Proposal of Multimodal Program Optimization Benchmark and Its Application to Multimodal Genetic Programming

Tomohiro Harada
Faculty of System Design
Tokyo Metropolitan University
Tokyo, Japan
harada@tmu.ac.jp

Kei Murano
Graduate School of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
is0284fv@ed.ritsumei.ac.jp

Ruck Thawonmas
College of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
ruck@is.ritsumei.ac.jp

***Abstract*—Multimodal program optimizations (MMPOs) have been studied in recent years. MMPOs aims at obtaining multiple optimal programs with different structures simultaneously. This paper proposes novel MMPO benchmark problems to evaluate the performance of the multimodal program search algorithms. In particular, we propose five MMPOs, which have different characteristics, the similarity between optimal programs, the complexity of optimal programs, and the number of local optimal programs. We apply multimodal genetic programming (MMGP) proposed in our previous work to the proposed MMPOs to verify their difficulty and effectiveness, and evaluate the performance of MMGP. The experimental results reveal that the proposed MMPOs are difficult and complex to obtain the global and local optimal programs simultaneously as compared to the conventional benchmark. In addition, the experimental results clarify mechanisms to improve the performance of MMGP.**

***Index Terms*—Multimodal program optimization, genetic programming, benchmark, multimodal search, symbolic regression**

I. Introduction

In recent years, multimodal optimization that aims at obtaining a global optimal solution and a plurality of local optimal solutions simultaneously using evolutionary algorithms (EAs) has been studied. Here, a local optimal solution is a solution that has a lower fitness than a global optimal one but is optimal in a local region [1]. We have introduced the concept of multimodal optimization into genetic programming (GP) [2], which targets program optimization, and proposed multimodal genetic programming (MMGP) [3] that obtains global and local optimal programs simultaneously. Since programs are represented by a tree structure in GP, MMGP divides a population into multiple clusters by clustering using the similarity of the tree structure and performs optimization considering the clusters to enable the multimodal search.

The previous study on MMGP [4] has been tested on the multimodal program optimization (MMPO) benchmark proposed in the previous work [3]. However, in the conventional benchmark, the global and local optimization programs are both simple and the number of the local optimal program is only one. This makes it relatively easy to find multiple optima. Therefore, the evaluation of the performance of MMGP was not enough, and it is unclear whether it can be applied to problems with more complex global and local optimal programs and/or problems with multiple local optimal programs.

In order to overcome this problem, this research proposes new MMPO benchmark problems for enabling algorithm evaluation of multimodal genetic programming. In particular, we propose new benchmarks that have different characteristics of the complexity of programs, the similarity between the global and local optimal programs, and the number of local optima. In this paper, MMPOs are designed by connecting multiple formulas used in the symbolic regression problem, referring to the GP benchmark problems provided in the work of McDermott et al. [5].

We apply the simple GP, which does not consider the multimodality, and MMGP to the proposed MMPO benchmarks to investigate the difficulty and the effectiveness of the proposed benchmarks. In addition, we evaluate the performance of MMGP and reveal the direction to improve its performance.

II. Background

A. Multimodal Optimization

The multimodal optimization problem [1] is a problem that aims at obtaining not only a global optimal solutions but also all local optimal solutions. A local optimal solution is an optimal solution in the local region. Fig. 1 shows an example of a multimodal optimization problem. In Fig. 1, the x-axis indicates the search space, while the y-axis indicates the fitness or the objective function. The higher the fitness, the better the solution. In this figure, the solution indicated by the red circle is the global optimal solution because it has the best fitness value in the entire search space. On the other hand, the solutions indicated by the blue circle have lower fitness than the global optimal solution, but is defined as the local optimal solutions because they are the best solution in a certain local

This work was supported by Japan Society for the Promotion of Science Grant-in-Aid for Scientific Research B Grant Number JP16KT0103.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

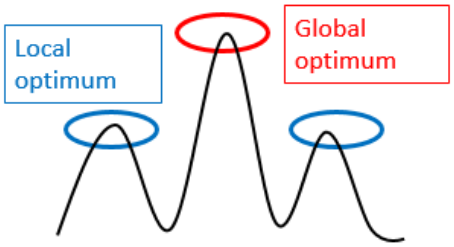


Fig. 1: An example of a multimodal optimization. The x-axis indicates the search space, while the y-axis indicates the fitness (maximization). The solution indicated by the red circle is the global optimal solution, while the solutions indicated by the blue circles are the local optimal solutions.

