Improving the robustness of a genetic programming learning method by function set tuning

Maria Prateeptongkum[1] and Prabhas Chongstitvatana[2]
Department of computer engineering
Chulalongkorn University
Phayathai road, Bangkok 10330,Thailand
Phone (662) 218-6982, Fax (662) 218-6955
maria@ncstc.mua.go.th[1], prabhas@chula.ac.th[2]

Abstract

This paper presents robustness improvement of robot programs generated by genetic programming. The technique to find a robustness solution is by function set tuning. The main hypothesis is a program that can express a wide variety of behaviors according to different situations should be more robust. Based on this idea, a special probabilistic function -- 2-way selection -- is defined. The experiment is performed on robot navigation problems moving from a starting point to a target point within a closed-area environment which contains obstacles. The result supports the hypothesis. It can be shown that a program with a special function extension behaves more robustly.

1. Introduction

Genetic Programming (GP) is a problem solving method that is inspired by natural evolution. Recently, it has been applied to various problem domains. GP has been using as a method for robot learning. In [1], GP is used to generate robot programs that solve visual reaching tasks. The solution is evolved under a simulated world because of the time constraint in using a real robot. A problem that often arises is the failure to transfer the success of a solution in a simulated world to the real-world because such solutions are fragile. We can not establish all environments and operations in a real-world situation completely similar to those in a simulated world.

To reduce failure of transferring solutions from simulated world to execute in real-world, it is essential to generate robust solutions which are strong enough to withstand errors in the real-world that may happen. Attempt to improve solution to be more robust, some GP researchers injected noise into system during process of evolution. Noise could be disturbance in environment, changing initial conditions or errors in sensor and actuator. [3] had limit success result in the box moving problem, whereas [5] found unsuccessful solution for corridor-following task. Another approach to improve robustness of the solution is to generate solutions under several training environments without putting any noise into system and [2] shown that solution which take more training environment has more robust result.

This paper examines robustness improvement of robot programs using function set tuning. We constructed a special probabilistic function in order to support our hypothesis that claims a program which can express a wide variety of behaviours according to different situations should be more robust. The special function is a kind of 2-way selection function. Executing this function resulting in a random selection of the path. The result shows that a program with a special function extension behaves more robustly. The experiment is performed on robot navigation problems. Robot programs are generated by GP to navigate mobile robot from a starting point to a target point among spread obstacles within a closed-area environment. The rest of this paper describes the experiment and the result.

2. The experiment

The test case of our method is the robot navigation problem. A mobile robot learns to travel from a starting point to a target point in a cluttered environment (fig. 1) negotiating obstacles. GP is used as the robot learning method. The aim is to generate robot programs that successfully perform the task with high robustness. The robustness is defined as the ability to perform well (achieving the goal) despite the disturbance.

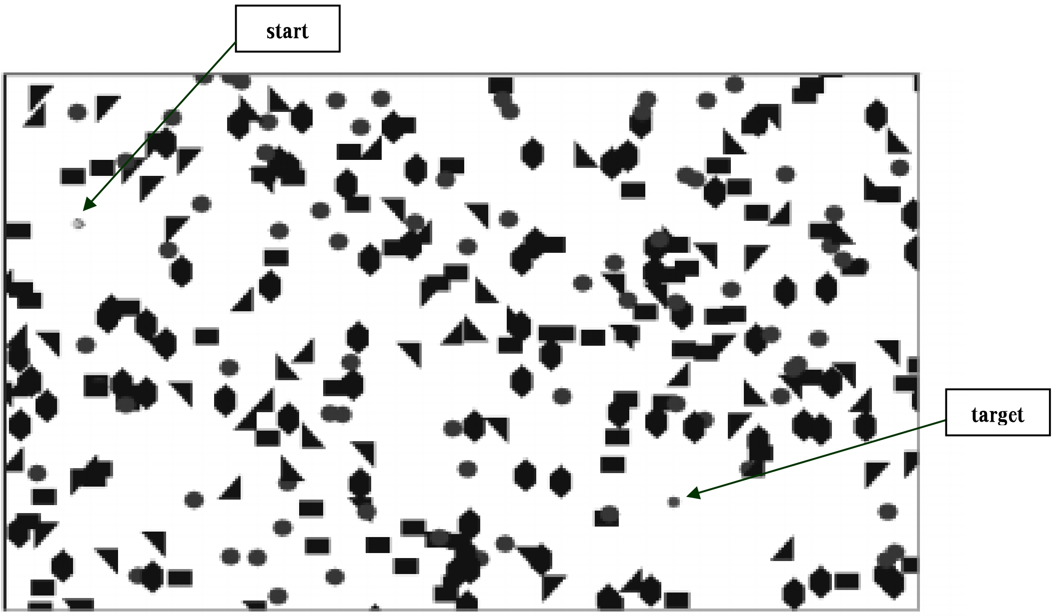

Figure 1 the environment of robot navigation problem

The experiments were carried out under a simulation, the details are as follows. The robot base is a circular shape with the radius 5 units. The robot can move forward, turn left and right. Its sensor can report whether the robot is getting nearer to the target. There are bumpers surrounding the base which can detect collision with obstacles. The step size of forward motion is 1 unit and the turning is 22.5 degrees. The robot environment is the grid of size 750 x 500 units. The obstacles have several shapes with the average size of 20 x 20 units and are placed randomly in the robot's world. The total area of the obstacles is 20% of the robot world.

