DeMaMech 2005 Exchange Project Report

EU and Japan Pilot Cooperation in Higher Education

Program DeMaMech

Seigo Maekawa

Contents

1. Personal Data

Name: Seigo Maekawa

E-Mail: seigo@sophie.q.t.u-tokyo.ac.jp

Home Institute:

Name: The University of Tokyo

Faculty: Graduate School of Engineering

Department: Quantum Engineering and System Science

Address: Eng-Build. No. 12, 7-3-1 Hongo, Bunkyo-ku, Tokyo,
113-8656 Japan

Supervisor: Professor Dr. Masaharu Nakazawa
nakazawa@q.t.u-tokyo.ac.jp

Host Institute Main:

Name: Delft University of Technology

Faculty: Mechanical, Maritime and Materials Engineering

Department: Bio Mechanical Engineering

Address: Mekelweg 2, 2628 CD Delft, the Netherlands

Supervisor: Professor Dr. Tetsuo Tomiyama
t.tomiyama@wbmt.tudelft.nl

Host Institute 2nd:

Name: The Technical University of Denmark

Department: Manufacturing Engineering and Management

Address: Anker Engelundsvej 1, 101A, DK-2800 Kgs. Lyngby

2. Executive Summary

I left Japan for the Netherlands on August 16th 2005. I went to TU Delft to do my research for 4 months. I knew my supervisor, Prof. T. Tomiyama is famous in Engineering and he had been the professor of the University of Tokyo before a few years. I wanted to study about neural networks at his laboratory which made cellular machines system. Both cellular machines system and neural networks are kinds of the self-organization systems. I thought the research here made the neural networks, the theme of my master thesis more flexible and more meaningful.

First month, it is September, I had a meeting with my supervisor and I decided my theme of this project at TU Delft, which is about Deadlock Detection of Cellular machines using neural networks. Deadlock is one of the biggest problems for cellular machines. So it is necessary how to detect and to avoid from that. What I call cellular machines is the machines which has a pallet on itself and can communicate with its adjoining cells. That is to have only local information. These features are redundancy. Not to spoil these merits we suggest using neural networks from local information to detect the deadlock. On next 2 months I was learning the new programming language for me, Python, and made the simulation of the cellular machines then applied neural networks system to this simulation. And on the last month I wrote the report about the result and the study.

3. Travel Schedule

Date	From	To
16 August 2005	Tokyo	Amsterdam
23 December 2005	Delft	Paris
2 January 2006	Paris	Copenhagen
26 January 2006	Copenhagen	Tokyo

4. Research

Research technical report

Deadlock detection using neural networks

4.1. Abstract

This research proposes a new type algorithm to solve problems the present cellular machine have. The cellular machine consists of homogeneous cells with modularity and autonomy, and employs a local communication mechanism. In this distribution system deadlocks may generally happen. Therefore we need the algorithm to detect and avoid deadlock. The main purpose of this research is making the new algorithm that the each cell detects its own deadlock state by itself if possible, without global information.

4.2. Cellular Machines

Cellular Machines have the following three features.

1. Homogeneity.

   It is homogeneously composed of identical units called cells. Since homogeneity realizes redundancy of the entire system, this is indispensable for achieving fault-tolerance.

2. Extremely High Modularity.

   Each cell is an independent, autonomous module.

3. Intelligence.

   Each cell is equipped with its own CPU and has necessary intelligence. This is essential for making the system autonomous.

Cellular Automatic Warehouse

This chapter describes the “Cellular Automatic Warehouse” we developed as an example of the cellular machines. The total function of an automatic warehouse is to take in “packets” and store them until they are called for by a user.

