Statistical Genetic Programming: The Role of Diversity

Maryam Amir Haeri, Mohammad Mehdi Ebadzadeh and Gianluigi Folino

Abstract In this chapter, a new GP-based algorithm is proposed. The algorithm, named SGP (Statistical GP), exploits statistical information, i.e. mean, variance and correlation-based operators, in order to improve the GP performance. SGP incorporates new genetic operators, i.e. Correlation Based Mutation, Correlation Based Crossover, and Variance Based Editing, to drive the search process towards fitter and shorter solutions. Furthermore, this work investigates the correlation between diversity and fitness in SGP, both in terms of phenotypic and genotypic diversity. First experiments conducted on four symbolic regression problems illustrate the goodness of the approach and permits to verify the different behavior of SGP in comparison with standard GP from the point of view of the diversity and its correlation with the fitness.

1 Introduction

Maintaining diversity in the genetic programming is important, because it helps to prevent the GP process from a premature convergence. The lack of diversity may lead to convergence towards local optima or towards a not optimal behavior in dynamic environments. Therefore, experimental analysis of diversity can give us a better perspective about the population transition and the search process in GP.

M. Amir Haeri (✉) · M. M. Ebadzadeh
Department of Computer Engineering and Information Technology,
Amirkabir University of Technology, Tehran, Iran
e-mail: haeri@aut.ac.ir

M. M. Ebadzadeh
e-mail: ebadzadeh@aut.ac.ir

G. Folino
ICAR-CNR, Rende, Italy
e-mail: folino@icar.cnr.it

V. Snášel et al. (eds.), *Soft Computing in Industrial Applications*,
Advances in Intelligent Systems and Computing 223, DOI: 10.1007/978-3-319-00930-8_4,

According to this, diversity in genetic programming is studied by many researchers working in the GP field. Some of them tried to define appropriate phenotypic or genotypic diversity measures. Rosca [9, 10] suggested a phenotypic measure based on the number of different fitness values in the population. Analogously, Langdon [7] defined genotypic diversity as the number of different structures in the population. Some of the genotypic diversity measures have been defined on the basis of the edit distance between structures in the GP population [2, 3].

Folino et al. [5] analyzed the effectiveness of parallel genetic programming models in maintaining diversity in a population, i.e. island and cellular GP, using phenotypic and genotypic entropy. Their study confirms that the considered parallel models help to promote diversity but the authors conclude no relation between diversity measures and goodness of the fitness can be obtained. Jackson [6] investigated the effects of mutation operator on enhancing the diversity in GP population. He reported that the role of mutation operator in enhancing the diversity depends on the nature of the problem. In three of his test problems mutation did not have a significant effect on any diversity measures, while in one case, mutation operator had a strong influence on improving the structural diversity.

Burke et al. [1] analyzed different types of diversity measures and investigated the importance of these measures and their correlation with the fitness in genetic programming. Their results demonstrate that there is a correlation between fitness and diversity. In particular, a positive correlation between the phenotypic diversity and the fitness and a negative correlation between the genotypic diversity and the fitness were observed in many problems. However, they concluded that this correlation must not be interpreted as a factor of causality, i.e. "...higher diversity does not necessarily cause better performance, but better performance is seen with higher diversity." Finally, in regression problems, they discovered the weakest values of correlation and that is one of the reasons why we decided to explore more deeply the behavior in terms of diversity in this kind of problems.

In recent years, both analyzing diversity and correlation and improving genetic programming has become the focus of many researchers. Among the desired properties, a GP-based algorithm should reduce the code growth and efficiently explore the huge search space of real hard problems considered.

To this aim, a new GP algorithm, named Statistical Genetic Programming (SGP) is introduced in this chapter. The novelty of the method is based on the exploitation of statistical information obtained in the structure of the individuals and in the building of new powerful genetic operators. SGP introduces three new operators, Correlation Based Crossover, Correlation Based Mutation and Variance Based Editing. The effect of these three operators is to decrease the rate of the code growth, while maintaining efficacy in exploring the search space. SGP is particularly apt to cope with symbolic regression problems; however we would like to remark that the algorithm can be also used for other kinds of problems, if the function associated to a node can be computed as a function of the input variables. It will be clearer in the next section. To study the behavior of the search process in SGP, and in regression problems in particular, the population diversity and its correlation with the fitness is analyzed, using phenotypic and genotypic measure of diversity.

