Speedup of Evolutionary Behavior Learning with Crossover Depending on the Usage Frequency of a Node

Daisuke KATAGAMI and Seiji YAMADA

CISS, IGSSE, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku Yokohama 226-8502, JAPAN
{katagami,yamada}@ymd.dis.titech.ac.jp

ABSTRACT

In this paper, for on-line robot behavior learning, we propose heuristics using node usage for speedup of evolutionary learning, and verify the utility experimentally. Genetic Programming (GP) is an evolutionary way to acquire a program through interaction with an environment. Since behaviors of a robot are described with a program, researches on applying GP to robot behavior learning have been activated. Unfortunately, in most of the studies, the behavior learning is done off-line using simulation, not a real robot. Because convergence of GP is slow, and this makes operation of a real robot quite expensive. However, since situations out of simulation easily happens in a real world, the behavior learning with a real robot (called on-line learning) remains very significant. Thus, in order to make on-line behavior learning with GP practical, we propose a novel crossover method for speedup of GP using node usage of a program.

1 Introduction

Recently, a robot learning using Genetic Programming (GP)[1] actively has been researched. However, unfortunately most of their researches use simulation, it hasn't been still experiment using a real robot in real environment. As for on-line learning using a real robot, it has been experimented to learn obstacle avoidance behavior using GP[5].

For the efficient way to search by GP, methods using a subroutine such as ADF (Automatically Defined Functions)[3] or MA (Module Acquisition)[2] have been developed. These methods have the advantage that a partial structure of a solution can be preserved from a destructive crossover.

However there isn't attempt to improve a destructive crossover in itself, and crossover points are determined randomly. In a case of using a conditional sentence like a If-sentence in GP, we observe the difference among nodes (functions) in frequency of usage. Only a part of nodes is used very often, and the other is hardly executed. Also this tendency keeps without the extreme change of an environment. It is significant that if crossover points are determined in nodes of low usage frequency both in parents, the crossover does not influence to children's behavior. Thus we consider the effectiveness of crossover depends on the usage frequency of the crossover points.

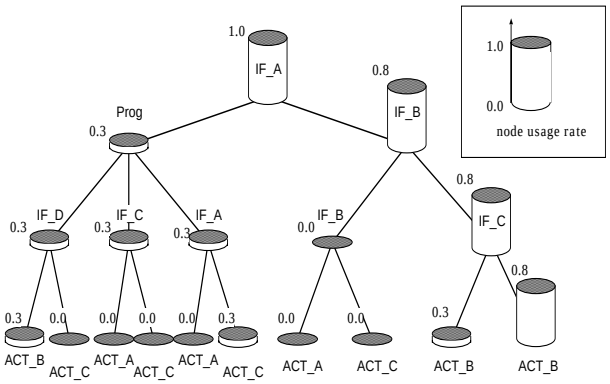

Figure 1: Distribution of node usage frequency.

Hence we propose a novel crossover using the usage frequency of nodes.

We develop three crossover methods: a crossover with crossover points in two nodes having high usage frequency, a crossover with crossover points in a node having high usage frequency and a node having low usage frequency, and a crossover with crossover points in two nodes having low usage frequency. We apply three crossovers to on-line behavior learning of a real mobile robot Khepera in two tasks: obstacle avoidance and box-pushing, and make systematic experiments for investigating the efficiency of learning. As a result, we found out our approach is promising for speedup of on-line behavior learning using GP.

2 The method of crossover depending on the usage frequency of a node

2.1 The usage frequency of a node

In this study, we defined the number of the usage of nodes as the usage frequency of nodes. First of all, we studied that the usage frequency of nodes varied each a node by a preliminary experiment. We experiment that a mobile learns box pushing behavior. Fig.1 shows the usage rate of nodes(18th individual, 21st generation) learn by the experiment. A height(a column) of nodes show the usage rate of nodes. the usage rate of nodes calculated A right tree is chosen by IF_A which is the first condition divergence sentence by this figure in the case of most, and a left tree is hardly used.

