mathematics

MDPI

Article

A Bilevel Optimization Strategy for the Mapping Process in Grammatical Evolution

Blanca Verónica Zuñiga-Núñez [1], Erick Israel Guerra-Hernández [2], Patricia Batres-Mendoza [2], Itandehui Belem Gallegos-Velasco [1] and Marco Aurelio Sotelo-Figueroa [3,*]

[1] Facultad de Medicina y Cirugía, Universidad Autónoma Benito Juárez de Oaxaca, Oaxaca de Juárez 68120, Mexico; zunb941006.fmc@uabjo.mx (B.V.Z.-N.); itanbel668@gmail.com (I.B.G.-V.)
[2] Facultad de Sistemas Biológicos e Innovación Tecnológica, Universidad Autónoma Benito Juárez de Oaxaca, Oaxaca de Juárez 68120, Mexico; ghernandez.cat@uabjo.mx (E.I.G.-H.); bmendoza.cat@uabjo.mx (P.B.-M.)
[3] División de Ciencias Económico-Administrativas, Universidad de Guanajuato, Guanajuato 36250, Mexico
* Correspondence: masotelo@ugto.mx

Abstract

Grammatical Evolution is a technique for evolving structured solutions using a genotype-to-phenotype mapping based on formal grammar. Optimization is typically performed on the genotype values. However, the use of permutations to guide mapping has been shown to improve solution performance. Traditionally, the focus has been on single-level optimization of the genotype values or the order in which codons are consumed during mapping. This motivates the use of a bilevel approach that enables the simultaneous optimization of genotype values and their ordering. This paper proposes a bilevel Grammatical Evolution approach called BiGE, in which the upper level optimizes the genotype values, whereas the lower level optimizes the order in which these same values are used in the mapping process. The approach was implemented using both Depth-First (DF) and Breadth-First (BF) mapping processes, which are the canonical mapping strategies in Grammatical Evolution. The proposed approach was evaluated on the Symbolic Regression Problem and compared with state-of-the-art approaches under identical experimental conditions. Statistical tests were performed to determine whether significant differences existed among the compared approaches. The results revealed statistically significant differences, indicating that incorporating a bilevel optimization strategy into the mapping process can improve the quality of the generated solutions.

Keywords: grammatical evolution; mapping process; bilevel optimization; symbolic regression

MSC: 68W50

Check for updates

Academic Editor: Jonathan Blackledge

Received: 26 May 2026
Revised: 16 June 2026
Accepted: 18 June 2026
Published: 26 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Evolutionary Computation (EC) encompasses evolutionary optimization techniques inspired by the principles of natural evolution, such as selection and inheritance. EC has been used to address a wide range of tasks, including the automatic generation of structures that must satisfy specific constraints [1,2]. In this context, it is not sufficient to explore the search space arbitrarily; therefore, several techniques have been developed to efficiently guide the search process toward feasible solutions. An important approach focuses on incorporating domain knowledge into the process, enabling the generation of solutions that comply with specific rules.

Among these approaches, Grammatical Evolution (GE) [3,4], a variant of Genetic Programming (GP), combines evolutionary search with formal grammars to incorporate domain knowledge. Through the use of grammars, GE enables the automatic construction of solutions that satisfy syntactic rules while embedding domain knowledge into the search process. Unlike general-purpose evolutionary or swarm-based algorithms such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), or Differential Evolution (DE), GE explicitly separates the genotype from the phenotype. More importantly, GE provides a well-defined grammar-guided process whose behavior can be isolated, modified, and studied. GE has the following main components [5]: the problem instance to be solved, a Grammar (G) that specifies the syntactic rules that determine how solutions are constructed, a Search Engine (SE) responsible for evolving candidate solutions, and a Mapping Process (MP) that translates genotypes into phenotypes according to grammatical rules. These elements have been studied separately as they affect the performance of GE [3,4,6,7]. Given the influence of GE's components on its performance, this work focuses on improving the MP. This proposal is inspired by πGrammatical Evolution (πGE) [8], which proposes using one part of the genotype as normal and using the other part to choose the expansion order. This process can be assessed as bilevel optimization, as it implies two decisions: the value of the codon and the order in which these codons are used.

The Bilevel Programming Problem (BLPP) generalizes the Stackelberg leader–follower game [9]. In bilevel optimization, the upper-level problem (leader) is constrained by the set of solutions of a lower-level optimization problem (follower). The objective is to optimize (minimize or maximize) a given function subject to constraints related to an optimal solution set of another optimization problem.

Some algorithms, such as the Memetic Algorithm (MA), share similarities with the BLPP [10,11]. In both the BLPP and the MA, multiple optimization processes are involved; however, they differ in their objectives. The MA combines a local search process that improves convergence with a global search strategy, such as an evolutionary algorithm; the local search does not represent an independent optimization problem but instead serves as an intensification mechanism. BLPP, on the other hand, is distinguished by a hierarchical leader–follower structure in which the best answer to a lower-level optimization problem determines the solution to the upper-level problem.

Although previous studies have shown that the MP has a significant effect on the performance of GE [6–8], most existing approaches modify the mapping order through fixed strategies or random permutations. These approaches demonstrate that the codon consumption order is relevant; however, they do not explicitly formulate the interaction between codon values and codon consumption order as two interdependent optimization problems.

This research proposes a bilevel optimization strategy for the MP in GE, designated BiGE, which the interaction between genotype values and their mapping order as a BLPP. The lower level optimizes the permutation that determines the codon consumption order during the MP, and the upper level optimizes the genotype values. Therefore, the lower level is an independent optimization problem whose resolution directly affects the evaluation of the upper level, rather than a local improvement process.

The flexibility of GE, derived from the use of formal grammars, allows it to be applied to a wide range of problems. In this work, the Symbolic Regression Problem (SRP) [12] is used to validate the proposed approach, as it provides a suitable scenario to analyze the impact of the MP on GE's performance. The SRP is the process of searching for a mathematical expression that best fits a given data set without making prior assumptions about the structure of the model. The search space consists of all possible mathematical expressions that can be built by combining basic expressions, making it a highly challenging

problem. In this context, the MP plays a crucial role in translating genotype values into syntactically valid expressions. In addition, the SRP is a widely used benchmark problem for Evolutionary Computation (EC), particularly for GP [13], owing to these characteristics.

The proposed approach is evaluated against state-of-the-art methods under the same experimental conditions to ensure a fair comparison, using the same number of function calls and instances. The results show that incorporating a bilevel optimization approach into the MP improves the quality of the solutions and enables this process to the classical MPs.

The main research question addressed in this work is as follows: can an MP be improved by simultaneously optimizing codon values and the order in which they are consumed using a bilevel optimization framework?

To address this research question, the present study makes the following main contributions to the literature:

1. A bilevel optimization strategy where the genotype and the codon consumption order are modeled as interconnected optimization problems;
2. BiGE, a new strategy for GE's MP to optimize codon values and their order of consumption modeled as a bilevel optimization problem;
3. A comparison between BiGE and state-of-the-art GE approaches using classical SRP instances;
4. A statistical analysis using the Friedman test and the post hoc Nemenyi test, along with an ablation study to evaluate the contribution of each level to the overall performance of the proposed approach.

The remainder of this article is organized as follows. Section 2 presents a review of the literature on GE and recent developments, the SRP and its relevance in GP, recent approaches in BLPP, and recent research on GE components. Section 3 describes the proposal. Section 4 presents and interprets the results obtained with the proposed approach. Finally, Section 5 presents the conclusions and describes potential future work.

2. Literature Review

This section presents a review of the relevant literature on GE, bilevel optimization, and related research that provides the foundation for this work.

GE is an algorithm that allows the construction of solutions that meet specific rules. It is a flexible tool for finding different types of structures depending on the problem and the grammar used, and it has been applied in diverse situations. One of the most popular uses is the design of topologies for Artificial Neural Networks (ANNs), in which the more complex the problem is, the more difficult it is to find the correct topology and parameters. The goal is to find the best topology and parameter configuration without requiring expert knowledge. For example, in [14], GE and a Genetic Algorithm (GA) are used to create simpler ANN topologies that achieve higher accuracy; in [15], the authors present a new method for neural network evolution that evolves both the network topology and the network parameters. Similarly, GE has been used to create topologies of other types of ANNs, such as Convolutional Neural Networks (CNNs) [16].

Beyond topology design, other studies focus only on hyperparameter optimization for CNNs. For instance, Ref. [17] proposes HyperGE, a two-stage model for automatically tuning hyperparameters driven by GE. In the first stage, the algorithm analyzes the behavior during evolution; then, using this analysis in the second stage, the grammar is refined and the search space reduced. Another example is a study [15] that proposes using GE that first focuses on finding a promising interval for the optimal parameters and later uses a standard optimization technique to train the network using this interval.