The rest of the chapter is organized as follows: In the Sect. 2, Statistical Genetic Programming is introduced. Section 3 presents the diversity measures used in this chapter. Section 4 is devoted to the description of the test problems and to the experimental results. Section 5 concludes the chapter.

2 Statistical Genetic Programming

In this section, a new GP algorithm named *Statistical Genetic Programming* (SGP) is introduced. The SGP utilizes statistical information to improve the performance of the standard GP. Before introducing the operators of SGP, firstly, it should be clarified what we mean by the statistical information of a GP tree.

2.1 Statistical Information of a GP Tree

Statistical information in a GP tree can be exploited in order to drive the evolutionary process in the case in which each node in the GP tree is a function of the input variables, i.e. in symbolic regression problems.

The SGP algorithm computes, for each node of all its subtrees, the following values: $E[g_i] = \frac{1}{M}\sum_{j=1}^{M} g_i(X_j)$, $E[g_i^2] = \frac{1}{M}\sum_{j=1}^{M} g_i^2(X_j)$ and $E[g_i.y] = \frac{1}{M}\sum_{j=1}^{M} y_j g_i(X_j)$, where $g_i(X_j)$ is the function of node i. $X_j = (x_{j1}, x_{j2}, ..., x_{jn})$ is the vector of input variables and n is the number of variables. M is the number of training data and $y_j = f(X_j)$ is the value of the (to be estimated, in the following named regression) function f at the point X_j. In order to compute these values the mean (m) and variance (σ^2) of each node and the correlation coefficient (ρ) of each node with f can be computed as follows:

$$m = E[g_i] \tag{1}$$

$$\sigma^2 = E[g_i^2] - E[g_i]^2 \tag{2}$$

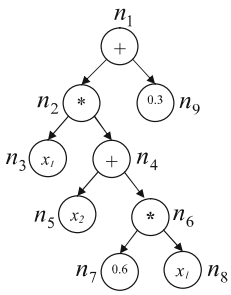
$$\rho = \frac{E[g_i.y] - E[g_i]E[y]}{\sigma_{g_i}\sigma_y} \tag{3}$$

An example of a GP tree and its statistical information is shown in Fig. 1. Suppose that we have a symbolic regression problem with regression data (fitness cases) as shown in Fig. 1, and let the depicted tree represent an individual of the GP population. Each node of the tree implies some function; for instance, the function of

node n_6 is $g_6 = 0.6 * x_1$. The "tree function" is the function implied by the root of the tree. Based on the regression data, one can compute statistical information—mean or variance—for each node of the tree (i.e. the function implied by it). For instance, for node n_4 ($g_4 = x_2 + x_1 * 0.6$) and the given regression data, the mean of output of g_4 is equal to $E[g_4] = 0.68$. All relevant data are tabulated in Fig. 1. Another useful statistical information is the correlation coefficient of the outputs of each node function with the desired output values of f (Regression function). This measure can indicate the relation between the function of each node and desired function, and shows how much a subtree is effective in constructing the desired function.

Regression Data		
x_1	x_2	$f(x_1,x_2)$
0.1	0.3	1.38
0.4	0.5	1.93
0.2	0.2	1.42
0.8	0.6	3.42
0	0.9	1.65

Node	Function (g_i)
n_1	$g_1(x_1,x_2) = x_1(x_2 + 0.6x_1) + 0.3$
n_2	$g_2(x_1,x_2) = x_1(x_2 + 0.6x_1)$
n_3	$g_3(x_1,x_2) = x_1$
n_4	$g_4(x_1,x_2) = x_2 + 0.6x_1$
n_5	$g_5(x_1,x_2) = x_2$
n_6	$g_6(x_1,x_2) = 0.6x_1$
n_7	$g_8(x_1,x_2) = 0.6$
n_8	$g_8(x_1,x_2) = x_1$
n_9	$g_8(x_1,x_2) = 0.3$



x_1	x_2	$g_1(x_1,x_2)$	$g_2(x_1,x_2)$	$g_3(x_1,x_2)$	$g_4(x_1,x_2)$	$g_5(x_1,x_2)$	$g_6(x_1,x_2)$	$g_7(x_1,x_2)$	$g_8(x_1,x_2)$	$g_9(x_1,x_2)$
0.1	0.3	0.336	0.036	0.1	0.36	0.3	0.06	0.6	0.1	0.3
0.4	0.5	0.596	0.296	0.4	0.74	0.5	0.24	0.6	0.4	0.3
0.2	0.2	0.364	0.064	0.2	0.32	0.2	0.12	0.6	0.2	0.3
0.8	0.6	1.164	0.864	0.8	1.08	0.6	0.48	0.6	0.8	0.3
0	0.9	0.3	0	0	0.9	0.9	0	0.6	0	0.3