2.1 Terminal set

The terminal set in this experiment is {forward, left, right, isnearer} where forward, left, right are the robot motion commands. isnearer is the sensing command which reports the improvement of the robot progress.

2.2 Function set

Our work focus on the function set. The basic function set composed of {if-and, if-or, if-not}. The functions that are varied in the experiment are {prog2, prog3, prog4, eio2}. The semantic of these functions are as follows: if-and has 4 parameters, p1..p4, if the return values from both p1 and p2 are true then performs p3 else performs p4, if-or is similar but p1, p2 is OR-ing, if-not has 3 parameters, p1..p3, if the return value from p1 is not true then perform p2 else perform p3, prog2 performs its two parameters in sequence, prog3 is similar but with 3 parameters, prog4 is similar but with 4 parameters and eio2 has 2 parameters, p1..p2, and perform either p1 or p2 based on random choice. The main hypothesis is that a program which can express a wide variety of behaviours according to different situations should be more robust. This eio2 (2-way Either Or function) is defined to test this hypothesis.

2.3 Fitness function

Fitness function is used to evaluate each of robot program. The fitness measure is based on the distance of the final position of robot to the target and the number of executed terminals. Smaller value is better for this experiment. The fitness function is

$$f = (10{,}000 \times fd) + t \tag{1}$$

where fd is Euclidean distance of the final position to the target point, fd = 0 if the robot reaches the target and t is the number of executed terminals.

2.4 Genetic Parameters

The parameters for GP were set as follows: population size 1500, maximum number of generation 20, the average initial size of an individual is 200, reproduction rate 10%, crossover rate 90%, maximum steps executing in the simulation before time-out is 6000. Two genetic operations we use are reproduction and crossover. All experiments are repeated 20 times to arrive at the reported statistics.

2.5 Robustness test phase

Robustness is measured by selecting the best individual from the maximum generation and evaluates it under new perturbed environments that are variant of the original. Perturbed environment was built up from original environment by selecting obstacles randomly and moving them in random direction of up, down, left and right by 6 units. The percent of disturbance is the number of obstacles that is moved by the total number of total obstacles.

Robustness value is calculated by evaluating the best robot program under these 1000 perturbed environments. The percent of disturbance is varied from 0-100%.

3. Result and discussion

Robustness of program composed of variety of function type is shown in fig.2. It is obvious that programs with eio2 function extension behave more robustly at every level of disturbance.