Like this, it knows the thing that the big difference comes out to the usage rate by the node. The node used by using the condition divergence used for the non-terminal sets of GP well in this environment, and the node which isn't used are made. Therefore it can be thought to be the results that it is accumulated. So far as there is no big change of the environment, the node which isn't used by that condition divergence means that it isn't used after that as well as for this thing. It pays attention to the correlation with the usage frequency and the experiment environment of that node, and aims at improving learning efficiency by including it into heuristic of the crossover operation by this research.

2.2 The crossover depending on the usage frequency of a node

The usage rate of nodes is calculated with the following formula.

$$R_i^m = \frac{C_i^m}{N_{max}}$$

The usage frequency of the node m in the individual i, N_{max} are the maximum number of the node with R_i^m with the usage rate of the node m in the individual i, C_i^m in the population here. In most research of GP, the case that an crossover point is decided at random like one-point crossover in Genetic Algorithm is most about the crossover operator. We call general crossover methods like that as *random crossover*. We prepared for the following three setups that crossover probability was changed by the usage rate of the node.

1. The setup of increasing the crossover probability with high node of the usage frequency and low node(The crossover function 1 : H1)
2. The setup of increasing the crossover probability with high nodes of the usage frequency(The crossover function 2 : H2)
3. The setup of increasing the crossover probability with low nodes of the usage frequency(The crossover function 3 : H3)

Crossover function 1 The following formula is made an evaluation function to increase the crossover probability at a low node and a high node of the usage frequency.

$$H_{mn}^{ij} = (R_i^m - R_j^n)^2$$

$$H1_{mn}^{ij} = \frac{H_{mn}^{ij}}{\sum_{i,j} H_{mn}^{ij}}$$

Where $H1_{mn}^{ij}$ are the crossover probability that node m of the individual i, j, n become the points of the crossover with H_{mn}^{ij} here. The landscape is shown in Fig.2.

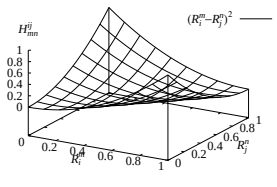

Figure 2: Crossover function 1(H1).

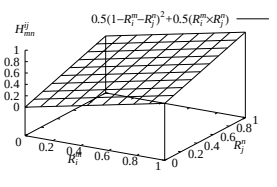

Figure 3: Crossover function 2(H2).

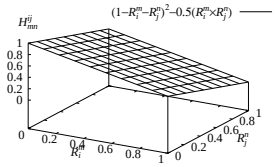

Figure 4: Crossover function 3(H3).

Crossover function 2 The following formula is made an evaluation function to increase the crossover probability with high nodes of the usage frequency. The landscape is shown in Fig.3.

$$H_{mn}^{ij} = 0.5(1 - R_i^m - R_j^n)^2 + 0.5(R_i^m \times R_j^n)$$

$$H2_{mn}^{ij} = \frac{H_{mn}^{ij}}{\sum_{i,j} H_{mn}^{ij}}$$

Crossover function 3 The following formula is made an evaluation function to increase the crossover probability with low nodes of the usage frequency. The landscape is shown in Fig.4.

$$H_{mn}^{ij} = 1.0(1 - R_i^m - R_j^n)^2 - 0.5(R_i^m \times R_j^n)$$

$$H3_{mn}^{ij} = \frac{H_{mn}^{ij}}{\sum_{i,j} H_{mn}^{ij}}$$

The experiment which two tasks were different from was done to examine the effect of each evaluation function by using these evaluation functions. Each is made the Exp.A, the Exp.B, and it explains in the following.

3 Setup of the Exp.A and Exp.B

3.1 An environment and task

We made experiments with a standard autonomous miniature robot Khepera. It is equipped with eight infrared proximity sensors. The mobile robot has a circular shape with 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

Table 1: Parameters of GP(Obstacle Avoidance).

	Obstacle Avoidance	Box Moving
Terminal	Forward,Right,Left	Forward,Right,Left ,Right_Round ,Left_Round
Function	IF_Front,IF_Right ,IF_Left,IF_Back	IF_Front,IF_Right ,IF_Left,IF_Back ,Prog3
population size	50	40
step	50	60
selection method	tournament	
tournament size	4	

a PID controller. 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.

