Optimal solution is the main purpose of solving engineering problems. However, problems often have a large degree of freedom, so that optimal solution cannot be obtained in a reasonable amount of time [1]. A significant progress was made in this area with the invention of two new algorithms: simulated annealing(SA) and microcanonical simulation(MCS) [2, 3]. Both algorithms use the analogy between optimization problem and magnetic material in the statistical physics. SA has received more attention and successfully solves the optimization problem using the theory of interpreting magnetic material [1]. Another recent achievement was made by Hopfield neural networks(HNN) [4]. In the HNN, a neuron is modeled as an independent variable of the optimization problem just as an atom in the SA and the role of neuron is similar to that of atom. Therefore the search method of SA can be applied to HNN for updating the state of neuron. However, the probabilistic search rule of SA is difficult to implement in HNN, and often requires an excessive computation. One of the promising suggestion is a mean field annealing(MFA) which is more amenable to implementation by HNN and requires less computation than SA with nearly equal quality of solution [5].

MFA starts at high temperature, and slowly reduces the temperature. However, an excessive computation is required by slow cooling from high initial temperature. Previous research shows the existence of critical temperature, T_c, near which neurons begin to move to their final states. For the speed-up of MFA, it is desirable to increase the cooling speed above T_c , and decrease it near T_c . This adaptive cooling requires knowledge of T_c . Although efforts were devoted to find T_c for specific examples, there is no general technique to estimate T_c by analyzing the structure of HNN. However during the simulation of MFA, it is easy to determine T_c by observing the energy function.

CH 3065-0/91/0000-0941 $1.00 © IEEE

MCS is an algorithm that computes the energy function of HNN. We apply this property to control the cooling speed of MFA and propose a new algorithm called microcanonical mean field annealing(MCMFA). MCMFA adapts the cooling speed based on the current temperature, so that the amount of computation time is reduced without degradation of performance. Therefore MCMFA requires less computation than MFA. The adaptive cooling schedule also solves the problem of choosing the initial temperature. These properties are examined with analysis and simulation of MCMFA. MCMFA is applied to graph bipartitioning problems and results are compared with MFA.

2 Microcanonical Mean Field Annealing

MCS uses an imaginary particle called demon for simulating magnetic material [3]. The demon has deterministic energy, E_d, and flips the state of neuron. Statistical physics shows that the demon's energy will become Boltzmann distribution

$$Prob[E_d] \propto e^{-E_d/T} \tag{1}$$

where T represents the temperature of demon [3].

Then the deterministic update rule of MCS should be changed to probabilistic rule as follows:

$$\begin{aligned} Prob[s_i = +1/s_i = -1] &= Prob[E_d > H(s_i = +1) - H(s_i = -1)] \\ Prob[s_i = -1/s_i = +1] &= Prob[E_d > H(s_i = -1) - H(s_i = +1)] \end{aligned} \tag{2}$$

where s_i represents the state of i^{th} neuron and $H(s)$ represents the energy function of HNN. The update rule of Eq. 2 is similar to that of SA. However, in the MCS, the total energy is conserved. The decrease of HNN energy causes the increase of demon energy as follows:

$$\beta < E_d > <= \beta < E_d > -\Delta < H >, \tag{3}$$

where β is the number of demon and <> represents the average value. In the equilibrium state, the average demon energy is the same as temperature.

$$< E_d >= T. \tag{4}$$

Therefore Eq. 3 becomes:

$$T <= T - \Delta < H > /\beta. \tag{5}$$

We combine the cooling schedule of MFA with Eq. 5 to propose a new algorithm called microcanonical mean field annealing(MCMFA).

- **Microcanonical Mean Field Annealing Algorithm:**

1. *Initialization* : $T = T_{initial}$.
2. *while*$(T > T_{minimum})$
   (a) *select neuron i for update.*
   (b) *update* i^{th} *neuron as follows:*

$$< s_i >= \frac{1}{1 + e^{-\Delta < H > /T}} \tag{6}$$

(c) *adjust temperature as follows:*

$$T <= T - \alpha - \Delta < H > /\beta. \tag{7}$$

3. *all states are equilibrated to* $+1$ *or* -1 *according to their signs.*

The update rule of Eq. 6 is same as that of MFA. The difference is the cooling schedule of Eq. 7. In the MFA, α determines the cooling speed and quality of solution. The MCMFA adds $\Delta < H >$ term to the cooling schedule of MFA. β determines the influence of $\Delta < H >$ term on the cooling speed. If β is small, $\Delta < H >$ plays an important role on the cooling speed. As the value of β increases, the effect of $\Delta < H >$ term decreases. If β is infinity, MCMFA behaves as MFA.

