Interactive Evolutionary Robotics from Different Viewpoints of Observation

Daisuke Katagami[1] and Seiji Yamada[2]

[1] *Tokyo Institute of Technology, CISS, IGSSE*
4259 Nagatsuta, Midori-ku Yokohama 226-8502, Japan, katagami@ntt.dis.titech.ac.jp
[2] *National Institute of Informatics, Japan, seiji@nii.ac.jp*

Abstract

In this paper, we describe influence of viewpoints of observation in an interactive evolutionary robotics system. We have been proposed a behavior learning system ICS (Interactive Classifier System) using interactive evolutionary computation. In this system, a mobile robot is able to quickly learn rules by direct teaching of a human operator. ICS is a novel evolutionary robotics approach using a classifier system. We classify teaching methods into internal observation and external one, and investigate influence of observation methods. We have experiments based on our teaching methods in two kinds of tasks. We found that teaching methods from different viewpoints of observation change teaching efficiency because of the difference between a robot's recognition and an operator's one in an environment.

1 Introduction

In previous robot learning studies, optimization of control parameters has been applied to acquire suitable behaviors in an real environment. Also in most of such researches, a model of human evaluation has been used for validation of learned behaviors. However, since it is very difficult to build a human evaluation function and adjust control parameters, a system hardly learns behaviors intended by a human operator.

In contrast with modeling human evaluation analytically, we introduce another approach in which a system learns suitable behaviors using human direct evaluation without its modeling. Such an interactive method with *Evolutionary Computation* (EC) as a search algorithm is called Interactive EC (IEC)[1] , and a lot of researches on it have been done thus far[2][3]. Consequently it can performs subjective evaluation by human sense. However an operator suffers from cognitive and physical load. In case of a robot reinforcement learning in a real environment, some researches have been worked[4]. Unfortunately it makes a lot of time for convergence of learning.

To cope with these problems, we have been proposed Interactive Evolutionary Robotics (IER)[5][6]. It is an interactive EC learning method to design a robot using EC methods like genetic algorithm, genetic programming and evolutionary strategy. We expect IER to perform high emergent property of ER and subjective adaptability of IEC. This method quickly learns effective rules by simple instructions of a human operator. The objective of IER is to make initial learning more efficient and learn behaviors that a human operator intended through interaction with him/her.

First of all, we developed a learning system based on *classifier system*[7] on IER framework, which is able to adapt to multiplicity of an environment and a variable dynamic state. we call it as *Interactive Classifier System* (ICS). Fig.1 give an environment of teaching using ICS. The major difference between ICS and a traditional learning classifier system is to introduce an interactive method. Accordingly, we expect that the system efficiently performs initial learning in an actual environment and can operate concentrative incremental learning. However there is few framework that an operator observes from robot's view. Therefore, the system can not make the best use of the learning. In this paper, we propose interactive method based on internal observation in order to solve the problem.

Nordin et al.[8][9] work a robot learning using genetic programming in a real environment. They argue a real robot learning is possible in a task of an obstacle avoidance. Unfortunately, it take a long time to learn the efficient behavior yet. Nicolescu et al.[10] researches learning by interacting with a human like as our approach. However, it is very different from our approach for reasons of what it applies experience of the interaction to evolutionary learning. Mondada et al.[11] works a robot learning with evolutionary

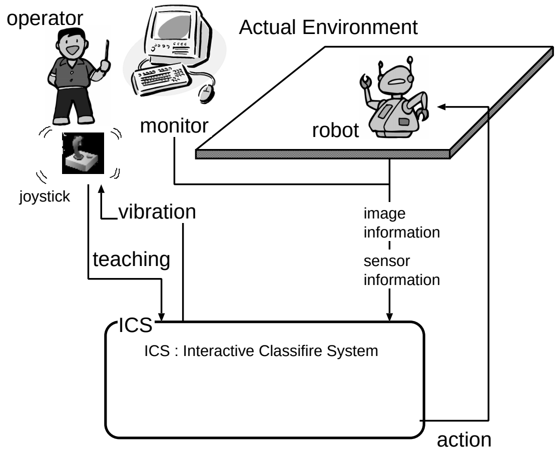


Figure 1: Teaching environment

learning as well as Nordin. However they report that the learning takes scores of hours to converge.