First, the experiment of the obstacle avoidance was done in the Exp.A to examine the effect of each evaluation function. Obstacle avoidance is a comparatively easy task because an answer can be expressed by mapping of the behavior toward the condition. Therefore, it is suitable for comparing the effect of the above-mentioned 3 setup. We prepared for 2 obstacles of the cube on the table of the flat rectangle of $110cm \times 90cm$, and installed the wall made of the white plastic board surrounded by them.

Next, the task of the acquisition of the box pushing behavior was done by using the evaluation function 2 that good results came out from the results of the Exp.A. This is an Exp.B. An light source and the box of the cube made of the transparent plastic board on the table of the flat rectangle of $110cm \times 90cm$ were put, and the experiment of the acquisition of the box pushing behavior was done from two initial positions where it varied in this environment in the light source direction.

A function like Table.1 was designed in the terminal sign of GP and the non-terminal sign.

4 Measures of the Exp.A and Exp.B

We divided an experiment into test phase with training phase. A robot learning in training phase, and the best individual which it could get in that process was evaluated again in test phase. As for the experiment A, a robot does learning through the a series of behavior. It begins in the initial position from the place where it is different every time for that reason.

It was arranged with the training at random in two initial positions, and learning was done and evaluated about those two places with the box push of the Exp.B by the test. We built the system of GP based on the evolution algorithm [4][5]which Nordins were using as the system of this research.

That procedure is shown in the following.

1. Select four arbitrary programs from the population.
   (a) Run each programs, and read proximity and light sensors.
   (b) For each of calculate a fitness value F.
2. Make two offspring of the two individuals with highest fitness and let the copies be subject to crossover and mutation. The following heuristic is carried out in the case of crossover.
   (a) For the combination of all nodes of two individuals calculate evaluation value H^{ij}_{mn}.
   (b) Calculate the $H1^{ij}_{mn}(or H2^{ij}_{mn}, H3^{ij}_{mn})$ crossover probability on the combination of all nodes from evaluated value.
   (c) Decide crossover points using the roulette strategy.
   (d) Carry out crossover in the decided point, and make two new offspring.
3. Replace the two individuals with worst fitness with the two new offspring.
4. Repeat step 1 to 4.

We did an experiment about the random crossover and this technique which the crossover function 1,2,3 by using this procedure respectively.

5 Acquisition of Obstacle Avoidance Behavior:Exp.A

First, the experiment of the obstacle avoidance was done as the Exp.A to examine the effect of each evaluation function.

5.1 Fitness function

An fitness function was defined as follows.

$$F_{std} = \alpha s_{max} + \beta(|10 - m_1| + |10 - m_2| + |m_1 - m_2|)$$

$$F_{adj} = \frac{1}{1 + F_{std}} \qquad Fitness : F = \frac{F_{adj}}{\sum_{i=0}^{p} F_{adj}}$$

s_{max} returns the biggest value $(0 - 1023)$ in eight proximity sensor value here. Moreover, m_1 and m_2 shows the speed of the left wheel and the right wheel respectively, and takes binary here $-10 - 10$. Big value is taken, and this function lowers evaluation as much as to be close to the obstacle. Big value is taken, and

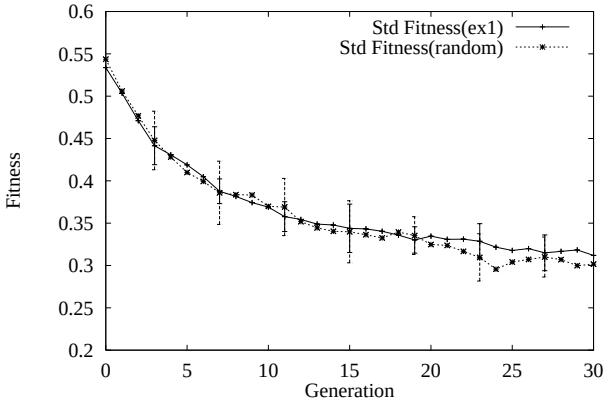

Figure 5: Ave. of fitness: $H1$.

