Random Sampling Technique for Overfitting Control in Genetic Programming

Ivo Gonçalves[1], Sara Silva[2,1], Joana B. Melo[3], and João M.B. Carreiras[3]

[1] ECOS/CISUC, DEI/FCTUC, University of Coimbra, Portugal
[2] INESC-ID Lisboa, IST, Technical University of Lisbon, Portugal
[3] GeoDES, Tropical Research Institute (IICT), Lisbon, Portugal
icpg@dei.uc.pt, sara@kdbio.inesc-id.pt,
joana.lx.bm@gmail.com, jmbcarreiras@iict.pt

Abstract. One of the areas of Genetic Programming (GP) that, in comparison to other Machine Learning methods, has seen fewer research efforts is that of generalization. Generalization is the ability of a solution to perform well on unseen cases. It is one of the most important goals of any Machine Learning method, although in GP only recently has this issue started to receive more attention. In this work we perform a comparative analysis of a particularly interesting configuration of the Random Sampling Technique (RST) against the Standard GP approach. Experiments are conducted on three multidimensional symbolic regression real world datasets, the first two on the pharmacokinetics domain and the third one on the forestry domain. The results show that the RST decreases overfitting on all datasets. This technique also improves testing fitness on two of the three datasets. Furthermore, it does so while producing considerably smaller and less complex solutions. We discuss the possible reasons for the good performance of the RST, as well as its possible limitations.

Keywords: Genetic programming, Overfitting, Generalization.

1 Introduction

Genetic Programming (GP) is an evolutionary computation paradigm that automatically solves problems without needing to know the structure of the solution in advance [1]. One of the areas in GP that has been recently recognized as an open issue that needs to be addressed in order for GP to realize its full potential is the one of generalization [2]. Generalization is the ability of a solution to perform well on unseen cases. Achieving good generalization is one of the most important goals of any Machine Learning (ML) method such as GP. Overfitting is said to occur when a solution performs well on the training cases but poorly on the testing cases. This indicates that the underlying relationships of the whole data were not learned, and instead a set of relationships existing only on the training cases were learned, but these have no correspondence over the whole known cases.

A. Moraglio et al. (Eds.): EuroGP 2012, LNCS 7244, pp. 218–229, 2012.

Other non-evolutionary ML methods have dedicated a far larger amount of research effort to generalization than GP, although the number of publications dealing with overfitting in GP has been increasing in the past few years. Notably, in Koza [3] most of the problems presented did not use separate training and testing data sets, so performance was never evaluated on unseen cases [4]. Part of the lack of generalization efforts can be related to another issue occurring in GP - bloat. Bloat can be defined as an excess of code growth without a corresponding improvement in fitness [5]. This phenomenon occurs in GP as in most other progressive search techniques based on discrete variable-length representations. Bloat was one of the main areas of research in GP, not only because its occurrence hindered the search progress but also because it was hypothesized, in light of theories such as Occam's razor and the Minimum Description Length, that a reduced code size could lead to better generalization ability. Researchers had a common agreement that these two issues were related and that counteracting bloat would lead to positive effects on generalization. This, however, has been recently challenged. Contributions show that bloat free GP systems can still overfit, while highly bloated solutions may generalize well [6]. This leads to the conclusion that bloat and overfitting are in most part two independent phenomena. In light of this finding, new approaches to improve GP generalization ability are in need, particularly those not based on merely biasing the search toward shorter solutions.

In this work we study the potential of a simple technique, the Random Sampling Technique (RST), to control overfitting in hard real world applications. Recently used with success on a simple benchmark problem [31], the RST is based on the idea of never using the entire training set in any given generation of the search process.

Section 2 reviews the state of the art of generalization in GP. Section 3 describes the RST and the experimental settings. Section 4 presents and discusses the results, advancing ideas for future work. Section 5 concludes.

