

Parsimony Measures in Multi-objective Genetic Programming for Symbolic Regression

Bogdan Burlacu

Josef Ressel Centre for Symbolic Regression
Heuristic and Evolutionary Algorithms Laboratory
University of Applied Sciences Upper Austria
Hagenberg, Austria
bogdan.burlacu@fh-hagenberg.at

Michael Kommenda

Josef Ressel Centre for Symbolic Regression
Heuristic and Evolutionary Algorithms Laboratory
University of Applied Sciences Upper Austria
Hagenberg, Austria
michael.kommenda@fh-hagenberg.at

Gabriel Kronberger

Josef Ressel Centre for Symbolic Regression
Heuristic and Evolutionary Algorithms Laboratory
University of Applied Sciences Upper Austria
Hagenberg, Austria
gabriel.kronberger@fh-hagenberg.at

Michael Affenzeller

Institute for Formal Models and Verification
Johannes Kepler University
Linz, Austria
Heuristic and Evolutionary Algorithms Laboratory
University of Applied Sciences Upper Austria
Hagenberg, Austria
michael.affenzeller@fh-hagenberg.at

ABSTRACT

Symbolic regression (SR) with genetic programming (GP) evolves models with the explicit goal of minimizing prediction error. However, practical applications usually require a balance between numerical accuracy and model interpretability. Furthermore, the success of this process depends also on implicit, indirect goals such as model parsimony and population diversity. GP is often extended with heuristics and techniques aimed at reducing bloat, improving diversity at the semantic or structural level, and preventing premature convergence to local optima.

We investigate in this paper the suitability of multi-objective algorithms for SR, where desired properties of parsimony and diversity are explicitly stated as optimization goals. We use two established multi-objective approaches, namely the NSGA-II and MOEA/D algorithms, and evaluate different secondary objectives such as length, complexity and diversity. We use two- and three-objective combinations on a selection of regression benchmark problems and compare the results with standard GP used as a baseline method.

We conclude based on empirical testing that multi-objective SR represents a viable and effective approach for obtaining models with improved accuracy and generalization ability. Our results show that explicit pursuit of diversity provides substantial benefits to the search process. Multi-objective configurations combining diversity and parsimony objectives provide the best balance of numerical accuracy and model parsimony, allowing practitioners to select suitable models from a diverse set of solutions on the Pareto front.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '19, July 13–17, 2019, Prague, Czech Republic

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

genetic programming, symbolic regression, multi-objective optimization, parsimony, diversity

ACM Reference Format:

Bogdan Burlacu, Gabriel Kronberger, Michael Kommenda, and Michael Affenzeller. 2019. Parsimony Measures in Multi-objective Genetic Programming for Symbolic Regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2019 (GECCO '19)*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

1.1 Symbolic Regression

Symbolic regression (SR) is a grey-box modeling technique where an appropriate mathematical structure of the regression model is found by exploring the space of all possible expressions, usually by employing genetic programming to evolve an initially-random population of expression tree solution candidates.

Since the model structure is derived from data, SR typically tends to produce large, complex models that are not easily-interpretable and prone to overfitting and poor prediction performance on unseen data. For this reason practitioners often have to accept a compromise between accuracy and complexity.

Methods to control complexity are connected with aspects of GP evolutionary dynamics such as loss of diversity, gradual increase in tree size, occurrence of bloat and introns. These phenomena do not occur individually but rather share a common set of facilitating conditions. The main cause for size increase is considered to be fitness-based selection [17]. Besides a detrimental effect on model quality and interpretability, bloat also increases the computational complexity of fitness evaluation, further slowing down the search.

Basic approaches to control tree size include static tree depth and length limits [16], dynamic limits [23], parsimony pressure [19, 21], or fixed length distributions [11]. Their effectiveness depends on appropriate parameterization by the user, according to the specific requirements of the problem.

More sophisticated approaches include mechanisms for diversity control at the phenotypic or genotypic level [3, 5, 6, 25]. Diversity, particularly at the behavioral (phenotypic) level, is closely tied to the phenomenon of bloat. Structurally different individuals can exhibit similar or even identical behavior when the population accumulates introns (non-functional code fragments in their tree structure), potentially leading to excessively large models, poor search performance or overfitting.

In this context, our motivation for this contribution is to explore new ways of improving the symbolic regression process as a whole via multi-objective approaches. Model simplicity and interpretability are main requirements in industrial applications of symbolic regression, justifying approaches where these goals are explicitly stated as optimization objectives.

We follow two promising research directions:

- (1) Explore the possibility of using combinations of secondary objectives (eg., parsimony and diversity) to improve the desired characteristics of models
- (2) Explore the effectiveness of different multi-objective optimization paradigms for symbolic regression

The remainder of this paper is organized as follows: Section 1.2 provides an overview of multi-objective genetic programming and a synthesis of previous work in this area. Section 2 introduces our proposed methodology and briefly discusses the main features of the MOEA/D and NSGA-II algorithms. Section 3 describes empirical results and discusses the effectiveness of the proposed multi-objective approach. Finally, Section 4 summarizes our main results and suggests potential avenues for future research.

1.2 Multi-objective Genetic Programming

Multi-objective optimization problems (MOP) can be stated as the simultaneous optimization of a collection of objectives, as follows:

$$\text{minimize } F(x) = (f_1(x), \dots, f_m(x))^T, x \in \Omega \quad (1)$$

where Ω is the *decision (variable) space*, $F : \Omega \rightarrow \mathbb{R}^m$ consists of m real-valued objective functions and \mathbb{R}^m is the objective space.

Generally speaking, a non-trivial MOP does not have a solution that simultaneously optimizes each objective. In such cases when the objectives contradict each other (are said to be conflicting), the best trade-offs are identified using the concept of Pareto optimality, which states that two solutions represent a trade-off if they don't *dominate* each other according to the domination criteria described by Equation 2. The set of all Pareto optimal solutions is called the *Pareto front* (PF).

$$u, v \in \mathbb{R}^m, u \text{ dominates } v \iff \forall i \in [1, \dots, m], u_i \leq v_i \text{ and } \exists j \in [1, \dots, m], u_j < v_j \quad (2)$$

Since for some MOP the PF might be very large or even infinite, multi-objective optimization algorithms (MOA) try to find a set of Pareto optimal solutions that are evenly distributed along the PF

and are representative of the possible trade-offs. Established multi-objective evolutionary algorithms (MOEAs) such as NSGA-II [10], SPEA2 [29] or MOEA/D [28] take special care to ensure diversity of the Pareto front.

