2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Introduction and Improvement of Genetic Programming for Intelligent Fuzzy Robots

Yasuyuki Murai[1], Koki Matsumura[2], Hisayuki Tatsumi[3], Hiroyuki Tsuji[1] and Shinji Tokumasu[1]
Department of Information and Computer sciences, Kanagawa Institute of Technology[1]
Department of Information and Knowledge Engineering, Tottori University[2]
Department of Computer Science, Tsukuba College of Technology[3]

1030, Shimoogino, Atsugi, Kanagawa 243-0292, Japan
E-mail: murai@tok.ic.kanagawa-it.ac.jp

Abstract **- We've been following research on the obstacle avoidance that is based on fuzzy control. We previously proposed a new method of automatically generating membership functions, which play an important role in improving accuracy of fuzzy control, by using genetic programming (GP). In this paper, we made two improvements to our proposed method, for the purpose of achieving better intelligence in fuzzy robots. First, the mutation rate is made to change dynamically, according to the coupled chaotic system. Secondly, the population partitioning using deme is introduced by parallel processing. The effectiveness of these improvements is demonstrated through several computer simulations.**

1. Introduction

In the case that a robot avoids the obstacle that moves randomly in unknown environment, it is needed to adapt to the diversity of environment and the change of situation.

In consideration of this case, we are developing the realization of the obstacle avoidance system that applies new intelligent information technology [1]. Conceivably, the fuzzy control is desirable as one of the method for that.

In the fuzzy control, generally, we use the function of a triangular shape for the membershipfunction and several kinds of conceivable control rules as a rule set for the simplification of system construction. However, in the case that precise fuzzy control is demanded, you need to construct membershipfunctions and rules that influence precision systematically and then you need to tune them up appropriately.

Thereupon, we proposed [2] an effective method that utilized the genetic programming [3] that is one of the optimization method with the function of the evolutionary computation, for the purpose to generate appropriate membershipfunctions automatically. However, because of a tree structure in the genetic programming, the size of a genetic type, in other words, the number of nodes of a tree, increases, as the genetic operation goes on. Therefore, the time of the genetic operation also increases when tracing the pointer and accessing to each node. Also, because the memory size that houses the tree structure is easy to become big, the individual size of the genetic operation target group cannot be taken so big. Therefore, as genetic operation goes on and it is conceivable that the diversity of the group becomes low and the search efficiency of the solution will fall. Thereupon, we propose next two methods, to solve this problem in this paper.

Proposal method one changes dynamically the mutation

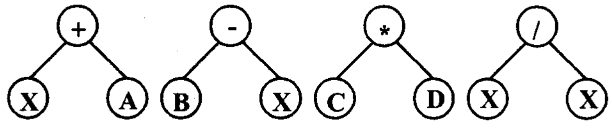


Figure 1: Example of the embryo

rate for the diversity maintenance inside the group, and uses the coupled chaotic system that combines several chaoses, to change the mutation rate.

Proposal method two divides the group into several sub groups (deme), processes them parallelly on several computers and also exchanges the individuals between the groups. This contributes the solution of problems of individual size and the diversity of the group as roused before.

Also, as the case of the obstacle avoidance problem, a goal object searching robot is taken up, where the robot departs from the start position and searches the goal object while avoiding the obstacle.

2. Genetic Programming

We have already announced the method that the membershipfunction is generated automatically by using genetic programming in the paper [2]. This paper intends the speed-up in membershipfunction generation based on the method. We will mention the outline about the method here.

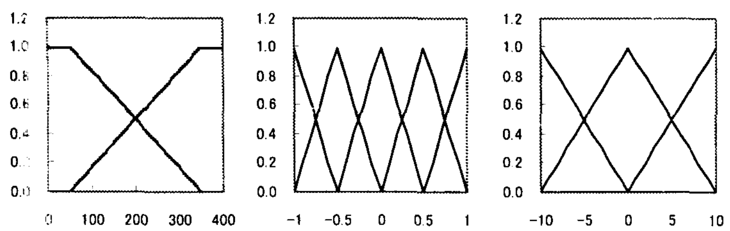
2.1 Concept of the genetic programming

The application method of genetic programming (we say after, GP) of this paper is based on the Kosa's solution of the asymptotic problem of functions [3]. The fundamental expression method of GP is applied in this paper. A membershipfunction is expressed by using tree structure that is composed of a numerical expression. And, the function is evolved by the genetic operation and the membershipfunction is generated automatically.