2 State of the Art

The most common approaches to reducing overfitting in GP are those based on biasing the search toward shorter solutions. Becker and Seshadri [7] proposed adding a complexity penalty factor to the fitness function. Mahler et al. [8] explored to what extent Tarpeian bloat control affects GP generalization ability. Gagné et al. [9] tested the application of parsimony pressure. Cavaretta and Chellapilla [10] used a low-complexity-bias algorithm that uses a modification in the fitness function meant to penalize larger individuals. Zhang et al. [11] also addressed the relationship between size and generalization performance by using a fitness function with two components: fitting error and size.

More recent approaches bias the search process toward less complex solutions. In these approaches complexity is not simply defined as solution size. Vladislavleva et al. [12] proposed a complexity measure called order of nonlinearity. This measure adopts the notion of the minimal degree of the best-fit polynomial, approximating an analytical function with a certain precision. The main objective

behind the proposed complexity measure is to favor smooth and extrapolative behavior of the response surface and to discourage highly nonlinear behavior, which is unstable toward minor changes in inputs and is dangerous for extrapolation. Vanneschi et al. [13] proposed a functional complexity measure based on the classic mathematical concept of curvature. Informally, the curvature of a function can be defined as the amount by which its geometric representation deviates from being straight. This complexity measure expresses the complexity of a function by counting the number of different slopes.

Also recently, approaches based on similarities between solutions have started to appear. Uy et al. [15] proposed a Semantic Similarity based Crossover approach which is based on the Sampling Semantics Distance between two trees (or subtrees), which is calculated by choosing N random points (fitness cases) and calculating the mean absolute difference between each corresponding points on the two trees. The authors argue that the exchange of subtrees is most likely to be beneficial if the two subtrees are not too similar or too dissimilar. Vanneschi and Gustafson [16] proposed avoiding solutions similar to already known overfitted solutions. The proposed method (repGP) keeps a list of overfitting individuals (called repulsors) and prevents any new individual to enter the next generation if they are similar to any of the known repulsors.

Various other approaches were proposed. A simple and elegant idea was proposed by Da Costa and Landry [17]. The idea is to relax the training set by allowing a wider definition of the desired solution which translates into considering not only the desired output y correct but allow a more broader range to be considered, i.e. allow any output in the range $[ymin, ymax]$. Chan et al. [18] proposed a statistical method called Backward Elimination that works by eliminating insignificant terms in polynomials models such as those produced by GP. Nikolaev et al. [19] proposed several techniques to balance the statistical bias and variance. In the context of financial applications, Chen and Kuo [20] proposed a measure of degree of overfitting based on the extracted signal ratio. Foreman and Evett [21] proposed Canary Functions, where the idea is to measure overfitting during the run by using a validation set. When overfitting starts to occur the search process is stopped. Vanneschi et al. [22] argued that using GP with a multi-optimization approach can enhance the generalization ability of the resulting solutions. This approach uses two other criteria besides the traditional sum of errors. These are: the correlation between outputs and targets (to maximize) and the diversity of pairwise distances between outputs and targets (to minimize). Robilliard and Fonlupt [23] applied a method called Backwarding that goes back as much as needed in the evolution process until the point that overfitting is not yet very relevant. This is achieved by saving two copies of the solutions: one copy for the best solution on the training set and another copy for the best solution on the validation set. At the end of the GP process the best saved solution for the validation set is returned. Finally, in the context of the Compiling Genetic Programming System, Banzhaf et al. [24] showed the positive influence of the mutation operator in generalization ability.

Although these and a number of other works have addressed the issue of overfitting in GP, they appear as a set of isolated efforts scattered along the years and among different applications. Nevertheless, as GP matures and slowly becomes a mainstream ML approach, also the overfitting issue is slowly becoming a central research subject.

3 Experiments

This section describes the datasets used, the Random Sampling Technique and the experimental parameters used.

3.1 Datasets

Experiments are conducted on three multidimensional symbolic regression real world datasets, the first two on the pharmacokinetics domain and the third one on the forestry domain.

