A BLOAT CONTROL IN GENETIC PROGRAMMING FOR BREAST CANCER DAIGNOSIS

Arzoo Khan, Medhavi Chouhan

Information technology Dept, SVITS, Indore, RGPV Bhopal(M.P.)

24arzoo.khan@gmail.com,8602219918

Abstract— Breast cancer affects several people at present time. Diagnosis which determines whether the cancer is benign or malignant requires a lot of effort from doctors and physician. Early diagnosis may save many lives. Accurate classification plays an important role in medical diagnosis. Genetic programming is a machine learning algorithm which now days excelling in classification field. But Genetic programming generally face the problem of code bloating in which an increase in average tree size is found without a corresponding increase in fitness. In this paper we are proposing a new technique for solving the problem of bloat and for increasing classification accuracy. The technique is known as intelligent crossover and mutation technique. This technique is a combination of hill climbing and conventional method which will be applied on both crossover and mutation operator. To demonstrate this, we had taken WBC dataset from UCI repository which has 2 classes and 9 features and we have compared classification accuracy of our method with standard crossover and FEDS crossover. Our classification accuracy was 96.5% for 50-50 training and testing methodology and 97.6% for 10 fold cross validation technique. This shows our method can be used for medical diagnosis as it provides good results.

Keywords— Bloat, Genetic Programming, Crossover, Fitness, Point Mutation, Breast Cancer Diagnosis, Wincosin Breast Cancer dataset, Intelligent Crossover and Mutation Technique.

INTRODUCTION

Breast cancer is the most common invasive cancer in females worldwide[2]. Breast cancer is a kind of a cancer that develops from breast cells. It accounts for 16% of all female cancers and 22.9% of invasive cancers in women. 18.2% of all cancer deaths worldwide, including both males and females are from breast cancer[1]. Because of social and cultural considerations, breast cancer ranks highest among women's health concerns. It is the most frequent diagnosed cancer in women. After thyroid cancer, melanoma, and lymphoma, breast cancer ranks fourth in cancer incidences in women between 20 to 29 years. The factors that cause this disease are many and cannot be easily determined. Breast cancer begins with the uncontrolled division of one cell inside the breast and results in a visible mass, called as tumor. The tumor can be either benign or malignant. The diagnosis process which determines whether the cancer is benign or malignant also requires a great deal of effort from doctors and physician. Several tests are involved in the diagnosis of breast cancer, such as lump thickness, uniformity of cell size, uniformity of cell shape…etc; the ultimate result may be difficult to obtain, even for medical experts. The accurate diagnosis for determining whether the tumor is benign or malignant can result in saving lives. Early diagnosis helps to save thousands of disease victims. Therefore, precise classification is needed in clinic as classification plays an important role in breast cancer.

Genetic programming approaches in medical domains is increasing rapidly due to the improvement effectiveness of these approaches to classification and prediction systems, especially in helping medical practitioners in their decision making[3]. In addition to its importance in finding ways to improve patient outcomes, reduce the cost of medicine, and help in enhancing clinical studies. Genetic programming is one of the machine learning algorithm performs classification [5][6]which is the most essential and important task. Many experiments are performed on medical datasets for example WBC for breast cancer using multiple classifiers which shows good classification accuracy. This importance of GP has been motivated for the last 25 years, when scientists began to realize the complexity of taking certain decisions to treat particular diseases. The use of machine learning and genetic programming as tools in medical diagnosis becomes very effective and one of the critical diseases in medicine where the classification task plays a vital role is the diagnosis of breast cancer. Therefore machine learning techniques such as genetic programming can help doctors to an accurate diagnosis for breast cancer and make the correct classification of being benign or malignant tumor. There is no doubt that the decisions of doctors and specialists are the most important in the diagnosis but machine learning tools for classification also help doctors and specialists in a great deal.

