Parallel Implementation of One Boltzmann Machine's Algorithm

H. Zhu[1], F. Ren[2], N. Sun[3], K. Eguchi[3] and T. Tabata[3]
[1]Department of Computer Science, Faculty of Engineering,
Hiroshima Kokusai Gakuin University,
Tel. +81-82-820-2649, Fax.: +81-82-820-2640
E-mail : kohe-shu@m.ieice.org
[2] Dept. of Information Science and Intelligent Systems
Faculty of Engineering, The University of Tokushima
[3]Department of Information and Computer Sciences
Department of Electronic Engineering
Department of Electronic Control
Kumamoto National College of Technology

Abstract: Parallel-computation is very interesting topic. This paper describes that we apply it into the the Boltzmann machine with the Parallel-Transit-Evaluation Method successfully.

1. Introduction

The Boltzmann machine – introduced by Geoffrey Hinton, David Ackley and Terrence Sejnowski (1984) - is a kind of neural network which is called as a discrete time Hopfied network or as a stochastic network in which the neuron state is binary and decided by a propabability. It can rigorously established that the Boltzmann machine will probabilitically converge to a global optimum point. The Boltzmann machine can be quite useful for solving certain type of combinatorial problems [1], [2], [3]. And the Boltzmann machine can be used in the another fields of knowledge representation and learning [4], etc.

The huge runtime requirements of the Boltzmann machine have limited its applications in real cases. Long runtime is mainly due to the annealing task performed by the network. Annealing is the physical process of heating up a solid until it melts, followed by cooling it down until it crystallizes into a state with a perfect lattice. During this process, the free energy of the solid is minimized. Practice shows that the cooling must be done carefully in order not to get trapped in locally optimal lattice structures with crystal imperfections.

With the temperature of the simulated annealing lowered, the network selects randomly neuron and makes if it change its state or not in the light of a propabability. As seen above-mentioned, this processing must go on slowly to make the final stable configuration of the Boltzmann machine on the global configuration or nearest to it. Therefore, it is the important investigating topic for enhancing the processing speed. Some papers about it have been reported up to now [5], [6], [7], [8], [9], [10]. In the Reference [6], we have proposed a novel method—the Parallel-Transit-Evaluation Method, to resolve the problem of the long runtime requirement.

The Parallel-Transit-Evaluation Method is to divide all neurons into some groups, evaluate simultaneously neurons wanting to move their state in every groups and select one of candidates to move its state. The method has the efficiency of speed-up of the Boltzmann machine through the increment of number of candidates. But the method has been implemented at one computer. Here, for further speeding-up of the Boltzmann machine, we present one parallel implementation of the method with a lot of computers connected with Internet, called a cluster.

A cluster, consisting of computers connected with Internet, has rapidly grown in recent years and has a strong appeal to the researchers, because it has the characteristic of the power and low prices. There are some famous clusters in World, such as Avalon (http://cnls.lanl.gov/avalon/) of USA, RWCP (http://pdswww.rwcp.or.jp/) of JAPAN. With the idea of cluster, we constructed one neural-network-computer (*NNCt*), composed of 30 off-the-shelf Sun-Workstations, to implement the Boltzmann machine with the Parallel-Transit-Evaluation Method. With *NNCt* the efficiency of speed-up the Boltzmann machine was confirmed with an experiment of well-known combinatorial optimization problem–TSP (Travelling Salesman Problem). Moreover, NNCt has the features of no-cost and easy expansion of scale.

This paper is organized as: a description of the Parallel-Transit-Evaluation Method and its parallel implementation—*NNCt*, a presentments of some experimental results and a discussion and some recommendations about NNCt.

2. Pallel-Transit-Evaluation Method

For convenience, first we present an outline of the standard algorithm of the Boltzmann machine, and then give the parallel-transit-evaluation algorithm.

