Balancing Learning and Overfitting in Genetic Programming with Interleaved Sampling of Training Data

Abstract. Generalization is the ability of a model to perform well on cases not seen during the training phase. In Genetic Programming generalization has recently been recognized as an important open issue, and increased efforts are being made towards evolving models that do not overfit. In this work we expand on recent developments that showed that using a small and frequently changing subset of the training data is effective in reducing overfitting and improving generalization. Particularly, we build upon the idea of randomly choosing a single training instance at each generation and balance it with periodically using all training data. The motivation for this approach is based on trying to keep overfitting low (represented by using a single training instance) and still presenting enough information so that a general pattern can be found (represented by using all training data). We propose two approaches called interleaved sampling and random interleaved sampling that respectively represent doing this balancing in a deterministic or a probabilistic way. Experiments are conducted on three high-dimensional real-life datasets on the pharmacokinetics domain. Results show that most of the variants of the proposed approaches are able to consistently improve generalization and reduce overfitting when compared to standard Genetic Programming. The best variants are even able of such improvements on a dataset where a recent and representative state-of-the-art method could not. Furthermore, the resulting models are short and hence easier to interpret, an important achievement from the applications' point of view.

Keywords: Genetic Programming, Overfitting, Generalization, Pharmacokinetics, Drug Discovery

1 Introduction

Genetic Programming (GP) [1] is now a mature technique that routinely produces results that have been characterized as human-competitive [2]. However, a few open issues remain, one of them being the lack of generalization, or overfitting, of the evolved models [3]. It is said to occur when a model performs well on the training cases but poorly on unseen 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. Notably, in Koza [4] most of the problems presented did not use separate training and testing datasets, so performance was never evaluated on unseen cases [5]. Other non-evolutionary machine

learning methods have dedicated a 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. For a review of the state-of-the-art in avoiding overfitting in GP the reader is referred to [6].

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 [7]. 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, on the same problem, bloat free GP systems can still overfit, while highly bloated solutions may generalize well [8]. 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 needed, particularly ones not based on merely biasing the search towards shorter solutions.

In this work we build on recent developments in this domain. We explore how we can balance keeping overfitting low and sti ll reaching models with general patterns. We do that by interleaving the usage of the training data between a single instance and all the instances. This approach is inspired by a state-of-the-art method to control overfitting called Random Sampling Technique (RST) [6]. In order to experimentally validate our approach, we apply it to hard high-dimensional problems in the field of pharmacokinetics, comparing the results with the ones obtained by standard GP and by RST. The three problems addressed are the prediction of median lethal dose, protein-plasma binding levels, and human oral bioavailability of medical drugs [9]. Section 2 describes the proposed approaches and the experiments conducted. Section 3 presents and discusses the results and section 4 concludes.

2 Approaches and Experiments

This section describes the motivation, the proposed approaches, the experimental parameters and the datasets used.

2.1 Motivation

Using a varying subset of the training data was previously shown to have positive effects. In [10] it was shown that this type of approach could reduce the speed of a GP run and still achieve similar results to the standard GP approach of using all training data in a static manner. In a particular configuration, it was even possible to improve generalization. In [11] the usage of a varying subset of the training data was shown to reduce overfitting in a software quality classification

task. [10] used between 10% and 15% of the total training data depending on the variant, while [11] used 50%. More recently, even smaller percentages of the total training data were shown to be able to reduce overfitting and improve generalization. In particular, even using only a single training instance and changing it every generation was shown to be able to achieve these same outcomes. This was shown in [6] in high-dimensional symbolic regression real-life datasets, as well as in artificial datasets in [12] and [13]. In [6] besides the reduced overfitting and improved generalization, it was also shown that the evolved solutions were smaller than those from standard GP.

In this work, we are mainly interested in the idea of choosing the subset of the training data randomly. This kind of approach is called Random Sampling Technique (RST) or Random Subset Selection (RSS). Here, we will use the term RST. Particularly, we build upon the idea of using a single randomly chosen training instance at each generation and balance it with periodically using all the training data. The motivation for this approach is based on trying to keep overfitting low (represented by using a single training instance) and still presenting enough information so that a general pattern can be found (represented by using all training data). We propose two approaches called interleaved sampling and random interleaved sampling that respectively represent doing this balancing in a deterministic or a probabilistic way.

