Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation July 16-20, 2003, Kobe, Japan

Teacher's Load and Timing of Teaching based on Interactive Evolutionary Robotics

Daisuke Katagami
CISS, IGSSE, Tokyo Institute of Technology, JAPAN
www.ntt.dis.titech.ac.jp/home/katagami/

Seiji Yamada
National Institute of Informatics, JAPAN
research.nii.ac.jp/˜seiji/

Abstract

We have proposed a fast learning method that enables a mobile robot to acquires autonomous behaviors from interaction between human and robot. In this research we develop a behavior learning method ICS (Interactive Classifier System) using interactive evolutionary computation considering an operator's teaching cost. As a result, a mobile robot is able to quickly learn rules by directly teaching from an operator. ICS is a novel evolutionary robotics approach using classifier system. In this paper, we investigate teacher's physical and mental load and proposed a teaching method based on timing of instruction using ICS.

1 Introduction

Under the situation that it's difficult to prepare the knowledge for action, the autonomous robot requires the ability to accomplish tasks in the environment where contents human activities. The situation, for example, may be an dynamic environment or unexpected interaction from human. Therefore, the study for acquisition of autonomous behavior and adaptation to various environments become necessary.

Recently, reinforcement learning and evolutionary computation technique were used as the framework of learning and adaptation. Moreover, the research that enables a robot get a controller autonomously has attracted attention. When making interaction dynamics with robot's embodiment and environment reflect in construction of a controller, one of the purposes of these techniques is eliminating the unsuitable and unnecessary bias by the designer. Therefore, in the former, it has been made usual to learn by trial and error to an agent, without putting in prior knowledge in the framework of reinforcement learning. However, the execution speed becomes a problem in applying to a real environment.

Then, some researches of the approach using interaction with the human who exists in environment has been carried out. Particularly for the robots that do not have a priori knowledge or commit trial and error in the initial stage, human instruction is the very effective acquisition technique of autonomous behavior. However, in a certain level of autonomous robot, it is not necessary to follow instruction from human all the time. In the stage which does not need instruction, robot should demonstrate its autonomy based on the instruction rules stored by interaction with human without putting a burden on human. Therefore, we need to the technique of establishing a robot's autonomy from through interaction between human and a robot is required.

Asoh et al. [1] proposed the framework that the map information of the unknown environment is built by a mobile robot, called Jijo-2 which performs a communication by voice conversation with human. However, it doesn't get the behavior of the robot by the interaction through human and a robot. Ishiguro et al. [2] built the state space of the mobile robot by reinforcement learning. However, it is learning by using as a sample action that the introduction human taught. After that, a robot only builds an internal state and there is no interaction with human. Horiguchi et al. [4] used the idea of the mutual leadership pattern interaction as the design of the interaction of the robot with the human and realized the cooperation behavior of the automation process of a mobile robot and human operations by using power feedback. However, the result of learning didn't reflected on the behavior acquisition of the robot. Inamura et al. [5] indicate acquirement behavior of a robot using Bayesian Network based on a dialog with a user. It is different from our technique to get behavior gradually by the evolutionary computation technique.

Our purpose is realizing a robot's autonomy by receiving the instruction information as a suitable act from human, and gaining act rules evolutionally with the state recognition which can solve a task. We call such a framework Interactive Evolutionary Robotics (IER).

In this paper, we propose some method about a teacher's load and timing of teaching which is key point in the framework of IER.

2 Interactive Teaching

2.1 Teacher's Load

Generally, in interactive evolutionary learning, the more it is taught, the better the performance is. However, human labor is not unlimited. It is clear that it is tradeoff like it is better as instruction cost lowers. Human's labor has a limit in cooperating with a machine without tiredness, carrying out comparison evaluation of many individuals (or rules) for every generation, and inputting an evaluation value. This has been a serious practical problem.

Moreover, as the second problem, the number of individuals and the number of search generations must be lessened as compared with the usual EC search in order to reduce physical and mental load in case human evaluates individuals. It makes convergence worse.

