

Tree-Structure-Aware Genetic Operators in Genetic Programming

Kisung Seo[†] and Chulhyuk Pang^{*}

Abstract – In this paper, we suggest tree-structure-aware GP (Genetic Programming) operators that heed tree distributions in structure space and their possible structural difficulties. The main idea of the proposed GP operators is to place the generated offspring of crossover and/or mutation in a specified region of tree structure space insofar as possible by biasing the tree structures of the altered subtrees, taking into account the observation that most solutions are found in that region.

To demonstrate the effectiveness of the proposed approach, experiments on the binomial-3 regression, multiplexor and even parity problems are performed. The results show that the results using the proposed tree-structure-aware operators are superior to the results of standard GP for all three test problems in both success rate and number of evaluations.

Keywords: Genetic programming, Genetic operators, Crossover, Balanced tree structures

1. Introduction

GP (Genetic Programming) [1, 2] is an extension of the GA (Genetic Algorithm) [3], using evolution to optimize actual computer programs or algorithms to solve some task, typically involving a tree-type representation. The tree representation of GP chromosomes, as compared with the string representation typically used in GA, gives GP more flexibility to encode solution representations for many real-world design and optimization applications [4-6].

The recombination operator in GP is regarded as a main driving force for the success of a GP run. Many recombination operators have been proposed to improve its performance. These approaches include context preserving [7], context-aware crossover [7], GP uniform crossover [8] and, depth-dependent crossover [9]. These approaches are mainly to minimize destructive effects of standard crossover, or preserve the position of genetic material in the genotype, and/or to accumulate building blocks. Although it is very important to preserve or minimize destruction of genetic material, there is no known implicit direction to favor.

The regions of a tree structure's space by a distribution of tree shapes has been investigated [10]. It is shown that tree structure can have a substantial impact on determining problem difficulty in standard GP, and most solutions in standard GP are found in a specific region of the space of tree structures [10].

We suggest new recombination operators using the above idea of tree structure space. The proposed GP operators are to place the generated offspring of crossover and/or mutation in a specified region of tree structure space insofar as possible, based on the observation that most

solutions are found in that region. To enable that, the proposed operators are designed to utilize information about the region to which the parents belong and node/depth statistics of the subtree selected for modification.

2. Structural Difficulty in GP

It was shown that structure alone can pose great difficulty for a standard GP search [10-12]. In particular, they classified four regions of the search space of tree structures, as shown in Fig. 1.

The regions of tree structures are summarized as follows, classified according to number of nodes and tree depth. There exist at least four distinct regions for depths 0–26. Region I contains most solutions in standard GP. Far fewer individuals than in Region I appear in Regions II (IIa and IIb). Only partial mixing of size / shape subtrees occurs

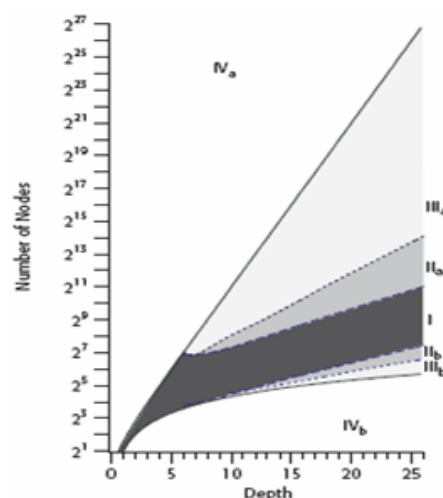


Fig. 1. Node and depth regions - Reprinted with permission from [10]

[†] Corresponding Author: Dept. of Electronics Engineering, Seokyeong University, Korea.(ksseo@skuniv.ac.kr)

^{*} Green Energy Division, AP Systems, Hwaseong, Korea
Received: July 27, 2013; Accepted: November 11, 2013

here, with mixing becoming non-existent towards the boundaries furthest away from Region I. Region III (IIIa and IIIb) is a place where even fewer individuals typically appear. Regions IVa and IVb are regions that are structurally precluded from binary trees.