2.2 Membership function

The membershipfunction is expressed with the binary tree. It has any one of an arithmetical operations operator (+, -, *, /) in the parents node like, Fig.1 and has one variable or constant each in the child node of the right and left. A necessary number of these functions are generated as the initial individuals. After that these initial individuals are evolved by genetic operation and tuned up to an optimal membershipfunction.

The example of the fuzzy control membershipfunctions that we use in this research is shown in Fig.2. Fig.2 (a) is the one



(a) distance (b) direction (c) movement quantity

IF	distance = near, direction = left side	then central
IF	distance = near, direction = a little bit left side	then right side
IF	distance = near, direction = central	then left side
IF	distance = near, direction = a little bit right side	then left side
IF	distance = near, direction = right side	then central
IF	distance = far, direction = left side	then central
IF	distance = far, direction = a little bit left side	then central
IF	distance = far, direction = central	then left side
IF	distance = far, direction = a little bit right side	then central
IF	distance = far, direction = right side	then central

(d) example of fuzzy rules

Figure 2: Membershipfunctions and rules

that means the distance between the robot and object (obstacle or goal) by two labels "near, far". Fig.2 (b) is the one that means the direction from the robot to the obstacle, by five labels "left side, a little bit left side, central, a little bit right side, right side". Fig.2 (c) is the one that means the grade of movement quantity (direction and quantity of movement) of the robot in the y axis direction in unit time, by three labels "left side, central, right side". Finally, Fig.2 (d) shows an example of fuzzy rules.

2.3 Obstacle avoidance model

The obstacle and goal exist in the workspace where the robot runs. The robot avoids the obstacle and move toward the goal from the start position. The fundamental movement of the robot is modeled in the following manner.

(1) The robot course is corrected by adjusting movement quantity with the fuzzy control on the basis of the position of the robot, obstacle and goal.

(2) The robot does the movement with constant quantity always to the workspace right direction (direction of positive x axis), but is possible in increase or decrease of movement quantity in top and bottom (y axis) direction only due to obstacle avoidance and goal approach.

(3) The position and movement of the obstacle is taken by the sensor.

The workspace and condition for the computer simulation are shown in Table 1.

2.4 Generation of the membershipfunction

The membershipfunctions are generated by the following genetic operation. A lot of individuals of the expression of two clauses shown in Fig.1 are randomly generated as the initial population and crossed over among them by roulette selection. The individuals generated by crossover are replaced with the individuals of lower fitness. Also, the one optional individual is selected in the group by the random number as the target of mutation. Then, the node position for mutation is determined by the random number on the basis of the length of the individual and its contents are changed. Furthermore the mutation rate changes dynamically by using the coupled chaotic system in proposal method one and on the other hand, is left fixed in proposal method two.

2.5 Fitness

Using the individual as the membershipfunction, the avoidance simulation is conducted. Then, the fitness is given higher value, if the robot takes shorter route without colliding the obstacle. Actually, fitness is taken as the summation of those values given by the designated number of simulation with different position of the obstacle. Yet, we evolve the individuals without doing the avoidance simulation in the early stage of evolution, for saving useless time when the robot cannot reach the goal. This is because the successful possibility that the robot reaches the goal is very low, while the membershipfunctions are not improved to some extent as in the early stage. Thereupon, discerning the shape of the individual as a function, the fitness is given according to the evaluation. After this value exceeds a preset value we do the avoidance simulation. In the case of the simulation, the fitness shall be the sum of that evaluated in shape of individual and that described before as evaluated regarding whether or not the robot is able to reach the goal as a result of the simulation and how short the length of the route is.