Node	Statistics		
	$m = E[g_i]$	$\sigma^2 = E[g_i^2] - E[g_i]^2$	$\rho = \frac{E[g_i f] - E[g_i]E[f]}{\sigma_{g_i}\sigma_f}$
n_1	0.552	0.130496	0.980357
n_2	0.252	0.130496	0.980357
n_3	0.3	0.1	0.926148
n_4	0.68	0.111	0.793819
n_5	0.5	0.075	0.324068
n_6	0.18	0.036	0.926148
n_7	0.6	0	0
n_8	0.3	0.1	0.926148
n_9	0.3	0	0

Fig. 1 An example of a GP tree and its statistical information

Although SGP computes some additional information during the evolution, these computations do not load considerable overheads and they are not very time consuming. Because most of the statistical information is reusable, and those need updating can be computed simultaneity and in parallel with updating the fitnesses.

In practice, SGP uses statistical information of the population to drive the search process. SGP has three operators that use this information: (1) Correlation Based Crossover, Correlation Based Mutation and Variance Based Editing, described in detail in the next subsections.

2.2 Correlation Based Crossover

In the standard crossover, two individuals are selected using a particular selection method and, from each of the parent trees, a subtree is randomly selected and swapped with the subtree of the other parent.

In correlation based crossover (CB crossover), for each parent, the subtrees that are more correlated to the regression function f (i.e. the ones having the maximum value of the correlation coefficient between the subtree root and the regression function f, using the absolute value) have more chance to be selected as swapping subtree. As in tournament selection, the subtrees of each parent compete with each other based on their absolute value of the correlation coefficient with f. The winner subtree of each parent is replaced with the winner of the other parent. The tournament size is proportional to the tree size. On the basis of experimental tries, the tournament size was set from 10 to 20 % of the tree size.

Using this kind of crossover, the nodes, which are more correlated to f, have more chance to be selected as crossover points; so it is more likely that the crossover points are located in the most effective parts of the parent trees. Therefore, the probability of neutral crossover, i.e. is a crossover that results in generating offspring that is not different from its parents, is decreased. Furthermore, more effective subtrees are selected as a swapping genetic material and this should lead to the relocation of valuable subtrees in the population and increase the probability of constructive crossover (crossover generating an offspring that is fitter than its parents).

2.3 Correlation Based Mutation

In the standard mutation, after that an individual is selected, one of its subtrees, randomly selected is replaced by a new random subtree. In CB mutation, the subtrees of the selected individual that are less correlated to the regression function f are more likely to be chosen as the point of mutation. In practice, the probability of choosing each node for mutation is inversely proportional to its absolute correlation. The subtree corresponding to the chosen node is replaced by a random

subtree. Unlike the standard mutation, CB mutation selects the mutant subtree *non*-uniformly at random. If a subtree has less correlation (considering the absolute value) with f, it has less influence in constructing the solution tree. Thus, changing this subtree may be productive.

2.4 Variance Based Editing

One of the problems of the GP is code bloat, i.e. producing code which is slower and larger, without a significant improvement in terms of fitness. More precisely, code bloat is a considerable increase in the average code size of the population with no significant change of the fitness. In this work, we use a method based on the editing of the tree in order to perform bloat reduction. In practice, variance and mean of each node are used to edit the trees. Every subtree of each GP individual whose variance of its root is zero is replaced with the mean of its root.

Most of the subtrees of GP trees are introns or just for constructing a numeric constant. The variance of these subtrees is equal to zero. Thus, this editing operator can restrict the code growth significantly.

3 Diversity in Genetic Programming

One of the objectives of this chapter is to try to understand the correlation between the performance of our algorithm and some diversity measures, i.e. the phenotypic and genotypic diversity. This section presents the diversity measures which are used in this chapter.

Phenotypic diversity is related to different fitness values in the population. In this chapter phenotypic entropy is utilized as a phenotypic diversity measure. The phenotypic entropy of the population P can be calculated as follows [9]:

$$H_p(P) = -\sum_{j=1}^{N} p_j \log(p_j)$$

where p_j is the portion ($\frac{n_j}{N}$) of the population P that have fitness j and N is the number of different fitness values in the population P.