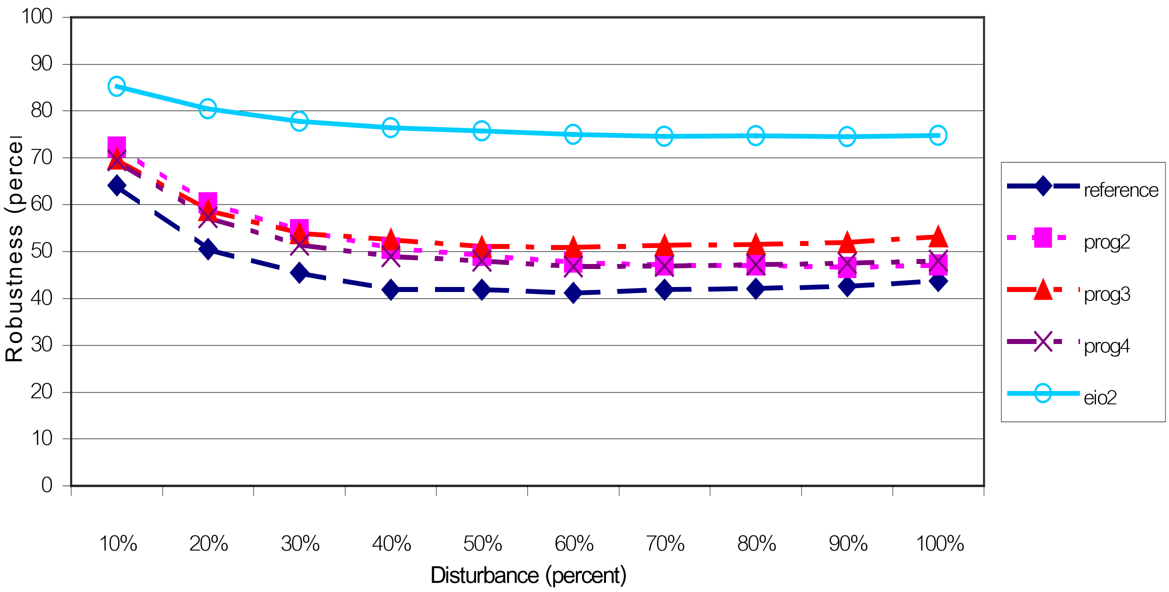

Figure 2 robustness of robot program with various function set

We analyse the result of robustness of program with a special function extension by counting the "path variety" of program while execute in simulation during robustness test phase. On the assumption that there is correspondence between the wide variety of behaviours and the path variety of a program. A path is a preorder traversal node string from root to leave of a program tree. Executing a program in simulation before achieving a goal or time-out occurs take several times in executing tree repeatedly and produces many paths. A path variety is the number of unique path in the set of all paths.

To find path variety of programs, we perform experiment varying disturbance from 10% to 50% because the robustness value don't change much after 50%. The result is shown in fig.3. A program with high robustness also has higher path variety value; therefore, our hypothesis is confirmed.

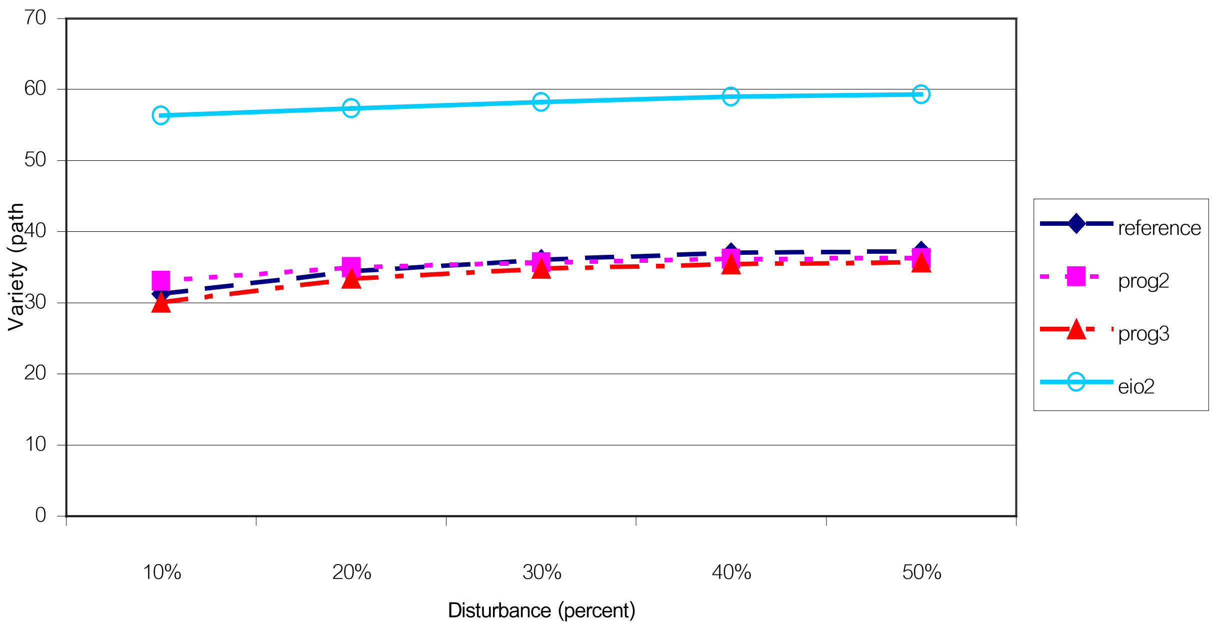

Figure 3 path variety of robot program with various function set

4. Conclusion

This paper explores an approach to improve robustness of robot programs using function set tuning on robot navigation problems. The result indicates that function set affects robustness of programs. The program with a special function behaves more robustly than other type of programs. The analysis shows that robustness is associated with the path variety. The future work will study effective of transferring robust solution from simulated world to real-world and adapt function set tuning to other problems.

References

[1] Chongstitvatana, P. and Polvichai, J., “Learning a visual by genetic programming”, Proc. of IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems, Osaka, Japan, 1996.

[2] Chongstitvatana, P., “Improving robustness of robot programs generated by genetic programming for dynamic environments”, Proc. of IEEE Asia-Pacific Conf. on Circuits and Systems (APCCAS98), Chiangmai, 1998, pp. 523-526.

[3] Ito, H. Iba and M. Kimura, “Robustness of robot programs generated by genetic programming”, Proc. of Conf. Genetic Programming 96, MIT Press, 1996, pp. 321-326.

[4] J. R. Koza., Genetic programming., MIT Press, 1992.

[5] Reynolds, C. W., “Evolution of Obstacle Behaviours: using noise to promote robust solutions”, Advances in Genetic Programming, MIT Press, 1994, pp. 221-224.