3. Improvement of Genetic operation in GP

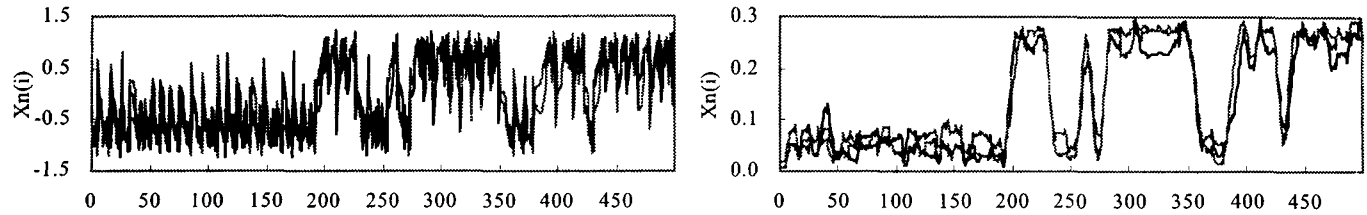
3.1 Adjustment of the GP parameter by the coupled chaotic system

The coupled chaotic system is the one that connects several chaoses. Many of the chaos phenomena in the real world, are not the behavior by the chaos of a simple substance, but are composed of several chaoses [4]. We are using GP. The coupled chaotic system, seen in the evolution of ecosystem, is thought to be effective for improving the GP.

As genetic operation goes on in GA and GP, similar individuals occupies overwhelmingly in the group due to crossover and the process is apt to fall into the local solution. Thereupon, it forcibly causes the gene change by the mutation, in order to escape from the local solution and maintain the diversity inside the group. However, when the mutation rate is set up high, the possibility becomes high that aggravates the good individuals by excessive mutation. On the other hand, when it is set up low, the diversity of individuals is getting low because of the strong influence of crossover. Therefore, its appropriate value is necessary. However, it is a difficult problem to give such an optimal value that the GA runs well in efficiency.

Table 1: Workspace and conditions

Parameter	Value
Size of workspace	640×400
Obstacle	Circle (Radius 40)
Width of the robot	50
Start position of the robot	(60,200)
Position of the goal	(560,200)



(a) before adjustment (x axis is a generation) (b) after adjustment

Figure 3: Coupled chaotic system

In this paper, by changing the mutation rate dynamically during genetic operation, the efficiency of the solution search is improved without losing the effect of crossover. In order to do this, the coupled chaotic system is taken which is mentioned before. The coupled chaotic system in this paper is specialized so that the change of attractor breaks out by connecting chaoses (Fig.3). In this coupled chaotic system, the attractor is changing the position up and down, as time goes on, without staying at a regular position. Moreover, the time interval of the change is not constant. By treating this attractor as the mutation rate, it is able to change dynamically. Thus, the mutation rate changes up and down with time according to the change of attractor of the coupled chaotic system.

The coupled chaotic system that we used in this paper is the one that connected three chaoses. The coupled chaotic system is shown with the expression (1) and Fig.3 (a). This is the one that connected three logistic functions and has a change of attractor like Fig.3 (a), where nonlinear shape coefficient a is 3.9 and connection coefficient e is 0.6. However, we are difficult to use it as the mutation rate as it is. Thereupon, the mutation rate is taken as a moving average of changes of the attractor and adjusted between 0～0.3 (Fig.3 (b)).

$$f(x) = ax(1 - x^2)$$
$$x_{n+1}(i) = (1-e)f(x_n(i)) + \frac{e}{2}(f(x_n(i+1)) + f(x_n(i-1)))$$
$$a = 3.9, e = 0.6 \qquad i = 1,2,3 \qquad (1)$$

3.2 Parallel processing by deme

The individual group regarded as the target of genetic operation is divided to several sub groups and genetic operation is conducted on each sub group. This is an adequate even for the evolution model of an actual creature and is effective in the parallel processing of GA and GP [5]. Furthermore the diversity inside the group is maintained effectively, by crossing over individuals among the groups.

