Multimodal Genetic Programming Using Program Similarity Measurement and Its Application to Wall-Following Problem

Shubu Yoshida
Ritsumeikan University
Kusatsu, Shiga, Japan
is0242rv@ed.ritsumei.ac.jp

Tomohiro Harada
Ritsumeikan University
Kusatsu, Shiga, Japan
harada@ci.ritsumei.ac.jp

Ruck Thawonmas
Ritsumeikan University
Kusatsu, Shiga, Japan
ruck@is.ritsumei.ac.jp

ABSTRACT

In this paper, we examine the effectiveness of multimodal genetic programming (MMGP) on the wall-following problem, which is a well-known benchmark problem of genetic programming (GP). MMGP aims to obtain multiple local optimal programs, including global optimal programs, that is, programs that achieve the same goal with different program structures. In this paper, we apply MMGP to the wall-following problem. The purpose of the wall-following problem is to find a program to control a robot having twelve distance sensors and four movements to follow irregular walls. We expect that there are several local optimal programs in the wall-following problem, which use different combinations of sensors. An experiment is conducted to investigate whether MMGP can get local optimal programs simultaneously for the wall-following problem. This experiment compares MMGP with a simple GP. The experimental results reveal that MMGP can achieve higher acquisition ratio of local optimal programs than the simple GP.

CCS CONCEPTS

• **Computing methodologies → Genetic programming**; *Evolutionary robotics*; • **Software and its engineering → Genetic programming**; • **Information systems** → Similarity measures;

KEYWORDS

Multimodal optimization, Genetic Programming, Wall-Following Problem, Program Similarity

ACM Reference Format:
Shubu Yoshida, Tomohiro Harada, and Ruck Thawonmas. 2019. Multimodal Genetic Programming Using Program Similarity Measurement and Its Application to Wall-Following Problem. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322063

1 INTRODUCTION

Our previous researches proposed multimodal genetic programming (MMGP) [2, 4], an extension of genetic programming (GP) to solve multimodal program optimization problems. MMGP divides the GP population into clusters using program structure similarity. MMGP enables to find global and local optimal programs with different genotypes by executing GP in consideration of clusters.

In this paper, we investigate the capability of MMGP using the wall-following problem [1], which is closer to the real-world problem than the problem in our previous researches. The aim of this problem is to find a program that controls the robot with twelve sensors and four movements to follow irregular walls. The wall-following problem can be expected to include several local optimal programs that can make the robot follow the wall without using some of the sensors used in the global optimal program. We conduct an experiment to compare MMGP with a simple GP and demonstrate the higher capability of MMGP in the wall-following problem.

2 MULTIMODAL GENETIC PROGRAMMING

To the authors' investigation, most conventional multimodal optimization methods target real-valued optimization, but there is no multimodal optimization method targeted for program optimizations. From this fact, our previous researches have proposed multimodal genetic programming (MMGP) [2, 4]. MMGP divides the GP population into clusters according to the similarity of program structure and optimizes each cluster for acquiring global and local optimal programs simultaneously.

Hierarchical clustering is used as a clustering algorithm of MMGP. As the indicator of the program similarity, MMGP employed tree similarity proposed by Yang et al. [3]. The crossover is performed to two parents, in particular, one parent is selected from the target cluster, while another parent is selected from the target cluster with a 50% probability, otherwise, another one is selected from the randomly selected cluster. This selection allows you to optimize individuals locally within each cluster while equally referring to all the clusters. After generating new solutions, the clustering is performed on the combined population, and half the solutions in each cluster are removed using negative tournament selection.

3 EXPERIMENT

In order to investigate the effectiveness of MMGP for the wall-following problem, this paper conducts an experiment to compare MMGP with a simple GP.

3.1 Experimental setup

The map of the wall-following problem used in this experiment is the same as in the work of Koza [1]. Table 1 shows the function and the terminal nodes used in this experiment. Unlike the work of Koza, this experiment does not use the constant variable SS, which will return the smallest value in all sensors. This is because the purpose

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3322063

Table 1: Function and terminal nodes used in the experiment

Function	IFLTE, PROGN2
Terminal	S0 – S11, EDG, MSD, FORWARD (F), BACKWARD (B), TURN_RIGHT (R), TURN_LEFT (L)

Table 2: Success ratio and cover ratio

	Simple GP	MMGP
Success ratio	34%	76%
Cover ratio (all)	9%	52%
Cover ratio (success)	27%	68%

of the multimodal search in this experiment is to find a program that uses different sensor values, but SS implicitly uses all sensor values. When the movement nodes (F, B, R, and L) are executed, they returns the smaller value of S2 and S3 sensors, which are the front sensors of the robot.