Genetic Programming (GP) is an evolutionary learning technique that offers a great potential for classification. Genetic Programming is essentially considered to be a variant of Genetic Algorithms (GA) that uses a complex representation language to codify individuals[7]. The most commonly used representation schema is based on trees, although other options exist. The original goal of GP, as its name implies, was the evolution of computer programs. However, nowadays GP is used to evolve other abstractions of knowledge, like mathematical expressions or rule-based systems, for example. GP individuals are usually seen as parse trees, where leaves correspond to terminal symbols (variables and constants) and internal nodes correspond to non terminals (operators and functions). The set of all the non terminal symbols allowed is called the function set, whereas the terminal symbols allowed constitute the terminal set. Two conditions must be satisfied to ensure that GP can be successfully applied to a specific problem: sufficiency and closure. Sufficiency states that the terminals and non terminals (in combination) must be capable of representing a solution to the problem. Closure requires that each function of the non terminal set should be able to handle all values it might receive as input. In practice, we often need to evolve programs that handle values of different types, and this makes it difficult to meet the closure requirement.

Crossover (sexual recombination) is recognized as the primary genetic operator for improving program structures in tree based Genetic Programming (GP). It plays a critical role in deriving optimal solutions as shown by the large number of studies related to crossover operators since the 1990s. The standard crossover operator picks a random crossover point in each of two parent program trees, and swaps the two sub trees rooted at the chosen crossover points to generate two new programs[8]. This has been seen as problematic: most crossover events in the standard crossover produce offspring with less than half of the fitness of their parents. Further, crossover points in the standard crossover are implicitly biased towards the leaves of a program tree because there are generally more nodes in that part of the tree, giving them a higher cumulative probability of being selected . Although an alternative crossover point selection with a preference towards function nodes was introduced in it may still be biased towards leaf node. The bias issue aggravates the destructiveness of the standard crossover operator and causes the code bloat problem in GP. In particular, it was noted that very often the average size (number of nodes) of the programs in a population, after a certain number of generations in which it was largely static, at some point would start growing at a rapid pace. Typically the increase in program size was not accompanied by any corresponding increase in fitness. The origin of this phenomenon, which is known as bloat[4]. Bloat is not only surprising, it also has significant practical effects: large programs are computationally expensive to evolve and later use can be hard to interpret, and may exhibit poor generalization. For these reasons bloat has been a subject of intense study in GP.

To solve this problem of bloat a new technique is going to proposed by us known as intelligent crossover and mutation technique. In this method, we are making our crossover and mutation operators intelligent and name given to them are intelligent crossover operator and intelligent mutation operator. In intelligent crossover operator, we will divide our individuals into two parts. On half of the individuals we will apply hill climbing and on rest of the individuals we will apply standard crossover. By doing this, we will get better and different variety of solutions as well as get our result earlier. In intelligent mutation operator, we will again divide our individuals into two parts. On half of the individuals we will apply hill climbing and on rest of the individuals we will apply standard point mutation. Because of which we will reach final stage faster and earlier and we will get much better and diverse solutions.

RELATED WORK

During the evolution of solutions using genetic programming (GP) there is generally an increase in average tree size without a corresponding increase in fitness—a phenomenon commonly referred to as bloat. Bloat is basically a problem that occurs during crossover and mutation in which after a certain limit only the depth of tree will increase but not its fitness. Bloat represents the destructive nature of conventional genetic operations. Code bloat is basically of two types- structural bloat and functional bloat. Structural bloat is one of the type of code bloat which takes place when no optimal solution can be found by set of programs with bounded length. Whereas functional bloat takes place even when optimal solution lies in search space and due to which program length keeps increasing.

Over the years a range of methods have been introduced to manage bloat: treating fitness and size as a multiobjective optimization [9]; using disassortative mating [10] based on two species (one selected on fitness, the other on fitness and size); explicitly reducing the fitness of above average-sized individuals (referred to as the Tarpeian method) [11]; eliminating programs where the parent and child fitness are similar (a property termed *resilience*) [12], using a modified tournament selection operator that uses either fitness, depth or an ordered combination of both for selection; placing a form of resource constraint on the population so that larger individuals are discourage[13]; using a waiting room for individual entry into a population, with time proportional to size [15]; biasing selection for removal from a population based on size [13];