As genetic operation goes on in GP, the length of an individual has a tendency to increase. Even genetic operation time increases, when the length of the individual increases. Therefore, it increases a processing time, to increase the number of genetic operation for searching a good solution. Also, there is a method that increases the individual number inside the group, improves search efficiency. However, even the time of genetic operation would increase when the number of individuals increases. Thereupon, we adopted the parallel processing by deme to solve these problems.

In this paper, each individual group is assigned to a personal computer connected on LAN and genetic operation is conducted to the parallel. This system configuration is shown in Fig.4. This system is composed of clients and a server that performs the exchange of the individuals among demes and synchronizes the clients for genetic operation in demes. The experiment of this paper is done with five computers (one server and four clients). The individual exchange of the server and clients is done with the next procedure.

(1) The clients connect to the server.

(2) After the connection of all the clients is completed, the genetic operation start command is transmitted from the server to clients.

(3) Each client does genetic operation.

(4) Each client transmits a best individual to the server and waits for receiving the individual from the server.

(5) After best individuals are received from all the clients, the server exchanges the individuals and transmits them to the clients. The individual exchange of the case complies with next expressions.

Cs= (Cr+1) Mod n,

Cs : sender client number, Cr : receiver client number, where n is the number of clients.

(6) The client receives the individual from the server.

The individual that was received is added to the individual group of each client, which becomes the genetic operation target of the next time.

(7) The operation of from (1) to (6) are repeated specific times.

4. Experiments and Discussions

Membershipfunctions were generated by GP by using two proposal methods, and the efficacy of the proposal methods was verified in comparison with the established method.

4.1 Generation of the membershipfunction

4.1.1 The case with coupled chaotic system

The proposal method and that with constant mutation rate were compared, by running simulations 15 times with

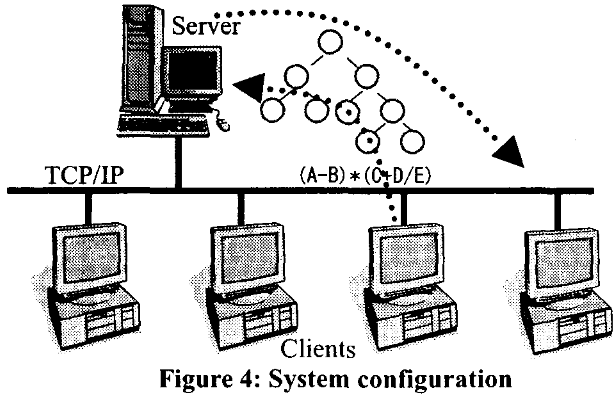


Figure 4: System configuration

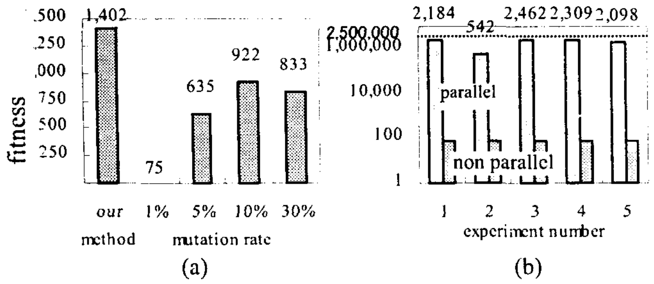


Figure 5: Comparison of supreme fitness

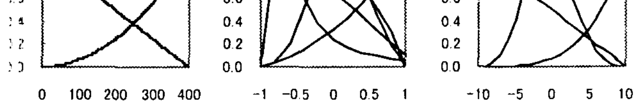
different seeds of the random number. The coupled chaotic system used the one with logistic functions shown in the expression (1). Also, the constant mutation rate takes 1%, 5%, 10%, and 30% in this experiment. Furthermore, it is assumed that the crossover rate is 5%, the genetic operation number is 4000, and the population size is 100. The result of the experiment is shown in Fig.5 (a). This figure is the average of the supreme fitnesses of the experiments, where genetic operation was done the specific times. The upper limit of fitness is set 2,500,000. From this figure, you see improvement of search efficiency in the proposal method, where the average fitness is becoming higher in comparison with the case that the mutation rate is fixed.

