An Empirical Study on the Feature Selection Ability of SLIM-GSGP

Davide Farinati
dfarinati@novaims.unl.pt
NOVA Information Management School (NOVA IMS),
Universidade Nova de Lisboa, Campus de Campolide
1070-312 Lisboa, Portugal

Leonardo Vanneschi
lvanneschi@novaims.unl.pt
NOVA Information Management School (NOVA IMS),
Universidade Nova de Lisboa, Campus de Campolide
1070-312 Lisboa, Portugal

Abstract

Feature Selection (FS) is a key characteristic of any Machine Learning method. Genetic Programming (GP) performs it inherently, using evolution pressure to exclude redundant or irrelevant features. However, this ability is lost in Geometric Semantic Genetic Programming (GSGP), where Geometric Semantic Operator (GSO) keep adding genetic material to the individuals, inevitably adding noisy features. This work focuses on comparing the FS abilities of GSGP and Semantic Learning algorithm based on Inflate and deflate Mutations (SLIM), a promising new variant that employs Deflate Geometric Semantic Mutation (DGSM), a genetic operator that is able to remove genetic material while still inducing an unimodal fitness landscape. The experimental results show how SLIM has superior FS abilities compared to GSGP.

CCS Concepts

• **Computing methodologies** → **Genetic programming**.

Keywords

Genetic Programming, Geometric Semantic Genetic Programming, Feature Selection, Symbolic Regression

ACM Reference Format:

Davide Farinati and Leonardo Vanneschi. 2025. An Empirical Study on the Feature Selection Ability of SLIM-GSGP. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3712255.3726642

1 Introduction

Machine Learning models must distinguish useful signals from noise to improve performance [1]. Feature Selection (FS) enhances models by retaining relevant data while filtering out noise. It is typically performed before training using methods like Entropy, Mutual Information, and SVM-RFE, though no single method consistently outperforms others [1]. Thus, models capable of performing FS inherently during training are beneficial.

Genetic Programming (GP) naturally performs FS due to its stochastic initialization and evolutionary operators [2]. It constructs initial solutions using randomly drawn features and applies mutation and crossover, implicitly filtering out irrelevant ones. This ability makes GP effective for high-dimensional or noisy datasets by reducing overfitting and enhancing generalization [3].

Geometric Semantic Genetic Programming (GSGP) [4], a GP variant, employs geometric semantic operators, creating a unimodal fitness landscape that eliminates local optima. However, these operators continuously add genetic material, increasing solution size and reducing interpretability. As later generations incorporate all dataset features, FS capabilities diminish, leading to noisy solutions [5].

Semantic Learning algorithm based on Inflate and deflate Mutations (SLIM) [6, 7], a recent GSGP variant, introduces Deflate Geometric Semantic Mutation (DGSM), a Geometric Semantic Operator (GSO) that removes genetic material while preserving a unimodal fitness landscape. This study compares the FS capabilities of GSGP and SLIM to assess whether DGSM improves feature selection by steering evolution toward noise-free solutions.

This work evaluates SLIM and GSGP on datasets augmented with noisy features. The contributions include an in-depth analysis of their FS abilities.

The paper is structured as follows: Section 2 introduces GSGP and SLIM. Section 5 presents parameters and datasets, while Section 4 details the experimental methodology. Section 5 discusses results, and Section 6 concludes with future research directions.

2 Background

This section briefly introduces GSGP and SLIM. It begins with an overview of GSGP, explaining its core mutation operator i.e., Inflate Geometric Semantic Mutation (IGSM). Then, it introduces the studied variants of SLIM and its new mutation operator i.e., DGSM, also focusing on a convenient way of representing SLIM's individuals by means of linked-list structures.