As in our case, fitness is a continuous quantity, in order to discretize the fitness values, we used an adaptive procedure, in which the ranges are determined *on the fly*, while the fitness values become known gradually. In practice, for each generation, the first fitness value computed becomes the representative for the first range. Subsequently, we compute the following quantity for each fitness range i:

$$\delta_i = \left| \frac{\text{new fitness value} - \text{avg. fitness in the range } i}{\text{avg. fitness in the range } i} \right|,$$

and if it is less than a predefined threshold τ, we put the new fitness value into that range (in case of ties, the i having minimum δ_i wins). Otherwise, if no such i is found, a new range is created. In the experiments, τ is set to 0.02.

In order to measure the genotypic diversity, the genotypic entropy is used in the chapter. Genotypic diversity is related to the different structures in the population. A tree distance measure is needed to keep into account the different structures. We use the tree edit distance measure, as defined by Ekárt and Németh [4].

The distance between two trees T_1 and T_2 can be computed as follows:

$$dist(T_1, T_2) = \begin{cases} d(a,b) & \text{if neither } T_1 \text{ nor } T_2 \text{ has any children} \\ d(a,b) + K \times \sum_{l=1}^{m} dist(s_l, t_l) & \text{otherwise} \end{cases},$$

where a and b are the roots of T_1 and T_2. T_1 and T_2 have m possible subtrees s and t. The parameter K is set to $1/2$. $d(a,b)$ is 0 if the nodes a and b are equal, 1 if they are different. The edit distance is calculated for each individual against the best individual in the run so far (note that it is different from the best individual in the current population).

As in the case of the phenotypic entropy, genotypic entropy is computed as follows:

$$H_g e(P) = -\sum_{j=1}^{N} ge_j \log(ge_j)$$

where ge_j is the portion of the population that has a given distance from the best individual in the run so far.

4 Experimental Results and Discussion

This section is devoted to assessing the performance of SGP and the effects of the new genetic operators. Specifically, we aim to understand the effect of the new operators on the diversity in the population, using the measure of genotypic and phenotypic diversity, introduced in the previous section.

4.1 Test Problems and GP Parameter Settings

Four real valued symbolic regression problems were chosen in order to perform an experimental evaluation. The benchmark functions were selected from [8, 11]. The benchmark problems are illustrated in Table 1. Each experiment were performed over 30 runs.

Table 1 Test problems

Benchmark number	Benchmark function	Function formula	Domain	Number of instances
Benchmark1	$f_1(x_1, x_2)$	$\frac{(x_1-3)^4+(x_2-3)^3+(x_2-3)}{(x_2-2)^4+10}$	$x_1, x_2 \in [-6, 6]$	50
Benchmark2	$f_2(x_1, x_2)$	$x_1x_2 + \sin((x_1 - 1)(x_2 + 1))$	$x_1, x_2 \in [-3, 3]$	20
Benchmark3	$f_3(x_1, x_2)$	$6\sin(x_1)\cos(x_2)$	$x_1, x_2 \in [-3, 3]$	20
Benchmark4	$f_4(x)$	$x^4 + x^3 + x^2 + x$	$x \in [-1, 1]$	20

Table 2 GP settings

Population size	100
Function set	$F = \{+, -, \times, \div\}$
Fitness function	Mean squared error (MSE)
Initial population method	Ramped half and half
Selection method	Tournament selection
Tournament size	4
Crossover rate	90%
Mutation rate	5%
Maximum tree depth in initial population	6
Maximum tree depth	17
Maximum generation	200
Number of runs	30 independent runs for each test

The function set is the set $F = \{+, -, \times, \div\}$. Note that the $\div$ represents *protected division*. The terminal set consists of random numbers, and of the function variables. Standard GP parameters are used, as shown in Table 2. The fitness function is the Mean Squared Error (MSE).

4.2 Accuracy Evaluation

In order to compare the accuracy of standard GP and SGP, we run GP and SGP for 200 generations using a population of 100 individuals on the above described benchmarks.

