

A Probabilistic Deterministic Genetic Programming Algorithm Using the Semantic Control Crossover

Usman Kaisser Moghal

Computer Sciences

FAST University

Islamabad, Pakistan

usman.kaisser@nu.edu.pk

Abstract—Genetic programming (GP) is an evolution of genetic algorithms that produces functional programs which solve a specific task, commonly represented as a tree. With the introduction of new crossover techniques, several variants of GP's have come about, with researches pushing the boundaries of efficiency.

Geometric semantic genetic programming (GSGP) is an example of one such variant. Unlike the traditional GP which will perform crossover/mutation at any randomly selected node, the GSGP uses a convex combination of two parental individuals at the root node, based on the positioning of the parents in semantic space, thus allowing a more deterministic approach.

In this paper, a new variant is proposed, which utilizes the semantic control crossover (SCC) technique to enhance a standard GSGP, such that it produces an solution, without the risk of finding a local instead of global optimal solution.

Index Terms—Genetic Programming, Evolutionary Computation, Symbolic Regression, Geometric Semantic Crossover, Semantic Control Crossover

I. INTRODUCTION

Genetic programming (GP) provides a powerful tool that can be used to approximate complex functions without user interference. It takes an input vector, and an expected outcome vector, and starts formulating a potential structure iteratively.

The process involves several genetic functions such as selection, crossover and mutation; which repeatedly modify the individuals of a population, and continue to do so until the population produces a value rich enough.

In a GP, the crossover method plays a heavy role as it is primarily responsible for the quality of the approximated solution. Therefore, it is a crucial point for improvement by researchers.

This traditional methodology has been refined over the years, and several optimized solutions produced. Some of them will be discussed in this paper.

In recent years, several new methods have come about that focus more on the behavior and semantics of the functions rather than its structure. Researchers have found that changing the structure of an individual will yield in uncertain offspring, while changing the behavior of an individual will yield a directional shift in the offspring.

This led to several techniques being introduced by researchers, where the semantics of individuals are explored and exploited. However, with each new technique came a new hurdle.

In this paper, after discussion of said techniques, a new method is proposed, which following a probabilistic recourse deterministically produces an optimal solution.

II. LITERATURE REVIEW

A. Traditional GP

Genetic programming (GP) provides a powerful tool that can be used to approximate complex functions without user interference. It takes an input vector, and an expected outcome vector, and starts formulating a potential structure iteratively. The process involves the following steps:

1) *Initialization*: This is the first step of the process, where program structures are created. The process varies based on the method used. There are 3 prominent methods: *grow method*, *full method*, and *ramped half-and-half method*. Considering a tree representation, all methods are similar in the sense that they create a tree using members of two sets. The first is a set consisting of functions (such as +, -, *, etc.), while the second is a set consisting of terminals (such as 0, 1, 2, etc.). The type of method used will decide which set to choose an element from and where to add said element into the tree. This process starts the GP, but is called again after every generation.

2) *Fitness Calculation*: Each program structure (individual) is assigned a fitness value, which represents the degree to which the structure solves the problem best. Two common fitness measures include the *raw fitness* and *standardized fitness*.

3) *Selection*: This is the operator which decides whether an individual is given a chance for reproduction, or whether it is cast aside (typically based on its fitness). If the individual is selected, then the genetic operations *crossover* and *mutation* can be applied on it.

4) *Crossover / Mutation*: These are two genetic operations that can be applied on an individual should they be selected. Either one of them is applied at a time. *Crossover* involves selecting a portion of an individual and adding its influence into another individual. This can be achieved in various ways. The *standard GP crossover* involves randomly taking a subtree of an individual, and swapping it with another such subtree of another such individual. *Mutation* involves changing the individual asexually. The *standard GP mutation* involves selecting a subtree at random, and replacing it with a newly generated subtree.

Throughout the process, the fitness of all individuals of a generation is calculated, and the error rate calculated. Typically, mean squared error (MSE) is used.

The algorithm will continue to create new individuals generation after generation, until a target fitness is reached, or until a certain number of iterations have passed. At the end of its run, the algorithm will return the individual with the best fitness.