Multi-objective GP approaches implement Pareto optimization techniques to exert more control on the evolutionary process, typically by employing some form of complexity measure as a secondary objective. It was shown, however, that explicitly optimizing for small tree size increases the risk of premature convergence without appropriate diversity control [8, 9].

Ekart and Nemeth use a Pareto non-domination criterion to shift the focus of selection towards accurate and small solutions. Their selection method is able to reduce code growth and processing time without significant loss of solution accuracy [12].

Bleuler et al. use size as a secondary objective within the SPEA2 algorithm and test their approach against several parsimony-oriented extensions of standard GP. The multi-objective approach successfully manages to produce more compact solutions [1].

de Jong et al. test an MOEA approach called FOCUS (Find Only and Complete Nondominated Sets) using fitness, size and diversity as objectives. Diversity is measured using a basic tree distance counting the number of overlapping nodes between two trees. They find that the diversity objective is essential in obtaining good performance, and that optimizing only for tree size leads to premature convergence of the algorithm. Overall, the multi-objective approach outperforms standard GP on the tested problems [9].

de Jong and Pollack further investigate the risk of premature convergence for size-focused multi-objective approaches. They show that diversity maintenance is required in order to prevent premature convergence to small trees [8].

Smits and Kotanchek propose a multi-objective GP algorithm called ParetoGP, using nested tree length (or visitation length) as a secondary objective to control complexity. They report major improvements over standard GP in terms of accuracy, robustness and computational efficiency [24].

Vladislavleva et al. use two different complexity measures within the ParetoGP framework. The expressional complexity measure favors compact models, while the order of non-linearity measure favors models with smoother response surfaces. They report an increase in the extrapolative capabilities of models generated using latter secondary objective, although compared to the former, this objective is less effective in reducing model length [26].

Schmidt and Lipson use age as a secondary objective in their AFPO (Age-Fitness Pareto Optimization) algorithm, to protect young, lower-fit individuals from replacement. This causes older, stagnant individuals that are already stuck to local optima to eventually be replaced by younger, fitter individuals. They find that while AFPO gives the overall best performance among all tested algorithms, it also exhibits a slight tendency towards bloat [22].

Burks and Punch develop a new distance measure based on the concept of genetic markers and use it to improve population diversity in a multi-objective setting. They replace individual age in the default AFPO algorithm with the genetic marker distance and observe that maintaining genetic diversity is greatly beneficial for the genetic algorithm [4, 5].

We employ in this work two well-known multi-objective evolutionary algorithms (MOEAs), MOEA/D [28] and NSGA-II [10].

The motivation for this comparison is to investigate how the two different paradigms – decomposition-based and domination-based – interact with symbolic regression. So far, all multi-objective methods in the area of GP have relied on the domination-based framework, where all objectives are optimized simultaneously. Here, the Pareto dominance principle plays a key role in the convergence of the algorithm. However, Pareto dominance can show its limitations when the number of objectives increases, due to most of the solutions in the population becoming non-dominated with one another, reducing selection pressure and hampering the evolutionary process [14].

The decomposition approach uses scalarizing functions to transform a MOP into single-objective optimization subproblems. A scalarizing function (also known as utility function or aggregation function) transforms a vector in objective space into a real value using a predefined weight vector. MOEAs based on decomposition perform selection based on the scalar aggregated fitness values of solutions.

2 METHODOLOGY

We employ the NSGA-II and MOEA/D algorithms together with a set of parsimony and diversity objectives. The algorithms differ from one another in the basic concept they employ for the search of Pareto optimal solutions. The MOEA/D algorithm uses a decomposition-based approach to achieve a uniform spread of solutions across the Pareto front, while the NSGA-II algorithm uses non-dominated sorting and crowding distance-based selection.

The two algorithms are implemented in HeuristicLab, an open-source framework for metaheuristic optimization [27]. Their respective implementations are adapted to use SR-specific tree recombination operators. Both algorithms utilize the same objective functions and genetic operators and differ only in the specific logic of their respective main loops.

2.1 MOEA/D

The Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) by [Zhang and Li](#) [28] decomposes a MOP into N scalar optimization subproblems, formulated via a scalarization approach using uniformly distributed weight vectors.

Different decomposition methods are possible; we employ the Chebyshev approach with objective scaling as suggested in [28], where the i^{th} subproblem is defined in the form:

$$\text{minimize } g^{tch}(x|\lambda_i, z^*) = \max_{1 \leq j \leq m} \left\{ \lambda_i^j \left| \frac{f_j(x) - z_j^*}{z_j^{\text{nad}} - z_j^*} \right| \right\} \quad (3)$$

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the ideal reference point, $z^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_m^{\text{nad}})^T$ is the nadir point in objective space and $\lambda_i, \dots, \lambda_N$ are the weight vectors associated with each subproblem.

$$z_j^* = \min \{f_j(x) | x \in \Omega\}, j = 1, \dots, m \quad (4)$$

$$z_j^{\text{nad}} = \max \{f_j(x) | x \in \Omega\}, j = 1, \dots, m \quad (5)$$

$$\lambda_i = (\lambda_i^1, \dots, \lambda_i^m)^T, \lambda_i^j \geq 0 \quad (6)$$

$$\sum_{j=1}^m \lambda_i^j = 1, \forall i = 1, \dots, N, \forall j = 1, \dots, m \quad (7)$$

In our experiments we set the reference point to zero (the best value for each objective). We generate uniformly distributed weight vectors λ_i by drawing m independent samples y_i^1, \dots, y_i^m from a $\Gamma(1)$ distribution and setting

$$\lambda_i^j = \frac{y_i^j}{\sum_j y_i^j}, j = 1, \dots, m \quad (8)$$

The MOEA/D algorithm main loop is the original one from [28], where the reproduction step is realized by subtree crossover, followed by mutation.

2.2 NSGA-II

The NSGA-II algorithm [10] uses the crowding distance between ranked non-dominated solutions to guide selection towards a uniformly spread PF. It employs elitism by filling a new population each generation with the best solutions from both parent individuals and generated offspring.