2 Teaching Method Based on Different Viewpoints of Observation

A robot hardly knows how it modifies behaviors of itself for a task. Furthermore it may be difficult to recognizes processing of a task. It is not an observed approach through an inside viewpoint of a system, but an observed one through an outside viewpoint to be able to recognize achieving of a task.

In case of an operator usually teaches to a robot, an external observer don't recognize its internal information. It is general that an operator teaches to a robot by guessing its internal information. Therefore, we had a significant difference between cognition of an operator and one of a robot, which is called a *perceptual aliasing problem.* It become a subject of discussion when a operator teach skills for a task to a robot.

In this paper, we prepare the simple setting based on the observation to examine how the difference influence to acquired rules by teaching. We call a method to observe through an outside viewpoint of a system as *teacher view (TV)* (Fig.2), and a method to observe through an inside viewpoint of a system as *learner view (LV)* (Fig.3).

We examine the difference by IER based on teaching with this TV and LV. To realize this IER, we applied these methods to developed ICS, which is a robot learning system based on interactive EC.

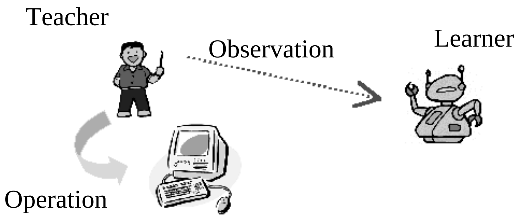


Figure 2: Teacher view

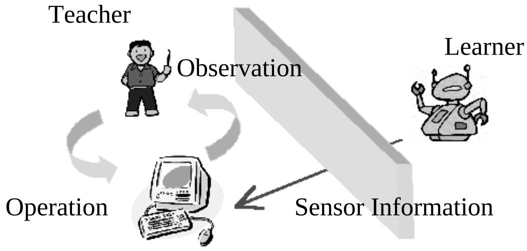


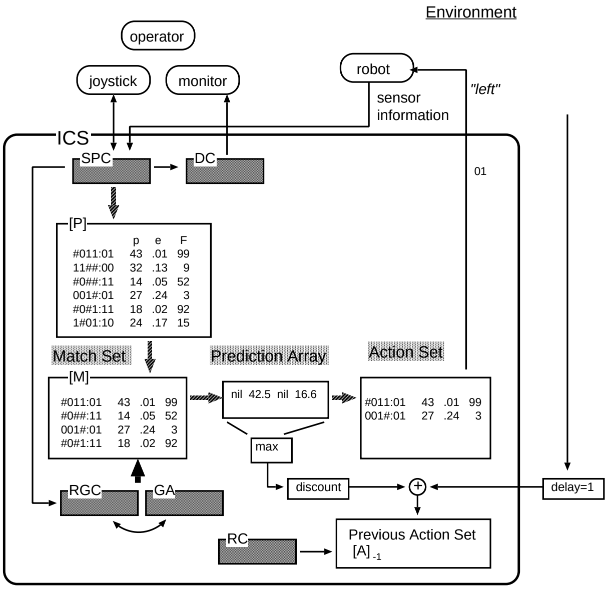
Figure 3: Learner view

3 Interactive Classifier System

3.1 System overview

ICS applies XCS[12], which is a kind of Learning Classifier System (LCS), as evolutionary computation and introduces a interactive function. XCS equipped a function to preserve classifiers from overgeneralization to make system's performance worse. Moreover, XCS applies restricted mating which is a kind of strategy based on *genetic algorithms.* For these reason, XCS improves learning performance of traditional CS. It is constructed as a robot learning model that can not only learn through teaching but also learn autonomously using XCS.

ICS mainly consists of a rule generation component (RGC), a sensor processing component (SPC), a display component (DC) and a reinforcement component (RC) (Fig.4). It was implemented with C language and GTK+ on Linux. It utilizes Video4Linux for image processing. The rule generation component makes a new classifier from teaching by the operator. The SPC processes each information of some sensors and camera, and through it for the RGC. The DC displays by GUI interface and processes the