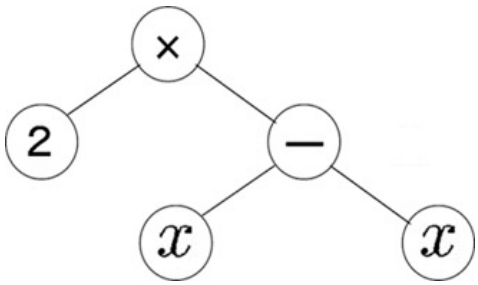


Fig. 2: An example of a tree representation of the mathematical expression $(2 \times (x - x))$ in GP

region. The final goal of multimodal optimization is to obtain all solutions indicated by the red and the blue circles in this example simultaneously in a single run.

B. Genetic Programming

GP is an extension of Genetic Algorithm (GA) [6] proposed by Koza and handles mathematical expressions and programs as optimization targets. Unlike genotype expression in GA is mainly an array, GP typically uses a tree structure. This makes it possible to represent structured data such as mathematical formulas and program codes that are hard to be represented by an array structure.

Fig. 2 shows an example of a tree structure expressing the mathematical expression $(2 \times (x - x))$. This tree structure consists of operators ("$\times$" and "$-$" in the figure), variables ("x" in the figure), and constants ("2" in the figure).

C. Multimodal program optimization

In our previous works, we invented a multimodal program optimization (MMPO) that applies a concept of multimodal optimization to program optimization. The purpose of MMPO is to obtain a global optimal program and local optimal programs with different genotypes (i.e., program structures) for one objective. For example, in a symbolic regression problem that identifies a function from a given input-output relationship, multiple functions (programs) using different operators and combinations of variables are obtained.

In our previous works, we proposed a benchmark problem for MMPO. The benchmark problem is defined as a symbolic regression problem based on the input and output values given by the following equations, taking the 4 variables x, y, z, w as inputs:

$$f(x, y, z, w) = x^2 + y^2 \quad (1)$$

$$z = (x + y + \alpha)^2 + \delta \quad (2)$$

$$w = \frac{\alpha^2}{2} + xy + x\alpha + y\alpha + \delta, \quad (3)$$

where α is a constant, while δ is the random noise given to the variables z and w. The domains of x and y are both defined as $[-1.0, 1.0]$. If δ is 0, this expression is reconstructed as:

$$\begin{aligned} f(x, y, z, w) &= x^2 + y^2 \\ &= z - 2w. \end{aligned} \quad (4)$$

From the formula in (1), when certain values of x and y are given, input/output values can be expressed as (4) using variables z and w. If δ is not 0 and the random noise is added to z and w, the value calculated by (4) has an error compared to the correct calculation result calculated by (1). For this, a program given by the formula of (4) can be regarded as a local optimal program in this problem.

D. Multimodal Genetic Programming

We proposed multimodal genetic programming (MMGP) [3] as a method for solving MMPOs. MMGP focuses on the similarity of the tree structure, which is an expression of programs, and realizes multimodal search by dividing the population into multiple clusters based on the tree structure similarity.

A pseudo-code of MMGP is shown in Algorithm 1. In the algorithm, P_g represents the population at the gth generation, while C_g represents a set of clusters constructed from P_g. C_g^i represents the ith cluster in C_g, which consists of one or more programs (solutions). P_c indicates the crossover probability.

In MMGP, new programs are generated through the crossover and the mutation operators in line 5–18. The crossover operator is applied to two parent programs, one is selected from the target cluster, while another is selected from randomly selected cluster. By this, all clusters are evenly referred, and the search is executed locally in the cluster. The mutation operator is also executed by considering the cluster. After generating new programs, the old and the new populations are merged, and the clustering based on the tree structure similarity is performed. Then, programs with low fitness values are eliminated from each cluster until the size of each cluster is reduced by half. These procedures are repeated until the number of generations reaches the maximum.

Algorithm 2 shows the pseudocode of the clustering in MMGP. In Algorithm 2, p^i indicates the ith program in the population P, while C^i indicates the ith cluster in the set of cluster C. MMGP uses the agglomerative hierarchical clustering. The similarity between two clusters is determined by the smallest similarity between programs in these clusters.