Whigham[4] has presented an implicit model of bloat control based on a spatially-structured population with local elitism; referred to as SS+E. Regular spatial structures(such as a ring or torus) maintain diversity and slow bloat by effectively reducing the population size. In addition, elitism reduces the growth of introns, especially once the population has largely converged and cannot easily find fitness improvements. Langdon and Poli[14] has explained the way to control bloat Using a fix size or depth limit (LGP) in which the bloat is controlled by applying the limit to the allowed individual size or depth. Individuals that exceed the limits are removed from the population. Because individual size or depth is calculated easily during evaluation, this approach requires only little additional computation. Stringer[16] has controlled bloat by explicitly setting an upper bound on the depth of evolved trees or by incorporating a parsimony pressure that adjusts the fitness of individuals by a tradeoff between performance and size. Bleuler et al.[17] proposed a nonparametric method, Double Tournament, this method is similar to a multi objective approach to bloat, however the objectives of fitness and size are treated separately. Hence, there are two tournaments: one based on parsimony, which produces an initial set of winners, and a subsequent tournament that selects a subset of those winners based on fitness.

Fernandez *et al.* , [18][19]specifically focused on bloat behavior using an island based model to introduce spatial structure to the population. This paper demonstrated that with an increasing number of islands (and therefore a reduced number of individuals in each island) bloat was effectively reduced. They also presented a theoretical argument for this property based on the assumption that if bloat is proportional to the square of the population size [i.e., for n individuals bloat(n) $\propto$ n2] then splitting n into smaller islands will reduce bloat. Rochat *et al.* also used an island-based approach with the additional complexity of varying population size. Individuals were added or deleted from an island based on measures of fitness change over a period of generations. Given the number of individuals to delete Ndel, the worst fitness of 2 *Ndel were sorted based on size and the first Ndel removed. Adding individuals was based on taking a proportion of the best from other islands, thus performing the role of migration. The results showed both an increase in the quality of solutions and a reduction in overall population size. However this approach involved tuning a number of parameters, such as the size of each island and migration rate.

Finally, implicit approaches to bloat control have been examined through various forms of elitism. Soule and Foster[21]introduced the concept of removal bias, arguing that neutral branches of code (i.e., introns) are likely to be small, however their replacement with crossover does not have this restriction. Hence, the children produced from neutral crossover events are likely on average to increase in size. To examine this property two forms of nondestructive crossover (NDC) were studied: a child would replace a parent if it was at least as fit as the parent, or in the strict version the child had to exceed the parent's fitness. These methods were tested with a maze navigation problem and a parity problem, with both examples showing a reduction in bloat and an improvement in convergence to fit solutions. However, since crossover is often destructive, strict elitism can reduce the effectiveness of crossover as a search mechanism, especially once the population has begun to converge. A second form of elitism, recombinative hill-climbing (RHC) was described by Hooper *et al*. With RHC the original population, termed the resident population, is copied to produce a second population, named the visitor population. Each member of the resident population is randomly paired with a member of the visitor population and genetic operators are applied to produced a child. If the child is at least as fit as the resident parent then the resident parent is replaced by the child.

Purohit et al.[20] has proposed FEDS crossover in which individuals are randomly selected from the population for the double tournament. In double tournament method there are two tournaments one based on fitness which produces an initial set of winners and a subsequent tournament that selects a subset of those winners based on depth and size limit. Then the best two parent individuals of the tournament are chosen for the crossover operation. From first parent, a sub tree is selected and placed at two different positions in the second parent to generate two children. Similarly, from second parent a sub tree is selected and placed in the first parent. In this way four individuals are generated from two parents. Then we calculate the fitness, elitism, depth limit and size of the 4 generated children and the two children which have the best result are transferred to the next generation and if the children does not have the better fitness than the parent/child will be retained to the next generation with a 0.5 probability.

PROPOSED WORK

In this paper, we are proposing a new approach to solve the problem of bloat. The approach focuses on making two important genetic operators intelligent and name given to this is Intelligent crossover and mutation technique.