2.2 Interleaved Sampling

This approach is based on deterministically interleaving between using one or all training instances. We propose three variants respectively naming them: interleaved, interleaved single and interleaved all. The first variant is based on using all training instances in the first generation, then changing to a single training instance in the next generation and proceeding with the same interleaving for the remaining generations. As such, and provided that the number of generations is even, this variant always evolves half of the generations with all training instances and the other half with a single instance. The interleaved single variant is based on giving preference to using a single training instance and can consequently be understood as interleaving with a bias towards a single training instance. A parameter is added in order to define how many generations using a single training instance are conducted for each generation where all training instances were used. The values tested for this parameter were 5%, 10%, 15%, 20% and 25%, where each value represents the percentage over the total number of generations. Conversely, the interleaved all variant is based on giving preference to using all training instances. The parameter for this variant is similar to the previous, and in this case defines how many generations using all training instances are conducted for each generation where a single training instance was used. The values tested for this parameter are the same as in the interleaved single variant.

2.3 Random Interleaved Sampling

This approach is based on probabilistically interleaving between using a single or all training instances. At each generation the decision of how many training instances to use is taken. The probability of using a single training instance is given as a parameter. The values tested for this parameter were 5%, 25%, 50%, 75% and 95%. It should be noted that using 100% as a parameter would be equivalent to the RST using a single training instance and changing it every generation. Similarly, using 0% as a parameter would be equivalent to the standard GP approach of always using all training data.

2.4 Parameters and Datasets

The experimental parameters used are provided in Table 2.4. Furthermore, crossover and mutation points are selected with uniform probability. Fitness is calculated as the Root Mean Squared Error between predicted and expected outputs. Statistical significance of the null hypothesis of no difference was determined with Mann-Whitney U tests at $p = 0.05$. Standard GP and RST 1/1 are used as baselines for comparison. Standard GP uses all the training data at every generation. RST 1/1 is also used as a baseline because it is a representative state-of-the-art method, as recently shown in [6]. It works by randomly choosing a new single training instance at each generation. For each dataset 30 different random partitions are used. Each method uses the same 30 partitions.

Table 1. GP parameters used in the experiments

Runs	30
Population	500
Generations	200
Training - Testing division	50% - 50%
Crossover operator	Standard subtree crossover, probability 0.9
Mutation operator	Point mutation, probability 0.1, mutation probability per node 0.05
Tree initialization	Ramped Half-and-Half, maximum depth 6
Function set	+, -, *, and /, protected as in [1]
Terminal set	Input variables, constants -1.0, -0.5, 0.0, 0.5 and 1.0
Selection for reproduction	Tournament selection of size 10
Elitism	Best individual always survives
Maximum tree depth	17

Experiments are conducted on three multidimensional symbolic regression real-life datasets, all of which on the pharmacokinetics domain. They have already been used in GP studies (e.g. [9]).

Toxicity. The goal of this application is to predict, in the context of a drug discovery study, the median lethal dose (represented as 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 [9]. The LD50 dataset consists of 234 instances, where each instance is a vector of 627 elements (626 molecular descriptor values identifying a drug, followed by the known LD50 for that drug). This dataset is freely available at `http://personal.disco.unimib.it/Vanneschi/toxicity.txt`. We will refer to this dataset as LD50.

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. Protein-plasma binding level (represented as %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 [9]. The %PPB dataset consists of 131 instances, where each instance is a vector of 627 elements (626 molecular descriptor values identifying a drug, followed by the known %PPB for that drug). We will refer to this dataset as PPB.