2.1 Geometric Semantic Genetic Programming

In GP, the term *semantics* refers to the vector of output values generated by a program for a given set of observations. GSGP [4] is a variation of GP that substitutes traditional syntax-based genetic operators with GSOs. These operators exhibit a predictable impact on the semantics of individuals, introducing geometric properties into the semantic space. Specifically, the Geometric Semantic Mutation (GSM) creates a mutated function from a parent $\mathcal{T} : \mathbb{R}^n \rightarrow \mathbb{R}$ as: $mathcalT_M = \mathcal{T} + ms \cdot (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions with outputs in $[0, 1]$, and ms is a parameter referred to as the *mutation step*. GSM functions as a ball mutation in the semantic space, producing offspring within a sphere of radius ms

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726642

centered around the parent's semantics. One of the key strengths of GSGP is that it creates a unimodal error surface, thereby simplifying the search process by removing local optima. For an in-depth explanation of how GSOs achieve a unimodal error surface, readers can refer to [8]. However, GSM have a notable drawback: by design, it always generates offspring that are larger than their parents. This leads to rapid growth in the size of individuals during evolution, posing challenges for computational efficiency and interpretability. This is why this GSM is also called "Inflate" GSM (IGSM).

In the last decade, different research have shown that mutation is a more powerful operator than crossover for GSGP, with mutation-only GSGP returning results that are comparable, or even better, than the ones obtained when both crossover and mutation are used [9]. For this reason, we consider exclusively GSGP with GSM in this study.

2.2 SLIM

SLIM-GSGP [6, 7], or more simply SLIM, is an extension of GSGP that preserves the property of inducing a unimodal error surface while generating smaller models. As its name implies, it uses two types of mutation: it maintains GSGP's IGSM and it introduces the DGSM, designed to generate offspring that are smaller than their parents.

The definition and validity of DGSM are based on two key observations: first, GSM can be reformulated as $\text{GSM}(T) = T + ms \cdot (T_{R1} - T_{R2}) = T - ms \cdot (T_{R2} - T_{R1})$, and second, the random trees T_{R1} and T_{R2} are interchangeable since they are independently sampled from the same distribution. Thus, the GSM can equivalently be expressed as: $\text{GSM}(T) = T - ms \cdot (T_{R1} - T_{R2})$.

This reformulation is key for the generation of offspring that are smaller than their parents. In fact, consider an individual T to which GSM is applied, say, three consecutive times. The resulting individual is: $T_M = T + ms \cdot (T_{R1} - T_{R2}) + ms \cdot (T_{R3} - T_{R4}) + ms \cdot (T_{R5} - T_{R6})$. Now, if we apply GSM again, but this time using subtraction instead of addition, the resulting individual becomes: $T_M = T + ms \cdot (T_{R1} - T_{R2}) + ms \cdot (T_{R3} - T_{R4}) + ms \cdot (T_{R5} - T_{R6}) - ms \cdot (T_{R7} - T_{R8})$.

It is now important to note that a commonly adopted approach in the literature is to reuse random trees instead of generating new ones [10]. For instance, instead of generating new trees T_{R7} and T_{R8}, we could reuse T_{R3} and T_{R4}. This would yield: $T_M = T + ms \cdot (T_{R1} - T_{R2}) \cancel{+ ms \cdot (T_{R3} - T_{R4})} + ms \cdot (T_{R5} - T_{R6}) \cancel{- ms \cdot (T_{R3} - T_{R4})}$. After this simplification, the generated individual has a smaller genotype. Using this idea, the DGSM operator effectively removes one of the terms previously added by IGSM, while still performing a geometric semantic mutation.

Both IGSM and DGSM perturb the semantics of the individual within the same range, specifically $[-ms, ms]$, where ms is the mutation step parameter. This enables the reduction of individual sizes while preserving the semantic properties necessary for the search process.

2.2.1 SLIM Variants. In [7, 11], the authors investigated three methods (2SIG, ABS, 1SIG) for defining functions that return values within the range $[-ms, ms]$, as well as two distinct approaches for perturbing an individual to achieve ball mutation in the semantic space. Here, we define two of the three functions that return values within $[-ms, ms]$ as follows:

(1) Using two random trees T_{R1} and T_{R2}, each passed through the sigmoid function S (which maps outputs to $[0, 1]$): $2\text{SIG} = ms \cdot (S(T_{R1}) - S(T_{R2}))$, This corresponds to the traditional GSM.
(2) Employing a single random tree T_R normalized using the absolute value: $\text{ABS} = ms \cdot (1 - 2/(1 + |T_R|))$.
(3) Using a single random tree T_R passed through the sigmoid function S: $1\text{SIG} = ms \cdot (2 \cdot S(T_R) - 1)$.