In the field of Automated Machine Learning (AutoML), specifically in automatic workflow composition, GE has proven to be a tool capable of efficiently designing workflow

structures to optimize the structure and hyperparameters [18]. Along the same line of research, Ref. [19] proposes an AutoML approach that generates and optimizes machine learning pipelines with complex directed acyclic graph (DAG) structures using heuristic search guided by context-free grammars. GE has also been used to design different heuristics and hyper-heuristics, such as evolutionary multi-objective optimization and hyper-heuristics (MOHHs) [20].

A relevant benchmark in this field is the SRP, which has been widely used as a benchmark to evaluate GP and GE algorithms [12,21]. Its relevance lies in the need to evolve the structure and parameters of mathematical functions without assuming a specific function. These characteristics make the SRP a suitable problem for these algorithms, since it takes advantage of the genotype-to-phenotype MP to generate syntactically valid expressions. The SRP has been addressed using different methods; recently, it has gained popularity in the field of Machine Learning (ML) [22]. For example, in [23,24], this problem is assessed using transformers. In particular, in GE, it has been a tool for analyzing the performance of different proposals. For instance, in [25,26], the authors introduced AutoGE, a framework designed to help in the automatic configuration of GE parameters, with special emphasis on grammar design. Similarly, Ref. [27] proposed WebGE, a user interface that allows the use of GE and Differential Evolution (DE), specifically developed to address the SRP.

The BLPP is defined as a mathematical program in which an optimization problem contains another optimization problem as a constraint. This problem is useful in scenarios involving nested decision processes, where the solution to an upper-level problem depends on the response of a lower-level problem. This has motivated the development of tools to solve this type of problem [28–30]. The BLPP has gained attention in computer science for its ability to model complex tasks. Moreover, the BLPP has been successfully applied in ML, in tasks such as hyperparameter optimization [31–34] or reinforcement learning [35]. In [36], a framework is proposed in which neural networks are used to approximate value functions within a mixed-integer framework to efficiently solve the BLPP. This shows that it is possible to model ML problems using bilevel optimization.

Within GE, one important component is the MP, which is responsible for mapping a genotype to a phenotype using a grammar. This MP has been studied, and several proposals have been made to carry it out. Classical techniques include Depth-First (DF) [4], Breadth-First (BF) [37], and πGE [8]. Because MP significantly influences the performance of GE, it has been studied to improve its performance. For example, Ref. [7] proposes incorporating the use of permutations for codon selection into the standard process, showing that random permutations in the MP help to achieve better results. Taking this into account, Ref. [6] proposes a new approach to intensify the solution by using permutation heuristics to guide the codon selection order. The reviewed studies indicate that the MP is not a neutral component of GE; rather, the order in which codons are consumed can directly affect the structure and quality of the generated phenotypes. Classical strategies define a deterministic MP, while πGE and the permutation-based approaches show that modifying the order in which the codons are consumed can improve performance. However, existing studies mainly treat the MP as a predefined rule or an encoded mechanism. This motivates the proposed bilevel approach where the first level focuses on optimizing the value of the codons and the second focuses on optimizing the permutation.

3. Methods

3.1. Grammatical Evolution

GE is a variant of GP that incorporates a formal grammar to guide the generation of valid solutions and delimits the search space. This algorithm is inspired by biological

evolutionary processes, in which populations of individuals adapt to their environment through various mechanisms such as selection, mutation, and crossover. In the context of GE, each individual in a population is represented by a genotype, which is usually a string of integers that is later transformed into a phenotype through a MP. This phenotype is a representation that corresponds to a candidate solution to the problem being addressed. The MP is carried out by following a set of syntactic rules defined by the grammar, ensuring that the generated solutions are syntactically correct and therefore valid within the space of possible solutions. Figure 1 illustrates the general workflow of an MP. Given a genotype and a grammar, a derivation tree is constructed by expanding non-terminal symbols according to grammatical rules. From this tree, the phenotype is obtained as a valid expression that represents a candidate solution.

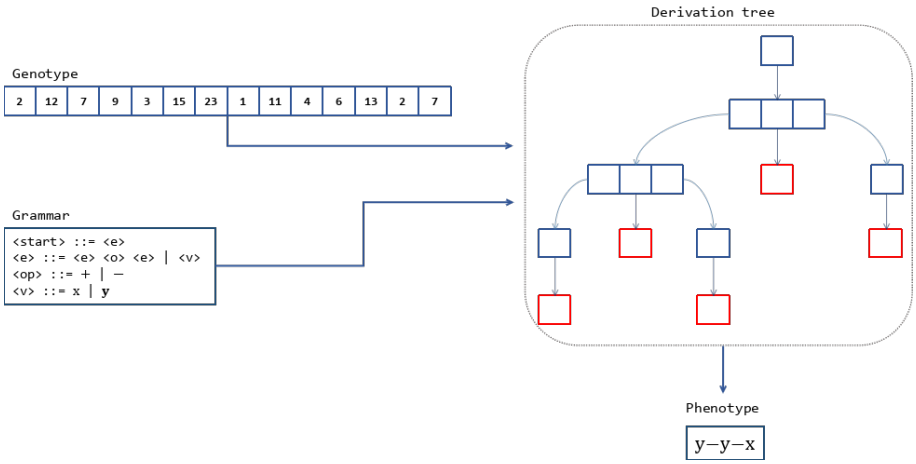


Figure 1. The MP combines the genotype and the grammar to construct a derivation tree. The leaves of this tree, highlighted in red, are then used to generate the phenotype.

3.2. Bilevel Optimization Problem

The proposed approach is inspired by the original Bilevel Programming Problem (BLPP) [9] and is implemented within the MP. In this process, two independent decisions are made: selecting codon values of the genotype and determining the order in which these codons are consumed during grammar expansion. This interaction could be seen as a hierarchical decision structure that can be modeled as a BLPP. In this proposal, the upper-level objective $F_U(x, y)$ optimizes the codon values of the genotype, and the lower-level objective $F_L(x, y)$ determines the order in which these values are used in the MP. Formally,

$$\min_{x \in S} \quad F_U(x, \min_{y \in S_n} F_L(x, y)) \tag{1}$$

subject to

$$y \in S_n \tag{2}$$

$$x \in [0, 255] \tag{3}$$

where the notation is defined as follows:

$F_L : \{x\} \times S_n \longrightarrow \mathbb{R}^+$ is the objective function of the lower level;
$F_U : S \times S_n \longrightarrow \mathbb{R}^+$ is the objective function of the upper level;
y denotes the permutation that controls the codon consumption order;
x is the genotype;
S is the search space;
S_n is the set of all n-element permutations.

Each genotype value, or codon, is restricted to the range $[0, 255]$, following the traditional GE representation in which codons are encoded as 8-bit integers [3,38]. This bound is not dictated by the formal grammar itself. During mapping, a codon selects a production rule through Equation (5). Consequently, increasing the codon range would not expand the set of selectable productions but would increase the number of codon values that map to the same production alternatives, potentially adding redundancy to the search space.

3.3. Proposed Bilevel Grammatical Evolution Approach

Canonical GE works on single-level optimization, for example, optimizing the codon values of the genotype or the order in which these values are used. In this proposal, a bilevel optimization approach was implemented within the MP, in which the two levels focus on optimizing genotype values and order, respectively.

Figure 2 shows the general workflow of BiGE. The upper-level searches for the best values for the integer genotype and, for each candidate solution, applies the lower-level optimization, searching for the best permutation that maps the genotype to the phenotype using the grammar, and evaluates its performance on the problem.