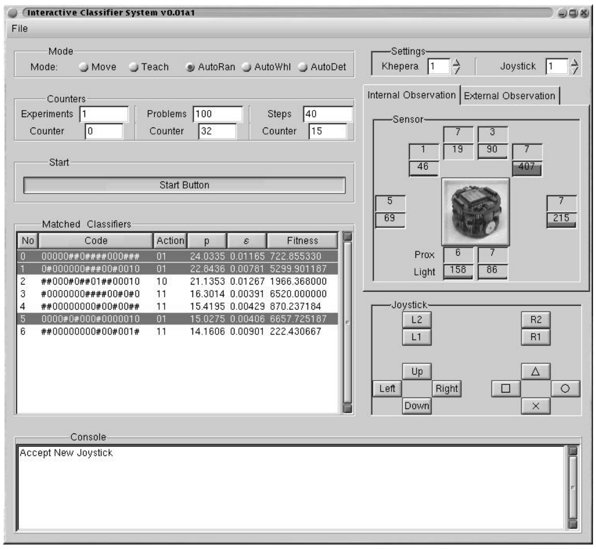


	p	e	F
#011:01	43	.01	99
11##:00	32	.13	9
#0##:11	14	.05	52
001#:01	27	.24	3
#0#1:11	18	.02	92
1#01:10	24	.17	15

#011:01	43	.01	99
#0##:11	14	.05	52
001#:01	27	.24	3
#0#1:11	18	.02	92

#011:01	43	.01	99
001#:01	27	.24	3

Figure 4: *Overview of Interactive Classifier System*



Mode: ○ Move ○ Teach ● AutoRan ○ AutoWhl ○ AutoDet

Settings: Khepera 1 Joystick 1

Counters: Experiments 1 Counter 0 | Problems 100 Counter 32 | Steps 40 Counter 15

Sensor: 7 / 3 / 1 / 19 / 90 / 7 / 46 / 407 / 5 / 69 / 7 / 215 / Prox 6 7 / Light 158 86

Matched Classifiers

No	Code	Action	p	ε	Fitness
0	00000##0####000###	01	24.0335	0.01165	722.855330
1	0#000000###00#0010	01	22.8436	0.00781	5299.901187
2	##000#0##01##00010	10	21.1353	0.01267	1966.368000
3	#0000000####00#0#0	11	16.3014	0.00391	6520.000000
4	##00000000#00#00##	11	15.4195	0.00429	870.237184
5	0000#0#000#0000010	01	15.0275	0.00406	6657.725187
6	##00000000#00#001#	11	14.1606	0.00901	222.430667

```
Accept New Joystick
```

Figure 5: *User interface*

input from a joystick. Finally the RC performs learning by updating parameters in ICS. Fig.5 shows the developed an interface for the system. The middle of the left of the interface indicates some matched classifiers in this time. The middle of the right indicates robot's sensor information and state of a joystick.

The experiments are made with a standard miniature mobile robot Khepera (Fig.6). The mobile robot has cylindrical shape, a diameter of 6 cm and a height of 5 cm. It possesses two motors and on-board power supply. The motors can be independently controlled by a PID controller. It is equipped with eight infrared proximity sensors. The eight infrared sensors are distributed around the robot in a circular pattern. They emit infrared light, receive the reflected light and measure distances in a short range: 2-5 cm. The robot is also equipped with a Motorola 68331 micro-controller which can be connected to a computer via serial cable. Moreover, the system utilizes SONY analog controller DUALSHOCK as a joystick. Fig.6 shows them respectively.

3.2 ICS procedures

We describe a learning procedure in ICS as follows.

1. At first, a human operates robot with a joystick by viewing sensor information displayed on GUI, and the DC processes it.

2. Next, the SPC gets operator's instruction and robot's sensor information.

3. The RGC makes new rules from them and adds them into a rule list. When nothing is input from the operator, a mobile robot executes autonomous behaviors.

4. Finally, the RC reinforces the classifiers by updating their parameters in the actions which were previously executed.

In traditional works of robot learning in a real environment, a learning takes pretty much time to converge because of learning by trial and error. In this work, we consider that this learning by trial and error is a problem in a real environment. However, it hardly prepare suitable a priori knowledge in an environment. For this reason, the ICS generate initial individuals by teaching from human-robot interaction. We can perform efficiently initial learning in this way.