Small perturbations of a numeric value can be achieved in two ways: by adding a small positive or negative quantity close to zero, or by multiplying by a factor slightly smaller or larger than one. Accordingly, given a function like 1SIG, 2SIG or ABS, there are two methods for perturbing an individual within the SLIM framework:

(1) IGSM adds the function, while DGSM subtracts it.
(2) IGSM multiplies by $1 +$ function, while DGSM divides by the same quantity.

By combining these two perturbation methods with the three mutation functions , we obtain six SLIM and GSGP variants. These are denoted as SLIM+, SLIM∗, GSGP+ or GSGP∗, depending on whether they use addition or multiplication, followed by 1SIG, 2SIG or ABS, indicating the function used for defining the perturbation. For example, the variant called SLIM+2SIG performs ball mutation using the sum of a small positive or negative value close to zero, with 2SIG as the function defining perturbations within $[-ms, ms]$.

3 Experimental Settings

We selected seven real-life symbolic regression problems as test cases, each characterized by datasets of varying dimensions. These problems are frequently used as benchmarks in the GP literature (e.g., [9, 12–14]). Among these, the *ppb*, *bioavailability*, and *ld50* datasets are known to be prone to overfitting [12], making them particularly useful for evaluating the generalization capabilities of the algorithms under study. The used datasets exhibit diverse characteristics. The *yacht* dataset has 307 observations and 6 features, while the *airfoil* dataset has 1,502 observations and 5 features. The *concrete slump* dataset has 102 observations and 9 features, and the *concrete strength* dataset has 1,029 observations with 8 features. For datasets with larger feature sets, *ppb* has 131 observations with 626 features, *bioavailability* has 359 observations and 241 features, and *ld50* has 234 observations with 626 features.

The experiments were conducted letting the methods evolve for 1,000 generations, with a population size of 100 individuals. Elitism was enabled, ensuring the best individual from each generation was carried over unchanged to the next. The maximum initial depth for individuals was set to 6. The function set consisted of basic arithmetic operations (addition, subtraction, multiplication, and protected division [15]), represented as $\{+, -, \times, \div\}$. Tournament selection [3] with a tournament size of 2 was applied, and the Ramped Half-and-Half initialization method [3] was employed. Following previous studies [6, 7], no numeric constants were included in the terminal symbol set. The probability of applying the IGSM operator was set to 0.3 for all datasets except *ld50* (where it was 0.1) and *concrete strength* (where it was 0.5). The probability of using the DGSM operator was always equal to 1 minus the IGSM probability. The mutation step was always obtained from a uniform distribution with range $[0, \tilde{y}]$, where $\tilde{y}$ is the median of the target variable for each dataset [16]. Monte Carlo cross-validation [17, 18] was employed, with each train-test split generated randomly for every run. For each split, 70% of the observations were allocated to the training set, and 30% to the test set.

The code for this study is publicly available in our library, hosted at the following web portal: slim[1].

4 Experimental Methodology

This work investigates and compares the feature selection (FS) capabilities of GSGP and SLIM. To achieve this, two sets of experiments are conducted.

The experiment conducted in this work, inspired by the study of Farinati and Vanneschi [5], augments the datasets presented in Section 3 with noise. Two types of noise are introduced in this experiment. The first type is pure random noise, drawn from a standard normal distribution, hereafter referred to as Random Noise (RN). The second type involves taking a copy of an existing feature (or column) and shifting it with Gaussian noise, proportional to the column's standard deviation, from now on referred to as Shifted Column Noise (SCN). Only one type of noise is added at a time, and either 1, 5, 10, or 50 noisy columns are introduced. GSGP and SLIM are then trained on datasets with and without noise, and their respective performance is compared. The assumption is that if the performance does not deteriorate when noise is added to the dataset, then the algorithm has strong FS capabilities and can effectively distinguish between noise and signal, and the opposite assumption also holds.

5 Experimental Results

Table 1 and Table 2 summarize the findings of this work.