However, the discrete nature of parsimony-focused measures for symbolic regression may impact the effectiveness of the NSGA-II algorithm for symbolic regression. In its default implementation, NSGA-II treats solutions with equal objective values as non-dominated. For example, tree length as a secondary objective may cause overcrowding of the PF with single-node non-dominated individuals that always have constant objective values. Another potential issue stems from the floating-point representation of the main objective value (usually an error or correlation measure), when many individuals of similar quality (up to many decimal places) and varying complexity artificially enlarge the Pareto front. A potential fix for this issue is rounding the objective value to a fixed number of decimal places [15].

To avoid the issues described above, we adopt the NSGA-II algorithm with the adaptations for symbolic regression proposed by [Kommenda et al.](#) [15].

2.3 Objective Functions

Typical parsimony measures for GP include tree length and complexity. These objectives are intended to complement the usual fitness measure and help the algorithm to:

- Evolve solutions faster by not having to process overly-large trees
- Increase solution parsimony, leading to better interpretability and lower risk of overfitting

We include in our testing a collection of three parsimony measures (tree length, tree visitation length, tree complexity) and combine them with a distance-based diversity measure.

2.3.1 Tree Size. We test two basic parsimony-oriented objectives, namely the tree length and the tree visitation length by [Smits and Kotanchek](#) [24], introduced as a way to simultaneously favor smaller, flatter, and more balanced structures.

Tree visitation length is defined as the sum of the lengths of all subtrees inside a tree.

$$\text{Length}(T) = \sum_{s \in_s T} 1 \quad (9)$$

$$\text{VisitationLength}(T) = \sum_{s \in_s T} \text{Length}(s) \quad (10)$$

where $s \in_s T$ defines the subtree relation and returns all subtrees s of tree T .

2.3.2 Tree Complexity. We use the recursive complexity measure by [Kommenda et al. \[15\]](#), abbreviated here as *Cxty*. This measure aims to produce simpler expressions by penalizing nesting of symbols inside the tree structure, as well as non-linear symbols. A sought effect is to push less complex functions towards the leaves of the tree and more complex ones towards the tree root.

$$\text{Cxty}(n) = \begin{cases} 1 & \text{if } \text{sym}(n) = \text{constant} \\ 2 & \text{if } \text{sym}(n) = \text{variable} \\ \sum_{c \in_c n} \text{Cxty}(c) & \text{if } \text{sym}(n) \in (+, -) \\ \prod_{c \in_c n} \text{Cxty}(c) + 1 & \text{if } \text{sym}(n) \in (\times, \div) \\ \text{Cxty}(n_1)^2 & \text{if } \text{sym}(n) = \text{square} \\ \text{Cxty}(n_1)^3 & \text{if } \text{sym}(n) = \text{sqrt} \\ 2^{\text{Cxty}(n_1)} & \text{if } \text{sym}(n) \in (\sin, \cos, \tan) \\ 2^{\text{Cxty}(n_1)} & \text{if } \text{sym}(n) \in (\exp, \log) \end{cases} \quad (11)$$

where $c \in_c n$ defines the child relation and returns all child nodes c of node n , the index n_i refers to the i^{th} child of node n and $\text{sym}(n)$ returns the symbol of node n .

2.3.3 Tree Similarity. We employ an efficient hash-based tree similarity measure, where the trees are hashed in a bottom-up traversal:

- Tree leaves are hashed according to their concrete type, i.e. constants are hashed based on value, variables are hashed based on variable name and associated weight coefficient.
- Function nodes are hashed according to their symbol and the hash values of their children.
- Children of commutative symbols are sorted before hashing to ensure correct identification of subtree isomorphisms during distance calculation.
- Actual hashes are computed as 64-bit unsigned integers using a simple and efficient hash function¹. No hash collisions were observed in our experiments.

After hashing, subtrees with the same hash value between two trees will be isomorphic. Tree distance is then defined as the number of common hash values between trees (equivalent to the number of common nodes). Inadvertently, our hash-based measure supersedes the structural diversity measure proposed by [Burks and Punch \[4\]](#), since node hash values implicitly act as genetic markers.

Similarity is defined as:

$$\text{TreeSimilarity}(T_1, T_2) = \frac{2 \cdot \text{TreeDistance}(T_1, T_2)}{\text{Length}(T_1) + \text{Length}(T_2)} \quad (12)$$

¹<http://www.partow.net/programming/hashfunctions/#DJBHashFunction>

Table 1: Two- and three-objective combinations

Two objectives	Three objectives
(R^2, length)	$(R^2, \text{length}, \text{diversity})$
$(R^2, \text{visitation length})$	$(R^2, \text{length}, \text{complexity})$
$(R^2, \text{complexity})$	$(R^2, \text{visitation length}, \text{diversity})$
$(R^2, \text{diversity})$	$(R^2, \text{visitation length}, \text{complexity})$
	$(R^2, \text{complexity}, \text{diversity})$

Since the resulting mapping of isomorphic subtrees depends on the results of hashing, and hashing depends on the numerical coefficients present in the tree, we consider our tree similarity as a *hybrid* measure, incorporating both structural and semantic information. Recent work suggests it may be worthwhile pursuing hybrid approaches for simultaneously preserving both structural and behavioral diversity [6].

We define the diversity objective as a minimization measure, namely to minimize the average similarity between the current individual and the rest of the population. A detailed analysis of this new similarity measure is available in [Burlacu et al. \[7\]](#) (preprint).

3 EMPIRICAL STUDY

We include the standard GA as a comparison baseline and test the MOEA/D and NSGA-II algorithms with different combinations of secondary objectives as shown in Table 1. We use a set of four synthetic benchmark problems and two real-world problems for symbolic regression, as described in Table 2. Each of the resulting 55 configurations was repeated 50 times.

Table 3 shows the algorithm parameterization. We use a high depth limit to investigate the possible influence of tree shape on the quality and parsimony of the results. The parameterization of the MOEA/D and NSGA-II algorithms is discussed below.

MOEA/D Parameters.

Neighbourhood size The size of the neighbourhood to select mates for recombination from, and to afterwards perform replacement in.

Neighbourhood selection probability Specifies the probability that recombination and replacement occur in the current subproblem's neighbourhood. Otherwise they occur at population level.

Maximum replaced solutions The maximum number of solutions to be replaced by a more fit child solution. A small number here helps preserve population diversity.