1. Intelligent crossover operator- After applying reproduction operator on P_r individuals in which top fitness individuals transfers to next generation we will apply crossover operator on P_c individuals. We will divide P_c individuals into two parts P_{hc} and P_{sc}. On half of the P_c individuals we will apply hill climbing and it will known as P_{hc} in which two parent programs will be selected and their fitness will be evaluated. Crossover operator will then select a crossover point for swapping after which swapping of two sub trees will be done to generate new offspring. After new offspring generation, their fitness will be evaluated. Offspring are allowed to enter the next generation only if their fitness value is greater than their parents otherwise discarded. This process will be continued until we get better individuals or termination criteria satisfied. Since by applying this method we will get better offspring.
   On rest of individuals we will apply standard crossover and it will known as P_{sc} which picks a random crossover point in each of two parent program trees and swaps the two sub trees to generate new offspring. By doing this, diversity will be achieved and we will get different variety of solutions. and by applying standard crossover on half of the individuals we will get diverse and variety of solutions and our probability of entering into local minima will become very less. By using this technique, we will reach to our solution very faster as well as we will get better results because offspring which are better than their parents are transferring to next generation this will result into elimination of problem of bloat.

2. Intelligent mutation operator- After applying intelligent crossover operator on P_c individuals we will apply point mutation operator on P_m individuals. In mutation, single offspring is generated from single parent. Here also we will divide Pm individuals into two parts P_{hm} and P_{sm}. On half of the Pm individuals we will apply hill climbing technique*(P_{hm})* where individual is selected and its fitness value is evaluated then function node of selected individual is randomly selected and replaced by a new randomly generated node which results into new offspring then fitness value of offspring is evaluated. Offspring is allowed to enter the next generation only if its fitness value is greater than its parent otherwise discarded.
   On rest of the Pm individuals we will apply standard point mutation(P_{sm}) where a function node of selected individual_is randomly selected and replaced by a new randomly generated function node which will results into new offspring. By this we will get several offspring mutated by several selected individuals. This will be done for achieving diversity. By applying both standard mutation and hill climbing mutation we will get better offspring because their fitness value is greater than its parents as well as we will achieve diversity so this will cause elimination of destructive nature of genetic operations.

ALGORITHM

1:Input WBC training data and GP parameters as shown in the table.

2:Output A Classifier for diagnosing breast cancer.

3:Begin

4:Initial Population Generate initial population of size k with input data.

5:while Number of fitness evaluations < Maximum number of fitness evaluation. **do**

6:Fitness Evaluation Calculate the fitness value of all individuals .

7:Reproduction Select top P_r individuals to be transferred to the next generation.

8:Crossover Apply intelligent crossover on Pc individuals which further will be divided into two parts P_{hc} and P_{sc}. In P_{hc} , hill climbing will be applied on half of individuals. In P_{sc}, standard crossover will be applied on rest of the individuals.

9:for all crossover pairs **do**

10: Repeat till we get better offspring than the parents.

11:Sort and Store Place the top offspring pairs into a table and then sorted according to fitness value.

12:end for

13:Selection Select the better P_c offspring and transfer them to the next generation.

14:Mutation Apply intelligent mutation operator on P_m individuals which like crossover will be divided into two parts P_{hm} and P_{sm}. In P_{hm}, hill climbing mutation will be applied on half of the individuals. In P_{sm}, standard point mutation will be applied on rest of the individuals.

15:for all mutation parents **do**

16: Repeat till we get offspring better than the parents.

17:end for

18:Selection Select top Pm offspring and transfer them to the next generation.

19:end while

20:return Best individuals with greater fitness value will be received which further will use for classification of breast cancer.

21:End

EXPERIMENTAL RESULTS

We have designed a Classifier to demonstrate our results and We have used wincosin breast cancer data set for training and validating our methodology.

A) Data Sets:-

Wincosin Breast Cancer:- we have considered wincosin breast cancer dataset for classification of breast cancer taken from UCI machine learning repository. WBC is basically used to distinguish malignant(cancerous) from benign(non cancerous). WBC consist of 2 classes and 9 features as shown in the table below

Name of Dataset	No of Classes	No of Features
WBC	2	9

WBC consist of 699 instances taken from fine needle aspirates(FNA) of human breast tissue. Out of 699 instances, 16 have missing attribute value so we generally prefer to discard them and consider the remaining 683 samples. Out of these 683 instances, 444 belong to benign class and rest of 239 belong to malignant class. Dataset is represented by ten attributes but out of these ten one attribute represents serial number so only 9 attributes will be considered and its class either benign or malignant correspond to each record. Each attribute is an integer value from 1 to 10. Value 10 indicate the most abnormal size.