3. Tree-structure Based Aware Operators

The main idea of the proposed tree-structure-based aware GP operator is to place generated offspring(s) resulting from crossover and/or mutation into region I as much as possible, using the observation of tree structure as guidance in specifying the shape of genetic material to be introduced [10,11]. Because most solutions of test problems found by standard GP search occur in region I, it is natural to say that it is easy for GP to find solutions to problems which exist in region I. Furthermore, biasing the operators to put generated offspring into region I seems to be plausible strategy to improve search capability in standard tree-based GP.

First, from the parent's region information to which a parent belongs, we can determine whether the offspring should move in some particular direction in shape space or stay in the current region, based on the idea that most solutions are found in region I. Second, the node/depth ratio of the chosen subtree is used to generate an offspring as close as desired to a specified structure by identifying a structurally appropriate subtree of parent2 for subtree

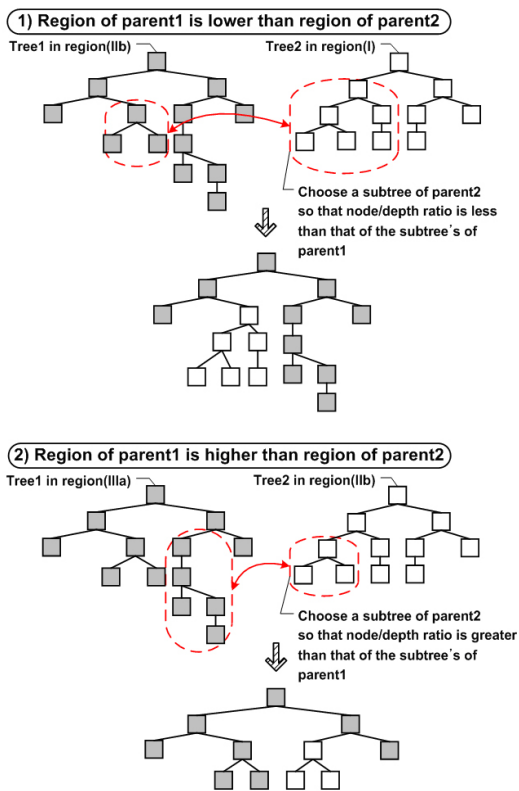


Fig. 2. Tree-structure-aware crossover

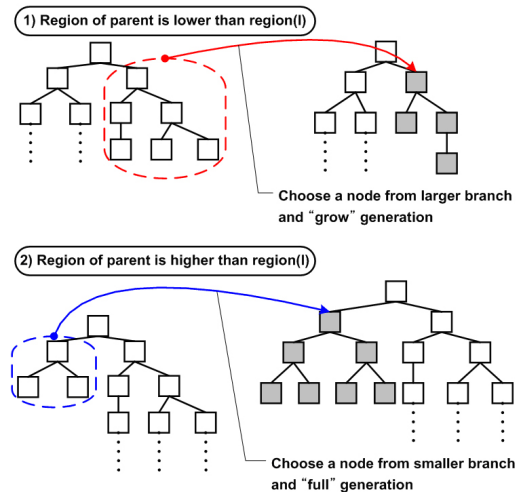


Fig. 3. Tree-structure-aware mutation

swapping.

The tree-structure-based aware crossover algorithm is described in Fig. 2. Two parents are selected by roulette wheel selection and the region of each parent is examined. A subtree of parent1 is chosen at random, and a subtree of parent2 is chosen for substitution according to the node/depth ratio of the subtree of parent1 and the regions of each parent.

For example, if the region of parent1 is lower than the region of parent2, a subtree of parent2 is chosen among possible candidates which have a smaller node/depth ratio than that of the subtree of parent1. This operation generates a child which has high probability of being in region I. In the other case, a subtree of parent2 is chosen among candidates which have larger node/depth ratios than that of the subtree of parent1. Unlike standard crossover, offspring are obtained from crossover of two parents, because it is difficult to control the shape of both offspring by the subtree swapping operation in the proposed crossover. Therefore, offspring from the proposed crossover is generated one at a time.