Bioavailability. In this dataset the pharmacokinetics parameter to predict is the human oral bioavailability. Human oral bioavailability (represented as %F) is the parameter that measures the percentage of the initial orally submitted drug dose that effectively reaches the systemic blood circulation after passing through the liver. Being able to reliably predict the %F value for a potential new drug is outstandingly important, given that the majority of failures in compounds development from the early nineties to nowadays are due to a wrong prediction of this pharmacokinetic parameter during the drug discovery process [14, 15]. The %F dataset consists of 359 instances, where each instance is a vector of 242 elements (241 molecular descriptor values identifying a drug, followed by the known value of %F for that drug). This dataset is freely available from the GP Benchmarks website, gpbenchmarks.org. We will refer to this dataset as Bio.

3 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 considered the overfitting measure described in [6]. According to this measure, overfitting is simply calculated as the absolute value of the difference between testing and training fitness. This measure is associated with the intuitive notion that overfitting is related to the discrepancy between the performance of a model on the data seen during the training phase and the unseen data. Tree size is calculated as the number of nodes of a solution. The evolution plots present the results based on the median of the fitness, overfitting, tree size and tree depth of the best individuals in the training data at each generation over 30 runs. These plots can be found in figures 1, 2 and 3.

3.1 Interleaved Single and Interleaved All Variants

The interleaved single and the interleaved all variants are not shown in the evolution plots as they are very similar to, respectively, RST 1/1 and Standard GP. These similarities apply regardless of the parameterization.

For the interleaved single, statistical results confirm that this variant is superior in terms of overfitting reduction, across all datasets, to standard GP, being also superior in testing fitness on the LD50 and the PPB datasets. There is no statistically significant difference in terms of testing fitness on the Bio dataset. The comparisons between the RST 1/1 and standard GP reach the same conclusions. Therefore, the interleaved single variant, in these tested parameterizations, can be seen as equivalent to the RST 1/1. It seems that the effect of presenting all training data with this periodicity to the algorithm is negligible. In terms of tree size and tree depth, the interleaved single variant produces smaller and shallower trees when compared to standard GP. These results are also statistically significant across all datasets.

The interleaved all variant produced similar results to standard GP, across all datasets, in terms of training and testing fitness and overfitting. The statistical results show that there are almost no statistically significant differences between these methods and standard GP. The only statistically significant differences in testing fitness and overfitting occurred on the Bio dataset where standard GP is superior in both measures when compared to parameterizations 15% and 25%. From these results we conclude that providing a bias towards using all training data and periodically using a single instance is not an effective approach of improving generalization and reducing overfitting.

3.2 Interleaved and Random Interleaved Variants

As we can see from the evolution plots, the random interleaved approach has the expected behavior in regard to its parameterization. The closer the parameter is to 100%, the closer the method behaves as the RST 1/1. Conversely, the closer the parameter is to 0%, the closer the method behaves as the standard GP approach. Statistical results confirm that the 5% parameterization is very similar to