3 Analysis and Simulation of MCMFA

In this section, we analyze the dynamics of MCMFA. We try simple case where $w_{ij} = w > 0$ for all i, j. Although this is not the general case, we can have a qualitative description of the behavior of general system from this specific example. In our case, the mean values of neuron states are all the same.

$$< s_i >=< s > \quad for\ all\ i. \tag{8}$$

In the equilibrium state, the temperature of neural networks should be identical to the temperature of demon. Therefore the mean value of final state is

$$< s >= \frac{1}{1 + e^{-2Nw<s>/<E_d>}}. \tag{9}$$

where N is the number of neurons. The average energy function becomes

$$< H >= -\frac{1}{2}N^2 w < s >^2 - N < s > \sum_i i_i. \tag{10}$$

Since our purpose is qualitative understanding of final state, a specific external input,i_i, does not make an important effect on our analysis. For the sake of convenient analysis, all external inputs are assumed to be '0'. Then

$$< H >= -\frac{1}{2}N^2 w < s >^2 . \tag{11}$$

From the energy conservation property, we have

$$\frac{1}{2}N^2 w < s >^2 = \beta(< E_d > - < E_{d,initial} >) - < H_{initial} > . \tag{12}$$

This equation can be solved graphically as shown in Fig. 1. The straight line corresponds to right hand side of Eq. 12 and the curve represents the left hand side. The initial state is reflected on the straight line and the final state is the intersection of two graphs. From Fig. 1, we can expect the behavior of MCMFA as follows. First, if the intersection of straight line and x-axis is greater than T_c, the final energy of HNN is '0'. The final temperature of demon is the intersection of the line and x-axis. Second, if this intersection is less than T_c, the final energy of HNN is less than '0'. If the slope of the straight line, β, is small, the final temperature is near T_c. As β grows larger, the final temperature becomes lesser. If β is infinity, then the behavior of MCMFA is the same as that of MFA.

MCMFA is simulated with HNN consisting of 100 neurons and 100 edges. The initial energy of HNN is chosen to be '0' and β is chosen to be '1'. Five temperatures are chosen to be initial

temperatures and the change of temperature is shown in Fig. 2. If the initial temperature is less than T_c, temperature converges near T_c. If the initial temperature is '2' which is greater than T_c, the temperature remains unchanged. This result corresponds to the graphical solution of Fig. 1.

The effect of β is also simulated and the result is shown in Fig. 3. The initial energy of HNN is chosen to be '0' and demon temperature is chosen to be '0.1'. If β is relatively small, the final temperature converges to critical temperature. As the value of β increases, the convergent temperature decreases. This simulation result is also same as what is expected from previous analysis.

4 Application

In this section, we apply MCMFA to graph bipartitioning problems. We investigate the quality of solution and convergence speed. We also apply MFA to graph bipartitioning problems and results are compared with MCMFA.

The energy function we use for graph bipartitioning problems is

$$H = -\frac{1}{2}\sum_i \sum_j s_i c_{ij} s_j + \frac{0.25}{2}\sum_i \sum_{j \neq i} s_i s_j \tag{13}$$

where

$$\begin{aligned} c_{ij} &= 1 \quad \textit{if nodes } i,j \textit{ of the graph are connected} \\ &= 0 \quad \textit{otherwise.} \end{aligned}$$

Our graph for the simulation has 100 neurons and 100 edges. These 100 edges are generated randomly. We tried this problem 100 times with different interconnections. The results are averaged over 100 trials. The detail of cooling schedule is described in what follows.

MFA : The cooling process starts with initial temperature '2' and stops at '0'. The initial temperature is chosen to be well above the critical temperature. A neuron is selected and its state is updated by Eq. 6. After state of neuron is updated, temperature is decreased by small amount α.

$$T = T - \alpha \tag{14}$$

In general, the convergence speed and solution quality depend on α . We try various values of α to compare the convergence speed and solution quality.

MCMFA : The initial temperature and stop condition are same as those of MFA. But Eq. 7 is used for cooling schedule instead of Eq. 14. As for β, we start with $\beta_{initial} = 0.1$ and increase it exponentially as cooling proceeds.

$$\beta = e^{\alpha}\beta_{initial}. \tag{15}$$

In order to compare MCMFA with MFA, we examine the quality of solution vs. the convergence speed. We observe the final value of energy function for the quality of solution and the number of iterations for the convergent speed. Fig. 4 shows that MCMFA produces better solutions than MFA. To reduce the energy of HNN to -42.0, MCMFA needs about 1400 iterations while MFA requires more than 2000 iterations. As for -43.0, MCMFA requires about 3000 iterations and MFA requires 5000 iterations.

In order to investigate the effect of $T_{initial}$, we try same problem with $T_{initial} = 0.4$. Fig. 5 shows the final value of energy function vs. number of iterations. With small $T_{initial}$, MFA converges quickly to poor solution. Thus the performance of MFA is better for the number of iterations is less than 2000. However, MFA cannot make any improvement by increasing the amount of computation. As for MCMFA, $T_{initial}$ does not make a large effect on the final solution. The

quality vs. iteration curve is similar to Fig. 4. MCMFA produces a solution with $T_{initial} = 0.4$ as good as with $T_{initial} = 2.0$. This simulation result shows that MCMFA does not have to start at temperature above T_c. Therefore, the problem of choosing $T_{initial}$ disappeared in the MCMFA.