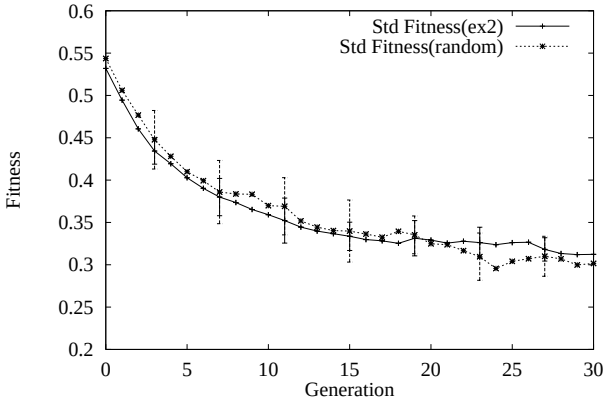

Figure 6: Ave. of fitness: $H2$.

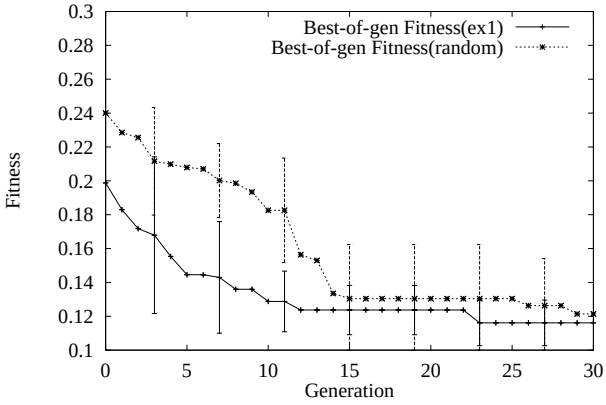

Figure 7: Max. of fitness: $H1$.

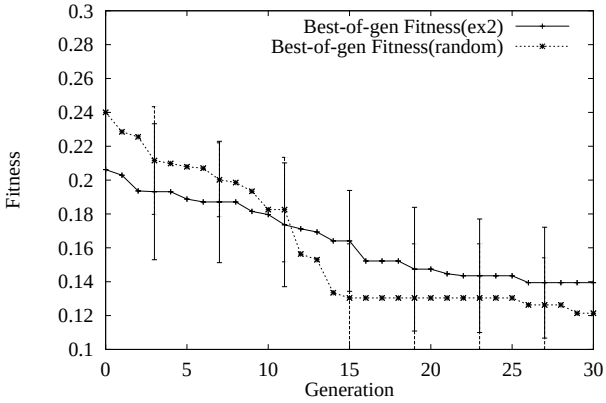

Figure 8: Max. of fitness: $H2$.

evaluation is lowered when a robot turn at fixed point without advancing. Fitness F_{std} obtained this experiment at population size p times. We use the results which normalized F_{std} and made the best value 0.0 as fitness F. α, β were coefficients, and it used $\alpha = 1.0$, $\beta = 10.0$ this time.

6 The results of Exp.A

6.1 Comparison with Random Crossover:Average of fitness

The experiment is made ten times by changing seed using crossover function 1,2 and 3, and we obtained the average of the mean value of the fidelity in the training. And, compared with the experiment using *random crossover*. Furthermore, the standard deviation of the experiment is shown every 4 generation in the graph. Fig.5 shows the change of average of the mean of fitness every generation in the experiment using crossover function 1 and *random crossover*. Fig.6 shows the change of average of the mean of fitness every generation in the experiment using crossover function 2 and using *random crossover*.

It was not the difference about the mean of adaptation of population using *random crossover* very much. This technique shows that the whole search of the group lacks an influence very much in this.