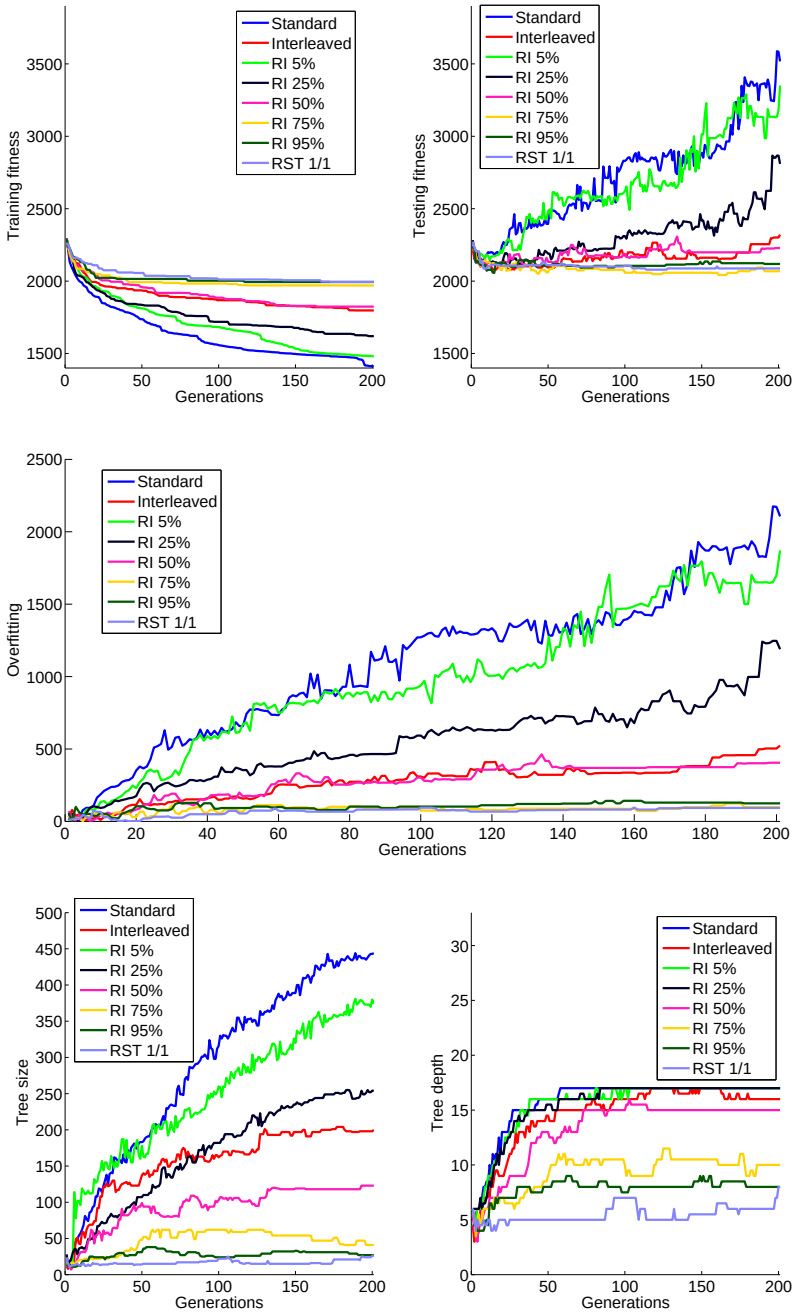

Fig. 1. Training fitness, testing fitness, overfitting, tree size and tree depth evolution plots for: Standard GP, Interleaved, Random Interleaved (RI) 5% 25% 50% 75% 95% and RST 1/1 on the LD50 dataset