NSGA-II Parameters.

Dominate on equal qualities Eliminates duplicate solutions (single-node trees that cannot otherwise be dominated) from the Pareto front

Decimal places Helps avoid overcrowding of the Pareto front in conjunction with the dominate on equal qualities parameter by eliminating solutions in very close proximity to one another

Crowded tournament size Tunes selection intensity towards more diverse (in terms of crowding distance) solutions in the Pareto front

3.1 Comparison Methodology

We first compare the algorithms against the baseline GA using only the R^2 objective value as a criterion. Then, we compare the two multi-objective algorithms in terms of their ability to produce Pareto fronts containing both diverse and numerically accurate solutions.

We measure multi-objective performance using the hypervolume indicator H [30], a measure of the volume enclosed by a Pareto front approximation and a reference point. The reference point must be dominated by every point in the Pareto front. We calculate hypervolume indicators H_L and H_C in the two-dimensional objective spaces defined by (quality, length) and (quality, complexity). The goal is to identify Pareto fronts containing small, simple and numerically accurate solutions.

To obtain comparable hypervolumes using the point $r = [0, 1]$ as reference point, both length and complexity are normalized beforehand:

- Tree length is divided by the maximum value of 50 nodes
- Since tree complexity is unbounded and grows exponentially, we clamp every ∞ value to `DOUBLE_MAX` and apply a logarithmic transformation.

The hypervolume calculation steps for each tested multi-objective configuration are as follows:

- (1) Retrieve the Pareto front in the original objective space
- (2) Calculate main objective values (Pearson's R^2 correlation) on either training or test data
- (3) Calculate the normalized length and complexity values for each solution in the Pareto front
- (4) Construct Pareto fronts in the new (quality, normalized length) and (quality, normalized complexity) objective spaces by removing dominated solutions
- (5) Calculate the H_L and H_C hypervolumes for each algorithm and problem. Since we consider both test and training data, we obtain four hypervolume indicators in total: $H_{L|Train}$, $H_{L|Test}$, $H_{C|Train}$ and $H_{C|Test}$.

3.2 Benchmark Results

Due to the large number of tested configurations we only show the median R^2 and aggregated algorithm ranking on the training and test data in Table 4. Calculated hypervolumes are shown in Table 5 for the training data and Table 6 on the test data. Table 7 additionally shows aggregated statistics for the Pareto fronts in the (quality, length) and (quality, complexity) planes, obtained with the method described in Section 3.1. Detailed results for each algorithmic configuration are available online².

Purely from a performance standpoint, the MOEA/D and NSGA-II algorithms are virtually indistinguishable on the tested problems. The standard GA algorithm ranks behind most multi-objective configurations in terms of training performance and places last in the ranking based on generalization capability on test data.

Although the relative difference in training R^2 values between the baseline approach (standard GA) and the top-ranked multi-objective approach (MOEA/D visitation length, diversity) is relatively small, the ranking is statistically significant, with a significance level $p = 0.0002$ for a one-tailed t-test of the respective algorithm rankings on each problem.

Our results confirm previous conclusions [4, 5, 8, 9] that diversity plays an important role in the evolutionary process. Algorithm ranking on both training and test data shows the effectiveness of parsimony–diversity objective combinations. The pair (visitation length, diversity) seems particularly well-suited for producing models with good accuracy and generalization capabilities, occupying the first positions in both training and test rankings. This also indicates that visitation length is a more effective measure for promoting solution parsimony.

Somewhat surprisingly, the (complexity, visitation length) combination works well for promoting solution generality, placing just below the diversity combinations in the test ranking (Table 4). This supports the intuition that small, balanced trees of low complexity are preferable to avoid overfitting.

We also notice that diversity alone as a secondary objective tends to overfit the training data, as indicated by the difference in training and test rankings for the corresponding MOEA/D and NSGA-II configurations. This tendency towards overfitting appears to correlate with low hypervolumes of the resulting Pareto fronts, as shown in Tables 5 and 6, where standalone diversity ranks behind every other configuration.

The hypervolume indicator shows that combinations of two secondary objectives tend to produce more diverse Pareto fronts. Pareto front statistics in Table 7 additionally show that multi-objective configurations using any one of the parsimony objectives produce smaller and less complex solutions. By comparison, standalone similarity as a secondary objective leads to dramatically increased complexity and a reduced number of non-dominated solutions in the Pareto front.

The Pareto fronts produced by the MOEA/D and NSGA-II algorithms are comparable in size. MOEA/D tends to produce slightly larger and more complex solutions in the Pareto front. Average Pareto front solution complexity suggests that the decomposition approach does not work very well with the tree complexity measure. By comparison, the NSGA-II algorithm using tree complexity is able to produce very simple solutions in the Pareto front.

Overall, our results show that the combination of relevant objectives such as length, complexity and diversity with state-of-the-art multi-objective algorithms represents a powerful approach for symbolic regression.

We did not include a detailed runtime analysis of the tested algorithms as several implementation aspects are not yet considered production-quality. Roughly, the slowest configuration was two times as slow as the standard GA. NSGA-II configurations using parsimony as an objective are 10% faster on average than standard GA, while MOEA/D configurations are between 10% and 40% slower. From the tested secondary objectives, population diversity is the slowest, and configurations including it are on average between 40% and 100% slower than standard GP.

²<https://dev.heuristiclab.com/trac.fcgi/wiki/AdditionalMaterial#GECCO2019>

Table 2: Definition of benchmark problems and training and test ranges.