2.1 Standard Algorithm of the Boltzmann machine

Assume that the Boltzmann machine consists of a number N of neurons. The neurons' state is associated with binary value "0" or "1", corresponding to "off" or "on", respectively. The i-th neuron's state x_i is depend on its activity u_i and is decided as follows:

$$P(x_i = 1) = \frac{1}{1 + e^{(-u_i/T)}} \quad (1)$$

$$P(x_i = 0) = 1 - P(x_i = 1) \quad (2)$$

where, $P(x_i = 1)$ and $P(x_i = 0)$ are the probability that the i-th neuron's state x_i take "1" and "0", respectively. T is a parameter called the temperature, is from high to low to simulate the annealing process and affects the quantity of probability $P(x_i = 1)$ and $P(x_i = 0)$. At each T, the main procedure of the Boltzmann machine is as follows:

- Selecting a neuron randomly;
- Evaluating the selected neuron with the Equ.(1);
- Moving the neuron's state according to the evaluation.

2.2 Pallel-Transit-Evaluation Algorithm

For enhancing the processing speed of the Boltzmann machine, all neurons have been divided into some groups randomly, which of the block diagram is shown at Fig.1. Under the administration of the Controller In Group (CIG), every group does simultaneously:

- Selecting a neuron in the group randomly;
- Evaluating the selected neuron with the Equ.(1);
- Applying to the Controller For Groups (CFG) for the movement of the neuron's state according to the evaluation through $\mathtt{REQ}_j (j = 1 \sim M)$.

where M is the number of groups. The CFG – a "supervisory console",

- selects only one from these applications randomly;
- makes this neuron moving its state.

other neurons calculate the new activity in the meantime; and then the network goes to next cycle – repeating the above-mentioned processing. The follow chart about the Pallel-Transit-Evaluation Algorithm is shown at **Procedure**. This Pallel-Transit-Evaluation Method has accelerated the processing of neural network (four times as fast as the standard Boltzmann machine)[6].

Procedure

```
PTE_Boltzmann_machine
begin
 t:=0; { t:iteration index }
 for i:=1 to N do x_i(t):=initial; { initialize }
 for i:=1 to N do u_i(t) := Σ_{j=1}^{N} W_ij x_j(t);
 while (t <= L) do
  begin
   for l:=1 to M do
    begin
     k(l) := random(1 + (l − 1) * N/M, l * N/M);
     Evaluate k-th neuron on Eq.(1);
     if fire then x̂ := 1 else x̂ := 0;
     if ( x̂ =! x_k(l)(t − 1) )
     then c(l) := 1 else c(l) := 0{recorded}
    end;
   Select m from J ={ l | c(l) = 1 } randomly;
   if ( x_k(m)(t − 1) =! 1)
   then x_k(m)(t) := 1 else x_k(m)(t) := 0; {update}
   for i := 1 to N do u_i(t + 1) := u_i(t) + Δ_i;
   t:=t+1
  end
end
```