In this research, in order to measure a teacher's load simply, it divides into mental load and physical load. We consider the timing of teaching as mental load and the number of times of teaching as physical load respectively. It is necessary to solve the load problem for utilization of interactive evolutionary learning. There are four measures in respect to that problem. It is 1)the design of the improvement of the input interface to a

0-7803-7866-0/03/$17.00 ©2003 IEEE

computer, 2)the improvement of the presentation interface on a computer, 3)speedup of EC convergence, and 4)the fusion method with the usual EC which is not interactive learning.

In this research, we consider the following things as instruction. First, direct operation of the robot by input equipment is performed. Next, a rule is automatically generated from the operation and the environment information at that time. In this framework, it is necessary to perform neither comparison evaluation of many individuals, nor the input of an evaluation value like the conventional interactive evolutionary learning. Thereby, it is expected physical and mental load is reduced sharply. These are considered to be the improvement of the input interface for a computer, and the improvement of the presentation interface from a computer in the framework of *Interactive Evolutionary Computation*.

Moreover, we will consider the case where interactive learning is applied to real robot environment. When a teacher directs by operating a robot intuitively from input equipment (teaching), a rule is created automatically, and a robot learns autonomously when there are no directions. We consider that this load problem is reduced by this method in the point that a system learns autonomously, the point that a rule is automatically created by human's intuitive instruction, and the point that additional study can be performed anytime.

2.2 Timing of Teaching

We think that the timing of teaching is greatly concerned with the above-mentioned teacher's load. However, since it depends on a system side for the timing which instruction performs, in order to teach in accordance with the timing, human has to wait. Not to mention the experiment in a simulation, a teacher's load increases further in the real environmental learning that needs more time for an experiment.

Compared with the above-mentioned general instruction study, we aim at teaching without recognizing that a teacher is teaching in IER. That is, a system will learn by gaining operation of a teacher as instruction information automatically only by a teacher operating a robot and performing a task. This means that it leads to reduction of an teacher's load. We investigate the effect by the timing of instruction, in order to realize such instruction. The timing of the conventional instruction shall be divided into the timing of following three instruction in this research.

- pre-teaching
- passive teaching
- anytime teaching

pre-teaching

Exploration in the framework of interactive evolutionary robotics learns from instruction information. Preteaching is the method of performing exploration by instruction at Teaching Mode beforehand, and performing exploitation at Autonomous Behavior Mode.

passive teaching

We define passive teaching method as the method of directing teaching at the time of the demand of a system to a user. Mishima and Asada el al. have improved that the efficiency of learning gets worse by passive teaching for a gap (Cross Perceptual Aliasing) of the environmental recognition produced between a teacher and a learner. In study efficiency, passive teaching has little futility of teaching and is considered to be a good method. However, the teacher has to be supervising until a system requires action. Moreover, since it does not know when the timing comes, it is thought that a mental load becomes large to the number of instruction.

anytime teaching

The problem how to treat the trade-off between exploration and exploitation is in one of the important topics in reinforcement learning. In order to obtain many rewards, you have to choose the optimal action preferentially under the present value function. In this case, it is said that the present knowledge about action value is exploit(ed). However, the optimal policy based on the present value function is not necessarily really optimal policy. In order to discover a better policy, it is necessary to raise the accuracy of the present value function. Therefore, each action needs to be tried several times. Moreover, in a dynamic environment, in order to acquire the knowledge which is adapted for new environment, it is necessary to also try the action that was not good before. It is referred to as exploring environment to try new action, in order to find a better policy. Since we cannot perform exploitation and exploration simultaneously, let both balance be a problem.

On the other hand, it is possible that a teacher gives instructs to a robot at favorite timing. In this research, this is called anytime teaching method. Seeing a robot perform autonomous action, a teacher operates a robot to favorite timing and makes a task. Thereby, teacher can instruct to a robot being unconscious of teaching, without worrying about whether he teaches by seeing a robot's action. Thereby, a teacher can teach without worrying about whether being conscious of teaching, whether it teaches, or not when he/she saw a learner's all actions. However, it is difficult to include such specification in a system side.