4 Experiments

4.1 Experiments with cognitive observation

We experimented in a real world to investigate difference in two teaching methods: TV and LV as mentioned earlier. Fig.7 shows an experimental environment. As an experimental task, ICS reduces the number of steps from any start points to a light source which set up as a goal in a field surrounded with white plastic plates. We compared two teaching methods (TV and LV) with a traditional method in which a robot autonomously learns by simple EC. To investigate influence by gap between operator's

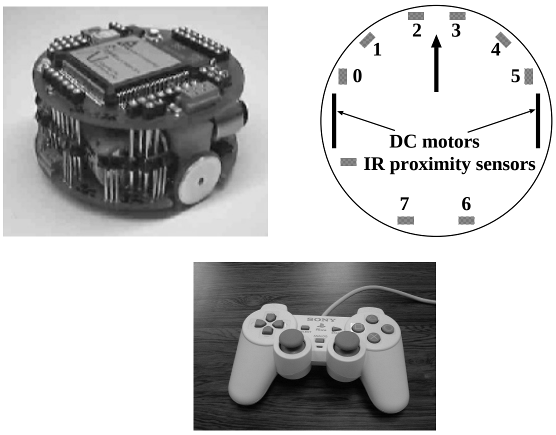


Figure 6: *A mobile robot Khepera and a joystick*

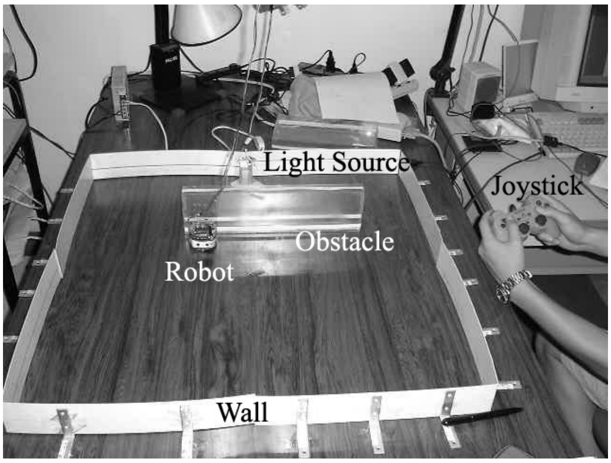


Figure 7: *Experimental Environment*

Table 1: *Experimental Parameters*

Parameters	Value
The number of problems in one experiment	30
The number of expriments	1
Maximum size of the population	300
Probability to do crossover	0.8
Probability of mutating one bit	0.04

recognition and robot's one, we prepared *experiment 1* without an obstacle and *experiment 2* with some obstacles.

We consider forty steps as a trial and begin teaching or autonomous exploration for five trials at random start points, and test a trial at each of five start points as examination of learned rules by the teachings or the exploration. It performs this procedure six times, and consequently we have thirty trials as examination. Table 1 shows experimental parameters.

An encoding of a classifier is as follows. A classifier is the twenty bit string "#000#100000###100#:01". The robot's condition is the left sixteen bit string, it represented eight infrared proximity (left eight bit) and light (right eight bit) sensors around the robot in a circular pattern respectively (Fig.6). The bit is "1" if a input sensor value larger than a threshold, or else "0". "#" is a "don't care" symbol which classifier system employed. The robot's previous action is the next two bit string, represented as forward "11", left-turn "01", right-turn "10 " and back "00". The robot's current action is the two bit string similarly.

A fitness function of ICS defined as follows. Reward F is computed by the sum of eight light sensors through a sigmoid function.

$$u = \sum_{i=0}^{7} light_i \frac{\alpha}{light_{MAX} \times 8} \qquad F = \frac{1}{1 + exp(-u)}$$

In the case of TV, the operator can not recognize a small obstacles as the robot can perceive although he/she looks a whole environment. To represent TV, an operator teaches a robot using a camera which looks a whole environment. Fig.8 shows information through GUI by a camera which sets up the environment.

In the case of LV, an operator can not look a whole environment although it can recognize directly a small obstacles and a recognition error which the robot can perceive. To implement LV, ICS uses GUI which indicates some sensor values. Since an operator hardly recognize an environment by reading only numerical values of sensor data, we developed GUI interface which can represent sensor values graphically. Fig.9 shows such sensor information through GUI interface.

In the experiment 1, we examined the number of reach times to a light source from five start points. Fig.10 shows the number of times to reach to a light source (goal of the task) at test trials from five start points. Though LV begins to success the task at early stage, it doesn't success later. On the other hand, TV improves robot's learning and achieves the task from all five start points before LV. TV improves a robot's learning in the simple environment which cognitive difference is little by teaching is easy because an operator looks a whole environment. However, the difference is not so large.