Attribute number	Attribute	Values	Mean	Standard Deviation
1.	Clump thickness	1-10	4.44	2.83
2.	Uniformity of cell size	1-10	3.15	3.07
3.	Uniformity of cell shape	1-10	3.22	2.99
4.	Marginal adhesion	1-10	2.83	2.86
5.	Single epithelial cell size	1-10	2.23	2.22
6.	Bare nuclei	1-10	3.54	3.64
7.	Bland chromatin	1-10	3.45	2.45
8.	Normal nucleoli	1-10	2.87	3.05
9.	Mitoses	1-10	1.60	1.73

In the Clump thickness benign cells grouped in monolayer, while malignant cells grouped in multilayer. In Uniformity of cell size/shape cancer cells varies in size and shape. That is why these parameters are important for determining whether the cells are cancerous or not. In Marginal adhesion normal cells will stick together whereas cancer cells will lose its ability so loss of adhesion is a sign of malignancy. In Single epithelial cell size, size relates to uniformity which is mentioned above. Epithelial cells which are large in size or which enlarge may refer malignant cells. The Bare nuclei are the nuclei which are not surrounded by cytoplasm and that is generally seen in benign tumors. The Bland chromatin represents a uniform texture of nucleus seen in benign cells. In cancer cells chromatin cells are usually coarser. The Normal nucleoli are small structures in the nucleus. In normal cells the nucleoli is very small but in cancer cells the nucleoli appears little larger than usual. Mitoses is basically a nuclear division plus cytokines and produce two identical daughter cells during prophase. In this process cell both divides and replicates. Cancer can be detected by counting number of mitoses.

B) Parameters

Intelligent crossover and mutation technique was applied to Wincosin breast cancer(WBC) dataset. Experimentation is carried out on the dataset with the parameters as shown in the table below-

Parameters	**Value**
Probability of crossover(Pc)	50%
Probability of reproduction (Pr)	25%
Probability of mutation(Pm)	25%
Population size(k)	100

Initialization method	Ramped half and half
Initial max depth of tree	6
Initial min depth of tree	3
Function set	+,-,*,/,^, sin, cos, sine and cosine
Terminal set	Feature variables from datasets, floating point constants(0.0,10.0)
Termination criteria	40,000 fitness evaluation

In machine learning algorithm like genetic programming, dataset is divided into two separate sets- a training set and a testing set. To evaluate the generalizability of our method and to compare our method with existing methods we will divide the training and testing data into two different partitions.

1. A standard 50-50 partition methodology where half of the samples are used for training and rest are for testing the classifier.
2. A 10-fold cross validation technique also used to calculate classification accuracy of our approach or method. In this method, dataset is divided into ten blocks of equal size approximately. We will use 90% of data to train our model and rest 10% for testing. This process will be repeated for 10 times with a different data block left out for testing every time so total 100 GP runs are evaluated.

Training-testing partition	Total training records	Benign records	Malignant records	Total testing records	Benign records	Mali-gnant record
50-50	**341**	**222**	**119**	**342**	**222**	**120**
10 fold Cross validation	**615**	**400**	**215**	**68**	**44**	**24**

C) Results

To evaluate performance of our classifier we calculate classification accuracy, specificity, sensitivity, confusion matrix, and ROC curves. Formulation are as follows:

Accuracy:- It can be defined as measure of the ability of classifier to produce accurate diagnosis.

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} * 100$$

"Classification accuracy of our method for 50-50 partition methodology and 10 fold cross validation is 96% and 97% respectively".

Specificity:- It can be defined as measure of the ability of classifier to separate the target class.

$$\textbf{Specificity} = \frac{TN}{TN+FP} * 100$$

“Specificity of our method for 50-50 partition methodology and 10 fold cross validation is 94.1% and 95.8% respectively”.