B. Semantic Space

The genetic functions of a traditional GP are probabilistic in nature. A node is randomly selected from a random tree, and is swapped with another such random node of another such random tree. The resulting offspring could produce a better or worse tree. Even if the parents fitness was high, there is no guarantee that the resulting offspring will have a high or even equal fitness.

From this lack of direction, an idea sprung up to use the semantics of a tree as a guide to a more deterministic approach. The semantics of a tree is defined as a set of outputs for a set of given inputs [1], [2].

Using semantics of a tree, a semantic space can be created, where if needed, every subtree can be compared and the optimal subtree selected, thus creating the potential for a more deterministic approach.

C. Geometric Semantic GP (GSGP)

An idea brought on by using the semantic space is the geometric semantic genetic programming (GSGP) [3], [4]. Instead of taking a subtree from each child, the GSGP algorithm takes both parents, and generates offspring via a convex operation of both parents. During this operation, a ratio 'R' is set, and applied as seen in 1 and 2, where 'R' is '0.7'.

The offspring of both parents can be calculated using the equation 1, where 'R' is a random constant between [0,1].

$$T_0 = (T_1 * R) + (T_2 * (1 - R)) \quad (1)$$

The equation for mutation within a GSGP is as shown in equation 2, where 'ms' is the mutation step parameter. If 'ms' is set to a small value, the offspring will be closer to their parents semantically, and if 'ms' is set to a larger value, the offspring will be very different to their parents semantically.

$$T_0 = T_1 + ms * (TR_1 - TR_2) \quad (2)$$

Based on the example shown in figure 1, and the resultant offspring shown in figure 2, all trees can be plotted in semantic space as seen in figure 3.

An inherent problem is visualized by the plot in figure 3. The tree 'P' represents the optimal tree. But the convex operation of GSGP can only provide an offspring that is in-between both parents in semantic space. In other words, the best offspring produced by standard GSGP can never have a fitness better than the fitness of the best parent. And in the case seen in figure 3 where the optimal tree 'P' is not between any parent tree, the standard GSGP algorithm can never find the optimal solution 'P'.