Name	Function	Training	Test
Breiman-1 [2]	$f(\mathbf{x}) = \begin{cases} 3 + 3x_2 + 2x_3 + x_4 & \text{if } x_1 = 1 \\ -3 + 3x_5 + 2x_6 + x_7 & \text{otherwise} \end{cases}$ $y = f(\mathbf{x}) + \epsilon \sim \mathcal{N}(0, 2)$ $P(x_1 = -1) = P(x_1 = 1) = \frac{1}{2}$ $P(x_m = -1) = P(x_m = 0) = P(x_m = 1) = \frac{1}{3}, m = 2, \dots, 10$	5000 samples $U[0, 1]$	5000 samples $U[0, 1]$
Friedman-1 [13]	$f(\mathbf{x}) = 0.1 \cdot e^{4x_1} + \frac{4}{1 + e^{-20 \cdot (x_2 - 0.5)}} + 3x_3 + 2x_4 + x_5$ $y = f(\mathbf{x}) + \epsilon \sim \mathcal{N}(0, 1)$	5000 samples $U[0, 1]$	5000 samples $U[0, 1]$
Friedman-2 [13]	$f(\mathbf{x}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 1/2)^2 + 10x_4 + 5x_5$ $y = f(\mathbf{x}) + \epsilon \sim \mathcal{N}(0, 1)$	5000 samples $U[0, 1]$	5000 samples $U[0, 1]$
Poly-10 [20]	$f(\mathbf{x}) = x_1 x_2 + x_3 x_4 + x_5 x_6 + x_1 x_7 x_9 + x_3 x_6 x_{10}$	250 samples $U[-1, 1]$	250 samples $U[-1, 1]$
Chemical	Real-world data	711 samples	355 samples
Housing	Real-world data	337 samples	169 samples

Table 3: Algorithm configuration

Genetic algorithm	(common for all algorithms)
Function set	Binary functions (+, -, ×, ÷) Trigonometric functions (sin, cos, tan) Exponential functions (exp, log) Power functions (square, sqrt)
Terminal set	constant, weight · variable
Max. tree depth	100 levels
Max. tree length	50 nodes
Tree initialization	Probabilistic tree creator (PTC2) [18]
Population size	1000 individuals
Max. generations	500 generations
Crossover probability	100%
Crossover operator	Subtree crossover
Mutation probability	25%
Mutation operator	Change symbol, single-point remove branch, replace branch
Selection operator	Tournament selector (tournament size 5)
Main objective	Maximize the Pearson's R^2 correlation coefficient
MOEA/D	
Decomposition method	Chebyshev
Neighbourhood size	20
Neighbourhood selection probability	10%
Max. replaced solutions	2
NSGA-II	
Dominate on equal qualities	True
Main objective rounding	5 decimal places
Selection operator	Crowded tournament selector (tournament size 5)
Selected parents	2000

4 CONCLUSION

We explored in this work different combinations of objective functions for multi-objective symbolic regression. Based on previous research, we selected a number of parsimony measures (tree length, visitation length and complexity) and combined them with a diversity measure aimed at preventing premature convergence of the algorithm.

We tested two multi-objective algorithms based on different optimization paradigms: the MOEA/D algorithm using a decomposition-based approach and the NSGA-II algorithm using a Pareto domination-based approach. We performed a detailed comparison using the hypervolume indicator as a performance measure for the generated Pareto fronts. Our methodology ensured a fair comparison between all tested configurations by moving Pareto solutions into the same (quality, length) or (quality, complexity) coordinates.

Our empirical study did not reveal any significant performance difference between the MOEA/D and NSGA-II algorithms. Both approaches successfully compare with standard GA in terms of training and test solution quality.

Empirical results support our original assumption that multiple secondary objectives can successfully work together towards more accurate and parsimonious solutions. Successful objective pairs such as (visitation length, diversity) likely own their success to a synergistic effect cancelling out their individual weaknesses.

Future directions in this area include empirical validation on a larger set of benchmarks, investigations of population dynamics in multi-objective symbolic regression, analysis of runtime performance, and the feasibility of many-objective approaches for symbolic regression.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support by the Christian Doppler Research Association and the Federal Ministry of Digital and Economic Affairs within the *Josef Ressel Centre for Symbolic Regression*

Table 4: Training and test qualities for each problem, expressed as median R^2 . $\overline{\text{Rank}}$ and $\widehat{\text{Rank}}$ represent the average and median rank calculated using training and test R^2 (rank on the training data followed by rank on test data inside the parentheses). Table rows ordered lexicographically based on median and average performance rank on the test data.

Algorithm	Secondary objectives	Breiman-I	Chemical-I	Friedman-I	Friedman-II	Housing	Poly-10	$\overline{\text{Rank}}$	$\widehat{\text{Rank}}$						
MOEA-D	visitation length, diversity	0.889	0.883	0.797	0.678	0.863	0.863	0.960	0.959	0.862	0.794	0.847	0.819	4 (8)	4 (4)
NSGA-II	visitation length, diversity	0.884	0.878	0.796	0.741	0.863	0.863	0.959	0.958	0.870	0.796	0.858	0.835	4 (6)	4 (5)
NSGA-II	length, diversity	0.883	0.877	0.796	0.707	0.863	0.863	0.960	0.959	0.864	0.815	0.857	0.840	5 (5)	4 (5)
NSGA-II	complexity, diversity	0.889	0.883	0.796	0.697	0.859	0.860	0.954	0.951	0.858	0.820	0.850	0.826	7 (6)	8 (6)
MOEA-D	complexity, visitation length	0.883	0.878	0.792	0.690	0.863	0.862	0.958	0.957	0.860	0.805	0.839	0.816	8 (7)	9 (7)
NSGA-II	complexity, visitation length	0.889	0.884	0.785	0.732	0.857	0.859	0.939	0.937	0.854	0.816	0.814	0.781	11 (8)	12 (8)
MOEA-D	visitation length	0.885	0.878	0.791	0.729	0.862	0.862	0.958	0.956	0.866	0.797	0.844	0.803	7 (9)	7 (9)
NSGA-II	diversity	0.870	0.864	0.805	0.688	0.863	0.863	0.956	0.956	0.868	0.797	0.848	0.825	7 (10)	5 (10)
NSGA-II	complexity, length	0.887	0.881	0.788	0.721	0.856	0.856	0.916	0.914	0.855	0.816	0.668	0.622	13 (11)	13 (10)
NSGA-II	complexity	0.888	0.883	0.782	0.697	0.842	0.847	0.893	0.887	0.839	0.821	0.772	0.732	15 (11)	17 (10)
MOEA-D	complexity, diversity	0.869	0.863	0.808	0.684	0.864	0.864	0.959	0.958	0.867	0.805	0.840	0.790	6 (10)	4 (11)
MOEA-D	length, diversity	0.884	0.877	0.777	0.709	0.857	0.859	0.933	0.930	0.854	0.793	0.833	0.806	13 (11)	13 (12)
MOEA-D	complexity	0.812	0.809	0.794	0.685	0.861	0.861	0.944	0.941	0.851	0.809	0.634	0.521	13 (12)	12 (12)
MOEA-D	diversity	0.871	0.864	0.802	0.669	0.864	0.864	0.958	0.956	0.871	0.799	0.830	0.797	6 (11)	5 (12)
MOEA-D	length	0.882	0.876	0.778	0.683	0.859	0.860	0.927	0.923	0.848	0.807	0.839	0.817	14 (11)	14 (12)
NSGA-II	visitation length	0.882	0.878	0.783	0.692	0.859	0.859	0.898	0.893	0.850	0.804	0.622	0.519	14 (13)	14 (13)
NSGA-II	length	0.881	0.876	0.779	0.684	0.858	0.859	0.926	0.924	0.849	0.805	0.607	0.468	15 (14)	15 (14)
MOEA-D	complexity, length	0.879	0.874	0.774	0.675	0.859	0.860	0.921	0.917	0.849	0.801	0.740	0.677	16 (14)	15 (14)
Standard GA	N/A	0.846	0.841	0.784	0.669	0.859	0.859	0.947	0.946	0.848	0.803	0.604	0.499	15 (16)	15 (18)