Both tables follow the same structure, displaying the number of times the model trained on noisy data was significantly outperformed by the model trained on standard data. The statistical significance was assessed using the Mann-Whitney U test [19] across 30 independent runs, with a significance level of $\alpha = 0.05$. The comparison includes both training and testing RMSE across all datasets considered. The maximum possible value is 4, indicating that the model trained on noisy data was outperformed under all noise addition scenarios: specifically, when 1, 5, 10, or 50 noisy columns were introduced. Conversely, the minimum possible value is 0, which indicates that the model trained on noisy data was never outperformed by the model trained on standard data. Having a high value indicates that the model is not able to identify and exclude noisy features, therefore showing a lack of FS ability. On the other hand, a low number indicates the opposite.

Table 1 presents the results when the RN noise creation method was used. From this table, it can be observed that all variants of SLIM exhibit either superior or comparable FS performance to their corresponding GSGP models across all datasets, with the sole exception of the yacht dataset when the +2SIG GSM is used. A similar pattern is evident in Table 2, which shows the results for the SCN noise creation method.

Across both noise creation methods, SLIM demonstrates better FS performance in 19 test cases and worse performance in only 2 cases (specifically, the yacht dataset when using the +2SIG mutation). In all other test cases, SLIM matches or outperforms the FS performance of GSGP, including 12 cases where both algorithms achieve a value of 0. This value indicates that the model trained on noisy data was never outperformed by the model trained on standard data.

Notably, most instances where a value of 0 is observed occur with the ppb, bioav, and ld50 datasets. This is attributed to these datasets having a large number of features—626, 241, and 626 respectively—compared to the remaining four datasets. In fact, a larger number

[1] https://github.com/DALabNOVA/slim

Table 1: The count of instances where the model trained on noisy data was significantly outperformed by the model trained on standard data (as determined by the Mann-Whitney U test across 30 runs with significance level $\alpha = 0.05$) according to the RMSE is reported. The results are provided for train and test performance. The method used to generate noise was RN.

		+2SIG		*1SIG		*ABS	
	dataset	**GSGP**	**SLIM**	**GSGP**	**SLIM**	**GSGP**	**SLIM**
Train	yatch	3	4	4	3	4	3
	airfoil	4	4	4	1	2	1
	slump	4	0	3	3	3	3
	strength	4	3	3	3	3	2
	ppb	4	0	0	0	0	0
	bioav	4	0	1	0	1	0
	ld50	4	0	0	0	0	0
Test	yatch	2	4	4	3	4	3
	airfoil	4	4	4	1	2	1
	slump	3	3	3	2	2	2
	strength	4	3	3	3	2	2
	ppb	2	0	0	0	0	0
	bioav	4	2	0	0	2	1
	ld50	4	0	0	0	0	0

Table 2: Results reported using the methodology followed in Table 1. The method used to generate noise was SCN.

		+2SIG		*1SIG		*ABS	
	dataset	**GSGP**	**SLIM**	**GSGP**	**SLIM**	**GSGP**	**SLIM**
Train	yatch	1	3	3	3	3	2
	airfoil	4	4	4	1	1	1
	slump	4	0	3	3	1	2
	strength	4	3	3	3	1	2
	ppb	4	0	0	0	0	0
	bioav	4	0	1	0	1	0
	ld50	4	0	0	0	0	0
Test	yatch	0	3	3	3	3	1
	airfoil	4	4	4	1	1	1
	slump	3	2	2	0	0	0
	strength	4	3	3	3	2	2
	ppb	2	0	0	0	0	0
	bioav	4	2	0	0	2	1
	ld50	0	0	0	0	0	0

of features diminishes the impact of noise addition, reducing the likelihood of a noisy feature influencing an individual's structure.

As mentioned earlier, SLIM+2SIG is the only SLIM variant with worse FS performance compared to its corresponding GSGP variant. This can be explained by the fact that the additive GSM has a smaller impact on offspring semantics than the multiplicative GSM. To convince oneself of this fact, one should consider, for instance a perturbation, given by 1SIG, 2SIG or ABS, equal to 0.1. If we consider an individual P with semantics $S_P = [3, 5, 6, 9]$, and we apply the sum, we obtain: $S_P + 0.1 = [3.1, 5.1, 6.1, 9.1]$, while if we apply the product we obtain $S_P \cdot 1.1 = [3.3, 5.5, 6.6, 9.9]$. From this example, we can clearly see that the multiplicative GSM produces a bigger perturbation of the semantics of the individual than the additive GSM. This smaller semantic impact of the additive GSM makes it less detrimental when noisy features are appended to a solution's structure in SLIM+2SIG, compared to SLIM*1SIG and SLIM*ABS. Consequently in SLIM+2SIG, solutions with noisy blocks are more likely to persist in the evolving population. Finally, Table 1 and Table 2 highlight that SLIM consistently outperforms GSGP in FS capabilities, underscoring one of the advantages of incorporating DGSM into the evolutionary process.