Figure 2a shows the result of the comparison. According to the figure, SGP performs better than the standard GP in terms of accuracy. Probably, lower probability of neutral crossover, higher constructive crossover rate and more effective mutation lead SGP to explore the GP search space more properly. Furthermore, variance based editing removes introns and decreases the computational cost by making the tree shorter. This can be seen in Fig. 2b, in which standard and statistical GP are compared in terms of average size of trees in the overall population.

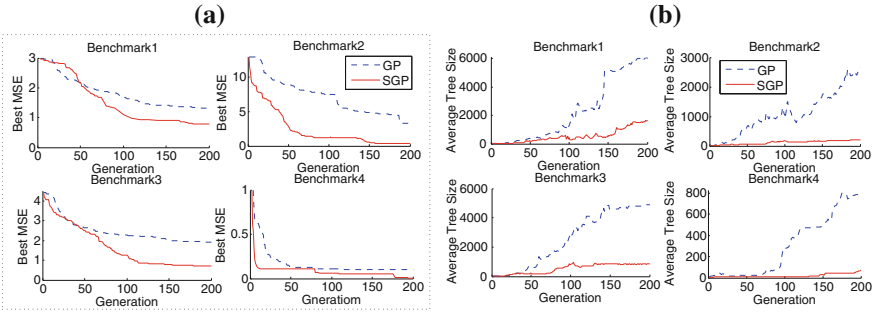


Fig. 2 **a** Comparison of SGP and GP accuracy, **b** Comparison of SGP and GP bloat control

4.3 Diversity and SGP

In this subsection, we want to investigate the correlation between diversity and fitness in SGP, using both the phenotypic and genotypic diversity as defined in the diversity section, and the Spearman correlation, defined later in this section.

Figure 3a shows the phenotypic entropy for SGP in comparison with the standard GP. It can be seen that SGP has a higher phenotypic diversity than the standard GP, probably because the CB crossover operator increases the rate of constructive crossover. In addition to that, CB mutation decreases the rate of ineffective mutation and increases the rate of constructive mutation. Thus, in SGP, the probability of generating offspring, which are better than its parents is higher than in standard GP. Furthermore, VB editing is effective in eliminating the introns and this could help to significantly decrease the rate of neutral genetic operations.

In a sub-optimal tree, higher nodes are more correlated to the regression function f. Hence, in CB crossover the higher nodes of the trees have more chance to be selected as a swapping subtrees. In the measure of genotypic diversity the higher nodes of trees have more influence, because the coefficient K is less than 1 (here is 0.5) . Therefore, as can be seen in Fig. 3b, in SGP the genotypic diversity is higher than in GP.

A second set of experiments aims to answer to the hard question whether populations with higher phenotypic or genotypic diversity can obtain a better solution. In practice, we want to investigate the correlation between the fitness and these measures of diversity.

Similar to [1], Spearman correlation is adopted in order to determinate if a relation between fitness and diversity exists. The Spearman correlation can be defined as $1 - \frac{6\sum_{i=1}^{N} d_i^2}{N^3 - N}$, where d_i is the difference between the rank of the best fitness and the rank of the diversity of population i. The population that have a better fitness (less MSE) have a greater fitness rank. Similarly, the diversity rank is higher for the population having higher diversity.

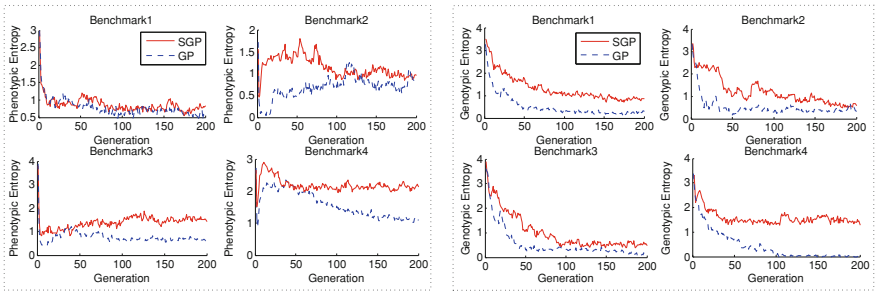


Fig. 3 Phenotypic and genotypic entropy in GP and SGP populations

Figure 4 illustrates the Spearman correlation between the fitness and the phenotypic entropy and the correlation between the fitness and the genotypic entropy. Each point in the graphs depicts the correlation between 30 populations, collected from 30 independent runs in the different phases of the evolutive process.