Toxicity. The goal of this application is to predict, in the context of a drug discovery study, the median lethal dose (LD50) of a set of candidate drug compounds on the basis of their molecular structure. LD50 refers to the amount of compound required to kill 50% of the considered test organisms (cavies). Reliably predicting this and other pharmacokinetics parameters would permit to reduce the risk of late stage research failures in drug discovery, and enable to decrease the number of experiments and cavies used in pharmacological research [25]. The LD50 dataset consists of 234 instances, where each instance is a vector of 626 molecular descriptor values identifying a drug. This dataset is freely available at `http://personal.disco.unimib.it/Vanneschi/toxicity.txt`.

Plasma Protein Binding. As in the toxicity application, also here the goal is to predict the value of a pharmacokinetics parameter of a set of candidate drug compounds on the basis of their molecular structure, this time the plasma protein binding level (%PPB). %PPB quantifies the percentage of the initial drug dose that reaches the blood circulation and binds to the proteins of plasma. This measure is fundamental for good pharmacokinetics, both because blood circulation is the major vehicle of drug distribution into human body and since only free (unbound) drugs can permeate the membranes reaching their targets [25]. This dataset consists of 131 instances, where each instance is a vector of 626 molecular descriptor values identifying a drug.

Biomass. The objective of this application is to estimate forest above-ground biomass (AGB) as a function of several metrics derived from synthetic aperture radar (SAR) data acquired by sensors on orbital platforms. Mapping and understanding the spatial distribution of forest AGB is an important and challenging task [26,27,28]. As it relates to the carbon stocks of a given ecosystem, these maps can be used to monitor forests and capture national deforestation processes, forest degradation, and the effects of conservation actions, sustainable management and enhancement of carbon stocks. The dataset is composed of 112 field measurements of forest AGB and corresponding 8 SAR metrics used to model AGB. This dataset has never been used in any GP studies.

3.2 Random Sampling Technique

The Random Sampling Technique (RST) has been previously used to improve the speed of a GP run [29], however in [30] it was used to reduce overfitting in a classification task in the context of software quality assessment. In the RST, the training set is never entirely used in the search process. Instead, at each generation, a random subset of the training data is chosen and evolution is performed taking into account the fitness of the solutions in this subset. This implies that only individuals that perform well on various different subsets will remain in the population. It is expected that, since these surviving individuals perform reasonably well on different subsets, they have captured the underlying relationships of the data instead of overfitting it. This work is a continuation of a previous work [31] that was done with the RST on a simple benchmark problem. In the mentioned work we have proposed a more flexible approach to the RST. Firstly, the size of the random subset can be defined as any percentage of the training set. Secondly, the rate at which a new random subset is chosen can be defined as either being at each N generations or as a percentage of the total number of generations. These two RST parameters are respectively labelled as Random Subset Size (RSS) and Random Subset Reset (RSR). In this extended approach they can be defined as any value, as opposed to their static nature in [30]. In this paper we build upon the previous work by exploring the RST on real world datasets with the best configuration found in that work (both RSS and RSR set to value 1, i.e. in each generation a single new random sample is chosen). Standard GP is used as the baseline for comparison.

3.3 Parameters and Statistical Tests

The experimental parameters used are provided in Table 3.3. Furthermore, crossover and mutation points are selected with uniform probability. Fitness is calculated as the Root Mean Squared Error (RMSE) between outputs and targets.

Statistical significance of the null hypothesis of no difference was determined with pairwise Kruskal-Wallis non-parametric ANOVAs at p=0.05. A non-parametric ANOVA was used because the data is not guaranteed to follow a normal distribution. For the same reason, the median was preferred over the mean in all the evolution plots shown in the next section. The median is also more robust to outliers.