3 Teaching based on Interactive Evolutionary Robotics

3.1 Interactive Evolutionary Robotics

Interactive Evolutionary Robotics (IER) is a framework aiming at performing efficient real environmental robot study using the evaluation capability of (*Interactive Evolutionary Computation*(IEC)). *IEC* is the method of including evaluation of human in the evaluation system of a system directly, and searching evolutionally. Moreover, IER is an approach which designs a robot interactively using the evolutional calculation techniques, such as a genetic algorithm, genetic programming, and an evolution strategy. The framework figure of IER is shown in Fig.1.

In general Evolutionary Robotics, it is considered as the object of a design of a robot's controller. ER is designing it through evolution process by selection pressure under an evaluation function rather than determines the detailed specification in top-down. In IER, evolution is promoted by gaining a rule by interaction with a designer further. For this reason, while expecting the high emergent of ER, it is possible to also expect the pliability and sensitivity of *IEC* in a complicated problem.

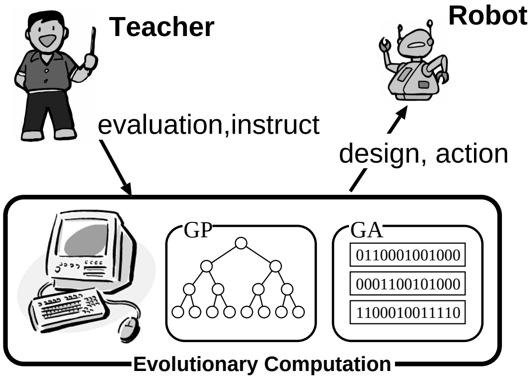


Figure 1: Interactive Evolutionary Robotics

We think that the method in this framework is effective in the learning of an initial stage that must be performed by trial and error. Moreover, we also expect the effect of obtaining a solution to the partial solution that cannot be solved only by human, through interaction between human and robot. Furthermore, since it is not dependent on learning algorithm, this technique is widely applicable to general evolution robotics.

3.2 Interactive Classifier System

By including the interactive function of *IEC* in (Learning Classifier System (LCS), ICS is the robot study model that can also perform study by instruction in addition to autonomous study. XCS[7] which Wilson proposed is used for LCS that is study algorithm. XCS adds the parameter that is what improved ZCS[8] and is called accuracy . Although the classifier was generalized by including #(don't care symbol) in a conditional part, the classifier system or ZCS of Holland were not able to perform it effectively.

This originates in not having a mechanism for the classifier system itself advancing generalization appropriately, and the phenomenon in which the performance of a system gets worse by the classifier generalized too much (overgeneral) has been reported[9]. In XCS, in order to control the classifier generalized especially too much, not only the conventional intensity but accuracy has determined the validity of a classifier. This accuracy is calculated by the error of the reward received as a result of performing a classifier, and its prediction value, and it is reported that a rule can be generalized appropriately by this, without becoming common too much[7]. The framework figure of the built system is shown in Fig.2.

ICS consists of a rule generation component (RGC), a sensor processing component (SPC), a display component (DC) and a reinforcement component (RC). All of them are developed on Linux. It is described by the C language and GTK+. Each module is explained below.

【RGC】 Rule Generation Component creates the rule by instruction. A teacher operates it using input equipment, looking at the information displayed on an interface in a robot. A sensor processing part (SPC) receives the operation history of a there, and the sensor information of the robot at that time, RGC creates a rule newly from it, and it adds to a rule list. The creation procedure of a rule was improved so that a rule could be created from instruction information (the action to which operator operated the robot) on the basis of XCS[7].