Behavior of a Cell

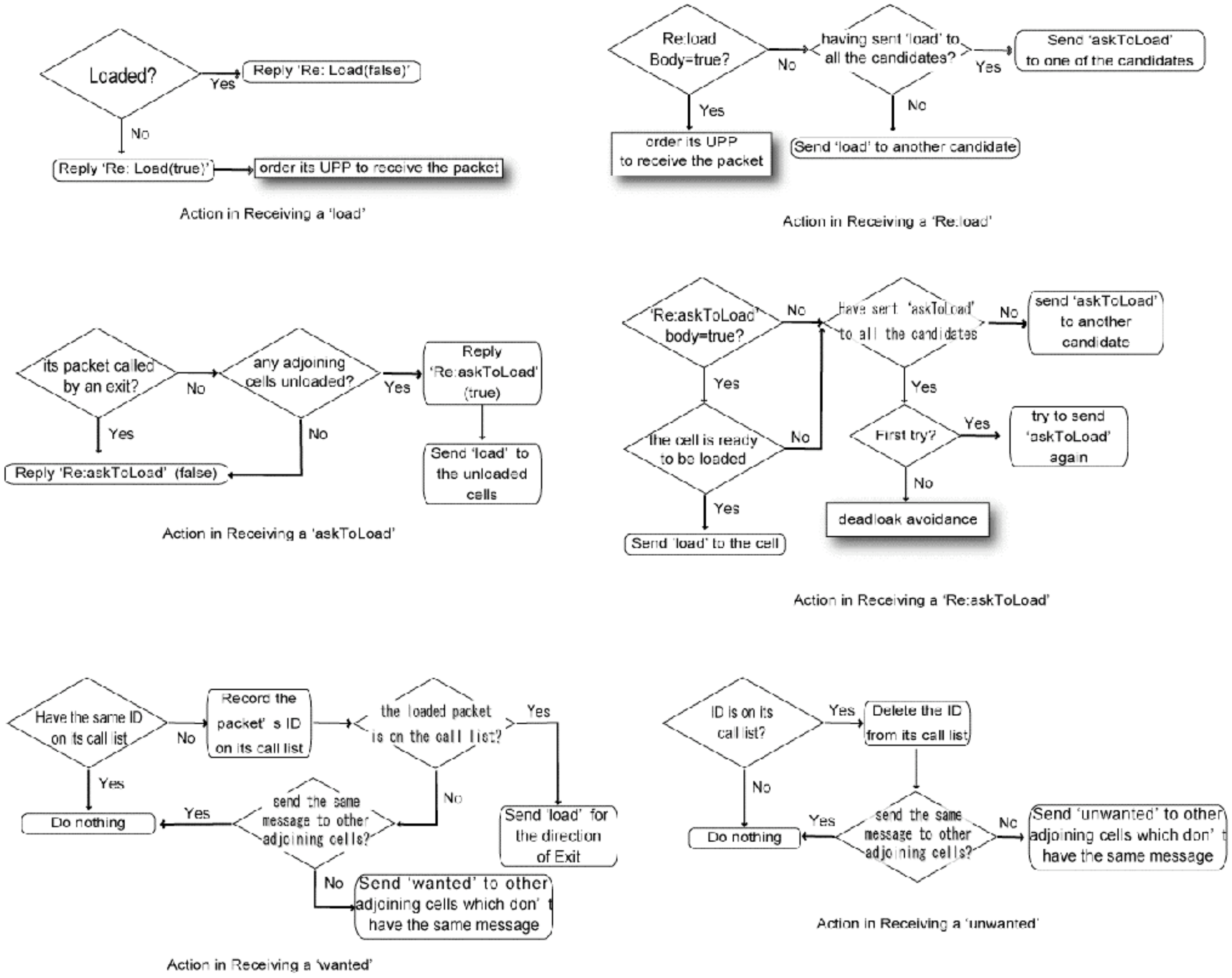
The cells behave according to exchanging message with each other. And the message use types listed in Tables 1. The messages of ‘wanted’ and ‘unwanted’ are issued by an exit cell in correspondence to the user’s request, and the other messages by any cells.

Table1 Types and Contents of Message

Message type	Contents	priority
Re: Load	Reply if it can receive the packet or not	1
Re: Ask to Load	Reply if it can remove its packet	2
Load	Ask if the adjoin cell can receive its packet	3

Ask to Load	Ask if the adjoin cell can remove its packet	4
Wanted	Transmit the ID called by the exit	5
Unwanted	Cancel an exit’s call for a packet	6

The action for each message is determined by the rules shown in Figures 2. A cell obtains a message with the highest priority from its message buffer.