As can be seen in Fig. 4a, for all benchmarks, at the beginning, as the population is randomly created there is no positive (or negative) correlation between fitness and diversity. Afterwards, a positive correlation can be found both for SGP and GP, then the correlation decreases and no significant correlation can be found. This is probably due, in accordance with the results found in the chapter of Burke, to the presence of many local optima.

In the case of genotypic diversity (see Fig. 4b) in SGP, in very early generations the correlation between the fitness and genotypic entropy is positive. After these early generations, the correlation becomes lower and close to zero and afterward becomes negative. In the experimental results of Burke et al., a similar behavior have been evidenced.

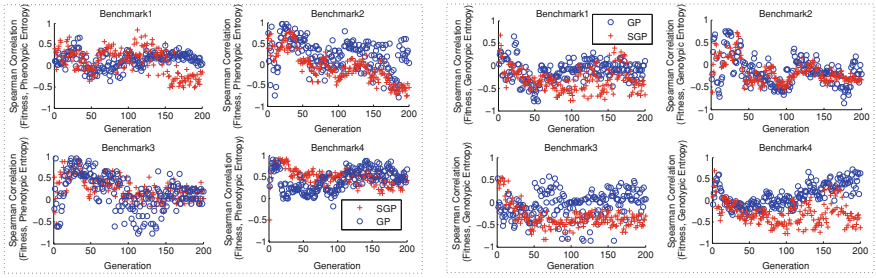


Fig. 4 **a** Correlation between fitness and phenotypic diversity, **b** Correlation between fitness and genotypic diversity

Our investigation results are similar to those of [1]. There is a positive correlation between the phenotypic entropy and fitness and a negative correlation between genotypic diversity and fitness.

It should be considered that, the correlation coefficient represents the association between fitness and diversity, not the causality. This means that, for example, higher phenotypic diversity is not necessarily the cause of better fitness. However, better performance is observed with higher phenotypic diversity. Burke et al. [1] expressed that crossover and selection methods have very important roles in constructing the structures of GP population. Any simple implementation difference may change the diversity of the population. Therefore, care must be taken when inferring causality from diversity.

5 Conclusions

This chapter proposed a new GP paradigm, Statistical Genetic Programming, exploiting the statistical information of the population in order to improve the accuracy of GP, mainly for symbolic regression problems. Experiments conducted on four symbolic regression problems confirm the improvement obtained using the new paradigm. A diversity analysis, based on genotypic and phenotypic diversity measures and on the study of correlation coefficients obtains results comparable with other classical model of GP, with the exception of the capacity of SGP to maintain a higher genotypic diversity. Future works will be conducted in order to try to understand better the relation between the performance of SGP and diversity and to study the different contributions of the three operators introduced.

References

1. Burke, E., Gustafson, S., Kendall, G.: Diversity in genetic programming: an analysis of measures and correlation with fitness. Trans. Evol. Comp. **8**(1), 47–62 (2004)
2. de Jong, E., Watson, R., Pollack, J.: Reducing bloat and promoting diversity using multi-objective methods (2001)
3. Ekárt, A., Németh, S.: A metric for genetic programs and fitness sharing. In: Genetic Programming, pp. 259–270. Springer, Berlin (2000)
4. Ekárt, A., Nemeth, S.: Maintaining the diversity of genetic programs. In: Genetic Programming, pp. 122–135 (2002)
5. Folino, G., Pizzuti, C., Spezzano, G., Vanneschi, L., Tomassini, M.: Diversity analysis in cellular and multipopulation genetic programming. In: CEC'03, vol. 1, pp. 305–311. IEEE (2003)
6. Jackson, D.: Mutation as a diversity enhancing mechanism in genetic programming. In: Proceedings of GECCO '11, pp. 1371–1378. ACM, New York, (2011)
7. Langdon, W.: Genetic programming and data structures: genetic programming+ data structures. Springer, New York (1998)

8. Pennachin, C.L., Looks, M., de Vasconcelos, J.a.A.: Robust symbolic regression with affine arithmetic. In: Proceedings of GECCO '10, pp. 917–924. ACM, New York (2010)
9. Rosca, J.: Entropy-driven adaptive representation. In: Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications, vol. 9, pp. 23–32. Tahoe City (1995a)
10. Rosca, J.: Genetic programming: exploratory power and the discovery of functions. In: Proceedings of Evolutionary Programming IV, pp. 719–736. Citeseer (1995b)
11. 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. Trans. Evol. Comp. **13**, 333–349 (2009)