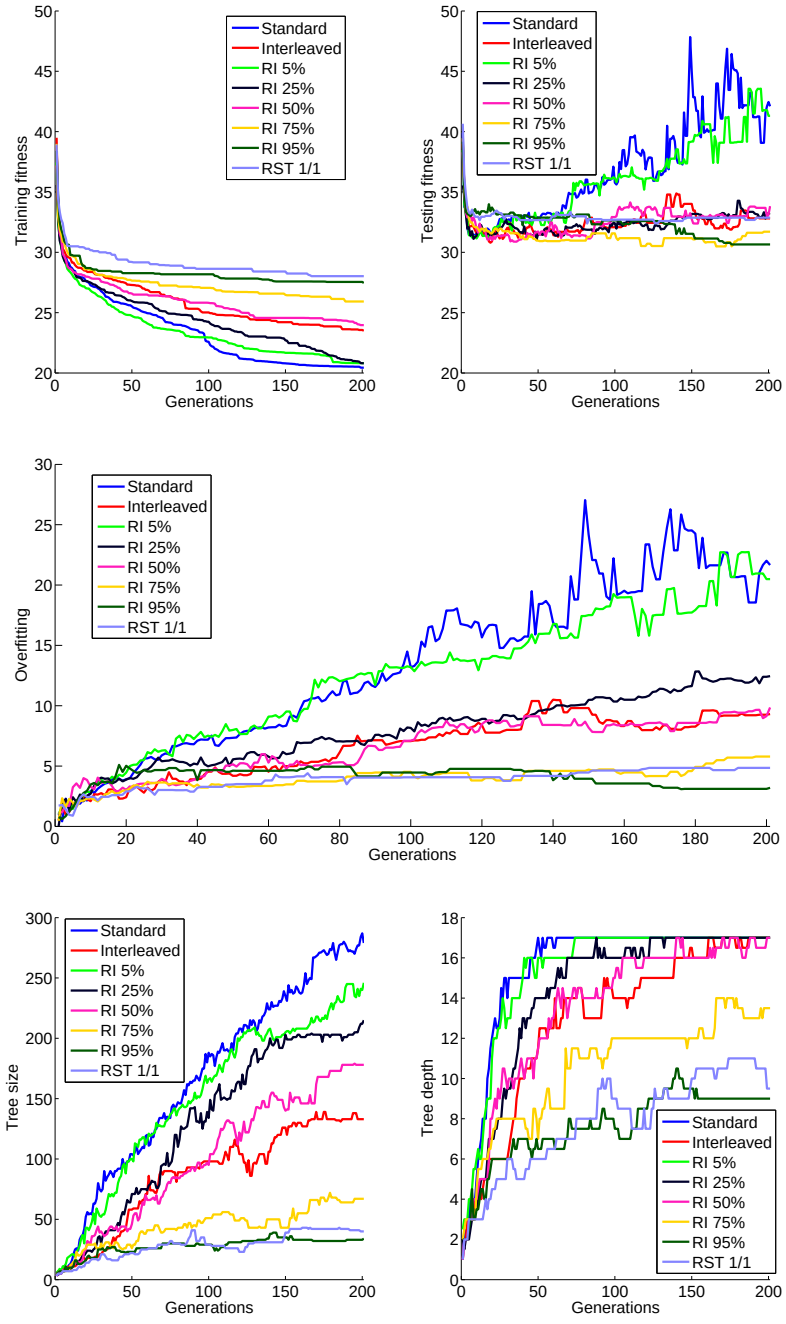

Fig. 2. Training fitness, testing fitness, overfitting, tree size and tree depth evolution plots for: Standard GP, Interleaved, Random Interleaved (RI) 5% 25% 50% 75% 95% and RST 1/1 on the PPB dataset