4 Results and Discussion

This section presents and discusses the results achieved. For the remainder of this paper, the terms training and testing fitness are to be interpreted in the following way: training fitness is the fitness of the best individual in the training set; testing fitness is the fitness of that same individual in the testing set. For the purpose of further comparisons we have defined a simple overfitting measure. According

Table 1. GP parameters used in the experiments

Runs	30
Population	500
Generations	100
Training - Testing division	70% - 30%
Crossover operator	Standard subtree crossover, probability 0.9
Mutation operator	Standard subtree mutation, probability 0.1, new branch maximum depth 6
Tree initialization	Ramped Half-and-Half [1], maximum depth 6
Function set	+, -, *, and /, protected as in [1]
Terminal set	Input variables, no constants
Selection for reproduction	Lexicographic Parsimony Pressure [32], tournaments of size 10
Elitism	Replication rate 0.1, best individual always survives
Maximum tree depth	17

to this measure, overfitting is simply calculated as the difference between testing and training fitness. This implies that if training fitness is better than testing fitness then overfitting is occurring, i.e. the measure retrieves a positive value. Negative values are allowed and occur when testing fitness is better than training fitness. This is rather uncommon but can still happen. In the evolution plots we have chosen not to take the absolute value of the difference between both fitness values as not to lose this potentially interesting information. For the statistical tests we take the absolute values since what we want is overfitting as close to zero as possible and not to have negative overfitting. This also prevents the unwanted effect of compensation between positive and negative overfitting values. Tree size is calculated as the number of nodes of a solution. Complexity is calculated based on the notion of curvature of a function as in [13]. The evolution plots present the results based on the median of the fitness, overfitting, tree size and complexity of the best individuals at each generation over 30 runs.

4.1 Results

The fitness and overfitting plots for Standard GP and RST can be found in Figure 1. The corresponding tree size and complexity plots can be found in Figure 2. The statistical results comparing training and testing fitness, overfitting, tree size and complexity between both techniques can be found in Table 4.1. In this table, s+ indicates that the RST is statistically better than Standard GP, while s- indicates the opposite, and $\sim$ indicates that no statistically significant difference was found. As we can see in the table, the RST achieves statistically better testing fitness on LD50 and on %PPB. On AGB the difference is not statistically significant. In training fitness, Standard GP is statistically better on

Table 2. Statistical results

	LD50	**%PPB**	**AGB**
Training Fitness	s-	s-	s-
	0.000000	0.000000	0.000000
Testing Fitness	s+	s+	∼
	0.000572	0.000058	0.208871
Overfitting	s+	s+	s+
	0.000095	0.000000	0.000001
Tree Size	s+	s+	s+
	0.000000	0.000002	0.000000
Complexity	s+	s+	s+
	0.000000	0.000001	0.000000

all datasets. In overfitting the RST achieves statistically lower overfitting than Standard GP on all datasets. The same statistical significance happens for tree size and complexity also on all datasets.

Looking at the evolution plots in Figure 1 we can see, across all the datasets, a constantly widening gap between training and testing fitness in Standard GP, which means that overfitting is occurring. The widening of this gap is particularly fast on the LD50 dataset. This gap is what our overfitting measure represents. We can see that the overfitting increase is steeper on the LD50 dataset, with AGB being the second most overfitted dataset and %PPB the least overfitted of the three, although still with the referred widening gap present. RST, however, can maintain a much smaller gap between training and testing fitness. This gap is also much more constant across all generations. LD50 is the dataset where the gap is smaller and in fact very close to zero. %PPB comes next in regard to overfitting and ABG is comparatively the most overfitted of the datasets with the RST.

Comparing the training fitness values of the Standard GP with the ones of the RST, we can see that Standard GP learns the training data much faster. This was expected, since in Standard GP each generation is allowed to see many more fitness cases than RST, that sees only one. However, the testing fitness values reveal that Standard GP is merely overfitting and not actually learning the underlying relationships of the whole known instances, a fact that becomes even clearer when looking at the corresponding overfitting plots. Despite being a slow learner, RST compensates in terms of overfitting as it maintains considerably lower values than Standard GP.