Fifty independent trials are performed for each method. The number of generations is 200, while the population size is 1000. The maximum number of steps is 400, and the maximum tree depth is 17. The fitness function uses the following fomula:

$$fitness = \begin{cases} \#tile - \#visited & \text{if } \#tile > \#visited \\ -Step & \text{otherwise,} \end{cases} \quad (1)$$

where $\#tile$ is the total number of tiles (56 in this experiment), while $\#visited$ is the number of visited tiles in 400 steps. $Step$ is the number of steps remaining after visiting all tiles in the map.

3.2 Evaluation criteria

We confirm the following three criteria: (1) the success ratio, (2) the cover ratio of all 50 trials, and (3) the cover ratio of the success trials. We define a success trial as a trial to acquire a program that can visit all tiles. The success ratio is defined as the percentage of the success trials in all 50 trials. On the other hand, the cover ratio is defined as the ratio at which a program can visit all tiles without using a specific sensor used by the global optimal program. We consider the cover ratio in all 50 trials and the success trials.

3.3 Result

Table 2 shows the success ratio, the cover ratio for all trials, and the cover ratio only for the successful trials. The "Simple GP" column indicates the result of the simple GP, and the "MMGP" column indicates the result of MMGP.

From the result of the success ratio, the simple GP succeeds in finding a program that can visit all tiles in 34% trials, while MMGP achieves 76%. This result shows that in the wall-following problem, the search capability of MMGP to find a global optimal program is improved compared to the simple GP. Second, the results of the cover ratio show that the simple GP only accomplishes 9% of the cover ratio in all trials. Even with a focus on the cover ratio in the success trials, the simple GP can only achieve 27% of the cover ratio. In contrast, MMGP achieves the higher cover ratio than the simple GP, in particular, 52% of the cover ratio in all trials and 68% of in the success trials.

Examples of the global and the local optimal programs obtained by MMGP are shown in Figs. 1 and 2. You can see that the structure

```
(IFLTE F EDG (PROGN2 (IFLTE (IFLTE R R S3 (PROGN2
B (PROGN2 B R))) R S5 (PROGN2 S8 S9)) (PROGN2 MSD
S6)) (IFLTE S0 EDG S0 (IFLTE (IFLTE S6 S1 S3 S5)
(PROGN2 S8 S11) S0 L)))
```

Figure 1: The global optimal program ($fitness = -239$)

```
(IFLTE F EDG (PROGN2 (IFLTE (IFLTE R R S3 (PROGN2
B R)) R S5 (IFLTE F EDG (PROGN2 B (IFLTE R B
(PROGN2 B EDG) S8)) S0)) (PROGN2 S8 S11)) (IFLTE
S0 EDG S0 (IFLTE S4 (PROGN2 S8 S11) S0, L)))
```

Figure 2: The local optimal program ($fitness = -222$)

of the global optimal solution and the local optimal solution is similar. The difference between the two is that the global optimal program of Fig. 1 uses the sensor values of S1, S6, and S9, while these sensor values are not used in the local optimal program. Instead of these sensors, the local optimal program uses the sensor value of S4 which is not used in the global optimal program.

These results indicate that MMGP has higher search capability of multimodal search than the simple GP, and MMGP can obtain the global and the local optimal programs with different genotypes.

4 CONCLUSION

This paper demonstrated the performance of MMGP in the wall-following problem by comparing the simple GP and MMGP. The experimental results showed that MMGP can acquire global and local optimal programs simultaneously with higher reliability than the simple GP. We also confirmed that although the global and the local optimal programs obtained by MMGP are structurally similar, they use different combinations of the sensors.

In this paper, we showed that MMGP can search the global optimal program and multiple local optimal programs that complement the global one. In the future, we plan to explore effective selection methods that take into consideration cluster similarity and dynamic cluster management using niching methods to improve the search accuracy of global and local optimal programs.

ACKNOWLEDGMENTS

This work was supported by Grant-in-Aid for Scientific Research B Grant Number JP16KT0103.

REFERENCES

[1] John R. Koza. 1994. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing* 4, 2 (01 Jun 1994), 87–112. https://doi.org/10.1007/BF00175355
[2] Kei Murano, Shubu Yoshida, Tomohiro Harada, and Ruck Thawonmas. 2018. A Study on Multimodal Genetic Programming Introducing Program Simplification. In *Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems (SCIS&ISIS 2018)*. 109–114.
[3] Rui Yang, Panos Kalnis, and Anthony K. H. Tung. 2005. Similarity Evaluation on Tree-structured Data. (2005), 754–765. https://doi.org/10.1145/1066157.1066243
[4] Shubu Yoshida, Tomohiro Harada, and Ruck Thawonmas. 2017. Multimodal genetic programming by using tree structure similarity clustering. In *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*. 85–90. https://doi.org/10.1109/IWCIA.2017.8203566