6 Conclusion and Future Work

This study investigated the Feature Selection (FS) capabilities of SLIM and its variants, with a focus on handling noisy datasets. Across different types of noise, we compared the performance of SLIM, GSGP.

The conducted experiments assessed the ability of SLIM and GSGP to perform feature selection in the presence of noise. The results demonstrated that SLIM consistently outperformed or matched the performance of GSGP in identifying and excluding noisy features, across all datasets and noise levels. However, SLIM+2SIG was the only variant that showed slightly worse performance in the presence of noise, particularly when using the +2SIG mutation on the yacht dataset. This result can be attributed to the smaller semantic impact of the additive Geometric Semantic Mutation (GSM), which makes it less effective at excluding noisy features compared to the multiplicative GSM variants.

In conclusion, SLIM demonstrates strong feature selection capabilities, outperforming traditional GSGP and offering a viable solution for noisy datasets.

Future work will focus on improving the DGSM, developing new techniques to identify harmful blocks in the solutions genotype. The extension of this analysis to other datasets should also be considered. Specifically observing the algorithms performance on epistatic datasets, such as the one created with GAMETES [20] and already used in other studies [21]. Also, future work should focus on analyzing the FS abilities of the newly introduced Deflating Geometric Crossovers [22].

Acknowledgments This work was supported by national funds through FCT (Fundação para a Ciência e a Tecnologia), under the project UIDB/04152/2020 (DOI: 10.54499/UIDB/04152/2020) – Centro de Investigação em Gestão de Informação (MagIC)/NOVA IMS.