Algorithm 1 A pseudo-code of MMGP

1: $P_0 \leftarrow \texttt{initialize}()$
2: $C_0 \leftarrow \texttt{clustering}(P_0)$
3: **for** $g = 0$ *to* G **do**
4: $\quad P'_g \leftarrow \emptyset$
5: $\quad$ **for** $i = 1$ *to* $|C_g|$ **do**
6: $\quad\quad$ **for** $j = 1$ *to* $|P_g|/|C_g|$ **do**
7: $\quad\quad\quad$ **if** $\texttt{rand}(0,1) < P_c$ **then**
8: $\quad\quad\quad\quad p_1 \leftarrow \texttt{tournament}(C_g^i)$
9: $\quad\quad\quad\quad r \leftarrow \texttt{randInt}(0, |C_g|)$
10: $\quad\quad\quad\quad p_2 \leftarrow \texttt{tournament}(C_g^r)$
11: $\quad\quad\quad\quad p' \leftarrow \texttt{crossover}(p_1, p_2)$
12: $\quad\quad\quad$ **else**
13: $\quad\quad\quad\quad p \leftarrow \texttt{tournament}(C_g^i)$
14: $\quad\quad\quad\quad p' \leftarrow \texttt{mutation}(p)$
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad P'_i \leftarrow P'_g \cup \{p'\}$
17: $\quad\quad$ **end for**
18: $\quad$ **end for**
19: $\quad P_{g+1} \leftarrow P_g \cup P'_g$
20: $\quad C_{g+1} \leftarrow \texttt{clustering}(P_{g+1})$
21: $\quad$ **for** $i = 1$ *to* $|C_{g+1}|$ **do**
22: $\quad\quad L_i \leftarrow \texttt{max}(1, |C_g^i|/2)$
23: $\quad$ **end for**
24: $\quad$ **while** $|P_{g+1}| > |P_g|$ **do**
25: $\quad\quad$ **for** $i = 1$ *to* $|C_{g+1}|$ **do**
26: $\quad\quad\quad$ **if** $|P_{g+1}| = |P_g| \vee |C_{g+1}^i| \leq L_i$ **then**
27: $\quad\quad\quad\quad$ *break*
28: $\quad\quad\quad$ **end if**
29: $\quad\quad\quad p \leftarrow \texttt{negative\_tournament}(C_{g+1}^i)$
30: $\quad\quad\quad C_{g+1}^i \leftarrow C_{g+1}^i \backslash \{p\}$
31: $\quad\quad$ **end for**
32: $\quad$ **end while**
33: **end for**

Algorithm 2 Clustering procedure

1: $C \leftarrow \{\{P^1\}, \{P^2\}, \ldots, \{P^N\}\}$
2: **for** $i = 1$ to $|C|$ **do**
3: $\quad$ **for** $j = i + 1$ to $|C|$ **do**
4: $\quad\quad D_{i,j} \leftarrow \texttt{similarity}(P^i, P^j)$
5: $\quad$ **end for**
6: **end for**
7: **while** $|C| > 1$ **do**
8: $\quad max_i \leftarrow 1, max_j \leftarrow 2$
9: $\quad$ **for** $i = 1$ to $|C|$ **do**
10: $\quad\quad$ **for** $j = i + 1$ to $|C|$ **do**
11: $\quad\quad\quad$ **if** $D_{max_i,max_j} < D_{i,j}$ **then**
12: $\quad\quad\quad\quad max_i \leftarrow i$
13: $\quad\quad\quad\quad max_j \leftarrow j$
14: $\quad\quad\quad$ **end if**
15: $\quad\quad$ **end for**
16: $\quad$ **end for**
17: $\quad$ **if** $D_{max_i,max_j} \leq d$ **then**
18: $\quad\quad break$
19: $\quad$ **end if**
20: $\quad C^{max_i} \leftarrow C^{max_i} \cup C^{max_j}$
21: $\quad$ **for** $i = 1$ to $|C|$ **do**
22: $\quad\quad D_{i,max_i} \leftarrow \min(D_{i,max_i}, D_{i,max_j})$
23: $\quad$ **end for**
24: $\quad C \leftarrow C \backslash \{C^{max_j}\}$
25: **end while**
26: **return** C

First, all programs in the population are set in one cluster from line 1. Then, from lines 2 to 6, the similarities between all clusters (programs) are calculated, where $D_{i,j}$ indicates the similarity between the ith and jth clusters (programs). Next, this algorithm finds the clusters with the highest similarity in lines 8 to 16, and they are merged in line 20. After that, the similarities between the cluster generated by the merge and others are updated according to the smallest similarity of programs in two clusters in lines 21 to 23. And finally, the merged cluster is removed from C in line 24. This procedure is repeated until the maximum similarity of clusters reaches the user-defined threshold d, and the clustered population is returned. As the tree structure similarity, the previous research employed one proposed by Yang et al. [7] (see [7] or [3] for detail).

III. Proposed benchmark problems

In the previous MMPO problem shown in (4), the global optimal program and the local optimal program are similar and simple. Furthermore, since it includes only one local optimal program, it is relatively easy to obtain the global and local optimal program simultaneously. Therefore, it cannot be said that the performance of MMGP has been sufficiently evaluated only by the previous MMPO problem.

In this paper, we propose new five MMPO problems with complex structure of global and local optimal programs and with multiple local optimal programs. Specifically, the proposed MMPO problems combine multiple symbolic regression problems with reference to the work of McDermott et al. [5], which summarizes the GP benchmark problems used in literature. This enables more detailed analyses of multimodal program optimization methods.