Besides achieving better testing fitness in two out of three datasets, and less overfitting in all datasets, the RST does so while producing smaller and less complex solutions. We can see from the tree size and complexity plots in Figure 2 that these present a similar behavior to the overfitting plots, i.e. RST presents rather constant values while Standard GP presents a somewhat constant increase in all values (overfitting, tree size and complexity). We have not yet explored

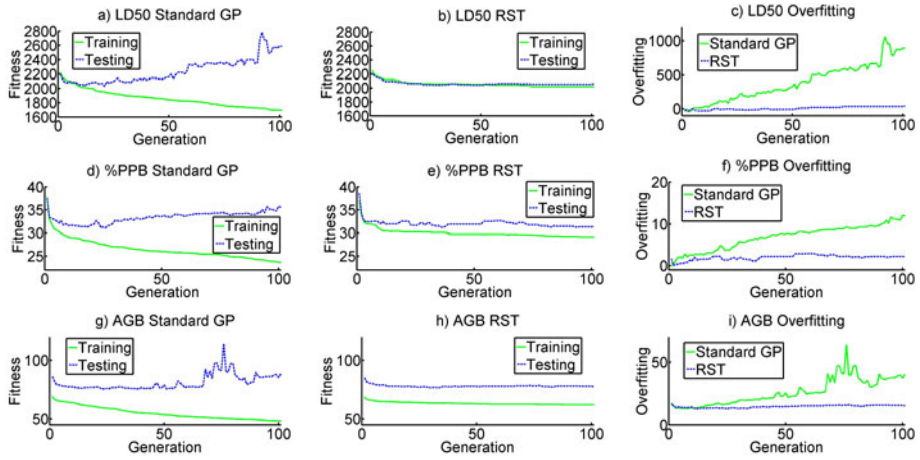

Fig. 1. Standard GP and RST fitness and overfitting evolution plots for: LD50 a) through c), %PPB d) through f) and AGB g) through i)

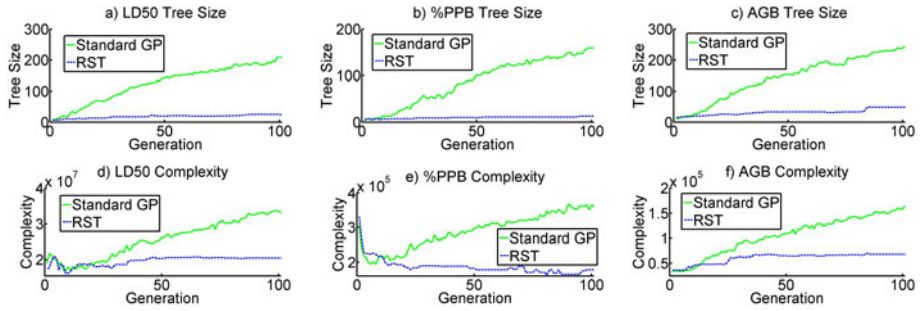

Fig. 2. Tree size and complexity evolution plots for: LD50 a) and d), %PPB b) and e) and AGB c) and f)

the relationship between these three measurements, or between these and the amount of bloat. At least one study has attempted it in the LD50 dataset [14] but did not reach solid conclusions. From the practical point of view, it is sufficient for now to state that the RST reveals clear advantages in the three real world applications studied here.

4.2 Discussion

One important point that the RST made us realize is that, in order to overfit, a solution does not necessarily need to find a set of relationships that occur across most of the fitness cases in the training set. We believe that, in most cases, the solutions that overfit are simply overfitting a few instances of the training set, but this is enough to achieve lower error (better fitness) than a solution that learns

the general trend of the training data without ever learning specific fitness cases so well. A solution that perfectly maps (i.e. has zero error) the relationship of a small number of fitness cases can have a much better fitness, thus higher chance of survival, than other solutions not as much overfitted. This can be one of the roots of the problem, since these overfitted, but low error solutions will likely remain in the population for a long time. On the other hand, RST has shown us that by using only one fitness case at the time and changing it frequently we can ultimately avoid this pitfall of the traditional approach.