6.2 Comparison with Random Crossover:The best of fitness

We obtained the change of every average generation of the best value of fitness in the experiment. It compared with the experiment using *random crossover*. Furthermore, the standard deviation of the experiment is shown every 4 generation in the graph. Fig.7 shows the change of average of the mean of best fitness every generation in the experiment using crossover function 1 and using *random crossover*. The best value gets the high value of the evaluation in the generation whose it is early. It is found clearly that the crossover of the learning efficiency which it is all right as for is done by giving it evaluation gradually. Fig.8 shows the change of average of the mean of best fitness every generation in the experiment using crossover function 2 and using *random crossover*. The best value got a little higher value than the method of *random crossover* with the

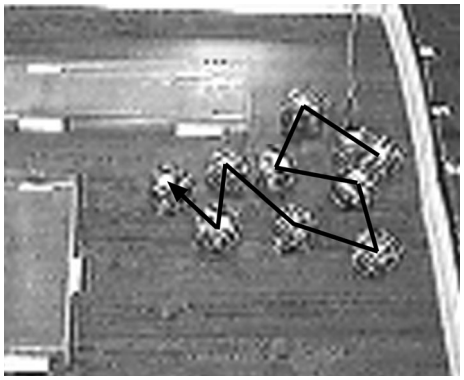

Figure 9: The moving locus(Ex.2, Gen.20).

generation whose it was early. However, it caught up in around the generation 10, and got the value that the method of using *random crossover* was better after that. The experiment which the crossover function 3 was used for showed worse value than before. As for the crossover with nodes which aren't used, it is found that it makes efficiency worse as for the search of the best value. Test value didn't almost change with before.

Fig.9 shows the example of a locus which a robot actually moves by using the evaluation function 1. This figures are the moving locus of the robot by the program which it gets with the generation 20 of the done Exp.A by using the evaluation function 2. When the robot seems to knock against the obstacle and the wall, it converting the direction without colliding and the evaluation has been given at generation 20. We did an experiment about three setups which made crossover probability change by the usage probability of a node by the task of the obstacle avoidance. It was evaluated about the mean of a degree of adaptation, the best value, the test value respectively toward three evaluation functions. As for the evaluation function 3, bad value was shown if it was the same in comparison with the method of using *random crossover*. Crossover with low node of the usage frequency didn't appear in the expression type easily. Therefore, this showed low evaluation as results as for the easy task when it was compared with the method of using *random crossover*. The graph showed a high evaluation respectively compared with *random crossover* about the evaluation function 1 and 2. This thing shows that this technique is effective. Moreover, when we observed the behavior of the robot which is an expression type, it was observed that the behavior which an evaluation function 2 was a little excellent about when it is compared with an evaluation function 1 was shown. The results of our experiment clearly shows that this technique was effective.

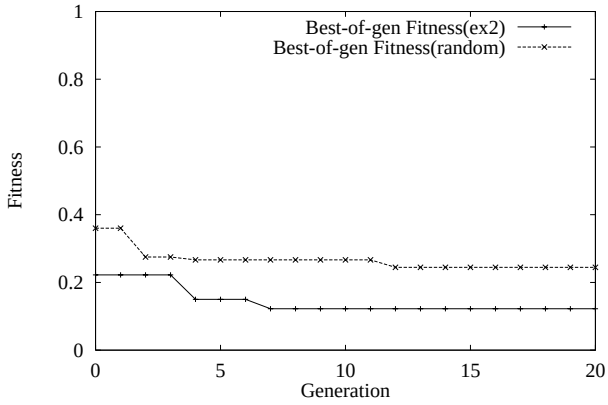

Figure 10: Average of best-of-gen fitness(Ex2).

7 Exp.B:Acquisition of box pushing behavior

We experimented about acquisition of box pushing behavior by using crossover function 2 with a good evaluation from the results of the Exp.A.

7.1 Fitness function

We defined a fitness function as follows.

$$F_{std} = \alpha \sum_{i=1}^{n} u_i + \beta \sum_{i=1}^{n} v_i + \gamma \sum_{i=0}^{n} w_i$$

$$F_{adj} = \frac{1}{1 + F_{std}} \qquad Fitness : F = \frac{F_{adj}}{\sum_{i=0}^{p} F_{adj}}$$

where u_i, v_i and w_i takes binary here 0,1. u_i evaluated the ith terminal sign and 1 is returned, when there was a reaction in the infra-red proximity sensor of the front this time. v_i returns 1 in the same way when there was a response in the ambient light sensor in front in addition. w_i returns 1 when a box is pushed in addition.