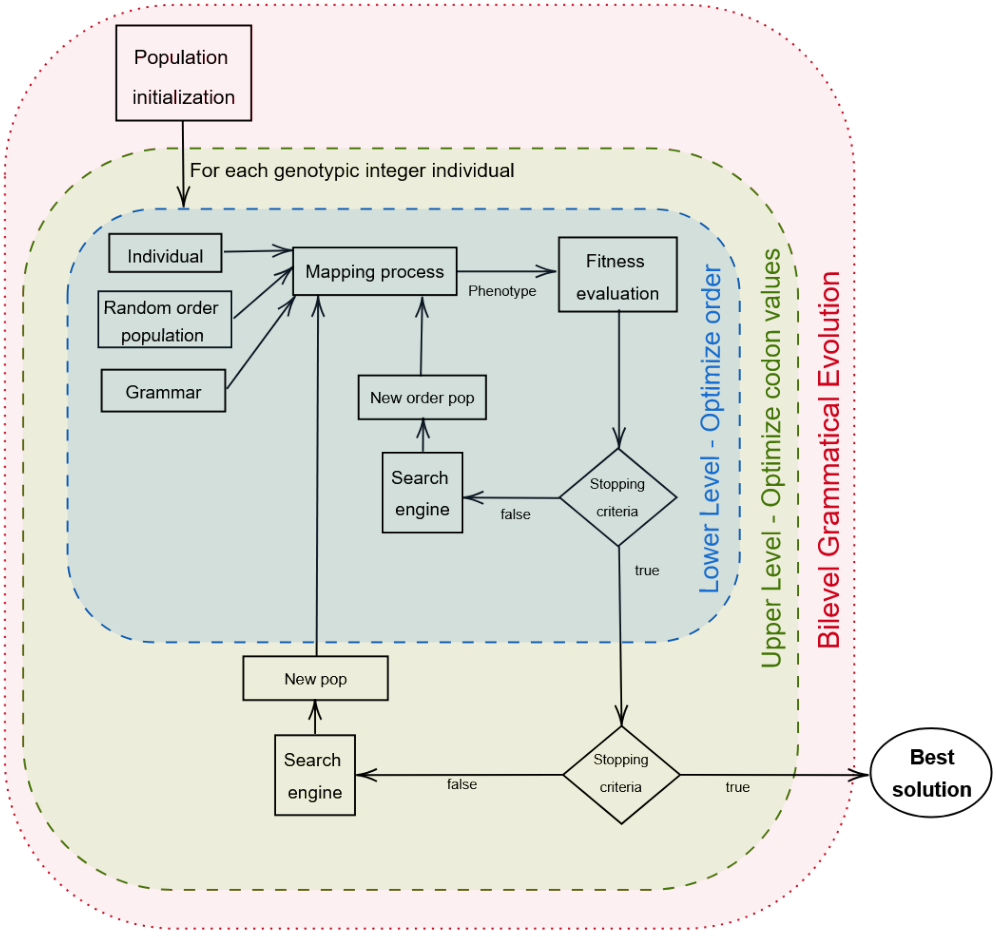


Figure 2. General schema of the proposed approach for GE using bilevel optimization.

3.3.1. Upper Level

The upper level optimizes the codon values of the genotype. Algorithm 1 shows the corresponding pseudocode. The notation is as follows:

- Grammar G is the grammar corresponding to each instance of the problem;
- f is the problem instance.
- n_var is the genotype size.

- UL_xl and UL_xu are the upper and lower bounds of the genotype values in the upper level.
- UL_pop_size is the population size in the upper level.
- Operators for the GA algorithm in the upper level:
  - Sampling *IntegerRandom*;
  - Crossover *TwoPoint*;
  - Mutation *Polynomial* for discrete values;
  - Generations UL_n_gen.
- $LL_xl = 0$ and $LL_xu = n_var - 1$ are the upper and lower bounds of the genotype values in the lower-level. This genotype is a permutation.
- LL_pop_size is the population size in the lower level.
- Operators valid for permutations for the GA algorithm of the lower level:
  - Sampling *PermutationRandomSampling*;
  - Crossover *OrderCrossover*;
  - Mutation *InversionMutation*;
  - Generations LL_n_gen.

Algorithm 1 Upper-level algorithm

Require: Grammar G, objective function f, n_var
Upper-level parameters: UL_pop_size, UL_n_gen, UL_xl, UL_xu
Lower-level parameters: LL_pop_size, LL_n_gen, LL_xl, LL_xu
Operators:
IntegerRandomSampling, TwoPointCrossover, PolynomialMutation
PermutationRandomSampling, OrderCrossover, InversionMutation
Ensure: *BestGenotype*, *BestPer*, *BestFitness*

```
Pop ← InitializePopulation(UL_pop_size)
for gen = 1 to UL_n_gen do
    for each x in Pop do
        (Fitness_x, Per_x) ← LowerLevel(
            x, G, f,
            PermutationRandomSampling
            OrderCrossover, InversionMutation
            LL_n_gen, LL_pop_size)
    end for
    New_Pop ←UL_GA(
            IntegerRandomSampling,
            TwoPointCrossover,
            PolynomialMutation)
end for
best ← argmin_{x∈Pop} Fitness(x)
return (BestGenotype, BestPer, BestFitness)
```

3.3.2. Lower Level

The lower level optimizes the order in which the codon values are taken using a permutation. Algorithm 2 shows the corresponding pseudocode. The permutation Per controls the order in which the codons of the genotype x are consumed during the MP.

The function $MappingProcess(x, Per)$ is based on [6]; it only modifies the order in which the codons are consumed. For example, given a genotype $x = [7, 10, 45, 5]$ and a permutation $Per = [3, 1, 4, 2]$, the codons are consumed in the order $[45, 7, 5, 10]$ during mapping. The grammar and the non-terminal expansion mechanism remain unchanged.

Algorithm 2 Lower-level algorithm

Require: x, Grammar G, objective function f
Operators:
PermutationRandomSampling, OrderCrossover, InversionMutation
Lower-level parameters: LL_n_gen, LL_pop_size
Ensure: $BestPer$, $BestFitness$

```
n ← length(x)
Per_pop ← InitializePopulation(LL_xl, LL_xu, LL_pop_size)
for gen = 1 to LL_n_gen do
    for each Per in Per_pop do
        phenotype ← MappingProcess(x, Per)
        Fitness_Per ← Evaluate(phenotype)
    end for
    Per_pop ←LL_GA(
          PermutationRandomSampling,
          OrderCrossover, InversionMutation)
end for
best ← argmin_{Per∈Per_pop} Fitness(Per)
return BestPer, BestFitness
```

In this study, the stopping criterion is defined by the number of function evaluations. To ensure a fair comparison with the other methods, all approaches were executed under the same evaluation budget of 250,000 function evaluations. It should be noted that, despite having the same evaluation budget, the proposed BiGE incurs additional computational overhead because selection, crossover, and mutation operators are executed at both the upper and lower optimization levels. Equation (4) shows how the number of function evaluations is calculated for the proposed bilevel approach.

$$Function_Calls = UL_{gen} \times UL_{pop_size} \times \left(LL_{gen} \times LL_{pop_size}\right) \tag{4}$$

3.4. Grammar

The grammar defines the rules and restrictions to form valid phenotypes for the specific problem. Specifically for GE, it is common to use the Backus–Naur Form (BNF–Grammar). Different grammars were used for each function:

- Grammar 1 for functions F_1 to F_8.
- Grammar 2 for functions F_9 to F_{10}.
- Grammar 3 for functions $Keijzer_1$ and $Keijzer_2$.
- Grammar 4 for functions $Keijzer_3$ to $Keijzer_5$.
- Grammar 5 for function $Keijzer_6$.

Since previous GE studies on SRP have employed this specific set of grammars, we adopted the same set to ensure a fair comparison with the literature. Accordingly, the analysis in this article focuses on improving the MP rather than grammar design.

It is important to note that protected versions of the exponential and logarithmic operators were used to avoid numerical instability and undefined evaluations. The protected exponential operator was defined as $pexp = \dfrac{1}{1+e^{-x}}$, and the protected logarithm operator as $plog(x) = log(1 + |x|)$.

3.5. Mapping Process

The MP is the procedure for creating a derivation tree using the grammar and the genotype, mapping the genotype into a phenotype. This process is guided by Equation (5).

$$next_productionrule = codon_value \,\%\, num_productionrules \tag{5}$$

```
⟨start⟩ ⊨ ⟨expr⟩
⟨expr⟩  ⊨ (⟨expr⟩⟨op⟩⟨expr⟩) | ⟨pre⟩(⟨expr⟩) | ⟨var⟩
⟨var⟩   ⊨ 1 | x
⟨pre⟩   ⊨ exp | log | sin | cos
⟨op⟩    ⊨ * | / | + | -
```

Grammar 1: Grammar for functions F_1 to F_8.

```
⟨start⟩ ⊨ ⟨expr⟩
⟨expr⟩  ⊨ (⟨expr⟩⟨op⟩⟨expr⟩) | ⟨pre⟩(⟨expr⟩) | ⟨var⟩
⟨var⟩   ⊨ 1 | x | y
⟨pre⟩   ⊨ exp | log | sin | cos
⟨op⟩    ⊨ * | / | + | -
```

Grammar 2: Grammar for functions F_9 to F_{10}.

```
⟨start⟩ ⊨ ⟨expr⟩
⟨expr⟩  ⊨ (⟨expr⟩⟨op⟩⟨expr⟩) | ⟨var⟩
⟨op⟩    ⊨ * | / | + | -
⟨var⟩   ⊨ x | 1
```

Grammar 3: Grammar for functions $Keijzer_1$ and $Keijzer_2$.

```
⟨start⟩ ⊨ ⟨expr⟩
⟨expr⟩  ⊨ (⟨expr⟩⟨op⟩⟨expr⟩) | ⟨var⟩
⟨var⟩   ⊨ 1 | x | y
⟨op⟩    ⊨ * | / | + | -
```