A. Problem description

The five proposed MMPO benchmark problems are detailed as follows:

MMPO1: The first benchmark, MMPO1, connects the following three formulas:

$$f(x) = \log(x+1) + \log(x^2+1) \quad (5)$$

$$g(y,z) = 2 - 2.1\cos(9.8y)\sin(1.3z) \quad (6)$$

$$h(w,v) = 2\sin(w)\cos(v). \quad (7)$$

To let these three have the same calculation result, the variables

y, z, w and v are decided from x as:

$$y = \frac{f(x)+1}{9.8\alpha} + \delta \quad (8)$$

$$z = \frac{1}{1.3}\sin^{-1}\left(\frac{2-f(x)}{2.1\cos\left(\frac{f(x)+1}{\alpha}\right)}\right) + \delta \quad (9)$$

$$w = \sin\frac{\log(x+1)}{\log(x^2+1)+1} + \delta \quad (10)$$

$$v = \cos^{-1}\left(\frac{f(x)}{2\sin w}\right) + \delta. \quad (11)$$

Similar to the previous MMPO problem, δ is the random noise given to each variable, while α is constant. The domain of x is defined as $[0.01, 0.5]$. If $\delta = 0$, the above formulas have the same calculation result.

MMPO2: The second benchmark, MMPO2, connects the following two formulas:

$$f(x) = x^3 + x^2 + x \quad (12)$$

$$g(y, z) = \sin(y^2) + \sin(z). \quad (13)$$

Same as MMPO1, the variables y and z are decided from x as following to let these two have the same calculation result:

$$y = \frac{x^6 + x^4 + x}{\alpha} + \delta \quad (14)$$

$$z = \sin^{-1}\left(f(x) - \sin(y^2)\right) + \delta. \quad (15)$$

The domain of x is defined as $[-0.5, 0.5]$.

MMPO3: The third benchmark, MMPO3, connects the following two formulas:

$$f(x) = \sin(x^2)\cos(x) - 1 \quad (16)$$

$$g(y, z) = 6\sin(y)\cos(z). \quad (17)$$

The variables y and z are decided from x as following to let these two have the same calculation result:

$$y = 2x^3 + 1 + \delta \quad (18)$$

$$z = \cos^{-1}\left(\frac{f(x)}{6\sin(y)}\right) + \delta. \quad (19)$$

The domain of x is defined as $[-0.5, 1.0]$.

MMPO4: The fourth benchmark, MMPO4, connects the following two formulas:

$$f(x) = \sin(x) + \sin(x + x^2) \quad (20)$$

$$g(y, z) = 2\sin(y)\cos(z). \quad (21)$$

The variables y and z are decided from x as following to let these two have the same calculation result:

$$y = \cos(f(x)) + \delta \quad (22)$$

$$z = \cos^{-1}\left(\frac{f(x)}{2\sin(y)}\right) + \delta. \quad (23)$$

The domain of x is defined as $[-0.5, 0.5]$.

TABLE I: Characteristics of the proposed MMPOs

	Similarity	Complexity	# locals
MMPO0	High	Low	1
MMPO1	Low	High	2
MMPO2	Low	Low	1
MMPO3	High	Low	1
MMPO4	Low	High	1
MMPO5	Low	High	2

MMPO5: The fifth benchmark, MMPO5, connects the following three formulas:

$$f(x) = x^4 + x^3 + x^2 + x \quad (24)$$

$$g(y, z) = \sin(y) + \sin(z^2) \quad (25)$$

$$h(v, w) = v^4 - v^3 + \frac{w^2}{2} - w. \quad (26)$$

To let these three have the same calculation result, the variables y, z, w and v are decided from x as:

$$y = \sin^{-1}(3x^3 + x^2) + \delta \quad (27)$$

$$z = \sqrt{\sin^{-1}(x^4 - 2x^3 + x)} + \delta \quad (28)$$

$$v = \frac{1}{1+x} + \delta, \quad (29)$$

$$w = 1 - \sqrt{1 - 2\left(-\frac{x}{(1+x)^4} - f(x)\right)} + \delta. \quad (30)$$

$$(31)$$

The domain of x is defined as $[0.0, 0.5]$.

B. Characteristics of the proposed benchmarks

The above five MMPOs have more complex structures than the previous benchmark and MMPO1 and MMPO5 have multiple local optimal programs. This makes it difficult to obtain global and local optimal programs simultaneously. Table I summarizes the characteristics of each benchmark (the similarity of the global and the local optimal programs, program complexity, and the number of local solutions). The benchmark proposed in the previous research is represented as MMPO0 hereafter.