Figure 3 exemplifies the abovementioned situation. In this figure we have a target and three possible solutions. In both examples solutions 2 and 3 have lower RMSE error than solution 1. However, although shifted up from the target, solution 1 represents the pattern of the target more precisely than solutions 2 or 3 and thus is more desirable to keep in the population. We can see that solutions 2 and 3 present a much higher risk of overfitting. However, in Standard GP they would be preferred over solution 1, consequently filling up the population with many similar solutions presenting the same overfitting risks. In RST, solution 1 is not necessarily discarded when compared to solutions 2 and 3, as the chosen instance for the fitness calculation in each generation can be any, and hence an instance where solutions 2 and 3 fail to fit can be chosen. This results in keeping solution 1 in the population and possibly allowing the search to find other more general solutions. In fact, a preliminary exploration of the population characteristics of both Standard GP and RST, has revealed that the genotypical diversity is generally higher with RST (with statistical significance), in particular among the best ranked individuals of the population. Additional observations have also revealed that the solutions produced by the RST tend to be smoother than the ones produced by Standard GP, much like the examples presented in Figure 3. Further investigation of these themes may help us develop better methods to control overfitting in the future.

Another issue that deserves further attention is that of the amount of search that is allowed to the RST when compared to Standard GP. One of the most interesting facts observed in the results is the evolution of the RST testing and training fitness in all datasets. These show that the RST is able to continually improve training fitness while not degrading the testing fitness. The improvement of its training fitness is much smaller than the improvement achieved by Standard GP, but the fact that the testing fitness and the overfitting values are better compensate for this fact. These results hint that the RST is indeed learning the underlying relationships of the whole known data (testing and training sets) instead of simply overfitting the training data. Even in the final generations the overfitting of the RST is close to zero. However, care must be taken when drawing conclusions, since it is not clear whether these low overfitting values can be kept if the runs are allowed to continue for more generations. We have not performed such experiments yet, but these will bring further insight on the full potential of the RST. If indeed the RST is learning the underlying relationships of the whole data, then similar overfitting values should occur in the extended runs. If not, we may have to conclude that the RST is only delaying the occurrence of overfitting,

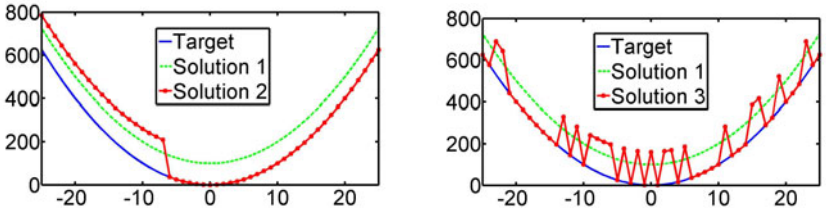

Fig. 3. A possible target and three different candidate solutions

simply because it sees very little of the training set in each generation. Note that, in a run of only 100 generations, it is not even possible for RST (with the settings used, RSS=1) to see all the instances of the training set in the LD50 dataset.

5 Conclusions

In this work we have studied the potential of the RST to control overfitting in hard real world applications. In the RST, the training set is never entirely used in the search process. Instead, at each generation, a random subset of the training data is chosen and evolution is performed taking into account the fitness of the solutions in this subset. In two multidimensional symbolic regression problem from the pharmacokinetics domain and one from the forestry domain, the RST was able to continually improve training fitness while not degrading the testing fitness, resulting in much lower overfitting values than the ones observed for Standard GP. Furthermore, the RST did so while producing considerably smaller and less complex solutions. From these facts we were able to claim that the RST, with the settings used in this work, reveals clear advantages in the three real world applications studied here.

Generalization has only recently been recognized as an important open issue of GP. Another contribution of this work is a brief summary of the published literature on this subject, which until now has been made of a series of isolated efforts scattered along the years and among different applications. While discussing the possible reasons for the good performance of the RST, as well as its possible limitations, we have advanced a few ideas for future research that we hope may help to build new methods to control overfitting in the future, thus contributing to the advancement of the state of the art of this subject.