Sensitivity:- It can be defined as measure of ability of classifier to identify the presence of target class precisely.

$$\textbf{Sensitivity} = \frac{TP}{TN+FN} * 100$$

“Sensitivity of our method for 50-50 partition methodology and 10 fold cross validation is 96.8% and 97.7% respectively”.

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

True positive (TP): An input is detected as a patient with breast cancer, as diagnosed by the expert clinicians.

True negative (TN): An input is detected as normal and also labeled as a healthy person by the expert clinicians.

False positive (FP): An input is detected as a patient with breast cancer, although labeled as a healthy person by the expert clinicians.

_ False negative (FN): An input is detected as normal, although diagnosed by the expert clinicians as having breast cancer.

(c)Confusion Matrix:- It contain information about actual and predicted classifications performed by classifier. To evaluate performance of classifier confusion matrix is used. Following table represents a confusion matrix-

	Predicted positive	**Predicted negative**
Actual positive	True positive	False negative
Actual negative	False positive	True negative

Following table will show confusion matrix of our method for 50-50 partition methodology-

	Predicted positive	**Predicted negative**
Actual positive	215	7
Actual negative	7	113

Following table will show confusion matrix of our method for 10 fold cross validation –

	Predicted positive	**Predicted negative**
Actual positive	43	1
Actual negative	1	23

Table 3. Comparison of Conventional Crossover and FEDS crossover method with Intelligent Crossover and Mutation technique

Partition methodology	Conventional crossover	FEDS crossover	ICMT
50-50	83.64%	85.56%	95.9%
10-fold cross validation	85%	86.3%	97.03%

CONCLUSION

One in every eight women is susceptible to breast cancer, at some point of time in her life. Early detection and effective treatment is the only rescue to reduce breast cancer mortality. Genetic programming approaches in medical domains is increasing rapidly due to the improvement effectiveness. Genetic programming is one of the machine learning algorithm performs classification which is the most essential and important task. Many experiments are performed on medical datasets for example WBC for breast cancer using multiple classifiers which shows good classification accuracy. Genetic programming provides good results but it also faces some problems and the most famous is problem of bloat in which only the depth of tree will increase but not its corresponding fitness. In this paper, we are proposing a new approach to solve the problem of bloat namely intelligent crossover and mutation technique. In this method we are making genetic operations such as crossover and mutation intelligent and name given to them is Intelligent crossover technique and Intelligent mutation technique. In Intelligent crossover technique, we will divide our individuals into two parts. On half of the individuals we will apply standard crossover and On rest of the individuals we will apply hill climbing. Similarly in Intelligent mutation technique , we will divide the individuals into two parts and on half we will apply standard point mutation and on the rest we will apply hill climbing. To demonstrate our approach we have designed a Classifier and presented the results on WBC dataset which has 2 classes and 9 features. After doing experimentations we got results in the form of classification accuracy, sensitivity, specificity and confusion matrix for both 50-50 partition methodology and 10 fold cross validation. And then we had compared our classification accuracy with conventional crossover and FEDS crossover's accuracy. Results provided by ICMT were much better than conventional and FEDS crossover.

From the above results, we conclude that our proposed ICMT model obtains very high accuracy in classifying the WBC breast cancer data. We believe that the ICMT approach can be a very helpful tool to assist the physicians to diagnose the patient or it can be used as a second opinion for their final diagnosis. This research has some limitations as we are working only on numeric data but in future we can work on images and signals as well as we have tested this technique only for medical dataset but we can apply this technique on different classification dataset and can obtain different classification results in various fields.