The previous benchmark, MMPO0, has only one local optimal program, and the global and local optimal programs are similar and simple, both of them are constructed with a few arithmetic operations. Three of the proposed benchmarks, MMPO2, MMPO3, and MMPO4, also have one local optimal program. On the other hand, MMPO1 and MMPO5 include two local optima. The characteristic of MMPO3 is similar to MMPO0, because its similarity is high and the complexity is low. In particular, The global and local optimal programs have the same structure, for example $\sin(\cdot) \times \cos(\cdot)$ in MMPO3.

IV. Experiment

To investigate the difficulty of the proposed MMPOs and investigate the search capability of MMGP proposed in the previous work, this paper conducts an experiment to solve MMPOs with MMGP and a simple GP.

A. Setting

In this experiment, we use MMPO0 and the five MMPOs proposed in this research, for a total of six benchmarks are used. Considering the complexity of the optimal programs, the maximum depth of the program tree is 4 for MMPO0, 10 for MMPO1, MMPO3, and MMPO5, and 6 for MMPO2, and MMPO4.

In MMGP, since the result differs depending on the threshold d at which clustering is stopped in MMGP, the threshold $d = \{0.5, 0.6, 0.7\}$ are used. A simple GP is regarded as a special case of MMGP with $d = 1.0$, i.e., the clustering is executed until all clusters are merged. The experiment is executed 20 independent runs for each method and each problem. The number of generations is 500 and the population size is 500. The crossover probability is 0.9, and the mutation probability is 0.1. The variables x, y, z, w are used for MMPO0, x, y, z, w, v for MMPO1 and MMPO5, x, y, z for MMPO2 and MMPO3 and MMPO4. Eight function nodes are used: $+, -, \times, \%$[1], $\sin, \cos, \tan, \log$ [2]. The random real constant value $c \in [-1, 1]$ as the terminal node other than the variable. The fitness function is defined as:

$$fitness = \sum_{i=1}^{D} |result_i - target_i| , \tag{32}$$

where D represents the number of the training data. $result_i$ represents the execution result of a program for the ith input value, while $target_i$ represents the output value respective to ith input value. The number of data D is set to $121, 50, 100, 150, 100, 50$ from MMPO0 to MMPO5, respectively, and distributed uniformly in the domain defined for each MMPO.

B. Evaluation criteria

In this experiment, the best individuals with the use of limited variables are identified in order to evaluate whether the global and local optimal programs with different genotypes can be obtained simultaneously. Concretely, among the programs obtained in each trial, the best program that uses only the variables included in the assumed local optimal program is considered. For example, when the local optimal program shown in (6) of MMPO1 has been acquired, even if the variables x, w, v are not used, the program using the variables y, z can achieve a low (excellent) fitness. The percentage of which the optimal program found for 20 trials and its transition are confirmed. This confirms whether the global and local optimal programs have been acquired simultaneously in one trial. We define that the global optimal program is found if the fitness is less than 0.02, while not found if it is more than 0.02. Moreover, we define that the local optimal programs are found if the fitness is less than the value calculated by (6) and (7) in MMPO1, (13) in MMPO2, (17) in MMPO3, (21) in MMPO4, and (25) and (26) in MMPO5 when using certain variables, respectively.

[1]Protected division.

[2]Since our previous research [3] does not use $\sin, \cos, \tan, \log$, the experimental results are different.

V. Result

A. Acquisition percentage of the global and local optimal programs

Table II shows the percentage of which the global and local optimal programs are found in the final population in 20 trials. Each column represents the global and local optimal programs. The "all" column represents the result when all variables are used, while the other columns represent the result when only some variables are used, for example, the "x" column indicates the result when the only variable x is used. Each row shows the threshold d of MMGP and the simple GP.

From the results of Table II, MMGP obtains the global and local optimal programs in most of trials in MMPO0, which is the benchmark proposed in the previous work. On the other hand, the simple GP obtains only the local optimal program in 15% trials, while does not find the global optimal program only using the variables x and y. From this, it is revealed that the benchmark proposed in the previous work is easily solved by MMGP.

In MMPO1 and 2, MMGP is more likely to find the global and local optimal programs than the simple GP. However, in MMPO1, all global and local optimal programs cannot be simultaneously found even with MMGP. For MMPO3, both MMGP and the simple GP hardly found the global and local optimal programs. In MMPO4, although the MMGP could find the local optimum program in all trials, the global optimal program could not be found. The simple GP does not also find the global optimal program. In MMPO5, MMGP does not only find the local optimal program, but not the global optimal program. On the other hand, the simple GP finds the global optimal program with a high percentage, but it does not find the local optimal program.