The proposed mutation operator has a similar concept as of crossover. The major feature of the tree-structure-based aware mutation is to control the replaced subtree by generation methods (grow, full, or half & half) considering the current shape of the parent tree.

4. Experiments & Results

4.1 Experimental setup

The tree-structure-based aware GP operators have been applied to three standard benchmark problems — binomial-3 regression, multiplexor and even parity problem — and compared with standard GP operators and their combinations. In all experiments, 20 runs were executed and the number of evaluations and hit ratio or success rate

of solutions found are recorded for standard crossover & mutation and the proposed crossover & mutation. We used lil-GP [13] for the standard GP runs, and modified it heavily to implement the new crossover and mutation operators. The GP parameters were as shown below.

- Number of generations : 500 for binomial-3, 2000 to 10000 for multiplexor, and 100 to 7000 for even-parity
- Population sizes : 500
- Initial population: half_and_half
- Initial depth : 2-6
- Max depth : 17, (25 for 11-multiplexor)
- Selection : Tournament (size=7) for standard roulette wheel for proposed
- Crossover : 0.9
- Mutation : 0.1

4.2 Binomial-3 regression

The binomial-3 [11] is a symbolic regression problem which is used for many applications. The problem involves seeking a function expressible as:

$$f(x) = x^3 + 3x^2 + 3x + 1 \quad (1)$$

The binomial-3 problem was defined as fitness cases for 50 equidistant points generated from Eq. (1), over the interval [-1,0). The raw fitness is the sum of the absolute errors. The ephemeral random constants (ERCs) are uniformly distributed over a specified interval of the form [- α , α], where α is a real number that defines the range for ERCs. Four values for α were used-namely: {0, 1, 3, 10}.

The experimental results for the binomial-3 problem are summarized in Table 1. The proposed methods produced better results than standard crossover & mutation, in terms of number of evaluations, and success rate. Because the binomial-3 problem is easier than the other two problems, it shows only slight differences in the success rate comparison. However, big improvements are shown for the number of evaluations, which is one of the most important indexes. The proposed crossover & mutation method reduces evaluations required approximately 70% compared to standard GP, except for the very easy case of no ERCs.

To investigate the distribution of offspring in tree

Table 1. Results of binomial-3 regression problem

Binomial-3 regression				
Operators	ERC range	Hit rate	No. of Evals	Success Rate
Standard crossover & Mutation	noERC	50.00	4,850.0	100%
	[-1, 1]	47.75	149,676.5	85%
	[-3, 3]	49.10	116,704.6	85%
Proposed crossover & Mutation	noERC	50.00	4,552.5	100%
	[-1, 1]	49.60	70,368.4	95%
	[-3, 3]	49.25	26,968.8	85%
	[-10, 10]	49.75	65,975.0	90%

structure space produced by the proposed crossover & mutation, data from a typical run for ERC[-100,100] are plotted in Fig. 3. During the initial random generation, most of the 500 individuals are placed within the left-hand portion of region I, and a few trees are located at the boundary with region IIb and on the border line in the lower left corner. As the generations are increased, the generated trees are moved to right and distributed evenly. From the results, we can confirm that most of proposed operations was executed according to the intention of tree-structure-based aware operation. One of the most

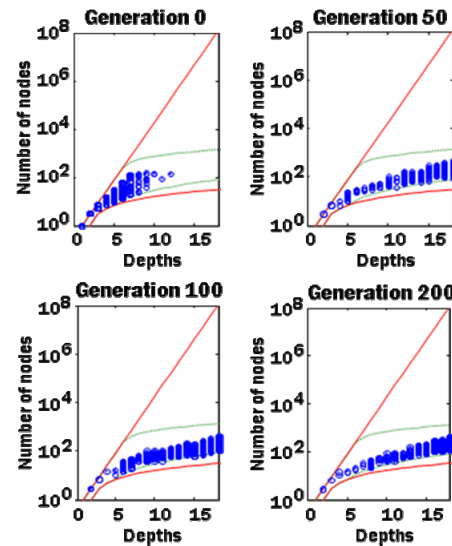


Fig. 3. Distribution of tree structures generated by the proposed crossover & mutation operators for 500 individuals by generation (binomial-3 regression with ERC[-100,100])