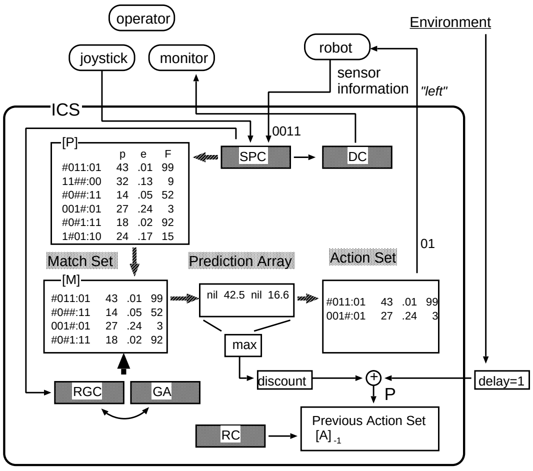


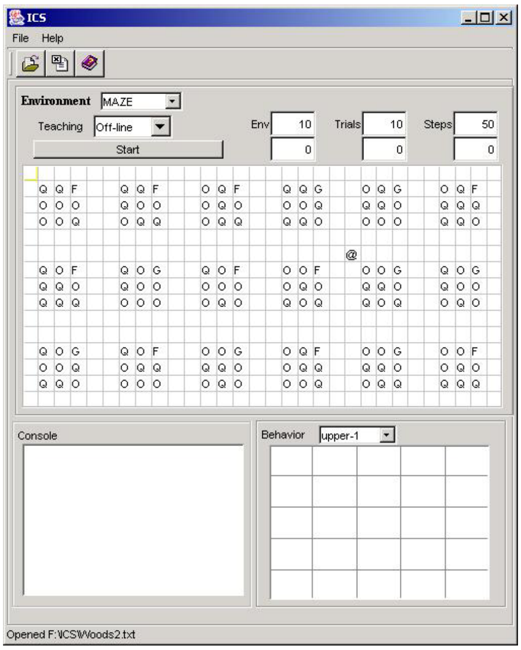
Figure 2: Overview of Interactive Classifier System

1. ICS receives a robot's sensor information X and instruction information a_t from SPC.
2. Some classifiers that matched X is moved from a group [P] to a match set [M]. ICS turns regularly the Prediction value of classifier which supports each act a_i in [M] with a Fitness value, and creates $P(a_i)$. The value of $P(a_i)$ is put on Prediction Array, and the act of classifier chosen by $P(a_i)$ is chosen by act selection methods. Act selection methods are performed by deterministic selection method or roulette wheel selection method.
3. If $a_j \neq a_t$ to compare act a_j chosen by act selection methods and act a_t obtained by teaching, the action part of the rule which has a_j in an action part in [M] will be rewritten to a_t. A change will not be made if $a_j = a_t$.
4. The action set [A] which consists of classifiers in [M] which supports selected act a_j is created. When act a_j or a_t is sent to an effect machine, and in case of a_t, reward r_{teach} is given immediately. When there is no input of a_t, remuneration r_{imm} is returned from environment.

【RC】 Reinforcement Component is a reinforcement learning part in classifier system. It learns by updating the parameter of classifiers chosen last time step. When there is no operation of a teacher, a robot can act autonomously from the rule created by then.

【DC】 Display Component takes charge of the display of the data processed by SPC. GTK+ is used for development of an interface. The developed interface is shown in Fig.3.

【SPC】 Sensor Processing Component performs processing of a robot's various sensors and processing of teaching information. It is sent to DC and RGC and the processed data is displayed and ICS creates classifiers from them.



ICS

File Help

Environment MAZE

Teaching Off-line

Start

Env 10 0

Trials 10 0

Steps 50 0

```
Q Q F    Q Q F    O Q F    Q Q G    O Q G    O Q F
O O O    Q O O    O Q O    O O Q    Q Q O    Q Q Q
O O Q    O Q Q    O Q Q    Q Q O    O O O    Q Q O

                                   @
Q O F    Q O G    Q O F    O O F    O O G    Q O G
Q Q O    Q O O    O O O    O Q O    Q Q O    Q O O
Q Q Q    O O O    O Q O    Q O Q    Q O Q    O Q O


Q O G    Q O F    O O G    O Q F    O O G    O O F
O O Q    O Q Q    Q Q O    O Q Q    Q Q O    O Q O
Q Q O    O O O    O Q O    O O Q    O Q Q    Q Q Q
```

Console

Behavior upper-1

Opened F:\ICS\Woods2.txt

Figure 3: User Interface

At first, a human operates robots with a joystick by referring to sensor information displayed on GUI interface, and the DC processes the information. Next, the SPC gets interaction and sensor information. The RGC make new rules from them and adds them into a rule list. When nothing is input from the operator, a mobile robot executes autonomous behaviors from interaction. Finally, the RC reinforces the classifiers by updating their parameters in the actions that were previously executed.