From this result, it is revealed that it was difficult to obtain the global and local optimal programs simultaneously for both the simple GP and MMGP when using the proposed benchmark. This indicates that the proposed MMPOs are more complex and difficult than the previous benchmark.

B. Transition of the acquisition percentage

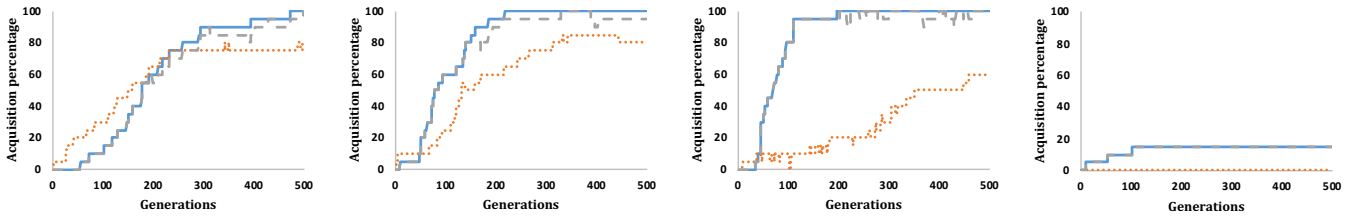
Fig. 3 to 8 show the generation transition of the percentage of the global and local optimal program acquisition for each method. The vertical axis shows the percentage of acquiring the global/local optimal program in 20 trials, while the horizontal axis shows the number of generations. In each figure, the type of line represents the acquisition percentage in consideration of the program using only certain variables corresponding to the global or local optimal program, as in Table II.

From the results of Fig. 3, it can be indicated that both the simple GP and MMGP do not find the local optimal program other than the final population. On the other hand, MMGP can acquire the global optimal program with a high percentage from the beginning of the search, while the simple GP stagnates its search and cannot acquire the global optimal program. From Fig. 4, in MMPO1, all methods can find the

TABLE II: The acquisition percentage of the global and the local optimal programs

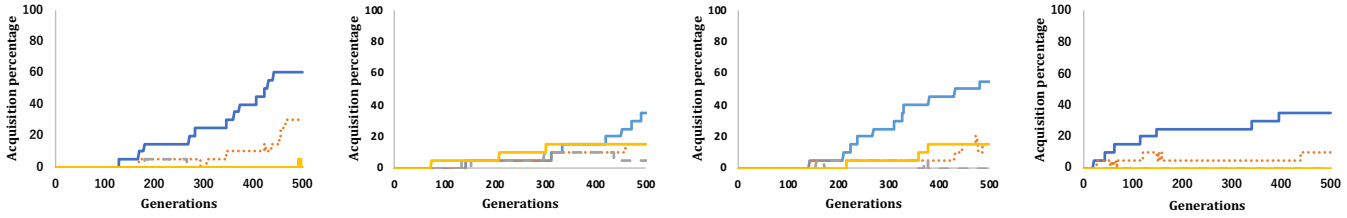
	MMPO0 (Depth=4)			MMPO1 (Depth=10)				MMPO2 (Depth=6)		
Method	all	x, y	z, w	all	x	y, z	w, v	all	x	y, z
MMGP $d = 0.5$	100%	80%	95%	60%	30%	0%	0%	75%	60%	100%
MMGP $d = 0.6$	100%	80%	95%	35%	15%	5%	15%	35%	25%	100%
MMGP $d = 0.7$	100%	60%	100%	55%	15%	0%	15%	70%	60%	100%
Simple GP	15%	0%	15%	35%	10%	0%	0%	50%	45%	10%

	MMPO3 (Depth=10)			MMPO4 (Depth=6)			MMPO5 (Depth=10)			
Method	all	x	y, z	all	x	y, z	all	x	y, z	w, v
MMGP $d = 0.5$	0%	0%	0%	0%	0%	100%	20%	15%	0%	0%
MMGP $d = 0.6$	0%	0%	0%	0%	0%	85%	0%	0%	10%	0%
MMGP $d = 0.7$	5%	5%	5%	15%	0%	75%	10%	0%	0%	0%
Simple GP	0%	0%	5%	0%	0%	30%	85%	80%	0%	0%



(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 3: Transition of the percentage of the global/local optimal program acquisition (MMPO0, $d = 6$), blue: use of all variables, orange: use of only x and y, gray: use of only z and w



(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 4: Transition of the percentage of the global/local optimal program acquisition (MMPO1, $d = 10$), blue: use of all variables, orange: use of only x, gray: use of only y and z, yellow: use of only w and v