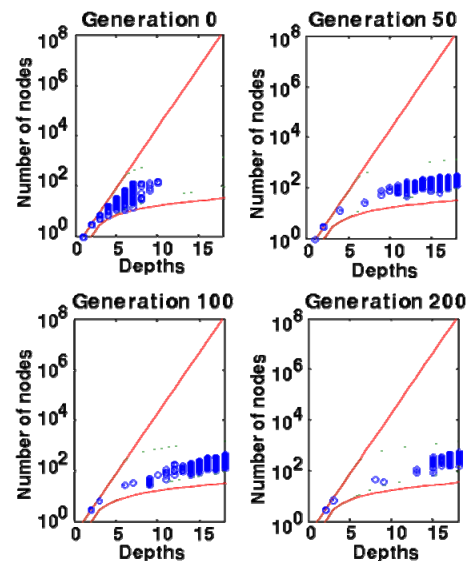


Fig. 4. Distribution of tree structures generated by standard crossover & mutation for 500 individuals by generation (binomial-3 regression with ERC[-100, 100])

Table 2. Results of binomial-3 regression problem

6-and 11-multiplexor				
Operators	Bits	Hit rate	No. of Evals	Success rate
Standard Crossover & Mutation	6bit	61.75 (max 64)	979,444.4	45%
	11bit	1,736.21 (max2048)	NA	0%
Proposed Crossover & Mutation	6bit	62.45	164,807.7	70%
	11bit	1,990.55	617,214.3	60%

interesting phenomena is the distribution of individuals shows much high diversity than the result of standard GP's in Fig. 4.

4.3 Multiplexor

The objective of the multiplexor problem [1] is to find a boolean function which performs multiplexing using an n-bit address. The experimental results for 6- and 11-bit multiplexor problems are summarized in Table 2.

For the 6-multiplexor, the proposed method is superior to standard GP for success rate and evaluations. Notably, the evaluations needed using the proposed crossover & mutation operators are only 17% of standard GP's. For the 11-multiplexor, the success rate of the proposed crossover & mutation is outstanding, at 60%; the other is 0%.

The distribution of trees in tree structure space using the proposed crossover & mutation for a typical run on the 6-multiplexor is very similar in the case of binomial-3 problem (not in shown in the paper because of space limitation).

4.4 Even-parity

The third experiment was on the even-n-parity problem [4]. It has been recognized as difficult for genetic programming to induce if no bias favorable to their induction is introduced in the function set, the input representation, or in any other part of the algorithm [14]. Table 3 summarizes the experimental results for 3-, 4- and 5-even parity problems. For the even-3 parity problem, the results of the proposed method is better than those of standard GP in terms of number of evaluations. The success rates of the proposed method is better than standard GP's

Table 3. Results of even parity problem

3-, 4-, and 5-even parity				
Operators	Bits	Hit rate	No. of evals	Success rate
Standard Crossover & Mutation	3bits	8.00 (max 8)	35,815.8	100%
	4bits	15.55 (max 16)	1,987,333.3	75%
	5bits	20.45 (max 32)	NA	0%
Proposed crossover & Mutation	3bits	8	15,266.7	100%
	4bits	15.90	237,631.6	95%
	5bits	31.35	2,924,100.0	75%

for the even-4 parity problem. Moreover, the number of evaluations of the proposed crossover & mutation is only 12% of standard GP's.