8 The eesults of Exp.B

8.1 Comparison the crossover fitness 2 with Random Crossover:The best fitness

Fig.10 shows the change of average of the mean of fitness every generation in the experiment using crossover function 2 and the method of using *random crossover*. This graph shows generally high evaluation in comparison with the method of using *random crossover*. As for this experiment, it is a difficult task to solve it by mapping of the behavior toward the condition. And, it increases in the 8×8 condition from 8 condition in the Exp.A even if sensor value is shown by adding the general idea of the light source direction by 2 value simply. It increases by 1 condition 1 behavior in the 8×8 condition from 8 condition in the Exp.A even if a

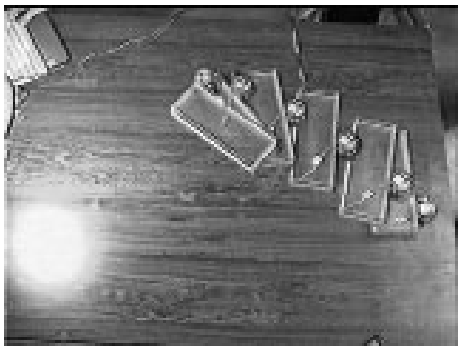

Figure 11: The moving locus(Ex2,Gen19).

thing to solve shows sensor value by adding a concept of the light source direction in addition to the difficult task by 2 value simply. This thing made the search of the answer of the task a difficult thing, and a difference became easy to appear by the crossover being used as the operation for the master conversely as results. It faces the problem of the acquisition of the best value, and it can think with the results which effect appeared in. A mean and test value showed the value which didn't almost change with before.

Fig.11 shows the example of a locus which a robot actually moves by using the evaluation function 2 in the Exp.B. A box is pushed, and it loses sight of a box soon for the first time as for the behavior of the robot of the generation whose it is early, too. This is so that it can't take suitable behavior when it is here with the program adapted to the change in the environment having not been made yet and a robot leaving a box. The robot of the generation 20 changes a direction when it is understood that a light source is in the left though it faces in the light source direction and a box is being pushed a little to right. It is understood more than this thing that the program which adapt with that change is being made with recognizing the direction which the light source direction and a box touch.

We did an experiment by using the crossover function 2 the behavior acquisition of the robot could be thought to be done better more than the results of the Exp.A by the task of the box push behavior again, and compared that results with the method of using *random crossover*. It was compared with the method of using *random crossover* in the best value, and it could get high evaluation with the generation whose it was early, and the effectiveness of this technique became clear as that results.

9 Conclusion

This paper proposed the novel crossover technique depending on the usage frequency of a node. We had the experiment on-line learning which an actual robot was used for with the system which this technique was applied. When the crossover function 1 and an evaluation function 2 in this study were used, the operation whose learning efficiency was very good could be done in comparison with *random crossover* as a result of the experiment. The validity of this method and its good performance are confirmed through the experiments.

REFERENCES

[1] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, dpunkt.verlag, January 1998.

[2] Kenneth E. Kinnear, Jr. Alternatives in automatic function definition: A comparison of performance. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 6, pages 119–141. MIT Press, 1994.

[3] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts, May 1994.

[4] Peter Nordin and Wolfgang Banzhaf. Genetic Programming Controlling a Miniature Robot. In E. V. Siegel and J. R. Koza, editors, *Working Notes for the AAAI Symposium on Genetic Programming*, pages 61–67, MIT, Cambridge, MA, USA, 10–12 November 1995. AAAI.

[5] P.Nordin and W.Banzhaf. An on-line method to evolve behavior and to control a mini ature robot in real time with genetic programming. In *Adaptive Behavior*, pages 107–140, 1996.