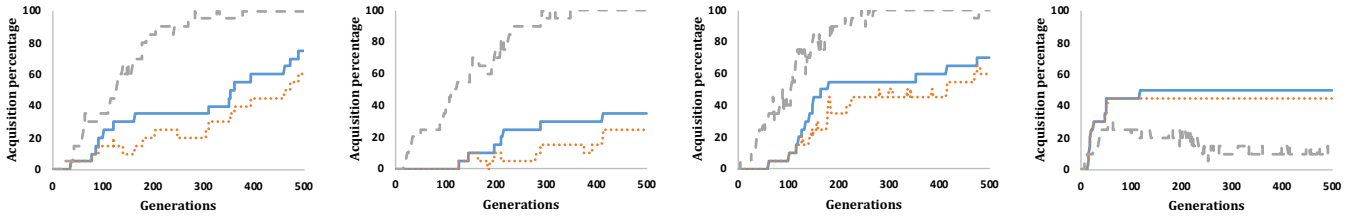
optimal solution when all variables are used from the early stage of the evolution, but it is indicated that the global optimal program consisting only of the variable x has not been obtained. Moreover, both the simple GP and MMGP hardly obtain the local optimal program during their search process. From Fig. 5, in MMPO2, the local optimal program using the variables y and z could be obtained early, and then global optimal programs begin to be obtained. From Fig. 6, in MMPO3, both the simple GP and MMGP cannot acquire both the global and local optimal programs at all. From Fig. 7, in MMPO4, the local optimal program using the variables y and z is obtained in the early stage of the evolution, but no global optimal program was found except for MMGP (d = 0.7). Regarding MMPO5, from Fig. 8 the simple GP could obtain the global optimal program in a high percentage, whereas MMGPs could not. In addition, the local optimal program has not been obtained for both the simple GP and MMGP.

VI. Discussion

The experimental results show that it is difficult to obtain the global and local optimal program simultaneously in the proposed MMPOs with the simple GP and MMGP. In this section, we consider the improvements required for MMGP to solve these MMPOs.

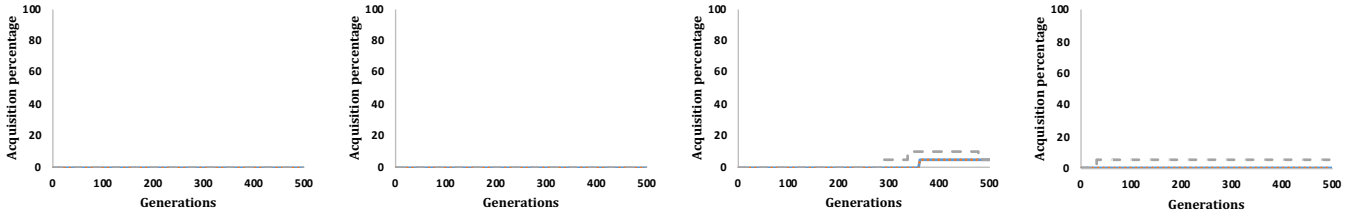
In MMPO1, the global optimal program using all variables was obtained, but the global optimal program using only the variables x, which is the only variable required in (5), was not obtained. This is caused by the inclusion of meaningless substructures (introns) that include the other unnecessary variables y, z, w, and v. From this, it can be indicated that a mechanism that realizes more appropriate clustering by removing unnecessary substructures after finding better programs is necessary.

In MMPO2 and MMPO4, the local optimal program can be acquired with a simple structure (a small number of nodes) compared to the global optimal one. This makes it easier to



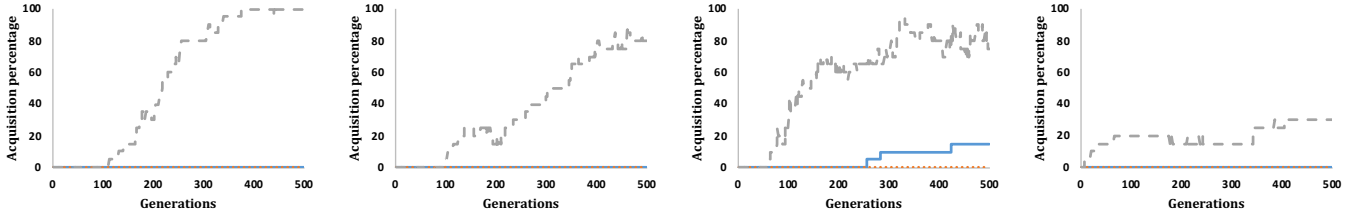
(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 5: Transition of the percentage of the global/local optimal program acquisition (MMPO2, $d = 6$), blue: use of all variables, orange: use of only x, gray: use of only y and z