For the even-5 parity problem, the success rate of the proposed crossover & mutation (75%) is quite superior to standard GP's (never found). The distribution graph of tree structures for a typical run on the even-3 parity problem displays similar results to those seen for the binomial-3 regression and multiplexor problems

4.5 Investigation of the depth distribution of individuals

Fig. 5 shows the depth distributions of parents that participated in GP crossover and mutation in the regression problem with ERC[-100,100]. The distribution using the newly proposed operators is more balanced than the results with standard operators. With standard operators, the parents' tree sizes of 16 and 17 are a larger proportion than with the proposed operators. The case of the 6-multiplexor problem is similar to that of the regression problem, as shown in Fig. 6.

The distribution of participating parents in the proposed GP operation is very widely spread in the 5-parity problem, as shown in Fig. 7. That seems to be closely related to the

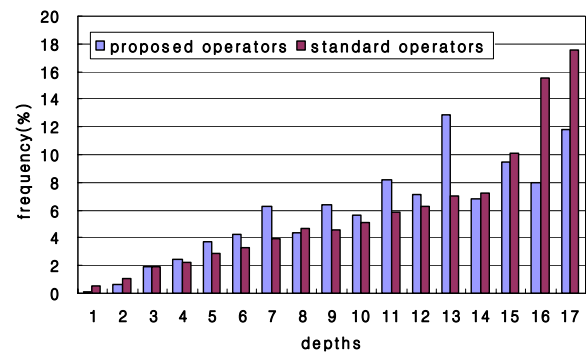


Fig. 5. Depth distribution of parents which participated in GP crossover & mutation, on the regression problem with ERC[-100,100]

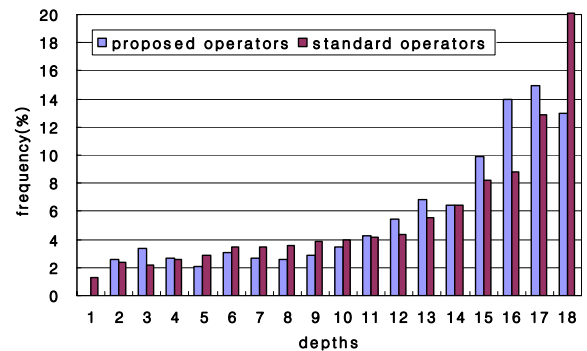


Fig. 6. Depth distribution of Parents which participated in GP crossover & mutation, on the 6-multiplexor problem

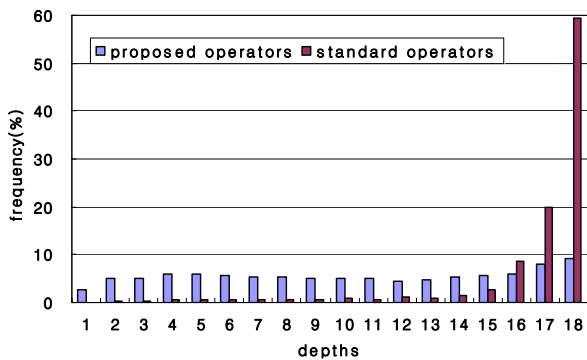


Fig. 7. Depth distribution of parents which participated in GP crossover & mutation, on the 5-parity problem

diversity of the population, because with the standard operators, most of individuals rapidly grew to large trees.

5. Conclusion

We have suggested new recombination operators based on tree distributions in structure space and structural difficulties. To demonstrate the effectiveness of our proposed approach, experiments on the binomial-3 regression, multiplexor and even parity problem were performed. The experimental results showed that the results using the proposed tree-structure based aware operators were superior to the results of standard GP for all three test problems in both success rate and number of evaluations.

Further study will aim at analysis, extension and refinement of the tree-structure based aware GP operators to validate their effectiveness more theoretically and to apply them to more complex and practical real-world problems.

Acknowledgements

This Research was supported by Seokyeong University in 2013.