Grammar 4: Grammar for functions $Keijzer_3$ to $Keijzer_5$.

```
⟨start⟩ ⊨ ⟨expr⟩
⟨expr⟩  ⊨ (⟨expr⟩⟨op⟩⟨expr⟩) | ⟨var⟩
⟨op⟩    ⊨ + | - | * | /
⟨var⟩   ⊨ x⟨c⟩ | 1
⟨c⟩     ⊨ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Grammar 5: Grammar for function $Keijzer_6$.

https://doi.org/10.3390/math14132279

This work utilizes the two most popular MPs in GE: Depth-First [4], considered the standard MP for GE, which begins with the start symbol and makes the expansion by taking the leftmost non-terminal symbol available in the derivation tree, and Breadth-First [37], which uses an expansion process slightly different from Depth-First. The expansion is done by expanding all the non-terminals at the same level (from left to right) before moving to the next level in the derivation tree.

4. Results

4.1. Experimental Setup

The SRP was selected to validate the proposed approach. Table 1 shows the benchmark functions used in the experiments.The proposal was compared with state-of-the-art methods that also solve the SRP using GE, such as the GE_opt_3 method proposed in [6] and the classical GE mapping processes DF, BF, and πGE. To ensure a fair comparison with these approaches, the benchmark suite employed in this work corresponds to $Nguyen\ F_1$ to $Nguyen\ F_{10}$ [39] and $Keijzer_1$ to $Keijzer_6$ [40], which are commonly used in the classical GE symbolic regression literature. These benchmark functions allow direct comparison with previously reported results. Nevertheless, more recent benchmark collections for SRP have been proposed [41]. In future work, the proposed BiGE approach will be evaluated in these newer benchmark suites to further assess its capacity for generalization.

Table 1. Symbolic regression functions used as instance set, chosen to compare the proposal with [6].

Function	Expression
$Nguyen\ F_1$	$f(x) = x^3 + x^2 + x$
$Nguyen\ F_2$	$f(x) = x^4 + x^3 + x^2 + x$
$Nguyen\ F_3$	$f(x) = x^5 + x^4 + x^3 + x^2 + x$
$Nguyen\ F_4$	$f(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x$
$Nguyen\ F_5$	$f(x) = \sin(x^2)\cos(x) - 1$
$Nguyen\ F_6$	$f(x) = \sin(x) + \sin(x + x^2)$
$Nguyen\ F_7$	$f(x) = \log(x + 1) + \log(x^2 + 1)$
$Nguyen\ F_8$	$f(x) = \sqrt{x}$
$Nguyen\ F_9$	$f(x, y) = \sin(x) + \sin(y^2)$
$Nguyen\ F_{10}$	$f(x, y) = 2\sin(x)\cos(y)$
$Keijzer_1$	$f(x) = 0.3x\sin(2\pi x)$
$Keijzer_2$	$f(x) = 1 + 3x + 3x^2 + x^3$
$Keijzer_3$	$f(x, y) = \frac{8}{2 + x^2 + y^2}$
$Keijzer_4$	$f(x, y) = \frac{x^4 - x^3 + y^2}{2 - y}$
$Keijzer_5$	$f(x, y) = \frac{\frac{x^3}{5 + y^3}}{2 - y - x}$
$Keijzer_6$	$f(x_1, \ldots, x_{10}) = 10.59x_1x_2 + 100.5967x_3x_4 - 50.59x_5x_6 + 20x_1x_7x_9 + 5x_3x_6x_{10}$

Both the upper and lower levels used a GA as a search engine for GE. The pymoo library, version 0.6.1.1 [42] was used for the GA implementation, and the Python version used was 3.11.9.

Table 2 presents the parameters used in the experiments. The parameter values used in the experiments were taken from [6]. This choice was made to ensure a fair comparison under the same experimental conditions and computational budget, taking into account the

bilevel behavior of the proposed approach. In this work, no additional parameter tuning was performed. The goal of this study was not to identify the optimal parameter configuration for BiGE but to isolate the effect of the proposed bilevel mapping strategy under the same conditions used in previous studies. The size of the upper-level population and the number of generations were intentionally kept small, since each upper-level individual performs a complete lower-level optimization process. Increasing these parameters would significantly increase the computational cost. Early stopping was not employed, as a fixed computational budget was required to maintain a fair comparison among the compared algorithms.

Table 2. Parameter settings for the upper and lower levels.

Parameter	Upper Level	Lower Level
Population size	10 (per level)	
Number of generations	10	250
Dimensions	100 (per level)	
Search engine	Genetic Algorithm (both levels)	
Selection operator	Random Integer	Random Permutation
Crossover operator	Two-Point Crossover	Order Crossover
Mutation operator	Polynomial Mutation (discrete)	Inversion Mutation
Total function evaluations	250,000 (total across both levels)	

Evaluation Metric

For the experiments presented in this work, both F_U and F_L were instantiated using the Root Mean Square Error (RMSE) of the generated symbolic expression. Thus, both optimization levels use the same fitness measure, but they operate on different decision variables. The upper level optimizes the genotype values, whereas the lower level optimizes the codon-consumption permutation used during the MP. The corresponding equation is shown in Equation (6).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i - y_i)^2} \tag{6}$$

where the terms are defined as follows:

n is the number of data points;

y_i is the real value;

t_i corresponds to the obtained value.

4.2. Results

A total of 33 independent runs were performed for each instance and MP using the proposed BiGE approach. The mean value and standard deviation of each set of runs were calculated and are reported in Table 3 for the compared methods.

Some instances are more complex because they generate wider and more irregular search spaces, which can lead the algorithm to converge to different local optima. This may explain the large standard deviations observed in instances such as $Nguyen\ F_7$ and $Keijzer_6$. Although the population size may also contribute to the observed variability, all compared methods were executed with the same population size and evaluation budget. Therefore, the variance cannot be attributed exclusively to the proposed bilevel framework. Nevertheless, BiGE achieves competitive performance across all instances.

Table 4 presents the median results obtained with the proposed approach, in the column for BiGE, and the compared methods.

Table 3. GE and πGE consists of the canonical MPs; GE_opt_3 [6] is an approach based on intensification codon selection; and *BiGE* corresponds to the proposed bilevel approach. Results are reported as the mean $\pm$ standard deviation. The best mean is in bold for each instance.