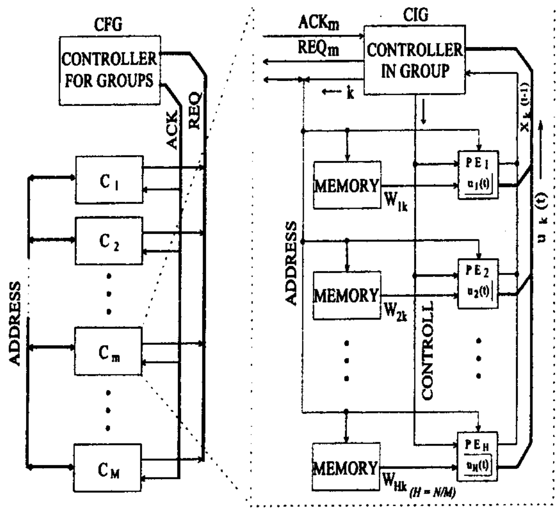


Figure 1. Block diagram of Parallel-Transit-Evaluation Method

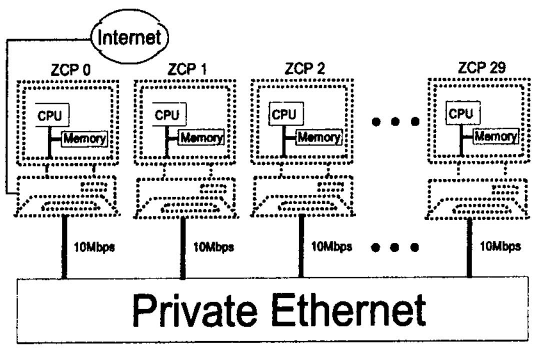


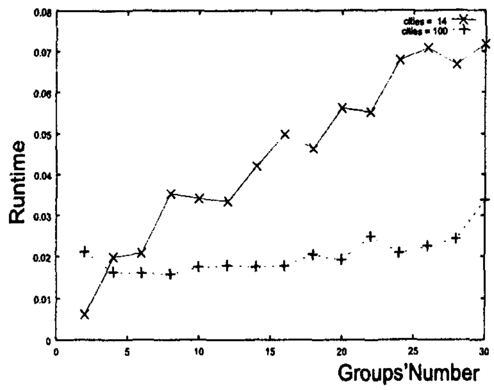
Figure 2. Block diagram of *NNCt*

3. Parallel Implementation

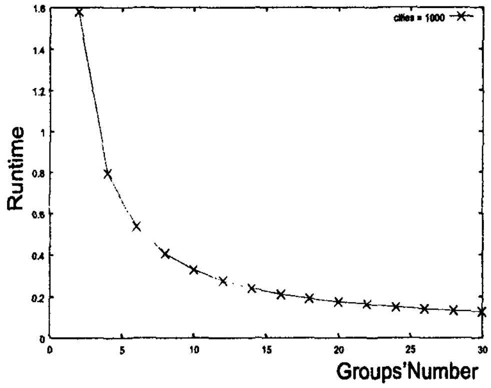
The experiments of the above-mention method have been implemented at one computer with one cpu. For further speeding-up and the preparation of integration of the Boltzmann machine, here we present a proposition that the Boltzmann machine with the Pallel-Transit-Evaluation Method is parallelly implemented by a lot of computers connected with Internet, which is called one neural-network-computer (*NNCt*).

The idea of *NNCt* comes from the cluster of computer science. The cluster, specially pc-cluster, consisting of computers connected with Internet, has rapidly grown in recent years and has a strong appeal to the researchers, because a parallel computing environment with a high-performance-computing is easily builded at a low price.

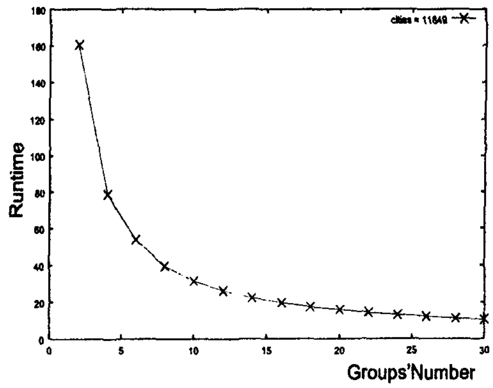
We have utilized the characteristic of the power & low prices and a similarity to human's brain to construct the *NNCt*, which is composed of 30 off-the-shelf Sun-Workstations (Sun SPARCstation LX). Each of the Sun-Workstations has:



(a) Cities=14 and 100



(b) Cities=1000



(c) Cities=11849

Figure 3. Relation of Runtime and Groups' Number

CPU:	SunGX
Motherboard:	Sun4m
Memory:	32MB
HDD:	1GB (SCSI)
Ethernet Adapter:	10Mbps

The *NNCt* is configured as one front-computer and some compute nodes. The compute nodes share a private ethernet; the front-computer is connected to the private ethernet and to the Internet. Of course, the front-computer partakes of parallel-computations. The block diagram of *NNCt* is shown at Fig.2, where ZCP0 is the front-computer and ZCP1~ ZCP29 are compute nodes.