Furthermore, in the case that the mutation rate is 1%, the fitness shows 75 with extremely small. As for this, the fitness based on the shape of the function did not exceed a preset constant value in 4000 times of genetic operation and the avoidance simulation was not done.

4.1.2 The case of parallel processing by deme

Fig.5 (b) shows comparative results on the largest fitness between the proposal method and the case without parallel processing. Fitness of the case of the parallel processing took the largest value among four clients. Experiments were done 5 times with different seeds of random number, where the crossover rate is 5%, the mutation rate is 5%, population size is 100, and the genetic operation number is 3000. In the case without the parallel processing, fitness based on shape of the function did not over the constant value in 3000 times of genetic operation and became a very low value. This is because the avoidance simulation was not done. However, 4 times during 5 times of experiments took two million or more and very high value in the case of the parallel processing and good fitness was obtained with genetic operation of the few times, verifying the effect of the parallel processing.

The example of the individual (the membershipfunction) is shown in Fig.6 that was created by the procedure.



(a) distance (b) direction (c) movement quantity

Figure 6: Example of the membershipfunctions

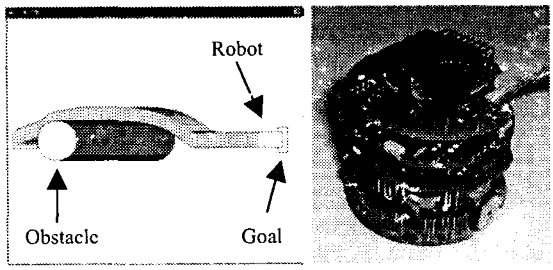


Figure 7: Computer simulation and real robot

4.2 Avoidance simulation and traveling experiment by the real robot

The moving obstacle avoidance simulation was run by using the membershipfunctions that were generated with GP. Also, we made a traveling robot on the basis of the principle mentioned above and examined the possibility of real robot realization. The example of the simulation result and the real robot are shown in Fig.7.

5. Conclusions

As a method that improves the problem of GP, we proposed the method that the mutation rate is modified dynamically by using the coupled chaotic system and, the parallel processing method that uses deme. And, the usefulness of these methods were examined by the computer simulation and the real robot travel experiment. The results are as follows

(1) The efficiency of the genetic operation of GP can be improved by the coupled chaotic system.

(2) The efficiency of the genetic operation of GP can be improved with the parallel processing by deme.

(3) Even the real robot travel experiment can obtain the fine result and it would be applicable to a real robot.

From this result, it is understood that the improvement of the genetic operation efficiency of GP is possible by the coupled chaotic system and deme. Finally, our future works include:

(1) Relating the coupled chaotic system to the internal state of GP, the control corresponding to the states inside GP is realized.

(2) The parallel processing by deme is to be done asynchronously. Also, genetic operating environment of each group is changed optimally.

REFERENCES

[1] Naito Hirohisa and Uchida Kenko, Emergent Neural Network and Application to Moving Obstacles Avoidance (in Japanese), IEEJ Trans., Vol.117-C (6), 768-776, 6, 1997.

[2] Koki Matsumura and Yasuyuki Murai, Obstacles Avoidance Model by the Fuzzy Robot based on Genetic Programming (in Japanese), IEICE Trans., Vol.J83-A (12), 1539-1551, 12, 2000.

[3] John Koza, Genetic Programming, MIT Press, Massachusetts, 1992.

[4] Kunihiko Kaneko and Ikegami Takashi, Scenario of the evolution of a complex system (in Japanese), Asakura Publishing, Tokyo, 1998.

[5] Matsui Kazuhiro, Correlative Selection Operators to Improve the Population Diversity in Genetic Algorithms (in Japanese), IEICE Trans., Vol.J84-DI (2), 232-236, 2, 2001.