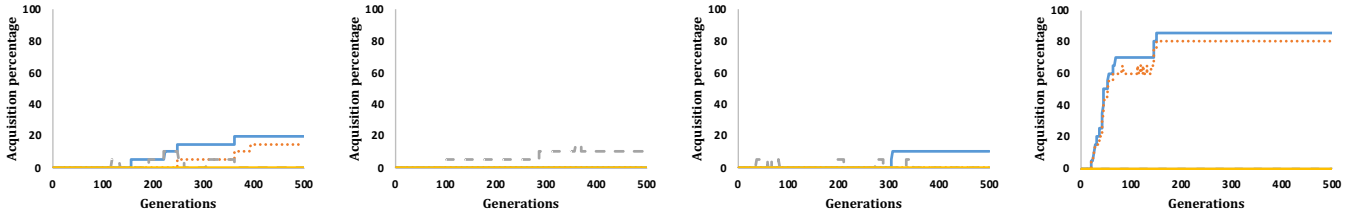
(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 6: Transition of the percentage of the global/local optimal program acquisition (MMPO3, $d = 10$), blue: use of all variables, orange: use of only x, gray: use of only y and z



(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 7: Transition of the percentage of the global/local optimal program acquisition (MMPO4, $d = 6$), blue: use of all variables, orange: use of only x, gray: use of only y and z



(a) MMGP with $d = 0.5$ (b) MMGP with $d = 0.6$ (c) MMGP with $d = 0.7$ (d) Simple GP

Fig. 8: Transition of the percentage of the global/local optimal program acquisition (MMPO5, $d = 10$), blue: use of all variables, orange: use of only x, gray: use of only y and z, yellow: use of only w and v

obtain the local optimal program than the global one. On the other hand, the global optimal program has low similarity with the local optimal program. Therefore, once the evolution converges to the local optimal program, it becomes difficult to search for the global optimal program thereafter. Therefore, in these MMPOs, the acquisition ratio of the local optimal program is high, and the acquisition ratio of the global optimal program is low.

In MMPO3, it was confirmed that the GP population tends to be filled with a program returning a negative constant (for example, $sin(c)$). Because such a simple structure can be easily generated, it is necessary to have a mechanism to maintain diversity so that it does not converge to a simple structure of a program.

In the case of MMPO5, the percentage to acquire the global optimal program in the simple GP is high, while it is low in MMGP. The global optimal program of MMPO5 is a polynomial consisting of "$+$" and "$\times$". When expressing this in a tree structure, various structures are possible depending on the branch point. However, according to the tree structure similarity used in the conventional MMGP, the similarity of the same expression is calculated low if these programs have different structures. Therefore, appropriate clustering becomes difficult, and the search performance is degraded.

From these facts, to solve the proposed benchmark, the following mechanisms should be introduced to MMGP:

- A simplification mechanism that removes unnecessary structures.
- A search mechanism that maintains the population diversity.
- A mechanism to search for a new structure when the search converges.
- Similarity considering not only the structure but also the semantics of the program.

VII. Conclusion

This paper proposed five new MMPO benchmarks with reference to the work of McDermott et al. [5]. The proposed benchmarks have different characteristics, the similarity between optimal programs, the complexity of optimal programs, and the number of local optimal programs. We applied MMGP and the simple GP to the proposed MMPOs to verify their difficulty and effectiveness, and evaluated the performance and problems of MMGP.

The experimental results showed that the proposed MMPOs are difficult and complex to obtain the global and local optimal programs simultaneously as compared to the conventional benchmark. In addition, the improvement required for MMGP was clarified from the experimental results.

In the future, we will devise problems with three or more local optimal programs and problems using common variables in the global and local optimal programs. Furthermore, using those benchmark problems, we will improve the search capability of MMGP that can search multimodal problems more reliably.

References

[1] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 518–538, Aug 2017.

[2] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," *Statistics and Computing*, vol. 4, no. 2, pp. 87–112, Jun 1994. [Online]. Available: https://doi.org/10.1007/BF00175355

[3] S. Yoshida, T. Harada, and R. Thawonmas, "Multimodal genetic programming by using tree structure similarity clustering," in *2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA)*, Nov 2017, pp. 85–90.

[4] K. Murano, S. Yoshida, T. Harada, and R. Thawonmas, "A study on multimodal genetic programming introducing program simplification," in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, Dec 2018, pp. 109–114.

[5] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. De Jong, and et al., "Genetic programming needs better benchmarks," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 791–798. [Online]. Available: https://doi.org/10.1145/2330163.2330273

[6] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning/david e," *Goldberg.–[USA]: Addison-Wesley*, 1989.

[7] R. Yang, P. Kalnis, and A. K. H. Tung, "Similarity evaluation on tree-structured data," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 754–765. [Online]. Available: https://doi.org/10.1145/1066157.1066243