References

[1] Matheus Cezimbra Barbieri, Bruno Iochins Grisci, and Márcio Dorn. Analysis and comparison of feature selection methods towards performance and stability. *Expert Syst. Appl.*, 249:123667, September 2024. ISSN 0957-4174. doi: 10.1016/j.eswa.2024.123667.
[2] Durga Prasad Muni, Nikhil R. Pal, and Jyotirmoy Das. Genetic programming for simultaneous feature selection and classifier design. *IEEE Trans. Syst. Man Cybern. B Cybern.*, 36(1):106–117, February 2006. ISSN 1083-4419. doi: 10.1109/tsmcb.2005.854499.
[3] Riccardo Poli, William B. Langdon, and Nicholas F. Mcphee. *A field guide to genetic programming.* March 2008.
[4] Alberto Moraglio, Krzysztof Krawiec, and Colin G. Johnson. Geometric semantic genetic programming. In Carlos A. Coello Coello, Vincenzo Cutello, Kalyanmoy Deb, Stephanie Forrest, Giuseppe Nicosia, and Mario Pavone, editors, *Parallel Problem Solving from Nature*, volume 7491 of *LNCS*, pages 21–31. Springer, 2012.
[5] Davide Farinati and Leonardo Vanneschi. A study on variations of Genetic Programming applied to time series forecasting: Machine Learning for Energy Consumption Forecasting, October 2022. URL https://run.unl.pt/handle/10362/145478?locale=en.
[6] Leonardo Vanneschi. SLIM_GSGP: The non-bloating geometric semantic genetic programming. In Mario Giacobini, Bing Xue, and Luca Manzoni, editors, *Genetic Programming*, pages 125–141, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-56957-9.
[7] Leonardo Vanneschi, Davide Farinati, Diogo Rasteiro, Liah Rosenfeld, Gloria Pietropolli, and Sara Silva. Exploring non-bloating geometric semantic genetic programming. *Genetic Programming Theory and Practice*, to appear.
[8] Leonardo Vanneschi and Sara Silva. *Lectures on Intelligent Systems.* Springer, 2023.
[9] Mauro Castelli, Luca Manzoni, Ivo Gonçalves, Leonardo Vanneschi, Leonardo Trujillo, and Sara Silva. An analysis of geometric semantic crossover: A computational geometry approach. pages 201–208, 01 2016. doi: 10.5220/0006056402010208.
[10] Leonardo Vanneschi, Mauro Castelli, Luca Manzoni, and Sara Silva. A new implementation of geometric semantic gp and its application to problems in pharmacokinetics. In *Genetic Programming: 16th European Conference, EuroGP 2013, Vienna, Austria, April 3-5, 2013. Proceedings 16*, pages 205–216. Springer, 2013.
[11] Illya Bakurov, José Manuel Muñoz Contreras, Mauro Castelli, Nuno Rodrigues, Sara Silva, Leonardo Trujillo, and Leonardo Vanneschi. Geometric semantic genetic programming with normalized and standardized random programs. *Genetic Programming and Evolvable Machines*, 25(1), feb 2024. ISSN 1389-2576. doi: 10.1007/s10710-024-09479-1. URL https://doi.org/10.1007/s10710-024-09479-1.
[12] Francesco Archetti, Stefano Lanzeni, Enza Messina, and Leonardo Vanneschi. Genetic programming for computational pharmacokinetics in drug discovery and development. *Genetic Programming and Evolvable Machines*, 8(4):413–432, 2007. ISSN 1573-7632.
[13] Sara Silva and Leonardo Vanneschi. Operator equalisation, bloat and overfitting: a study on human oral bioavailability prediction. In Franz Rothlauf, editor, *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009*, pages 1115–1122. ACM, 2009.
[14] Ivo Gonçalves, Sara Silva, Carlos M. Fonseca, and Mauro Castelli. Unsure when to stop? In *Proceedings of the Genetic and Evolutionary Computation Conference.* ACM, jul 2017. doi: 10.1145/3071178.3071328.
[15] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.
[16] Davide Farinati, Gloria Pietropolli, and Leonardo Vanneschi. Exploring the impact of data scale on mutation step size in SLIM-GSGP. In Bing Xue, Luca Manzoni, and Illya Bakurov, editors, *European Conference on Genetic Programming, EuroGP 2025*, LNCS, Trieste, 23-25 April 2025. Springer Nature. Forthcoming.
[17] Werner Dubitzky, Martin Granzow, and Daniel P. Berrar. *Fundamentals of Data Mining in Genomics and Proteomics.* Springer, 2006. ISBN 0387475087.
[18] Leonardo Vanneschi and Sara Silva. *Lectures on Intelligent Systems.* Natural Computing Series. Springer, 2023. ISBN 978-3-031-17921-1. doi: 10.1007/978-3-031-17922-8. URL https://doi.org/10.1007/978-3-031-17922-8.
[19] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Ann. Math. Stat.*, 18(1):50–60, March 1947. ISSN 0003-4851. doi: 10.1214/aoms/1177730491.
[20] Ryan J. Urbanowicz, Jeff Kiralis, Nicholas A. Sinnott-Armstrong, Tamra Heberling, Jonathan M. Fisher, and Jason H. Moore. GAMETES: a fast, direct algorithm for generating pure, strict, epistatic models with random architectures. *BioData Mining*, 5(1):1–14, December 2012. ISSN 1756-0381. doi: 10.1186/1756-0381-5-16.
[21] Nuno M. Rodrigues, João E. Batista, William La Cava, Leonardo Vanneschi, and Sara Silva. Exploring SLUG: Feature Selection Using Genetic Algorithms and Genetic Programming. *SN Comput. Sci.*, 5(1):1–17, January 2024. ISSN 2662-995X. doi: 10.1007/s42979-023-02106-3.
[22] Gloria Pietropolli, Davide Farinati, Luca Manzoni, Mauro Castelli, and Leonardo Vanneschi. Introducing crossover in SLIM-GSGP. In Bing Xue, Luca Manzoni, and Illya Bakurov, editors, *European Conference on Genetic Programming, EuroGP 2025*, LNCS, Trieste, 23-25 April 2025. Springer Nature. Forthcoming.