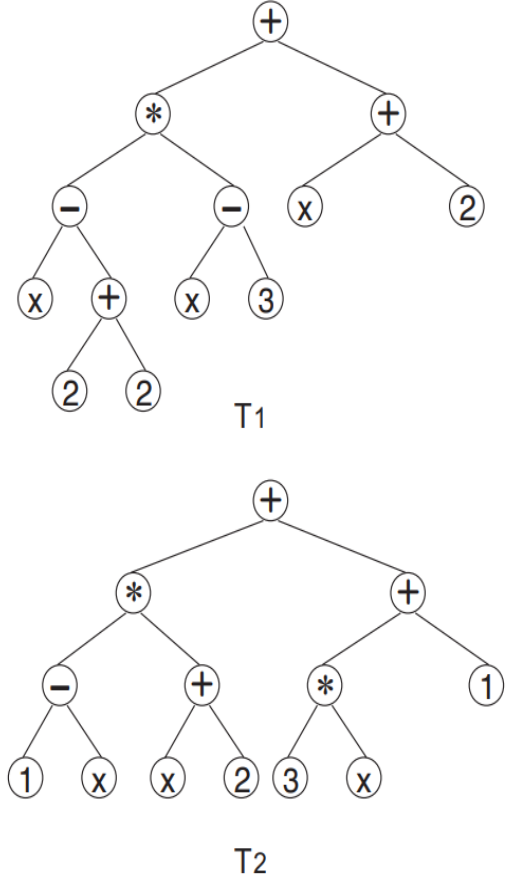


Fig. 1. Two parental individuals T1 and T2

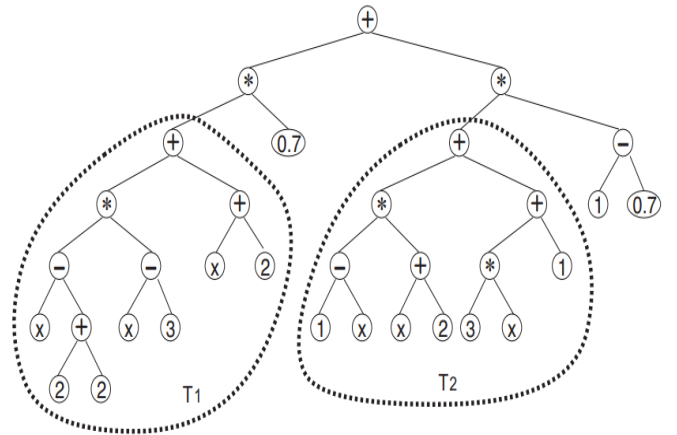


Fig. 2. An offspring created using GSGP with R set to 0.7

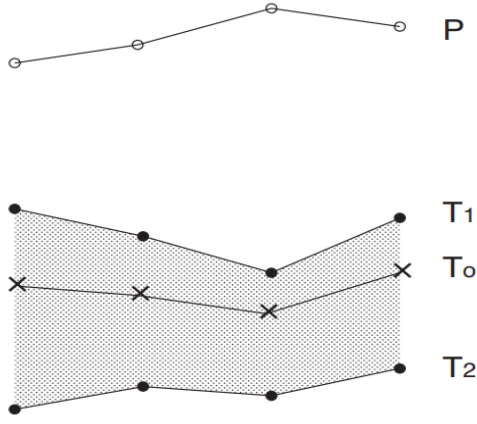


Fig. 3. Tree plots of standard GSGP in semantic space

D. Deterministic Geometric Semantic Crossover (DGSC)

Stemming from the problem faced in the standard GSGP algorithm, the deterministic geometric semantic crossover was born [5].

The main logic behind this crossover, is to make the process more deterministic by calculating the optimal value of 'R', and hence obtaining the best offspring. Moreover, in order to deal with the limitation of the standard GSGP, the value of 'R' is no longer set to [0,1]. It can now be less than 0, and even greater than 1. This will allow more flexibility to set 'R' to obtain optimal solution 'P'.

We can get the optimal value for 'R' by making use of the semantic space, and the positioning of the tree in said space. Doing so, provides us with the equation 3

$$R = \frac{|T_1 T_2| - |T_1 P| * \cos\theta}{|T_1 T_2|} \quad (3)$$

Even though this crossover allows a standard GSGP to be more deterministic, and even get to the optimal solution, it has a flaw. This crossover makes the algorithm converge too fast, and 'tailoring' the offspring to the potential optimal tree 'P' creates an environment where the algorithm can potentially find local instead of global optimal solution.

III. PROPOSED METHOD

The DGSC had the right approach, but was forcing the algorithm towards a potential solution. What we learn from this is that exploring the semantic space is the right approach, however we have to explore our option more when considering offspring, rather than bee-lining towards a potential solution.

In other words, a certain level of randomness is appropriate, as it allows the algorithm to explore a wider range in our semantic space and eventually find the global optimal solution.

With that in mind, a crossover technique was proposed in [6] called the semantic control crossover (SCC). This takes advantage of the semantic space to ascertain how similar or dissimilar individuals are with each other. If two parents which are similar to each other semantically, then their offspring

will be semantically similar to its parents. Similarly, if they are dissimilar, then the offspring will be dissimilar. The 'semantic distance' needed to ascertain semantic similarity can be calculated using the equation 4.

$$d_{AB_j} = \sqrt{\sum_{i=1}^n (semantics_i^{P_A} - semantics_i^{P_{B_j}})^2} \quad (4)$$

If a local search is required, then equation 5 is used.

$$\alpha(t) = \alpha_{max} + (-(\alpha_{max} - \alpha_{min})/G) * t \quad (5)$$

The process of SCC proceeds as shown in the figures 4 and 5.

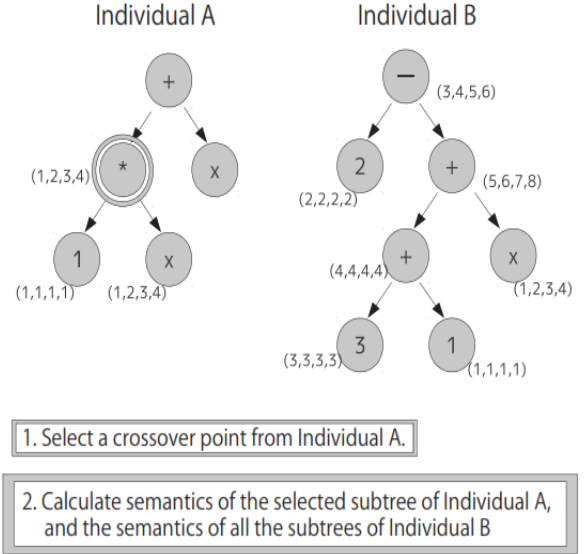


Fig. 4. Procedure of SCC step 1 and 2

This creates an opportunity for us to exploit. We can compensate the limitation of a standard GSGP through SCC.