In the experiment 2, we introduced an obstacle in the

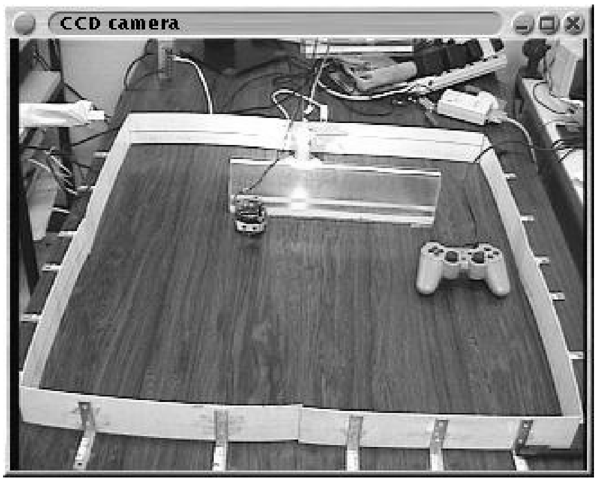
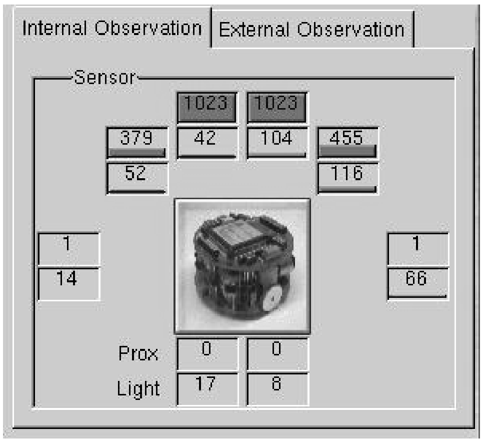


Figure 8: *External Observation Settings*



Sensor

1023 1023

379 42 104 455

52 116

1 14

1 66

Prox 0 0

Light 17 8

Figure 9: *Internal Observation Settings*

environment of experiment 1 to investigate the effect by a more difficult task. The obstacle was made of the transparent plastic board because of perception a direction of a light source. A robot must reach to the light source as avoiding the obstacle. We compared a teaching method by TV with LV in the same way as experiment 1.

We examined the number of steps to a light source. The number of steps to a light source shows Fig.11. Since ICS hardly uses information of proximity sensors and acquires effective rules by using only information of light sensors in the environment 1, the difference of cognition between an operator and a robot is a little. For this reason, the difference of two teaching methods was not so large.

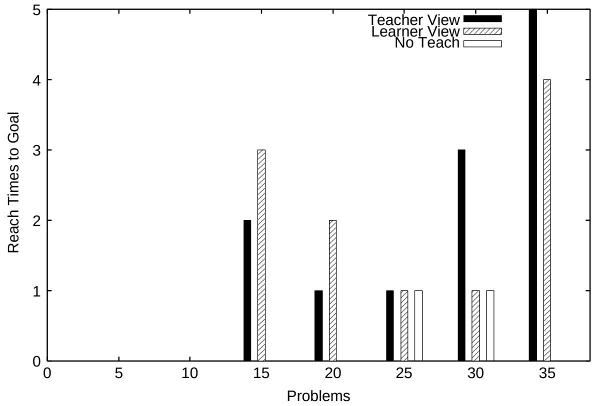


Figure 10: *Reach Times to Light Source*

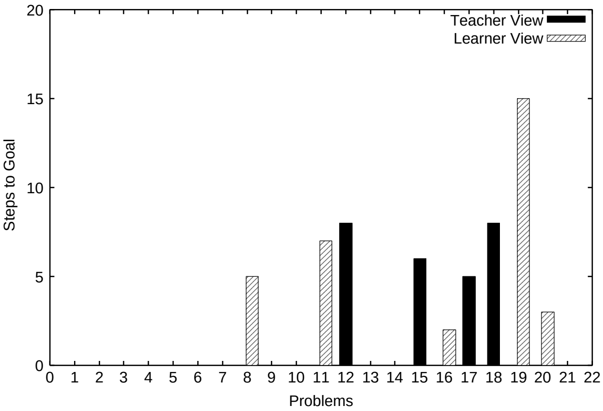


Figure 11: *Steps to Light Source*

However ICS hardly learns from teaching of an operator in case of experiment 2 because there is the difference between the situation of the robot which an operator estimated by observation from the outside and the one of the robot in the real world. LV actually creates effective rules because an operator performed teaching as he/she checks robot's internal status.