Table 5: Training data hypervolumes H_L (quality, normalized length) and H_C (quality, normalized complexity) median values (over 50 repetitions) for each problem. Ranking on each problem is calculated based on average hypervolume $H = \frac{H_C + H_L}{2}$. $\overline{\text{Rank}}$ and $\widehat{\text{Rank}}$ represent the average and median rank over all problems, respectively. Table rows ordered lexicographically based on median and average hypervolume rank.

Algorithm	Secondary objectives	Breiman-I		Chemical-I		Friedman-I		Friedman-II		Housing		Poly-10		$\overline{\text{Rank}}$	$\widehat{\text{Rank}}$
		H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C		
MOEA-D	visitation length, diversity	0.703	0.885	0.634	0.789	0.706	0.856	0.775	0.951	0.708	0.832	0.628	0.835	2.7	2.0
NSGA-II	complexity, diversity	0.700	0.886	0.631	0.793	0.700	0.854	0.748	0.949	0.700	0.854	0.629	0.845	3.3	3.5
NSGA-II	visitation length, diversity	0.705	0.873	0.637	0.782	0.707	0.853	0.775	0.949	0.711	0.820	0.636	0.836	3.8	3.5
MOEA-D	complexity, visitation length	0.697	0.876	0.632	0.783	0.703	0.854	0.771	0.947	0.704	0.838	0.623	0.830	5.7	5.5
MOEA-D	visitation length	0.691	0.877	0.629	0.782	0.702	0.854	0.770	0.949	0.705	0.828	0.625	0.833	6.5	5.5
NSGA-II	complexity, visitation length	0.704	0.886	0.632	0.783	0.701	0.852	0.739	0.928	0.704	0.851	0.607	0.810	5.7	6.5
NSGA-II	complexity, length	0.703	0.884	0.636	0.786	0.701	0.851	0.735	0.913	0.704	0.852	0.481	0.648	6.7	6.5
NSGA-II	length, diversity	0.702	0.859	0.635	0.779	0.708	0.845	0.776	0.951	0.710	0.816	0.637	0.832	7.7	7.5
MOEA-D	length, diversity	0.701	0.876	0.628	0.772	0.701	0.849	0.742	0.919	0.703	0.826	0.624	0.822	9.2	9.0
NSGA-II	visitation length	0.700	0.874	0.629	0.777	0.703	0.851	0.721	0.872	0.700	0.829	0.468	0.601	10.7	10.0
MOEA-D	length	0.698	0.876	0.627	0.773	0.700	0.851	0.737	0.903	0.699	0.830	0.625	0.821	10.3	11.0
MOEA-D	complexity, length	0.697	0.873	0.627	0.770	0.702	0.853	0.728	0.898	0.701	0.828	0.539	0.724	10.5	11.5
NSGA-II	complexity	0.691	0.884	0.631	0.779	0.684	0.834	0.708	0.890	0.691	0.837	0.557	0.768	11.3	11.5
NSGA-II	length	0.698	0.869	0.629	0.771	0.702	0.849	0.732	0.887	0.700	0.806	0.462	0.574	13.3	13.0
MOEA-D	complexity, diversity	0.630	0.786	0.608	0.780	0.686	0.850	0.734	0.941	0.682	0.793	0.568	0.798	12.7	14.5
MOEA-D	complexity	0.559	0.681	0.569	0.762	0.651	0.844	0.662	0.888	0.659	0.795	0.422	0.597	16.0	16.0
MOEA-D	diversity	0.511	0.646	0.535	0.768	0.590	0.759	0.623	0.863	0.608	0.733	0.434	0.514	17.0	17.0
NSGA-II	diversity	0.562	0.559	0.540	0.641	0.599	0.680	0.647	0.671	0.601	0.692	0.452	0.300	18.0	18.0

REFERENCES

- [1] Stefan Bleuler, Martin Brack, Lothar Thiele, and Eckart Zitzler. 2001. Multi-objective Genetic Programming: Reducing Bloat Using SPEA2. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*. IEEE Press, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 536–543. <https://doi.org/doi:10.1109/CEC.2001.934438>
- [2] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Taylor & Francis.
- [3] E. K. Burke, S. Gustafson, and G. Kendall. 2004. Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation* 8, 1 (Feb 2004), 47–62. <https://doi.org/10.1109/TEVC.2003.819263>
- [4] Armand R. Burks and William F. Punch. 2015. An Efficient Structural Diversity Technique for Genetic Programming. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, Madrid, Spain, July 11–15, 2015*, Sara Silva and Anna Isabel Esparcia-Alcázar (Eds.). ACM, 991–998. <http://dl.acm.org/citation.cfm?id=2739480>

Table 6: Test data hypervolumes H_L (quality, normalized length) and H_C (quality, normalized complexity) median values (over 50 repetitions) for each problem. Ranking on each problem is calculated based on average hypervolume $H = \frac{H_C + H_L}{2}$. Rank and Rank represent the average and median rank over all problems, respectively. Table rows ordered lexicographically based on median and average hypervolume rank.