REFERENCES:

[1] [Sakorafas et al., 2002] : Sakorafas , G . , Krespis , E . & Pavlakis , G .(2002) Risk estimation for
breast cancer development; a clinical perspective . Surgical Oncology Journal, 10(no issuenumber),pp 183-192.
[2] U.S. Cancer Statistics Working Group. United States Cancer Statistics: 1999–2008 Incidence and Mortality Web-based Report. Atlanta (GA): Department of Health and Human Services, Centers for Disease Control and Prevention, and National Cancer Institute; 2012.
[3] J. F. Winkeler and B. Manjunath, "Genetic programming for object detection," Genetic Programming, pp. 330–335, 1997.
[4]Peter A. Whigham, Member IEEE, And Grant Dick, Member IEEE, "Implicitly Controlling Bloat in Genetic Programming," IEEE Transaction on Evolutionary Computation, Vol. 14, NO. 2, APRIL 2010.
[5] M. Oltean and L. Dios¸an, "An autonomous gp-based system for regression and classification problems," Applied Soft Computing, vol. 9, no. 1, pp. 49–60, 2009.
[6]T. Loveard and V. Ciesielski, "Representing classification problems in genetic programming," in Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, vol. 2. IEEE, 2001, pp. 1070–1077.
[7] Z. Michalewicz, Genetic algorithms+ data structures= evolution programs. springer, 1996.

[8] J. R. Koza, Genetic Programming: vol. 1, On the programming of computers by means of natural selection. MIT press, 1992, vol. 1.
[9] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler, "Multiobjective genetic programming: Reducing bloat using SPEA2," in *Proc. 2001 Congr. Evol. Comput. (CEC '01)*, Piscataway, NJ: IEEE Press, 2001, pp. 536–543.
[10] C. Ryan, "Pygmies and civil servants," *Advances in Genetic Programming*, K. E. Kinnear, Jr., Ed. MIT Press, 1994, ch. 11, pp. 243-263.
[11] R. Poli, "A simple but theoretically-motivated method to control bloat in genetic programming," in *Proc. Genet. Programming (EuroGP '03)*, vol. 2610. Essex: Springer-Verlag, Apr. 14–16, 2003, pp. 204–217.
[12] M. J. Streeter, "The root causes of code growth in genetic programming," in *Proc. Genet. Programming (EuroGP '03)*, vol. 2610. Essex: Springer- Verlag, Apr. 14–16, 2003, pp. 443–454.
[13] N. Wagner and Z. Michalewicz, "Genetic programming with efficient population control for financial time series prediction," in *Proc. Genet. Evol. Comput. Conf. Late Breaking Papers*, 2001, pp. 458–462.
[14] W. B. Langdon and R. Poli, "Fitness causes bloat: Mutation,"in Proc.Theory Application Evolutionary Comput. (ET'97), London, U.K.:University College London, 1997, pp. 59–77.
[15] S. Luke and L. Panait, "A comparison of bloat control methods for genetic programming," *Evol. Comput.*, vol. 14, no. 3, pp. 309–344,2006.
[16] H. Stringer and A. Wu, "Bloat is unnatural: An analysis of changes invariable chromosome length absent selection pressure," Univ. Central Florida, Tech. Rep. CS-TR-04-01, 2004.
[17] S. Bleuler, M. Brack,L. Thiele, and E.Zitzler, "Multiobjective Genetic Programming : Reducing bloat using SPEA2," in Proc. 2001 Congr .Evol. Comput. (CEC '01), Piscataway, NJ: IEEE Press, 2001, pp.536–543.
[18]F. Fernandez, G. G. Gil, J. A. Gomez, and J. L. Guisado, "Control of bloat in genetic programming by means of the island model," in *Proc. Parallel Problem Solving Nature (PPSN VIII)*, vol. 3242. Birmingham, U.K.: Springer-Verlag, 18–22 Sep. 2004, pp. 263–271.
[19] F. Fernandez, G. Galeano, and J. A. Gomez, "Comparing synchronous and asynchronous parallel and distributed GP models," in *Proc. 5th Eur. Conf. Genet. Programming (EuroGP '02)*, vol. 2278. Kinsale, Ireland: Springer-Verlag, Apr. 3–5, 2002, pp. 326–335.
[20]A. Purohit, A. Bhardwaj, A. Tiwari, and N. S. Choudhari, "Removing code bloating in crossover operation in genetic programming," in Recent Trends in Information Technology (ICRTIT), 2011 International Conference on. IEEE, 2011, pp. 1126–1130.
[21] T. Soule and J. A. Foster, "Effects of code growth and parsimony pressure on populations in genetic programming," Evolutionary Computation, vol. 6, no. 4, pp. 293–309, 1998.