	Breadth-First			Depth-First			πGE
Instance	**GE**	**GE_opt_3**	***BiGE***	**GE**	**GE_opt_3**	***BiGE***	
Nguyen F_1	$5.79E{-}03 \pm 2.13E{-}02$	$7.76E{-}03 \pm 2.02E{-}02$	$\mathbf{3.09E{-}03} \pm 1.04E{-}02$	$5.36E{-}03 \pm 1.82E{-}02$	$7.14E{-}03 \pm 2.10E{-}02$	$3.42E{-}03 \pm 9.29E{-}03$	$5.14E{-}02 \pm 4.81E{-}02$
Nguyen F_2	$3.50E{-}02 \pm 5.45E{-}02$	$3.19E{-}02 \pm 4.72E{-}02$	$\mathbf{1.84E{-}02} \pm 1.67E{-}02$	$3.95E{-}02 \pm 4.29E{-}02$	$3.51E{-}02 \pm 4.12E{-}02$	$2.80E{-}02 \pm 2.15E{-}02$	$6.55E{-}02 \pm 5.98E{-}02$
Nguyen F_3	$7.39E{-}02 \pm 6.92E{-}02$	$7.40E{-}02 \pm 7.32E{-}02$	$\mathbf{5.52E{-}02} \pm 7.46E{-}02$	$8.71E{-}02 \pm 6.94E{-}02$	$8.36E{-}02 \pm 6.60E{-}02$	$6.92E{-}02 \pm 8.08E{-}02$	$1.11E{-}01 \pm 7.32E{-}02$
Nguyen F_4	$1.40E{-}01 \pm 1.50E{-}01$	$1.16E{-}01 \pm 1.14E{-}01$	$6.11E{-}02 \pm 3.20E{-}02$	$1.39E{-}01 \pm 9.56E{-}02$	$1.43E{-}01 \pm 9.73E{-}02$	$\mathbf{5.88E{-}02} \pm 2.00E{-}02$	$8.77E{-}02 \pm 6.74E{-}02$
Nguyen F_5	$6.99E{-}02 \pm 4.98E{-}02$	$7.88E{-}02 \pm 4.27E{-}02$	$\mathbf{7.25E{-}03} \pm 4.02E{-}03$	$5.05E{-}02 \pm 3.92E{-}02$	$5.07E{-}02 \pm 3.53E{-}02$	$1.13E{-}02 \pm 6.24E{-}03$	$1.54E{-}02 \pm 1.06E{-}02$
Nguyen F_6	$4.48E{-}02 \pm 4.16E{-}02$	$5.25E{-}02 \pm 4.06E{-}02$	$\mathbf{3.13E{-}02} \pm 1.50E{-}02$	$3.82E{-}02 \pm 3.22E{-}02$	$4.02E{-}02 \pm 4.41E{-}02$	$3.33E{-}02 \pm 1.90E{-}02$	$7.65E{-}02 \pm 3.48E{-}02$
Nguyen F_7	$2.35E{-}01 \pm 8.67E{-}02$	$2.47E{-}01 \pm 8.28E{-}02$	$1.48E{+}00 \pm 6.00E{+}00$	$2.21E{-}01 \pm 8.73E{-}02$	$\mathbf{2.14E{-}01} \pm 7.88E{-}02$	$2.60E{-}01 \pm 4.85E{-}01$	$8.59E{-}01 \pm 9.69E{-}01$
Nguyen F_8	$1.17E{-}01 \pm 4.09E{-}02$	$9.51E{-}02 \pm 4.12E{-}02$	$\mathbf{1.14E{-}02} \pm 2.40E{-}02$	$9.40E{-}02 \pm 3.96E{-}02$	$9.26E{-}02 \pm 3.50E{-}02$	$1.39E{-}02 \pm 2.17E{-}02$	$3.09E{-}02 \pm 4.84E{-}02$
Nguyen F_9	$8.13E{-}02 \pm 5.07E{-}02$	$6.60E{-}02 \pm 5.55E{-}02$	$4.33E{-}02 \pm 1.86E{-}02$	$5.92E{-}02 \pm 5.29E{-}02$	$5.60E{-}02 \pm 6.28E{-}02$	$\mathbf{2.78E{-}02} \pm 2.25E{-}02$	$8.72E{-}02 \pm 4.82E{-}02$
Nguyen F_{10}	$7.70E{-}02 \pm 5.84E{-}02$	$5.96E{-}02 \pm 4.38E{-}02$	$\mathbf{2.62E{-}02} \pm 1.52E{-}02$	$5.00E{-}02 \pm 3.38E{-}02$	$4.73E{-}02 \pm 2.76E{-}02$	$3.03E{-}02 \pm 1.61E{-}02$	$5.86E{-}02 \pm 4.87E{-}02$
Keijzer$_1$	$6.88E{-}02 \pm 2.44E{-}02$	$7.33E{-}02 \pm 2.22E{-}02$	$6.48E{-}02 \pm 8.18E{-}03$	$7.19E{-}02 \pm 2.53E{-}02$	$6.72E{-}02 \pm 2.29E{-}02$	$\mathbf{6.19E{-}02} \pm 6.66E{-}03$	$1.10E{-}01 \pm 2.36E{-}02$
Keijzer$_2$	$5.04E{-}03 \pm 2.94E{-}02$	$3.21E{-}16 \pm 3.40E{-}17$	$5.24E{-}16 \pm 4.89E{-}17$	$\mathbf{3.10E{-}16} \pm 2.90E{-}17$	$3.16E{-}16 \pm 5.15E{-}17$	$4.66E{-}16 \pm 5.82E{-}17$	$4.37E{-}16 \pm 5.18E{-}17$
Keijzer$_3$	$\mathbf{3.93E{-}01} \pm 1.96E{-}02$	$4.04E{-}01 \pm 2.39E{-}02$	$4.02E{-}01 \pm 5.82E{-}02$	$4.03E{-}01 \pm 2.42E{-}02$	$4.05E{-}01 \pm 2.96E{-}02$	$3.96E{-}01 \pm 5.03E{-}02$	$4.72E{-}01 \pm 4.01E{-}02$
Keijzer$_4$	$6.76E{-}01 \pm 8.73E{-}04$	$6.76E{-}01 \pm 2.16E{-}03$	$\mathbf{2.25E{-}01} \pm 2.63E{-}02$	$6.76E{-}01 \pm 8.60E{-}04$	$6.76E{-}01 \pm 6.99E{-}04$	$2.48E{-}01 \pm 4.80E{-}02$	$3.06E{-}01 \pm 9.66E{-}02$
Keijzer$_5$	$4.23E{-}01 \pm 4.17E{-}03$	$4.24E{-}01 \pm 3.40E{-}03$	$\mathbf{1.03E{-}01} \pm 5.55E{-}02$	$4.26E{-}01 \pm 3.70E{-}03$	$4.25E{-}01 \pm 3.87E{-}03$	$1.07E{-}01 \pm 4.80E{-}02$	$1.14E{-}01 \pm 5.99E{-}02$
Keijzer$_6$	$3.94E{+}00 \pm 3.49E{+}00$	$3.01E{+}00 \pm 2.17E{+}00$	$4.51E{+}00 \pm 2.39E{+}00$	$2.70E{+}00 \pm 1.64E{+}00$	$3.27E{+}00 \pm 1.42E{+}00$	$\mathbf{1.44E{+}00} \pm 9.64E{-}01$	$1.51E{+}01 \pm 2.28E{+}00$

Table 4. Median results for each instance. The results for GE_opt_3 are taken from [6], while *BiGE* corresponds to the proposed bilevel approach, and GE and πGE consists of the canonical MPs. The best results are in bold.

	Breadth-First			Depth-First			πGE
Instance	**GE**	**GE_opt_3**	***BiGE***	**GE**	**GE_opt_3**	***BiGE***	
Nguyen F_1	$\mathbf{0.00E+00}$	$5.83E-17$	$1.11E-16$	$\mathbf{0.00E+00}$	$6.21E-17$	$1.11E-16$	$4.74E-02$
Nguyen F_2	$\mathbf{0.00E+00}$	$1.56E-16$	$1.70E-02$	$3.29E-02$	$3.13E-02$	$2.48E-02$	$5.38E-02$
Nguyen F_3	$5.77E-02$	$5.87E-02$	$\mathbf{4.50E-02}$	$8.20E-02$	$6.51E-02$	$5.37E-02$	$9.23E-02$
Nguyen F_4	$1.18E-01$	$1.02E-01$	$\mathbf{5.99E-02}$	$1.43E-01$	$1.30E-01$	$6.01E-02$	$5.71E-02$
Nguyen F_5	$5.72E-02$	$8.45E-02$	$\mathbf{6.79E-03}$	$4.92E-02$	$4.61E-02$	$9.34E-03$	$1.26E-02$
Nguyen F_6	$3.33E-02$	$3.27E-02$	$3.19E-02$	$\mathbf{2.30E-02}$	$2.30E-02$	$3.22E-02$	$7.57E-02$
Nguyen F_7	$2.63E-01$	$2.93E-01$	$9.74E-02$	$2.54E-01$	$2.23E-01$	$\mathbf{9.44E-02}$	$6.53E-01$
Nguyen F_8	$1.07E-01$	$1.03E-01$	$\mathbf{5.28E-03}$	$9.50E-02$	$9.38E-02$	$8.55E-03$	$1.78E-02$
Nguyen F_9	$6.63E-02$	$6.46E-02$	$4.43E-02$	$3.58E-02$	$3.58E-02$	$\mathbf{2.37E-02}$	$7.84E-02$
Nguyen F_{10}	$5.26E-02$	$5.16E-02$	$\mathbf{2.88E-02}$	$4.39E-02$	$4.39E-02$	$3.66E-02$	$4.36E-02$
Keijzer$_1$	$\mathbf{6.07E-02}$	$\mathbf{6.07E-02}$	$6.36E-02$	$\mathbf{6.07E-02}$	$\mathbf{6.07E-02}$	$6.36E-02$	$1.26E-01$
Keijzer$_2$	$\mathbf{0.00E+00}$	$3.07E-16$	$5.19E-16$	$2.95E-16$	$2.95E-16$	$4.64E-16$	$4.08E-16$
Keijzer$_3$	$\mathbf{3.81E-01}$	$4.08E-01$	$3.86E-01$	$4.08E-01$	$4.04E-01$	$3.86E-01$	$4.86E-01$
Keijzer$_4$	$6.77E-01$	$6.77E-01$	$\mathbf{2.18E-01}$	$6.77E-01$	$6.77E-01$	$\mathbf{2.18E-01}$	$3.32E-01$
Keijzer$_5$	$4.22E-01$	$4.23E-01$	$\mathbf{7.92E-02}$	$4.27E-01$	$4.25E-01$	$1.24E-01$	$\mathbf{7.92E-02}$
Keijzer$_6$	$2.52E+00$	$2.55E+00$	$4.04E+00$	$2.45E+00$	$2.94E+00$	$\mathbf{1.57E+00}$	$1.56E+01$

Although BiGE achieved competitive performance across most benchmark instances, canonical GE obtained better results on *Nguyen* F_1 and *Nguyen* F_2. These functions correspond to relatively simple symbolic regression problems with limited search spaces, where high-quality solutions can often be identified efficiently without requiring additional optimization of the mapping process. In such cases, the bilevel optimization strategy may introduce exploratory overhead that provides little additional benefit. Overall, the results suggest that the advantages of BiGE become more evident as problem complexity increases, particularly in instances characterized by larger search spaces and more challenging genotype-to-phenotype mappings.