Softwares of the *NNCt* have been free and for all computers are as following:

OS:	Debian GNU/Linux 2.2r2 (Sparc)
Message Passing:	MPICH, PVM, and LAM/MPI
Compiler:	GNU C, C++ and FORTRAN (g77)

With the same construction and softwares installed for each of computer, the *NNCt* is much similar to human's brain processing and of great advantage to expand scale of system

4. Experiments and Results

With the *NNCt*, some experiments of well-known combinatorial optimization problem–TSP (Traveling Salesman Problem) have been done. TSP is a classic of difficult optimization. The problem is to find a closed tour which visits each city once, returns to the starting city, and has a short (or minimum) total path length. Today, many researchers confirmed the performance of their propositions using it. We also have selected it to examine the *NNCt*. In the experiments, the ZCP0 (front-computer) plays the role of the CFG (Controller For Groups) and the ZCP1~ ZCP29 (compute nodes) play the role of the groups and the CIG (Controller In Group).

Cities:	14, 100, 1000, 11849
Start Temperature:	100
End Temperature:	0.1
Magnification of Temperature:	0.99

Fig.3 shows the relation between the runtime of one simulation of the Boltzmann machine with the Pallel-Transit-Evaluation Method for TSP and the number of groups divided, equalling to the number of compute nodes joining to parallel-computations. From Fig.3, it is clear that the speed-up of processing is greatly increased when the number of cities is over 100, because the computation requiring time is much much large than the communication requiring time.

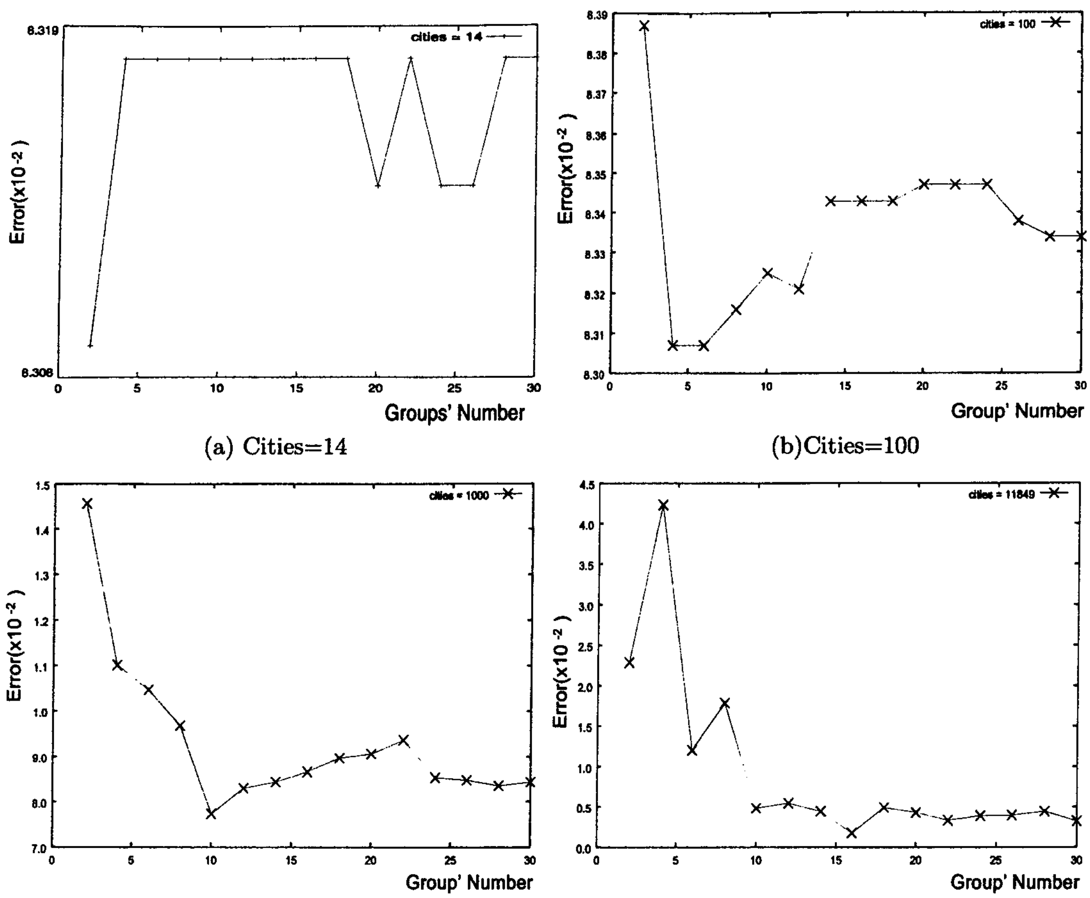
Fig.4 shows the relation between the errors of computation results and the number of groups divided. From Fig.4, it is evident that the errors go into decrement and near-stable status when the number of cities is over 100. The causation about it is studying.