5 Conclusions

In order to reduce the amount of computation, a new algorithm called MCMFA is proposed. We use the properties of MCS and combine it with MFA so that cooling speed can be scheduled adaptively during the annealing process. In the MCMFA, the cooling speed is increased at the temperature where computation makes a little contribution to final solution. Thus MCMFA with adaptive cooling schedule produces better solution than MFA with equal amount of computation. In addition, MCMFA solves the problem of choosing the initial temperature. If the initial temperature is lower than critical temperature, MCMFA can raise the temperature to critical temperature. Thus the initial temperature does not affect the quality of final solution. These advantages are justified by analysis and simulation experiments. We analyze the properties of MCMFA with simple HNN of which all interconnections have same weights. We also simulate the behavior of MCMFA with HNN. MFA and MCMFA are applied to graph bipartitioning problems and the speed and quality of solution are compared. Simulation results show that MCMFA requires less computation than MFA. If initial temperature is below critical temperature, MFA converges to local minimum and produces poor solution. But the solution quality of MCMFA is not affected by the initial temperature.

References

[1] E. Aartz and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley and Sons, Chichester, Great Britain, 1989.

[2] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing, " *Science* , vol. 220(4598), pp. 671-680, May 13, 1983.

[3] M. Creutz, "Microcanonical Monte-Carlo Simulation," Physical Review Letters, vol. 50, pp. 1411-1414, May, 1983.

[4] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. U.S.*, vol. 81, pp. 3088-3092, 1984.

[5] D. E. Van Den Bout and T. K. Miller, III, "Graph partitioning using annealed neural networks," *IEEE Trans. Neural Networks* , vol. 1, no. 2, pp.192-203, Jun. 1990.

[6] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the theory of neural computation* , Redwood, Addison-Wesley ,1991.

[7] G.E. Hinton and T.J. Sejnowski, "Learning and Relearning in Boltzmann Machines," in *Parallel Distributed Processing, Volume I*, Cambridge, MIT Press, 1986.

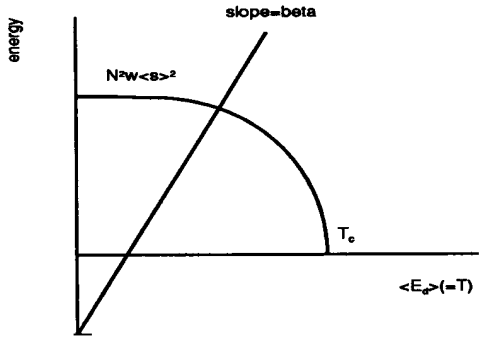

Figure 1: Graphical solution of final state.

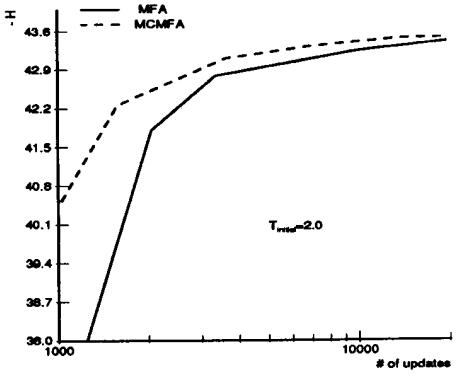

Figure 4: The performance comparison(initial temperature=2.0).

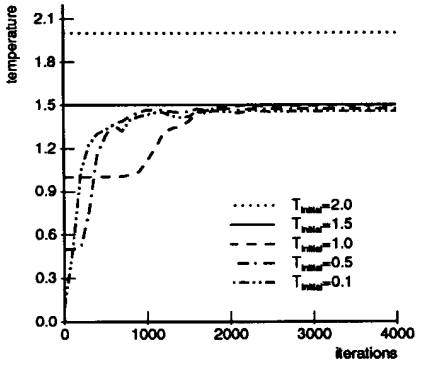

Figure 2: The change of temperature with various $T_{initial}$

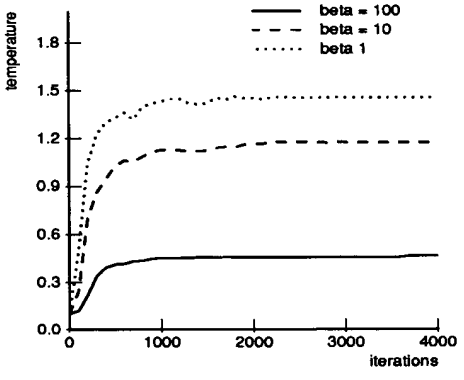

Figure 3: The change of temperature with various beta.

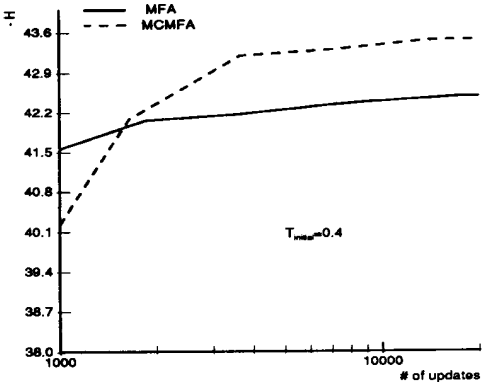

Figure 5: The performance comparison(initial temprature=0.4).