The proposed method achieved competitive results across all benchmark instances. However, different behavior was observed for *Keijzer*$_6$ depending on the mapping strategy. Under the BF MP, BiGE showed lower performance than the canonical approaches, whereas under the Depth-First mapping process, it achieved the best median result. This suggests that the interaction between permutation optimization and the mapping strategy plays a more important role in this benchmark than in the other instances.

The same number of function calls was used for all algorithms to ensure a fair comparison. The Friedman test was applied to the median performance values obtained by each algorithm across the benchmark instances to determine whether statistically significant differences existed among the compared methods. If there existed a significant difference, a Nemenyi post hoc test was used to determine which algorithm was different. The Friedman statistic was 15.999 with a p-value of 0.013754. Since the p-value was below the significance

level of 0.05, there was sufficient evidence to conclude that at least one of the algorithms had a statistically significant difference. Next, a Nemenyi post hoc test was performed to determine which algorithms differed in performance.

Table 5 shows the post hoc test ranking of the algorithms evaluated. The *BiGE* algorithm had the best performance for both MPs. Figure 3 shows the convergence of the upper level of the proposed approach in some instances using both MPs, and Figure 4 shows the convergence of the lower level in the same instances. The proposed method obtained competitive results in all cases except $Keijzer_6$, in which it performed poorly under the BF MP. This behavior warrants further analysis.

Table 5. Nemenyi post hoc test ranking. The prefix DF refers to the Depth-First MP, and BF refers to the Breadth-First MP. The proposed method is in bold and italics.

Algorithm	Ranking
DF_BiGE	2.84375
BF_BiGE	2.87500
BF_*GE*	4.00000
DF_GE_opt_3	4.06250
DF_*GE*	4.46875
BF_GE_opt_3	4.59375
πGE	5.15625

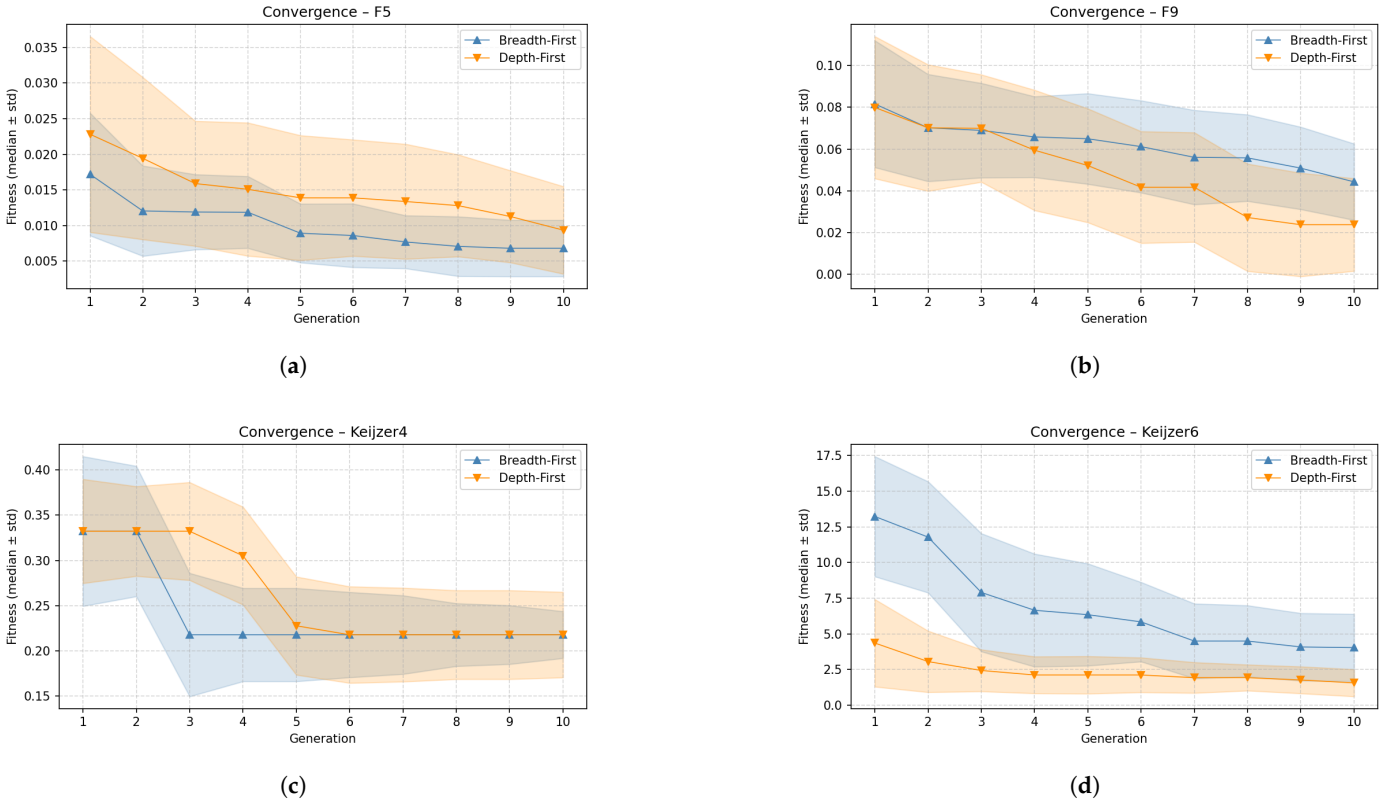


Figure 3. Convergence of the proposal for the upper level on instances $NguyenF_5$ (**a**), $NguyenF_9$ (**b**), $Keijzer_4$ (**c**), and $Keijzer_6$ (**d**) for the Breadth-First and Depth-First mapping processes. The plot presents the median fitness across generations in the upper level, with the shaded area indicating the standard deviation.

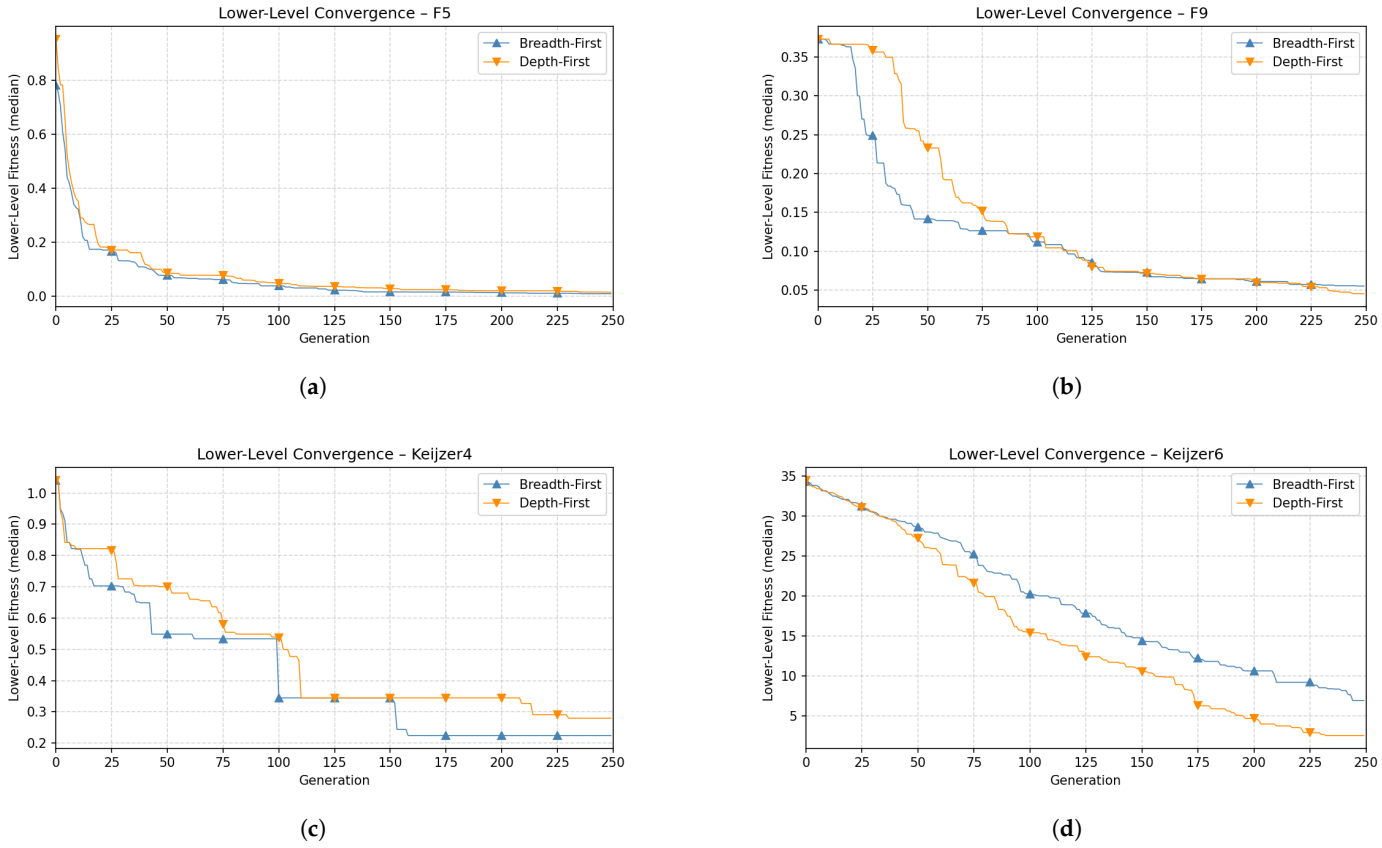


Figure 4. Convergence of the proposal for the lower level on instances $NguyenF_5$ (**a**), $NguyenF_9$ (**b**), $Keijzer_4$ (**c**) and $Keijzer_6$ (**d**) for the BF and DF MPs. The plot presents the median fitness across generations in the lower level.

4.3. Ablation Study

An ablation study (AS) [43] was conducted to assess the contribution of each optimization level to the overall performance of the proposed BiGE algorithm. The purpose of this analysis is to determine whether the improvements achieved by BiGE with the DF MP are primarily associated with the upper level, the lower level, or the interaction between both levels.

Table 6 presents the configurations considered in the AS. Four variants were evaluated: the full BiGE configuration, a variant combining random upper-level optimization with guided lower-level optimization, a variant combining guided upper-level optimization with random lower-level optimization, and a fully random configuration—all of them using the MP DF. Each configuration was executed independently 33 times on all benchmark functions listed in Table 1.

Table 6. Configurations used in the ablation study.

Method	Upper Level	Lower Level
BiGE	Guided	Guided
BiGE-UL-Random	Random	Guided
BiGE-LL-Random	Guided	Random
BiGE-Random-Random	Random	Random

Table 7 shows the obtained results with each instance at each configuration for AS. The results obtained in these experiments were analyzed using the non-parametric Friedman test to determine if there were significant performance differences between the evalu-

ated configurations. The Friedman test yielded a statistic of 30.7350 with a p-value of $9.66E-07$, indicating significant differences among the tested configurations. Since significant differences were detected, the Nemenyi post hoc test was applied to identify the configurations responsible for them. Table 8 presents the average rankings obtained using the Nemenyi test.

Table 7. Results of the tested configurations. The median of 33 independent experiments is reported for each configuration.

BiGE	BiGE-UL-Random	BiGE-LL-Random	BiGE-Random-Random
$1.11E-16$	$1.11E-16$	$9.98E-02$	$9.70E-02$
$2.48E-02$	$2.33E-02$	$1.45E-01$	$1.49E-01$
$5.37E-02$	$6.39E-02$	$1.73E-01$	$1.90E-01$
$6.01E-02$	$6.95E-02$	$4.57E-02$	$4.57E-02$
$9.34E-03$	$1.15E-02$	$2.97E-02$	$2.90E-02$
$3.22E-02$	$3.50E-02$	$6.95E-02$	$8.20E-02$
$9.44E-02$	$9.52E-02$	$5.17E-01$	$6.05E-01$
$8.55E-03$	$8.85E-03$	$3.43E-02$	$3.70E-02$
$2.37E-02$	$3.60E-02$	$4.37E-02$	$4.37E-02$
$3.66E-02$	$2.58E-02$	$5.02E-02$	$5.02E-02$
$6.36E-02$	$6.20E-02$	$7.88E-02$	$7.88E-02$
$4.64E-16$	$4.81E-16$	$4.78E-16$	$5.13E-16$
$3.86E-01$	$3.86E-01$	$5.08E-01$	$5.08E-01$
$2.18E-01$	$2.18E-01$	$3.32E-01$	$3.32E-01$
$1.24E-01$	$1.35E-01$	$1.68E-01$	$1.41E-01$
$1.57E+00$	$1.93E+00$	$1.95E+01$	$2.01E+01$

Table 8. Nemenyi post hoc test ranking for the ablation study.

Configuration	Ranking
BiGE	1.40625
BiGE-UL-Random	1.90625
BiGE-LL-Random	3.18750
BiGE-Random-Random	3.50000

The results indicate that the complete BiGE configuration achieved the highest overall ranking, providing evidence that jointly optimizing genotypic values and codon consumption order plays a key role in the performance gains achieved by the proposed approach.

5. Conclusions