ICS differs from *IEC* in that operators do not have to evaluate individuals each time. The operator can always operate a mobile robot directly, and such direct operation can take place of the fitness evaluations of each individual in *IEC*. Therefore the operator can do teaching with less load, and can always do concentrative additional learning for sub-tasks difficult to achieve.

3.3 Procedure of learning

ICS performs two modes: a teaching mode and an autonomous behavior mode by turns. The procedures of the two modes are shown in the following.

Teaching mode

Step 1: Prepare the robot's state space.

Step 2: It teaches depending on any of the procedure of the timing of three kinds of instruction they are.

Step 3: A rule is created by an operator's directions information and environmental information at the time.

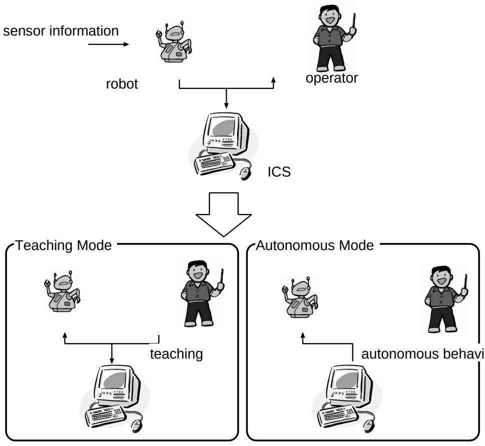


Figure 4: Teaching Mode and Autonomous Mode

Teaching mode

Step 4: If there is no rule belonging to the same cluster, it will add as a rule newly.

Step 5: If there is a rule belonging to the same cluster, a strength value will be updated by reward.

Autonomous behavior mode

Step 1: The robot behaves by conforming to stored rules in Rule List.

Step 2: If the average of the number of the time steps from GA of just before in a match set exceeds a threshold, GA will be performed to the match set.

Fig.4 shows the overview of a teaching mode and an autonomous behavior mode.

3.4 Procedure of the timing of teaching

The timing of teaching has three timing described in Chapter 2.2. Each procedure is shown below. Each is performed in **Step2: in *teaching mode.***

pre-teaching

1. **A teacher directs action to state space.**

passive teaching

1. **Act A will be performed if there is effective action A to state space.**
2. **If there are no directions, directions will be requested to a teacher.**

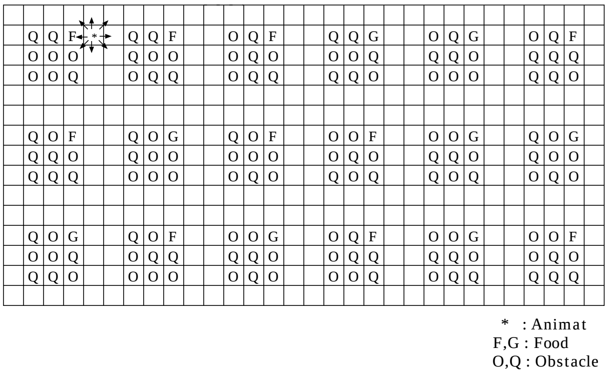


Figure 5: Woods2 Environment

anytime teaching

1. To state space, if there are directions from a teacher, it will perform.
2. If there are no directions, a robot will perform exploration autonomously.