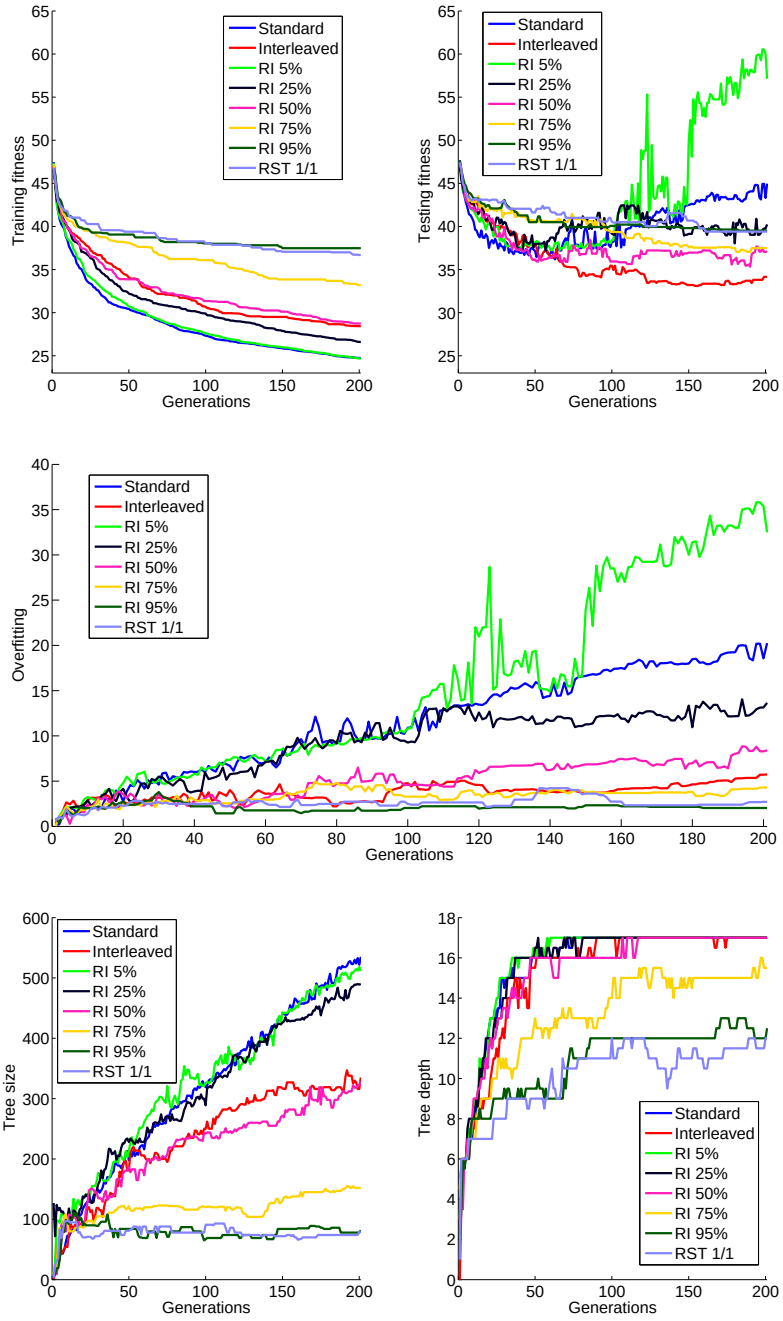

Fig. 3. Training fitness, testing fitness, overfitting, tree size and tree depth evolution plots for: Standard GP, Interleaved, Random Interleaved (RI) 5% 25% 50% 75% 95% and RST 1/1 on the Bio dataset

the standard GP approach and it is unable to improve generalization and reduce overfitting. All other parameterizations are able, with statistical significance, to improve generalization and reduce overfitting over the standard GP approach on the LD50 and PPB datasets. However, only the 50% and the 75% parameterizations can achieve an increase in generalization on the Bio dataset. The RST 1/1 is unable to achieve this same statistically significant result on this dataset. This shows that depending on the dataset, different probabilities of choosing a single training instance may be helpful. Nevertheless, from the results we can see that the most promising area for looking for a good parameterization to a given dataset revolves around the 50% parameterization. The interleaved results are similar to the random interleaved 50%, having also the same statistical significance over the standard GP approach. This was somewhat expected since both methods use on average the same number of generations with a single training instance. As we can see from the RST 1/1 results on the Bio dataset, although it is able to avoid overfitting, it also presents a slow learning of both training and testing data. In comparison, interleaved and random interleaved 50% and 75% are able to increase the rate of learning of the training data while also improving testing fitness. In terms of tree size and tree depth, the interleaved variant and the random interleaved variant with 50%, 75% and 95% parameterizations, produce smaller and shallower trees when compared to standard GP. These results are also statistically significant across all datasets.

3.3 Final Remarks

Overall, and across all the datasets, the methods that showed to be more consistent were: interleaved and random interleaved 50% and 75%. These three methods showed to be superior to standard GP in terms of reducing overfitting and improving generalization. Furthermore, they have also improved generalization where the RST 1/1 and the interleaved single methods could not: the Bio dataset. This dataset showed to be the most difficult of the three in terms of improving the testing fitness over standard GP. These facts allow us to conclude that these three methods are superior to standard GP and more robust than the RST 1/1 approach and hence contribute to an incremental improvement of the state of the art in this field.

From the point of view of the applications, the fact that these methods also produce relatively short models is a major advantage. At the end of the run, random interleaved 75% provides models with median size around 50 for the LD50 and PPB problems, and around 150 for the Bio problem. These are very short models when we consider the dimensionality of the data (626 features for LD50 and PPB, 241 for Bio). For the Bio problem this size is similar to the sizes obtained with the very successful bloat control technique Operator Equalisation (OpEq) [16]. For LD50 it is actually better, i.e. lower, than the sizes obtained by OpEq [17]. For PPB, to our knowledge no results are reported in the literature for the median tree size of the best individual.