GE is a flexible and powerful tool that relies on the use of a grammar and an MP to generate solutions. In this work, a bilevel optimization approach was introduced into the MP of GE. This proposal is inspired by the πGE MP, which uses a pair of codons, one of them controlling the expansion order. This proposal incorporates permutations, which have previously demonstrated improvements in performance.

The proposed *BiGE* approach was evaluated on the SRP using 16 widely used benchmark instances, and its performance was compared against state-of-the-art methods. The experimental results show that the proposal achieves competitive performance and obtains the best median performance in most benchmark instances. Furthermore, the close agreement between mean and median values suggests consistent behavior across independent runs. Although higher variance was observed in some challenging benchmark instances, the overall results indicate that the proposed method is robust and capable of maintaining competitive performance across problems with different characteristics.

It is important to note that these improvements were achieved while maintaining the same number of function evaluations, ensuring a fair comparison with existing approaches. Additionally, the proposed bilevel approach can be applied to both BF and DF mappings.

Some benchmark instances remain challenging for all approaches, suggesting inherent difficulty in their search spaces. An interesting behavior can be observed in the $Keijzer_6$ instance. The results indicate a limitation under the BF MP. This instance involves complex interactions among variables, which significantly increases the search space. In this scenario, the permutation optimization step may introduce additional exploration that can slow the convergence. This behavior suggests that future work could investigate adaptive strategies to regulate search intensity during the permutation optimization process.

Although some benchmark instances showed relatively high variability across independent runs, the statistical analysis indicates that the proposed approach remains competitive and achieves the best overall ranking among the evaluated methods.

Overall, the results indicate that introducing a bilevel optimization perspective into GE constitutes a promising method for the MP and maintains the same function-evaluation budget used by competing approaches. Future work may explore its application to other problem domains.

Author Contributions: Conceptualization, M.A.S.-F. and B.V.Z.-N.; methodology, M.A.S.-F.; software, B.V.Z.-N. and M.A.S.-F.; validation, B.V.Z.-N. and E.I.G.-H.; formal analysis, I.B.G.-V.; investigation, B.V.Z.-N.; writing—original draft, B.V.Z.-N.; writing—review and editing, E.I.G.-H. and P.B.-M.; supervision, E.I.G.-H. and M.A.S.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the source code files are available at Bilevel-Grammatical-Evolution https://github.com/blancazun/Bilevel-Grammatical-Evolution (accessed on 29 May 2026).

Acknowledgments: The authors want to thank the Secretaria de Ciencias, Humanidades, Tecnología e Innovación (SECIHTI) for providing support through scholarship number 889918 (B.V.Z.-N.).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dong, J.; Zhong, J. Recent Advances in Symbolic Regression. *ACM Comput. Surv.* **2025**, *57*, 290. [CrossRef] [PubMed]
2. Li, N.; Ma, L.; Xing, T.; Yu, G.; Wang, C.; Wen, Y.; Cheng, S.; Gao, S. Automatic design of machine learning via evolutionary computation: A survey. *Appl. Soft Comput.* **2023**, *143*, 110412. [CrossRef]
3. Ryan, C.; Collins, J.J.; Neill, M.O. Grammatical evolution: Evolving programs for an arbitrary language. In *European Conference on Genetic Programming*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; pp. 83–96. [CrossRef]
4. O'Neill, M.; Ryan, C. Grammatical evolution. *IEEE Trans. Evol. Comput.* **2001**, *5*, 349–358. [CrossRef]
5. Sotelo-Figueroa, M.A.; Puga Soberanes, H.J.; Carpio, J.M.; Fraire Huacuja, H.J.; Cruz Reyes, L.; Soria-Alcaraz, J.A. Improving the Bin Packing Heuristic through Grammatical Evolution Based on Swarm Intelligence. *Math. Probl. Eng.* **2014**, *2014*, 545191. [CrossRef]
6. Espinal, A.; Sotelo-Figueroa, M.A.; Soria-Alcaraz, J.A. Grammatical Evolution with Codons Selection Order as Intensification Process. *Comput. Sist.* **2024**, *28*, 837–846. [CrossRef]
7. Verónica Zúñiga, B.; Martín Carpio, J.; Aurelio Sotelo-Figueroa, M.; Espinal, A.; Jair Purata-Sifuentes, O.; Ornelas, M.; Alberto Soria-Alcaraz, J.; Rojas, A. Exploring Random Permutations Effects on the Mapping Process for Grammatical Evolution. *J. Autom. Mob. Robot. Intell. Syst.* **2019**, *14*, 65–72. [CrossRef]
8. O'Neill, M.; Brabazon, A.; Nicolau, M.; Garraghy, S.M.; Keenan, P. π-Grammatical Evolution. In *Genetic and Evolutionary Computation—GECCO 2004*; Lecture Notes in Computer Science; Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., et al., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3103, pp. 617–629. [CrossRef]
9. Bard, J.F. *Practical Bilevel Optimization*; Vol. 30, Nonconvex Optimization and Its Applications; Springer: Boston, MA, USA, 1998. [CrossRef]

10. Neri, F.; Cotta, C.; Moscato, P. (Eds.) *Handbook of Memetic Algorithms*; Vol. 379, Studies in Computational Intelligence; Springer: Boston, MA, USA, 2012. [CrossRef]
11. Neri, F.; Cotta, C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm Evol. Comput.* **2012**, *2*, 1–14. [CrossRef]
12. Augusto, D.; Barbosa, H. Symbolic regression via genetic programming. In Proceedings of the Sixth Brazilian Symposium on Neural Networks, Rio de Janeiro, Brazil, 22–25 November 2000; Volume 1, pp. 173–178. [CrossRef]
13. Kronberger, G.; Burlacu, B.; Kommenda, M.; Winkler, S.M.; Affenzeller, M. *Symbolic Regression*, 1st ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2024. [CrossRef]
14. Ahmadizar, F.; Soltanian, K.; AkhlaghianTab, F.; Tsoulos, I. Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. *Eng. Appl. Artif. Intell.* **2015**, *39*, 1–13. [CrossRef]
15. Tsoulos, I.G.; Tzallas, A.; Karvounis, E. Constructing the Bounds for Neural Network Training Using Grammatical Evolution. *Computers* **2023**, *12*, 226. [CrossRef]