Algorithm	Secondary objectives	Breiman-I		Chemical-I		Friedman-I		Friedman-II		Housing		Poly-10		Rank	Rank
		H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C	H_L	H_C		
MOEA/D	visitation length, diversity	0.698	0.879	0.554	0.747	0.711	0.857	0.772	0.949	0.701	0.817	0.641	0.838	3.0	2.5
NSGA-II	visitation length, diversity	0.700	0.866	0.580	0.773	0.711	0.854	0.772	0.947	0.702	0.812	0.642	0.841	3.5	2.5
NSGA-II	complexity, diversity	0.696	0.880	0.565	0.780	0.706	0.856	0.743	0.946	0.700	0.835	0.638	0.840	3.5	4.0
NSGA-II	length, diversity	0.697	0.853	0.580	0.747	0.712	0.849	0.773	0.950	0.705	0.813	0.645	0.839	5.0	4.0
NSGA-II	complexity, visitation length	0.699	0.879	0.543	0.748	0.706	0.854	0.736	0.924	0.701	0.831	0.616	0.831	6.0	6.5
MOEA/D	complexity, visitation length	0.691	0.870	0.552	0.733	0.707	0.856	0.768	0.946	0.697	0.817	0.631	0.836	6.7	6.5
MOEA-D	visitation length	0.686	0.871	0.549	0.737	0.707	0.854	0.767	0.947	0.697	0.812	0.630	0.836	7.7	7.0
NSGA-II	complexity, length	0.698	0.878	0.539	0.743	0.707	0.853	0.729	0.910	0.700	0.829	0.530	0.681	7.7	8.5
MOEA/D	length, diversity	0.696	0.870	0.544	0.721	0.707	0.853	0.738	0.915	0.699	0.809	0.633	0.838	8.5	9.0
MOEA/D	length	0.693	0.871	0.512	0.703	0.706	0.854	0.732	0.898	0.695	0.813	0.631	0.836	10.2	9.5
NSGA-II	visitation length	0.696	0.868	0.523	0.726	0.708	0.852	0.718	0.867	0.696	0.811	0.426	0.565	11.0	11.5
NSGA-II	complexity	0.686	0.880	0.516	0.714	0.693	0.839	0.701	0.884	0.695	0.826	0.573	0.762	10.8	12.0
MOEA/D	complexity, length	0.691	0.870	0.521	0.701	0.707	0.853	0.721	0.892	0.695	0.812	0.561	0.733	11.3	12.0
MOEA/D	complexity, diversity	0.626	0.786	0.509	0.722	0.693	0.852	0.729	0.939	0.687	0.799	0.575	0.822	12.2	13.0
NSGA-II	length	0.693	0.865	0.533	0.710	0.707	0.851	0.727	0.881	0.694	0.803	0.408	0.511	13.3	13.0
MOEA/D	complexity	0.554	0.667	0.403	0.668	0.662	0.847	0.653	0.883	0.671	0.793	0.405	0.568	15.8	16.0
MOEA/D	diversity	0.510	0.643	0.366	0.632	0.596	0.772	0.617	0.859	0.614	0.758	0.441	0.502	16.8	17.0
NSGA-II	diversity	0.559	0.561	0.383	0.438	0.609	0.702	0.641	0.658	0.601	0.709	0.442	0.317	18.0	18.0

Table 7: Length and complexity of Pareto front solutions in the (quality, length) and (quality, complexity) planes. Size here represents the size of the Pareto front. Results are averaged over all tested problems.

Algorithm	Secondary objectives	(quality, length)			(quality, complexity)		
		Length	Cxty	Size	Length	Cxty	Size
MOEA-D	complexity	21.3	67.5	13.8	24.0	49.5	23.3
MOEA-D	complexity, length	21.0	98.2	27.2	21.8	19.0	24.4
MOEA-D	complexity, similarity	21.7	183.5	18.8	19.5	50.4	22.7
MOEA-D	complexity, visitation length	22.3	135.4	28.2	21.4	28.1	26.6
MOEA-D	length	22.3	108.8	26.3	19.6	41.6	14.4
MOEA-D	length, similarity	21.2	135.9	26.0	18.8	39.2	15.2
MOEA-D	similarity	33.0	504.1	8.8	28.7	218.2	3.7
MOEA-D	visitation length	22.5	127.2	26.9	19.5	39.4	16.8
MOEA-D	visitation length, similarity	22.4	158.1	27.3	19.7	40.6	17.8
NSGA-II	complexity	20.7	4.0	19.8	22.9	4.2	34.1
NSGA-II	complexity, length	18.9	6.6	25.9	23.4	4.8	36.7
NSGA-II	complexity, similarity	21.8	12.7	24.3	26.5	7.4	42.4
NSGA-II	complexity, visitation length	18.6	10.3	24.3	22.7	4.6	36.6
NSGA-II	length	19.2	134.7	25.7	15.9	42.1	12.3
NSGA-II	length, similarity	20.6	297.4	27.9	16.5	70.1	13.4
NSGA-II	similarity	32.4	603.5	8.7	27.8	327.8	2.3
NSGA-II	visitation length	18.8	81.8	23.7	17.2	27.9	14.5
NSGA-II	visitation length, similarity	20.9	233.6	27.3	17.4	54.9	16.5

[5] Armand R. Burks and William F. Punch. 2017. An analysis of the genetic marker diversity algorithm for genetic programming. *Genetic Programming and Evolvable Machines* 18, 2 (June 2017), 213–245. <https://doi.org/doi:10.1007/s10710-016-9281-9>

[6] Armand R. Burks and William F. Punch. 2018. *An Investigation of Hybrid Structural and Behavioral Diversity Methods in Genetic Programming*. Springer International Publishing, Cham, 19–34. https://doi.org/10.1007/978-3-319-97088-2_2

[7] Bogdan Burlacu, Michael Affenzeller, Gabriel Kronberger, and Michael Komenda. 2019. Online Diversity Control in Symbolic Regression via a Fast Hash-based Tree Similarity Measure. *Accepted for publication in 2019 IEEE Congress on Evolutionary Computation abs/1902.00882* (2019). arXiv:1902.00882 <http://arxiv.org/abs/1902.00882> Preprint.