4 Experiment

4.1 Teaching effects and Teacher's Load

We test a preliminary experiment in order to evaluate the effectiveness our ICS. This is a very simple domain. We use Woods2 environment which one of Wood-like environments[7] as an environment in the experience. It used as a test-bed in several works based on classifier system. Fig.5 shows Woods2 environment. This environment is markovian multi-step problem. The left and right edges of Woods2 are connected, as are the top and bottom. Woods2 has two kind of "food" and two kind of "rocks". F and G are the two kind of food, with sensor codes 110 and 111, respectively. O and Q are the two kind of objects, with sensor codes 010 and 011, respectively. Blanks have sensor code 000. The system, here regarded as an animat or artificial animal, is represented by *. To sense its environment, * is capable of detecting the sensor codes of objects occupying the eight nearest cells. The encoding of a classifier is as follows. A classifier, for example, is the 24-bit string 000000000000000010010110. The left-hand three bits are always those due to the object occupying the cell directly north of *, with the remainder corresponding to cells proceeding clockwise around it. The animat's available actions consist of the eight one-step moves into adjacent cells, with the move directions similarly coded from 0 for north clockwise to 7 for northwest. If a cell is blank, * simply moves there. If the cell contains food, * moves to the cell, "eats" the food, and receives a reward($r_{imm} = 1000$).

We compare our teaching method with No-teaching method by four settings of the number of teaching to investigate relationships with teaching effect and load. The teaching operates by timing as same as pre-teaching. We try 50 teachings that have about 3 steps by one trial. A setup of the number of instruction prepared four setup with 5, 10, 20, and 50 instruction to one problem. Action for about 3 steps is built by one trial by instruction. When ten trial is performed, the classifier of about 30 will be created. The experiment is asking for 10 trial deed and its average about each setup. The parameter of an experiment is shown in Table 1.

Table 1: Experimental Parameters

Parameters	Value
number of problems in one experiment	5000
number of experiments	10
maximum size of the population	800
number of teaching	5, 10, 20, 50
probability to do crossover	0.8
probability of mutating one bit	0.04

4.2 Experimental Results

In this work, we investigated performance (in average steps to food) and system error (average absolute difference between the system prediction for the chosen action and P which is the sum total of maximum system prediction and the current reward in Fig.2). Fig.6 shows the steps to the foods. And, Fig.7 shows system error.

Especially in early stage of learning, ICS outperformed No-teaching system. Our teaching method is twice as good as Non-Teaching method in early stage(50 trials) in the steps to foods. Moreover, there is no effect on system error. ICS improves the early time learning due to have been given the human-robot interaction in advance. Although efficiency becomes good about the load of instruction so that the number of instruction goes up, it turns out that it is difficult to look for an appropriate point.

In order to investigate how much taught rule is spread and used effectively for a group, the rate in which the rule taught into the group or its posterity is contained was investigated for every number of problems. Here, the taught rule expresses the rule made from instruction information, and all the children made by Genetic Algorithms considering it as parents. The result is shown in Fig.8. In order to teach by setup of 10 beforehand, the rate is not filled in the stage to begin to 1%. However, if 500 problems are exceeded, 80% of a group will be occupied. The rule to which about 100% was taught in the stage beyond 1000 problems occupies. Signs that the taught rule is used very well and spreads in the group are known.

5 Conclusion

We proposed a fast learning method based on ICS which enables a mobile robot to acquire autonomous behavior from interaction between human and robot. We evaluated the efficiency of this work by the experiments in the multi-steps simulation environment.