16. Lima, R.H.; Pozo, A.T.; Santana, R. Automatic Design of Convolutional Neural Networks using Grammatical Evolution. In Proceedings of the 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil, 15–18 October 2019; pp. 329–334. [CrossRef]
17. Vaidya, G.; Kshirsagar, M.; Ryan, C. Grammatical Evolution-Driven Algorithm for Efficient and Automatic Hyperparameter Optimisation of Neural Networks. *Algorithms* **2023**, *16*, 319. [CrossRef]
18. Barbudo, R.; Ramírez, A.; Romero, J.R. Grammar-based evolutionary approach for automated workflow composition with domain-specific operators and ensemble diversity. *Appl. Soft Comput.* **2024**, *153*, 111292. [CrossRef]
19. Marinescu, R.; Kishimoto, A.; Ram, P.; Rawat, A.; Wistuba, M.; Palmes, P.P.; Botea, A. Searching for Machine Learning Pipelines Using a Context-Free Grammar. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 8902–8911. [CrossRef]
20. Juárez, J.; Terashima-Marín, H.; Coello Coello, C.A. Multi-objective hyper-heuristics: A survey. *Artif. Intell. Rev.* **2026**, *59*, 80. [CrossRef]
21. Gustafson, S.; Burke, E.; Krasnogor, N. On improving genetic programming for symbolic regression. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; Volume 1, pp. 912–919. [CrossRef]
22. Makke, N.; Chawla, S. Interpretable scientific discovery with symbolic regression: A review. *Artif. Intell. Rev.* **2024**, *57*, 2. [CrossRef]
23. Kamienny, P.a.; d' Ascoli, S.; Lample, G.; Charton, F. End-to-end symbolic regression with transformers. In *Proceedings of the Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 10269–10281. [CrossRef]
24. Shojaee, P.; Meidani, K.; Barati Farimani, A.; Reddy, C. Transformer-based Planning for Symbolic Regression. In *Proceedings of the Advances in Neural Information Processing Systems*; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 45907–45919. [CrossRef]
25. Ali, M.; Kshirsagar, M.; Naredo, E.; Ryan, C. AutoGE: A Tool for Estimation of Grammatical Evolution Models. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence—Volume 2: ICAART, Online, 4–6 February 2021; pp. 1274–1281. [CrossRef]
26. Ali, M.; Kshirsagar, M.; Naredo, E.; Ryan, C. Towards Automatic Grammatical Evolution for Real-world Symbolic Regression. In Proceedings of the 13th International Joint Conference on Computational Intelligence, Valletta, Malta, 25–27 October 2021; pp. 68–78. [CrossRef]
27. Colmenar, J.M.; Martín-Santamaría, R.; Hidalgo, J.I. WebGE: An Open-Source Tool for Symbolic Regression Using Grammatical Evolution. In *Applications of Evolutionary Computation*; Jiménez Laredo, J.L., Hidalgo, J.I., Babaagba, K.O., Eds.; Springer International Publishing: Cham, Switzerland, 2022; Volume 13224, pp. 269–282. [CrossRef]
28. Sinha, A.; Malo, P.; Deb, K. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Trans. Evol. Comput.* **2018**, *22*, 276–295. [CrossRef]
29. Dempe, S. Bilevel Optimization: Theory, Algorithms, Applications and a Bibliography. In *Bilevel Optimization*; Dempe, S., Zemkoho, A., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 161, pp. 581–672. [CrossRef]
30. Sinha, A.; Malo, P.; Deb, K. Evolutionary Bilevel Optimization: An Introduction and Recent Advances. In *Recent Advances in Evolutionary Multi-objective Optimization*; Bechikh, S., Datta, R., Gupta, A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; Volume 20, pp. 71–103. [CrossRef]
31. Franceschi, L.; Frasconi, P.; Salzo, S.; Grazii, R.; Pontil, M. Bilevel Programming for Hyperparameter Optimization and Meta-Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1568–1577. [CrossRef]
32. Shaban, A.; Cheng, C.A.; Hatch, N.; Boots, B. Truncated Back-propagation for Bilevel Optimization. *arXiv* **2018** . [CrossRef]
33. Bennett, K.P.; Kunapuli, G.; Hu, J.; Pang, J.S. Bilevel Optimization and Machine Learning. In *Computational Intelligence: Research Frontiers*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5050, pp. 25–47. [CrossRef]

34. Sinha, A.; Khandait, T.; Mohanty, R. A gradient-based bilevel optimization approach for tuning regularization hyperparameters. *Optim. Lett.* **2024**, *18*, 1383–1404. [CrossRef]
35. Hong, M.; Wai, H.T.; Wang, Z.; Yang, Z. A Two-Timescale Stochastic Algorithm Framework for Bilevel Optimization: Complexity Analysis and Application to Actor-Critic. *SIAM J. Optim.* **2023**, *33*, 147–180. [CrossRef]
36. Dumouchelle, J.; Julien, E.; Kurtz, J.; Khalil, E.B. Neur2BiLO: Neural Bilevel Optimization. In *Proceedings of the Advances in Neural Information Processing Systems*; Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2024; Volume 37, pp. 86688–86719. [CrossRef]
37. Esparcia-Alcázar, A.I.; Ekárt, A.; Silva, S.; Dignum, S.; Uyar, A.; Hutchison, D.; Kanade, T.; Kittler, J.; Kleinberg, J.M.; Mattern, F.; et al. (Eds.) *Genetic Programming: 13th European Conference, EuroGP 2010, Istanbul, Turkey, April 7–9. 2010. Proceedings*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6021. [CrossRef]
38. Ryan, C.; O'Neill, M.; Collins, J. (Eds.) *Handbook of Grammatical Evolution*; Springer International Publishing: Cham, Switzerland, 2018. [CrossRef]
39. Karaboga, D.; Ozturk, C.; Karaboga, N.; Gorkemli, B. Artificial bee colony programming for symbolic regression. *Inf. Sci.* **2012**, *209*, 1–15. [CrossRef]
40. Ryan, C.; Soule, T.; Keijzer, M.; Tsang, E.; Poli, R.; Costa, E.; Goos, G.; Hartmanis, J.; Van Leeuwen, J. (Eds.) *Genetic Programming: 6th European Conference, EuroGP 2003 Essex, UK, April 14–16, 2003 Proceedings*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2610. [CrossRef]
41. Imai Aldeia, G.S.; Zhang, H.; Bomarito, G.; Cranmer, M.; Fonseca, A.; Burlacu, B.; La Cava, W.G.; de França, F.O. Call for Action: Towards the next generation of symbolic regression benchmark. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '25 Companion, New York, NY, USA, 14–18 July 2025; pp. 2529–2538. [CrossRef]
42. Blank, J.; Deb, K. Pymoo: Multi-Objective Optimization in Python. *IEEE Access* **2020**, *8*, 89497–89509. [CrossRef]
43. Fawcett, C.; Hoos, H.H. Analysing differences between algorithm configurations through ablation. *J. Heuristics* **2015**, *22*, 431–458. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.