Acknowledgments. This work was partially supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds. The authors acknowledge projects EnviGP (PTDC/EIA-CCO/103363/2008, funded by FCT, Portugal) and CarboVeg GB (funded by the Ministry of Environment, Portugal), and also the Secretary of State of the Environment and Sustainable Development (SEAD, Guinea-Bissau), and Maria José Vasconcelos, Project PI, IICT, Lisbon (Portugal).

References

1. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to genetic programming (With contributions by J.R. Koza) (2008), http://lulu.com, http://www.gp-field-guide.org.uk
2. O'Neill, M., Vanneschi, L., Gustafson, S., Banzhaf, W.: Open Issues in Genetic Programming. Genetic Programming and Evolvable Machines 11, 339–363 (2010)
3. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press (1992)
4. Kushchu, I.: An Evaluation of Evolutionary Generalisation in Genetic Programming. Artificial Intelligence Review 18, 3–14 (2002)
5. Silva, S., Costa, E.: Dynamic Limits for Bloat Control in Genetic Programming and a review of past and current bloat theories. Genetic Programming and Evolvable Machines 10(2), 141–179 (2009)
6. Vanneschi, L., Silva, S.: Using Operator Equalisation for Prediction of Drug Toxicity with Genetic Programming. In: Lopes, L.S., Lau, N., Mariano, P., Rocha, L.M. (eds.) EPIA 2009. LNCS, vol. 5816, pp. 65–76. Springer, Heidelberg (2009)
7. Becker, L.A., Seshadri, M.: Comprehensibility and Overfitting Avoidance in Genetic Programming for Technical Trading Rules. Technical report, Worcester Polytechnic Institute (2003)
8. Mahler, S., Robilliard, D., Fonlupt, C.: Tarpeian Bloat Control and Generalization Accuracy. In: Keijzer, M., Tettamanzi, A.G.B., Collet, P., van Hemert, J., Tomassini, M. (eds.) EuroGP 2005. LNCS, vol. 3447, pp. 203–214. Springer, Heidelberg (2005)
9. Gagné, C., Schoenauer, M., Parizeau, M., Tomassini, M.: Genetic Programming, Validation Sets, and Parsimony Pressure. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 109–120. Springer, Heidelberg (2006)
10. Cavaretta, M.J., Chellapilla, K.: Data Mining using Genetic Programming: The implications of parsimony on generalization error. In: Proceedings of the 1999 IEEE Congress on Evolutionary Computation, pp. 1330–1337. IEEE Press (1999)
11. Zhang, B.-T., Mühlenbein, H.: Balancing Accuracy and Parsimony in Genetic Programming. Evolutionary Computation 3(1), 17–38 (1995)
12. Vladislavleva, E.J., Smits, G.F., den Hertog, D.: Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming. IEEE Transactions on Evolutionary Computation 13(2), 333–349 (2009)
13. Vanneschi, L., Castelli, M., Silva, S.: Measuring Bloat, Overfitting and Functional Complexity in Genetic Programming. In: Proceedings of GECCO 2010, pp. 877–884. ACM Press (2010)
14. Trujillo, L., Silva, S., Legrand, P., Vanneschi, L.: An Empirical Study of Functional Complexity as an Indicator of Overfitting in Genetic Programming. In: Silva, S., Foster, J.A., Nicolau, M., Machado, P., Giacobini, M. (eds.) EuroGP 2011. LNCS, vol. 6621, pp. 262–273. Springer, Heidelberg (2011)
15. Uy, N.Q., Hien, N.T., Hoai, N.X., O'Neill, M.: Improving the Generalisation Ability of Genetic Programming with Semantic Similarity based Crossover. In: Esparcia-Alcázar, A.I., Ekárt, A., Silva, S., Dignum, S., Uyar, A.Ş. (eds.) EuroGP 2010. LNCS, vol. 6021, pp. 184–195. Springer, Heidelberg (2010)
16. Vanneschi, L., Gustafson, S.: Using Crossover Based Similarity Measure to Improve Genetic Programming Generalization Ability. In: Proceedings of GECCO 2009, pp. 1139–1146. ACM Press (2009)
17. Da Costa, L.E., Landry, J.-A.: Relaxed Genetic Programming. In: Proceedings of GECCO 2006, pp. 937–938. ACM Press (2006)