We will continue to use GSGP, since it works fast as it only considers the semantic value of each tree only at the node, rather than at every node. This also simplifies the semantic space, and makes comparing different trees easier. And the nature of the convex operation utilizing semantic space allows a deterministic approach. The only limitation is a situation where the optimal solution is out of bounds of the parents.

This is where we use SCC. By observing the semantic space, we can observe if such a scenario has developed, where the optimal solution is out of bounds by solving linear equations (in semantic space). In such a situation we can use SCC, and depending on how far off the optimal solution is from the parents, we can adjust if we want a big change in offspring from the parents, or a minor change.

In other words, if the offspring being produced by GSGP are far away from the optimal solution, and if the optimal solution is also out of bounds, then we can perform SCC between

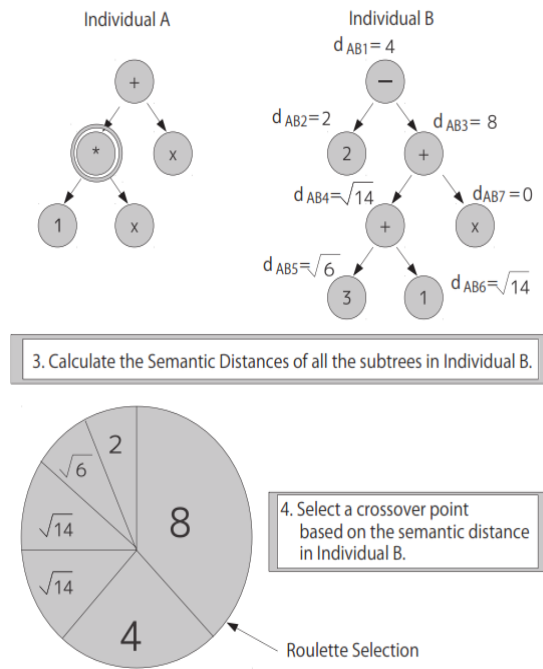


Fig. 5. Procedure of SCC step 3 and 4

two such parents who are semantically dissimilar. Thus, the produced offspring will be very different from the parents, and hence have a greater chance of lying in the semantic space such that they also encompass the optimal solution. Therefore, we can now continue using the standard GSGP.

This methodology follows a probabilistic recourse, in which the offspring produced by the SCC could potentially be better or worse. This element of randomness helps us in preventing the algorithm from converging too quickly and finding a local instead of a global optimal solution.

REFERENCES

- [1] Nicholas Freitag McPhee, Brian Ohs, and Tyler Hutchinson. "Enumerating building block semantics in genetic programming". In: (2007).
- [2] Nicholas Freitag McPhee, Brian Ohs, and Tyler Hutchinson. "Semantic building blocks in genetic programming". In: *European Conference on Genetic Programming*. Springer. 2008, pp. 134–145.
- [3] Alberto Moraglio, Krzysztof Krawiec, and Colin G Johnson. "Geometric semantic genetic programming". In: *International Conference on Parallel Problem Solving from Nature*. Springer. 2012, pp. 21–31.
- [4] Leonardo Vanneschi, Mauro Castelli, Luca Manzoni, et al. "A new implementation of geometric semantic GP and its application to problems in pharmacokinetics". In: *European Conference on Genetic Programming*. Springer. 2013, pp. 205–216.

- [5] Akira Hara, Jun-ichi Kushida, Ryo Tanemura, et al. "Deterministic crossover based on target semantics in geometric semantic genetic programming". In: *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE. 2016, pp. 197–202.
- [6] Akira Hara, Yoshimasa Ueno, and Tetsuyuki Takahama. "New crossover operator based on semantic distance between subtrees in Genetic Programming". In: *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2012, pp. 721–726.