Loaded?
Yes
Reply 'Re: Load(false)'
No
Reply 'Re: Load(true)'
order its UPP to receive the packet
Re:load Body=true?
No
having sent 'load' to all the candidates?
Yes
Send 'askToLoad' to one of the candidates
Yes
order its UPP to receive the packet
No
Send 'load' to another candidate
its packet called by an exit?
No
any adjoining cells unloaded?
Yes
Reply 'Re:askToLoad' (true)
Send 'load' to the unloaded cells
Yes
No
Reply 'Re:askToLoad' (false)
'Re:askToLoad' body=true?
No
Have sent 'askToLoad' to all the candidates
No
send 'askToLoad' to another candidate
Yes
Yes
The cell is ready to be loaded
No
First try?
Yes
try to send 'askToLoad' again
Yes
No
deadloak avoidance
Send 'load' to the cell
Have the same ID on its call list
No
Record the packet' s ID on its call list
the loaded packet is on the call list?
Yes
Yes
Do nothing
Yes
send the same message to other adjoining cells?
No
Send 'load' for the direction of Exit
No
Send 'wanted' to other adjoining cells which don' t have the same message
ID is on its call list?
Yes
Delete the ID from its call list
No
Do nothing
Yes
send the same message to other adjoining cells?
No
Send 'unwanted' to other adjoining cells which don' t have the same message

Action in Receiving a 'load'

Action in Receiving a 'Re:load'

Action in Receiving a 'askToLoad'

Action in Receiving a 'Re:askToLoad'

Action in Receiving a 'wanted'

Action in Receiving a 'unwanted'

Figure 2 Behavior rule of a cell

4.3. Neural Networks

Programming neural networks presents several problems, among them the algorithms which compute an output for the network and a method of training the network. This research, after a brief introduction to neural networks which can be skipped by those already familiar with the topic, describes code to compute with and train a particular type of neural network.

A neural network is a collection of simple processing units which are connected to each other. The output of one unit helps determine the output of other units in a potentially simultaneous system. The neural network originated as a model of neural activity and is inherently parallel, but can be modeled on a sequential machine. In this system the units are organized into layers; an input layer, a number of hidden layers, and an output layer.

Backpropagation

Backpropagation is the basis for training a supervised neural network. Static backpropagation is used to produce an instantaneous mapping of a static (time independent) input to a static output. These networks are often used to solve static classification problems such as pattern character recognition.

4.4. Report

First, I made the program of the cellular machine simulation. But at first it was not working because it had two main problems, which were just my mistakes. One was how to use “Resource” functions in SimPy. If you use the Resource functions such as “request”, “release” and etc, you have to use these functions only in the function which are activated by the main function. Since I didn’t do that, it couldn’t work accurately. The other was the cells don’t have their buffer on terms of making the algorithm. Then the message didn’t spread to all cells by overwriting the previous unread message. So I reformed the algorithm to have buffer of the message.

Second we used “bpnn” to implement back-propagation on neural networks. “bpnn” is named from Back Propagation on Neural Network. This tool is very easy to use backpropagation on Python. What you have to do is just to “import” this tool and to send input data and teaching signals to its class object. But in learning it by neural networks we should know whether the system is deadlock. We should judge whether the system state is deadlock or not by ourselves. So we decided that by settling the time span. If the called packet is not brought to exit within the settled time, we regard the state as deadlock. After all we judged it by initial state of the system. We can apply this method because our model of cellular machine has not randomness.

We simulated under the above mentioned condition. As a result, it can detect the deadlock under this condition.

But under this condition we don’t achieve the essential purpose of this research. So we need to change the following points;

1. Adding randomness to the cellular machine algorithm. You remake the program in order that the packets which are not called by exit move.
2. In this neural networks the input nodes connect all of the hidden nodes. This has no difference from the node gets the global information. So the input nodes connect only the hidden nodes which are close. Then using this networks you make it learn the

deadlock pattern.

3. In addition to No.2, we consider the connections to the hidden nodes from the output nodes to be adjusted to the positions.