References

- [1] J. R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection*, MIT Press, Cambridge, MA, USA, 1992
- [2] S. Luke, *Issues in Scaling Genetic Programming Breeding Strategies, Tree Generation and Code Bloat*, PhD thesis, University of Maryland, 2000
- [3] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.
- [4] J.W. Davidson, D.A. Savic, G.A. Walters, "Symbolic and numerical regression: experiments and applications," *Information Sciences*, Volume 150, Issues 1-2,

- March 2003, pp. 95-117
- [5] K. Seo S. Hyun E. D. Goodman, "Genetic Programming-Based Automatic Gait Generation in Joint Space for a Quadruped Robot," *Advanced Robotics*, Vol. 24, No. 15, pp. 2199-2214. 2010.
- [6] K. Seo, B. Hyeon, S. Hyun, and Y. Lee, "Genetic Programming-Based Model Output Statistics for Short-Range Temperature Prediction," *Lecture Notes in Computer Science*, Springer-Verlag, Vol. 7835, pp. 122-131, 2013
- [7] H. Majeed, C. Ryan, "On the Constructiveness of Context Aware Crossover," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'07)*, pp.1659-1666, London, England, United Kingdom, July 7-11, 2007
- [8] R. Poli, J. Page, "Solving High Order Boolean Parity Problems with Smooth Uniform Crossover," *Sub Machine Code GP and Demes*, *Genetic Programming and Evolvable Machines*, 1(1/2), pp. 37-56, April 2000
- [9] T. Ito, H. Iba, S. Sato, "Depth Dependent Crossover for Genetic Programming," in *Proceedings of the World Congress on Computational Intelligence*, pp. 775-780, Anchorage, AK, USA, May 4-9, 1998
- [10] J. M. Daida, A. M. Hilss, "Identifying Structural Mechanisms in Standard Genetic Programming," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2003)*, LNCS 2724, pp.1639-1651, Chicago, IL, USA, July 12-16, 2003.
- [11] J. M. Daida, J. A. Polito, S. A. Stanhope, R. R. Berttam, J. C. Khoo, "What Makes a Problem GP Hard? Analysis of a Tunably Difficult Problem in Genetic Programming," *Genetic Programming and Evolvable Machines*, ISSN 1389-2576, 2(2), June 2001, pp.165-191.
- [12] Ngyen, X. Hoai, B. McKay, D. Essam, "Representation and structural Difficulty in Genetic Programming," *IEEE Transactions on Evolutionary Computation*, 10(2), pp.157-166, April 2006
- [13] D. Zongker, B. Punch, *Lil-GP User's Manual*, Michigan State University, July, 1995
- [14] S. Sliva, E. Costa, "Resource Limited Genetic Programming: The Dynamic Approach," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'05)*, pp.1673-1680, Washington, DC, USA, June 25-29, 2005



Kisung Seo received the BS, MS, and Ph.D degrees in Electrical Engineering from Yonsei University, Seoul, Korea, in 1986, 1988, and 1993 respectively. He became Full Time Lecturer and Assistant Professor of Industrial Engineering in 1993 and 1995 at Seokyeong University, Seoul, Korea. He joined

Genetic Algorithms Research and Applications Group (GARAGe) and Case Center for Computer-Aided Engineering & Manufacturing, Michigan State University from 1999 to 2002 as a Research Associate. He was also appointed Visiting Assistant Professor in Electrical & Computer Engineering, Michigan State University from 2002 to 2003. He was a Visiting Scholar at BEACON (Bio/computational Evolution in Action CONSortium) Center, Michigan State University from 2011 to 2012. He is currently Associate Professor of Electronics Engineering, Seokyeong University. His research interests include genetic programming, evolutionary design of gait generation for quadruped and humanoid robots, and evolutionary prediction for weather system.



Cheolhyuk Bhang He received B.S. and M.S. degrees from Seokyeong University, Seoul, Korea, in 2007 and 2009, respectively. Currently, he is a R&D team manager with the AP Systems, Hwaseong, Korea. His research interests are genetic programming and control systems in power electronics.