5. Conclusion

In this paper, one parallel implementation of the Pallel-Transit-Evaluation Method have been proposed. And the efficiency of speed-up has been confirmed with the experiments of TSP.

References

[1] E. Aarts and J. Korst, "Boltzmann machines and their applications," *Lecture Notes in Computer Science, New York: Springer Verlag*, no.258, 1987.



(a) Cities=14 (b)Cities=100

(a) Cities=1000 (b)Cities=11849

Figure 4. Relation between Results' Errors and Groups' Number

[2] E. H. L. Aarts and J. H. M. Korst, "Combinatorial optimization on a Boltzmann machine," *Proc. European Seminar on Neural Computing*, London, UK, 1988.

[3] V. Zissimopoulos and V. Th. Paschos and F. Pekergin, "On the approximation of NP-complete Problems by Using the Boltzmann Machine Method: The Case of Some Convering and Packing Problems," *IEEE Trans. Computers*, Vol.40, No. 12, pp.1413-1418, 1991.

[4] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, D. E. Rumelhart, J. L. McCelelland, and the PDP Research Group, Eds., Cambridge,* MA: Bradford Books, 1986

[5] H.Zhu, M.Sasaki, F.Ueno and T.Inoue, "An approach of sequential-like parallel algorithm in Boltzmann machine," *Proc. of IEEE International Symposium on Circuits and Systems*, Vol.6, pp.439-442, 1994.

[6] H.Zhu and M.Sasaki, "Introducing a Pallel-Transit-Evaluation Method to the Sequential Boltzmann Machine," *IEICE Trans. Fundamentals*, Vol.J79-A, No.9. pp.1571-1581, 1996.

[7] H.Zhu, M.Sasaki, F.Ueno and T.Inoue, "A Parallel-Processing Structure for the Sequential Boltzmann Machine," *Proc. of The 1998 International Technical Conference on Circuits/Systems, Computers and Communications*, Vol.2, pp.1041-1044, 1998.

[8] H.Zhu, M.Sasaki and T.Inoue, "A Pipeline Structure for the Sequential Boltzmann Machine," *IEICE Trans. Fundamentals*, Vol.E82-A, No.6, pp.920-926, 1999.

[9] H. Zhu, M.Sasaki, N.Sun, K.Eguchi and T.Tabata, "A Improvement of Boltzmann machine," *Proc. of the 2001 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2001)*, Vol2, pp.1115-1117, 2001.

[10] H. Zhu, M.Sasaki, N.Sun, K.Eguchi and T.Tabata, "A Parallel Algorithm of Boltzmann Machine," *Proc. of the 2001 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2001)*, Vol2, pp.1122-1124, 2001.