To solve the above problems,

You can use the algorithm of the article written by Tomohiko Sakao [1]. First you enable the uncalled packet to move by apply the function of "mode". Furthermore we simplified the function more than his article. One is the point that the cells read directly the adjoining cell's state instead of sending "Ping" or "Re: ping". Other is the packet position are decided first instead of making the packet flow by entrance cell's sending. The other one is the deadlock avoidance is not applied.

If you don't simplify these functions, you can't judge the deadlock from the initial packet position. Thus you should judge the deadlock by your self, and using the data of this result you make the neural networks learn. Then you reconnect the neural network considering the cell's position.

4.5.Future research directions

1. Enable it to deal with the other type of deadlock which is hard to detect by this system, for example apparently it dynamically move but in fact it is not do any jobs. But in the present system this case rarely occurs.

2. Hardware error of cellular machines. Actually the hardware problems occur for example the machine is broken and the communication is not done and etc. In this case the deadlock may happen. So it is necessary that this type of deadlock is detected. This problem is relatively easy to solve because the cell know its neighborhood state of dead or alive.

3. Improve cellular machine algorithm more complexly and more functionally. For example the two or three cells work together and bring larger packet, etc.

4. Make the algorithm of deadlock avoidance when it detect deadlock.

5. Another type of neural networks algorithm is applied. Our model of neural networks is one of simplest algorithm. But it does not completely meet our purpose because this model consists of multi-layer networks, then it can not be applied larger scale system. So we have to consider about this problem. We suggest using a Learning Vector Quantization (LVQ) method. A LVQ is one of the most famous algorithms on neural networks. It is a competitive learning algorithm which said to be a supervised version of the Self-Organizing Map (SOM) algorithm by Kohonen. A LVQ algorithm is based on mutual information optimization. It has possible to detect deadlock by using local information.

5. Exchange student life

I studied at TU Delft in Netherlands from September to December for 4 months. This is the first experience to live abroad and to live by myself. My accommodation is one room for one person and has one kitchen, shower and toilet. The room is not so big but that is enough for me to live alone. Because my research needed only a personal computer to do the simulation, I could do my work at home. So my life cycle was quite free as I liked. The main patterns of my life were that I cleaned my room and went to the shopping in the morning until lunch time, and then I went to the university. In weekends I had a lot of parties with my friends. From a personal point of view, these were good and fun chances to communicate with the persons from other countries. I had two lectures in TU Delft. One is English course and the other is "Pattern Recognition". I went to the university three times per week. English course is really interesting and profitable. Before I went to Europe, I was terribly worried about my communication skill in English as my English is deadly poor. The English course difficulty was suitable for me. Its textbook and curriculum were thoughtful. So this lecture made my English better. On the other hand the lecture of Pattern Recognition was too difficult for me to understand in term of language. There were some specific terms. But this lecture has exercise class which has group work, and my partner of the group was a smart and kind man, who taught me the lecture. Unfortunately I couldn't take the exam due to the date which is on January.

About sightseeing, the Netherlands is very beautiful country. Especially Delft, where I stayed, has a lot of small canals and old buildings. It seems typical traditional Europe town. I love this town. And in Europe we can go everywhere by train or by bus. For Japanese that is very interesting because Japan is an island country. I went to Belgium or France by train. There are also the cheapest flights and in the most cases it is not necessary to show my passport; therefore we can cheerfully go to other countries in EU. I went to Switzerland, Germany, and Austria. In the different language country, I felt the people and building are quite different. I am glad that all of them were great experience for me.

6. Summary

For me, the most important and valuable experience is not only to do research but to meet some persons. Especially I am glad to meet my advisor who is a Ph.D student and advice for me about my research. Both my advisor and my supervisor are Japanese. But it is really valuable in my life to meet and talk with them. Their stories are all to my advantage. Their visions are always pointed at the world. In Europe that might be natural. But I felt they have bigger visions than my expectation. So the experience to hear their stories is of the greatest value.

And a life abroad is also unique and precious. I met many oversea students, and talk to them. In the world I saw there are various types of people. I am very grateful to have joined to this project and I hope it made a contribution to my personal life.