[8] Edwin D. de Jong and Jordan B. Pollack. 2003. Multi-Objective Methods for Tree Size Control. *Genetic Programming and Evolvable Machines* 4, 3 (Sept. 2003), 211–233. <https://doi.org/doi:10.1023/A:1025122906870>

[9] Edwin D. de Jong, Richard A. Watson, and Jordan B. Pollack. 2001. Reducing Bloat and Promoting Diversity using Multi-Objective Methods. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, Lee Specter, Erik D. Goodman, Annie Wu, W. B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahrnam Pezeshk, Max H. Garzon, and Edmund Burke (Eds.). Morgan Kaufmann, San Francisco, California, USA, 11–18. http://www.demo.cs.brandeis.edu/papers/rbpd_gecco01.pdf

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (April 2002), 182–197. <https://doi.org/10.1109/4235.996017>

[11] Stephen Dignum and Riccardo Poli. 2008. Operator Equalisation and Bloat Free GP. In *Proceedings of the 11th European Conference on Genetic Programming, EuroGP 2008 (Lecture Notes in Computer Science)*, Michael O’Neill, Leonardo Vanneschi, Steven Gustafson, Anna Isabel Esparcia Alcaraz, Ivanoe De Falco, Antonio Della Cioppa, and Ernesto Tarantino (Eds.), Vol. 4971. Springer, Naples, 110–121. https://doi.org/doi:10.1007/978-3-540-78671-9_10

[12] Aniko Ekart and S. Z. Nemeth. 2001. Selection Based on the Pareto Nondomination Criterion for Controlling Code Growth in Genetic Programming. *Genetic Programming and Evolvable Machines* 2, 1 (March 2001), 61–73. <https://doi.org/doi:10.1023/A:1010070616149>

[13] Jerome H. Friedman. 1991. Multivariate Adaptive Regression Splines. *Ann. Statist.* 19, 1 (03 1991), 1–67. <https://doi.org/10.1214/aos/1176347963>

[14] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. 2008. Evolutionary many-objective optimization: A short review. (2008). <https://doi.org/10.1109/CEC.2008.4631121>

[15] Michael Komenda, Gabriel Kronberger, Michael Affenzeller, Stephan Winkler, and Bogdan Burlacu. 2015. Evolving Simple Symbolic Regression Models by Multi-objective Genetic Programming. In *Genetic Programming Theory and Practice XIII (Genetic and Evolutionary Computation)*, Rick Riolo, William P. Worzel, M. Kotanchek, and A. Kordon (Eds.). Springer, Ann Arbor, USA. <https://doi.org/doi:10.1007/978-3-319-34223-8>

[16] John R. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA. <http://mitpress.mit.edu/books/genetic-programming>

[17] W. B. Langdon. 1998. The Evolution of Size in Variable Length Representations. In *1998 IEEE International Conference on Evolutionary Computation*. IEEE Press, Anchorage, Alaska, USA, 633–638. <https://doi.org/doi:10.1109/ICEC.1998.700102>

[18] Sean Luke. 2000. Two Fast Tree-Creation Algorithms for Genetic Programming. *IEEE Transactions on Evolutionary Computation* 4, 3 (Sept. 2000), 274–283.

- [19] Sean Luke and Liviu Panait. 2002. Lexicographic Parsimony Pressure. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska (Eds.). Morgan Kaufmann Publishers, New York, 829–836. <http://cs.gmu.edu/~sean/papers/lexicographic.pdf>
- [20] Riccardo Poli. 2003. A Simple but Theoretically-Motivated Method to Control Bloat in Genetic Programming. In *Genetic Programming*, Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 204–217.
- [21] Riccardo Poli. 2010. Covariant Tarpeian Method for Bloat Control in Genetic Programming. In *Genetic Programming Theory and Practice VIII*, Rick Riolo, Trent McConaghy, and Ekaterina Vladislavleva (Eds.). Genetic and Evolutionary Computation, Vol. 8. Springer, Ann Arbor, USA, Chapter 5, 71–90. <http://www.springer.com/computer/ai/book/978-1-4419-7746-5>
- [22] Michael Schmidt and Hod Lipson. 2010. Age-Fitness Pareto Optimization. In *Genetic Programming Theory and Practice VIII*, Rick Riolo, Trent McConaghy, and Ekaterina Vladislavleva (Eds.). Genetic and Evolutionary Computation, Vol. 8. Springer, Ann Arbor, USA, Chapter 8, 129–146. <http://www.springer.com/computer/ai/book/978-1-4419-7746-5>
- [23] Sara Silva and Ernesto 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 (2009), 141–179. <https://doi.org/doi:10.1007/s10710-008-9075-9>
- [24] Guido Smits and Mark Kotanchek. 2004. Pareto-Front Exploitation in Symbolic Regression. In *Genetic Programming Theory and Practice II*, Una-May O'Reilly, Tina Yu, Rick L. Riolo, and Bill Worzel (Eds.). Springer, Ann Arbor, Chapter 17, 283–299. https://doi.org/doi:10.1007/0-387-23254-0_17
- [25] Leonardo Vanneschi, Mauro Castelli, and Sara Silva. 2014. A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines* 15, 2 (01 Jun 2014), 195–214. <https://doi.org/10.1007/s10710-013-9210-0>
- [26] Ekaterina J. Vladislavleva, Guido F. Smits, and Dick den Hertog. 2009. Order of Nonlinearity as a Complexity Measure for Models Generated by Symbolic Regression via Pareto Genetic Programming. *IEEE Transactions on Evolutionary Computation* 13, 2 (April 2009), 333–349. <https://doi.org/doi:10.1109/TEVC.2008.926486>
- [27] S. Wagner, G. Kronberger, A. Beham, M. Kommenda, A. Scheibenpflug, E. Pitzer, S. Vonolfen, M. Kofler, S. Winkler, V. Dorfer, and M. Affenzeller. 2012. Architecture and Design of the HeuristicLab Optimization Environment. In *First Australian Conference on the Applications of Systems Engineering, ACASE (Topics in Intelligent Engineering and Informatics)*, Robin Braun, Zenon Chaczko, and Franz Pichler (Eds.), Vol. 6. Springer International Publishing, Sydney, Australia, 197–261. https://doi.org/doi:10.1007/978-3-319-01436-4_10 Selected and updated papers.
- [28] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (Dec 2007), 712–731. <https://doi.org/10.1109/TEVC.2007.892759>
- [29] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Technical Report.
- [30] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms — A comparative case study. In *Parallel Problem Solving from Nature — PPSN V*, Agoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 292–301.