4 Conclusions

In this work we expanded on recent developments in terms of overfitting reduction and generalization improvement. These developments have showed that using a small and frequently changing subset of the training data is effective in reducing overfitting and improving generalization. Particularly, we have built upon the idea of using a single randomly chosen training instance at each generation and balance it with periodically using all training data. The motivation for this approach is based on trying to keep overfitting low (represented by using a single training instance) and still presenting enough information so that a general pattern can be found (represented by using all training data). We have proposed two approaches called interleaved sampling and random interleaved sampling that respectively represent doing this balancing in a deterministic or a probabilistic way. Experiments were conducted in three high-dimensional real-life problems on the pharmacokinetics domain. The results have shown that most of the proposed approaches were able to consistently improve generalization and reduce overfitting when compared to the standard GP approach. In particular, three methods have shown these improvements even on a dataset where a state-of-the-art technique failed. These results were confirmed as being statistically significant. From the point of view of the applications, the winning methods have the additional advantage of producing relatively short models, hence easier to interpret.

In conclusion, we have found that both the deterministic and the probabilistic approach of balancing the usage of training data were helpful in improving generalization and reducing overfitting. We have also found that, in most cases, and in order to achieve these improvements, a preference has to be given towards using only a single training instance. The prevalence of this preference is dependent on the dataset but, in general, using a single training instance in more or less half of the generations is enough.

References

1. R. Poli, W.B. Langdon and N.F. McPhee (2008). A field guide to genetic programming, `http://lulu.com`, `http://www.gp-field-guide.org.uk`, (With contributions by J.R. Koza)
2. J. Koza (2010). Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines 11(3/4): 251–284
3. M. O'Neill, L. Vanneschi, S. Gustafson and W. Banzhaf (2010). Open Issues in Genetic Programming. Genetic Programming and Evolvable Machines 11(3/4): 339-363
4. J. Koza (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press
5. I. Kushchu (2002). An Evaluation of Evolutionary Generalisation in Genetic Programming. Artificial Intelligence Review 18: 3-14
6. I. Gonçalves, S. Silva, J. B. Melo, and J. M. B. Carreiras (2012). Random Sampling Technique for Overfitting Control in Genetic Programming. In 15th European Conference on Genetic Programming (EuroGP 2012). Springer, April 2012.

7. S. Silva and E. Costa (2009). 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
8. L. Vanneschi and S. Silva (2009). Using Operator Equalisation for Prediction of Drug Toxicity with Genetic Programming. In Proceedings of EPIA 2009, 65-76. Springer
9. F. Archetti, E. Messina, S. Lanzeni, and L. Vanneschi (2007). Genetic programming for computational pharmacokinetics in drug discovery and development. Genetic Programming and Evolvable Machines 8(4): 17–26
10. C. Gathercole and P. Ross (1994). Dynamic Training Subset Selection for Supervised Learning in Genetic Programming. In Proceedings of PPSN III, 312–321. Springer
11. Y. Liu and T. Khoshgoftaar (2004). Reducing Overfitting in Genetic Programming Models for Software Quality Classification. In Proceedings of the Eighth IEEE International Symposium on High Assurance Systems Engineering, 56–65. IEEE Press
12. W. B. Langdon (2011). Minimising Testing in Genetic Programming. Technical report, RN/11/10, Computer Science, University College London, Gower Street, London WC1E 6BT, UK, 2011
13. I. Gonçalves and S. Silva (2011). Experiments on Controlling Overfitting in Genetic Programming. 15th Portuguese Conference on Artificial Intelligence (EPIA 2011), Oct. 2011
14. I. Kola and J. Landis (2004). Can the pharmaceutical industry reduce attrition rates? Nat Rev Drug Discov, 3(8):711-716.
15. T. Kennedy (1997). Managing the drug discovery/development interface. Drug Discovery Today, 2(10):436-444.
16. S. Silva and L. Vanneschi (2012). Bloat free Genetic Programming: application to human oral bioavailability prediction. Int. J. Data Mining and Bioinformatics, 6(6):585–601.
17. L. Vanneschi and S. Silva (2009). Using Operator Equalisation for Prediction of Drug Toxicity with Genetic Programming. In Proceedings of EPIA 2009, 65-76. Springer.