18. Chan, K.Y., Kwong, C.K., Chang, E.: Reducing Overfitting in Manufacturing Process Modeling using a Backward Elimination Based Genetic Programming. Applied Soft Computing 11(2), 1648–1656 (2011)
19. Nikolaev, N., de Menezes, L.M., Iba, H.: Overfitting Avoidance in Genetic Programming of Polynomials. In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, pp. 1209–1214. IEEE Press (2002)
20. Chen, S.-H., Kuo, T.-W.: Overfitting or Poor Learning: A Critique of Current Financial Applications of GP. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 34–46. Springer, Heidelberg (2003)
21. Foreman, N., Evett, M.: Preventing overfitting in GP with canary functions. In: Proceedings of GECCO 2005, pp. 1779–1780. ACM Press (2005)
22. Vanneschi, L., Rochat, D., Tomassini, M.: Multi-optimization improves genetic programming generalization ability. In: Proceedings of GECCO 2007, p. 1759. ACM Press (2007)
23. Robilliard, D., Fonlupt, C.: Backwarding: An Overfitting Control for Genetic Programming in a Remote Sensing Application. In: Collet, P., Fonlupt, C., Hao, J.-K., Lutton, E., Schoenauer, M. (eds.) EA 2001. LNCS, vol. 2310, pp. 245–254. Springer, Heidelberg (2002)
24. Banzhaf, W., Francone, F.D., Nordin, P.: The Effect of Extensive Use of the Mutation Operator on Generalization in Genetic Programming using Sparse Data Sets. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 300–309. Springer, Heidelberg (1996)
25. Archetti, F., Messina, E., Lanzeni, S., Vanneschi, L.: Genetic programming for computational pharmacokinetics in drug discovery and development. Genetic Programming and Evolvable Machines 8(4), 17–26 (2007)
26. Baccini, A., Laporte, N., Goetz, S.J., Sun, M., Dong, H.: A first map of tropical Africa's above-ground biomass derived from satellite imagery. Environmental Research Letters 3, 045011 (2008)
27. Lucas, R., Armston, J., Fairfax, R., Fensham, R., Accad, A., Carreiras, J., Kelley, J., Bunting, P., Clewley, D., Bray, S., Metcalfe, D., Dwyer, J., Bowen, M., Eyre, T., Laidlaw, M., Shimada, M.: An Evaluation of the ALOS PALSAR L-Band Backscatter-Above Ground Biomass Relationship Queensland, Australia: Impacts of Surface Moisture Condition and Vegetation Structure. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 3(4), 576–593 (2010)
28. Saatchi, S.S., Harris, N.L., Brown, S., Lefsky, M., Mitchard, E.T.A., Salas, W., Zutta, B.R., Buermann, W., Lewis, S.L., Hagen, S., Petrova, S., White, L., Silman, M., Morel, A.: Benchmark map of forest carbon stocks in tropical regions across three continents. Proceedings of the National Academy of Sciences 108(24), 9899–9904 (2011)
29. Gathercole, C., Ross, P.: Dynamic Training Subset Selection for Supervised Learning in Genetic Programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 312–321. Springer, Heidelberg (1994)
30. Liu, Y., Khoshgoftaar, T.: Reducing Overfitting in Genetic Programming Models for Software Quality Classification. In: Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering, pp. 56–65. IEEE Press (2004)
31. Gonçalves, I., Silva, S.: Experiments on Controlling Overfitting in Genetic Programming. In: 15th Portuguese Conference on Artificial Intelligence (to appear)
32. Luke, S., Panait, L.: Lexicographic parsimony pressure. In: Proceedings of GECCO 2002, pp. 829–836. Morgan Kaufmann (2002)