After we have experiments TV and LV each twenty trials, we test a trial at each of five start points as examination using each created rules. Table 2 shows the experimental results.

The robot can not reach to a light source at any start points of five in the examination of TV. On the other hand, the robot reached to a light source by six steps and nine steps respectively at two of five start points in the examination of LV. We can see that ICS can create effective rules independently of start points

Table 2: *Experimental Results of Exploit*

	Step to Light Source					Reach to Goal
times	1	2	3	4	5	total
TV	-	-	-	-	-	0
LV	6	9	-	-	-	2

Table 3: *Created Rules by LV Method*

Condition	Action	Prediction	Teach
0#0#0######0#10##1#	10	413.7	2
0010###0##000#1###	01	364.0	1
00###0#000#1#0#01#	11	292.0	4
###000000101001011	11	256.0	1

since LV improves the robot's learning by teaching.

Table 3 shows ten rules which have best values of the system prediction in created ones after twenty trials with LV. ICS created effective and commonsense rules like as the robot moves forward when the light source faces in front of the robot and it moves right when the light source faces right. In this table, *Prediction* is a fitness of a classifier and *Teach* is whether it is taught by human or its offspring. Its number is the number of which a taught classifier alternate generations. We can see teaching improves very well because all the rules whose the system prediction is high created from teaching or its offspring.

Though an operator did not teach a robot to go back in the both experiments, the robot goes back for avoiding an obstacle when the robot collides with a wall and reaches to a light source. Because ICS created these rules by which a robot works in cooperation with human.

5 Conclusion

In this paper, we proposed a novel interactive method from the viewpoint of observation and investigated its effects in a real world experiments. It was found that internal observation increase the effects according to arise the difference between the robot's recognition and operator's one in more complex environment.

References

[1] R. Dawkins. *The Blind Watchmaker.* Longman, Essex, 1986.

[2] Y. Nakanishi. Capturing Preference into a Function Using Interactions with a Manual Evolutionary Design Aid System. Genetic Programming, pages 133–140, 1996.

[3] T. Unemi. A Design of Multi-Field User Interface for Sumulated Brdding. Asian Fuzzy System Symposium, pages 489–494, 1998.

[4] E. Uchibe, M. Asada, and K. Hosoda. Behavior coordination for a mobile robot using modular reinforcement learning. In IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96), pages 1329–1336, 1996.

[5] D. Katagami and S. Yamada. Interactive classifier system for real robot learning. In *IEEE International Wortkshop on Robot and Human Interaction*, pages 258–263, 2000.

[6] D. Katagami and S. Yamada. Real robot learning with human teaching. In *The Fourth Japan-Australia Joint Workshop on Intelligent and Evolutionary Systems*, pages 263–270, 2000.

[7] J. H. Holland and J. S. Reitman. Cognitive systems based on adaptive algorithms. In Donald A. Waterman and Frederick Hayes-Roth, editors, *Pattern-Directed Inference Systems*, pages 313–329, Orlando, 1978. Academic Press.

[8] P. Nordin and W. Banzhaf. Genetic Programming Controlling a Miniature Robot. In Working Notes for the AAAI Symposium on Genetic Programming, pages 61–67, 1995.

[9] P. Nordin and W. Banzhaf. An on-line method to evolve behavior and to control a miniature robot in real time with genetic programming. Adaptive Behavior, 5(2):107–140, 1996.

[10] M. Nicolescu and M. J Mataric. Learning and Interacting in Human-Robot Domains. IEEE Transactions on Systems, Man, Cybernetics, 31(5), 2001.

[11] F. Mondada and D. Floreano. Evolution and mobile autonomous robotics. In In Proceedings of the Evolutionary Engineering, pages 221–249, 1995.

[12] S.W. Wilson. Classifier fitness based on accuracy. Evolutionary Computation, 3(2):149–175, 1995.