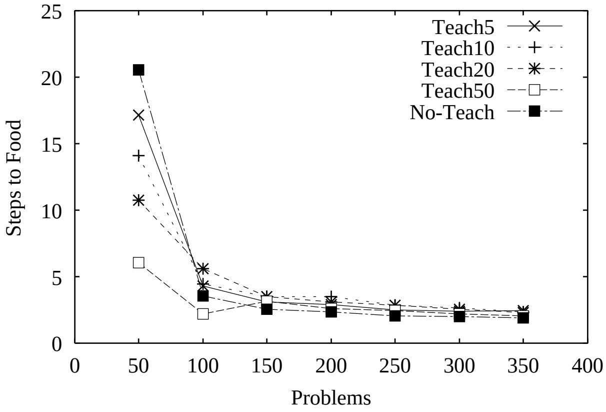


Figure 6: Step to Food

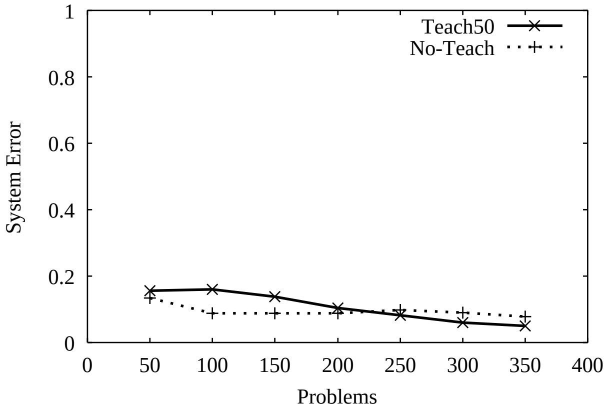


Figure 7: System Error

ICS has two major characteristics. First, the ICS generate initial individuals by teaching from human-robot interaction. We can perform initial learning efficiently in this way. Second, a user can add new rules to operate a robot directly at any time during the course of teaching in ICS. Therefore the user can perform teaching without much load, and can always do concentrative incremental learning for sub-tasks difficult to achieve.

About the timing of teaching, it is under construction at this time. In the near future, we will make experiments on the three teaching timing using real robot such as AIBO to inspect the effect of teaching and user's load.

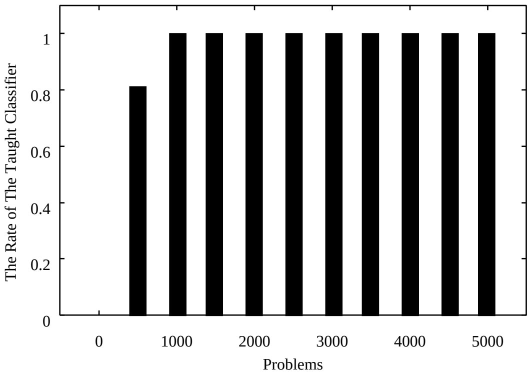


Figure 8: The Rate of The Taught Classifiers

References

[1] H. Asoh and Y. Motomura and I. Hara and S. Akaho and S. Hayamizu and T. Matsui: Combining probabilistic map and dialog for robust life-long offifce navigation; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 807–812 (1996)

[2] H. Ishiguro and R. Sato and T. Ishida: Robot Oriented State Space Construction; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1496–1501 (1996)

[3] C. Mishima and M. Asada: Active Learning from Cross Perceptual Aliasing Caoused by Direct Teaching; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1420–1425 (1999)

[4] Y. Horiguchi and T. Sawaragi and G. Akashi: Naturalistic Human-Robot Collaboration Based upon Mixed-Initiative Interactions in Teleoperating Environment; *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 876–881 (2000)

[5] T. Inamura, M. Inaba, and H. Inoue: User Adaptation of Human-Robot Interaction Model based on Bayesian Network and Introspection of Interaction Experience; *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2139–2144 (2000)

[6] D. Katagami and S. Yamada: Interactive Classifier System for Real Robot Learning; *IEEE International Wortkshop on Robot and Human Interaction*, pp. 258–263 (2000)

[7] S. W. Wilson: Classifier fitness based on accuracy; *Evolutionary Computation*, Vol. 3, No. 2, pp. 149–175 (1995)

[8] S. W. Wilson: ZCS: a zeroth order classifier system; *Evolutionary Computation*, Vol. 2, pp. 1–18 (1994)

[9] D. Cliff and S. Ross: Adding temporary memory to zcs; *Adaptive Behavior*, Vol. 3, No. 2, pp. 101–150 (1994)

[10] A. R. Cassandra and L. P. Kaelbling and M. L. Littman: Acting Optimally in Partially Observable Stochastic Domains; *12th National Conference on Artificial Intelligence*, Vol. 2, pp. 1023–1028 (1994)

[11] S. W. Wilson and D. E. Goldberg: A critical review of classifier systems; *The Third International Conference on Genetic Algorithms*, pp. 244–255 (1989)