Post-Hoc Refinement for Multitask Symbolic Regression via Consensus-Accelerated Shapley Analysis

Xinyue Li[1], **Wang Hu**[1], **Yu Zhang**[1*]

[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

xinyueli@std.uestc.edu.cn, huwang@uestc.edu.cn, yuzhang@uestc.edu.cn

Abstract

Multitask genetic programming (MTGP) is one of the primary methods for solving multitask symbolic regression (MTSR), the problem of discovering mathematical expressions for multiple interconnected tasks simultaneously. However, conventional MTGP approaches discard a wealth of valuable knowledge from the population of expressions due to their inherent "winner-take-all" selection criteria. To address this, we introduce MTGP with bidirectional cooperation and consensus-accelerated Shapley analysis (MTGP-BS), a method whose core is a novel post-hoc refinement framework that shifts from selection to synthesis. Our method first employs a consensus-accelerated Shapley analysis to reliably identify important subexpressions by multi-model attribution. Second, to supply this analysis with high-quality candidates, we design a bidirectional subexpression cooperative extraction method to create a refined archive of effective components by improving knowledge transfer and filtering out redundancies. These allow MTGP-BS to synthesize superior expressions by integrating knowledge dispersed throughout the entire population. On diverse MTSR problems, our algorithm statistically outperformed state-of-the-art approaches in 140 out of 160 direct comparisons, with its effectiveness and practical utility further verified by real-world case studies and in-depth ablation analyses.

Code and Extended version — https://github.com/yu-zhang576/MTGP-BS

Introduction

Genetic programming (GP) is an evolutionary computation paradigm for automatically learning computer programs and symbolic expressions (Al-Helali et al. 2024; Oltean et al. 2009). A prominent application is symbolic regression (SR), where GP excels at discovering explicit mathematical formulas from data without pre-specifying functional form (Gligorovski and Zhong 2023). This capability is vital for automated scientific discovery (Fang et al. 2023). Multitask genetic programming (MTGP) extends GP to solve MTSR problems with multiple interconnected tasks by leveraging shared knowledge (Huang et al. 2022; Zhang et al. 2022).

However, a critical limitation persists in conventional MTGP. For a set of tasks indexed by $t = 1, \cdots T$, MTGP generates a population of expressions $\{\mathbf{P}_t\}_{t=1}^{T}$. The final model for each task is typically the single best expression p_t^* chosen based on an isolated fitness criterion $f(\cdot)$,

$$p_t^* = \arg\min_{p_t \in \mathbf{P}} f(\mathbf{P}_t, t).$$

This "winner-take-all" approach discards a wealth of knowledge from suboptimal expressions, which may harbor valuable **subexpressions**, meaningful segments without coefficients obtained by splitting an expression using the "+" and "-" operators (e.g., "x_1^2" in "$2x_1^2 + \sin(x_2)$"). This practice limits the accuracy and generalizability of final expressions, as discussed in Appendix A.1.

To address this, we propose a post-hoc refinement framework that reconstructs expressions after MTGP evolution. It analyzes the entire evolved population $\{\mathbf{P}_t\}_{t=1}^{T}$ to synthesize a new, superior expression e_t for each task as

$$e_t = \hbar(\{\boldsymbol{R}_t\}_{t=1}^{T}, t),$$

where $\boldsymbol{R}_t$ is raw subexpressions from $\mathbf{P}_t$, and $\hbar(\cdot)$ is a synthesis function that constructs an expression e_t by combining the most valuable subexpressions from all tasks.

This vision presents two challenges. First, the raw set of subexpressions is massive and redundant, making analysis computationally intensive and prone to overlooking key subexpressions (Peijin et al. 2023). Second, credibly assessing subexpression importance is non-trivial, naive methods like Shapley additive explanations (SHAP) (Kumar et al. 2020) can produce unreliable results due to stochastic errors. Detailed analysis of these issues is in Appendix A.2.

To address these challenges, we propose MTGP with bidirectional cooperation and consensus-accelerated Shapley analysis (MTGP-BS). It introduces a novel post-hoc refinement framework that shifts from selection to synthesis, enabling the integration of dispersed knowledge to construct superior expressions. Our main contributions are as follows:

* *Corresponding author*: Yu Zhang

- We find that conventional "winner-take-all" selection discards valuable knowledge. We propose the MTGP-BS, which shifts from selection to synthesis to integrate knowledge dispersed throughout the entire population.
- We identify that the massive and redundant subexpressions hinder effective SHAP analysis. We design a bidirectional cooperative extraction method that constructs a compact, high-quality archive to solve this issue.
- We discover that naive importance assessment of subexpressions is unreliable. We develop a consensus-accelerated Shapley analysis that leverages multi-model attribution and consensus-based subexpression selection to reliably identify key components from the archive.

Related Work and Motivations

Our work is situated at the intersection of MTGP for SR and the integration of interpretability method SHAP with evolutionary algorithms. MTGP has shown promise in enhancing performance on problems with multiple interconnected tasks (Jaśkowski, Krawiec and Wieloch 2008) and has proven highly effective in diverse applications (Bi, Xue and Zhang 2022; Durasevic, Dumic and Gala 2025; Liu et al. 2025), but existing frameworks are hindered by a "winner-take-all" selection process that discards valuable knowledge from the entire population (Zhong et al. 2020).

To synthesize superior expressions, we must accurately assess the importance of their constituent parts. SHAP analysis offers a game-theoretic approach for this, assigning each feature an importance value (Choi, Shin and Shin 2025). An explanation model g is defined as:

$$g(z') = \varphi_0 + \sum_{m=1}^{M} \varphi_m z'_m,$$

where $z' \in \{0,1\}^M$ is a binary vector representing a feature coalition, M is the number of model features, and φ_m is the Shapley value for the m-th feature.

The synergy between SHAP and evolutionary computation has been explored for post-hoc interpretation (Greenwood, Abbass and Hussein 2023) and, more recently, to actively guide evolution (Li et al. 2024; Wang et al. 2025). While leveraging SHAP to quantify subexpression importance for post-hoc synthesis is a natural extension, a naive application is fraught with challenges, including the high dimensionality of redundant subexpressions and the unreliability of single-model SHAP estimations. Our work is the first to systematically address these fundamental limitations. Our post-hoc refinement approach also differs from recent works that instead modify the evolutionary process itself (Anthes, Sobania and Rothlauf 2025; França and Kronberger 2025). We introduce a novel post-hoc framework to MTGP for solving MTSR that operates through two synergistic mechanisms: a bidirectional cooperative extraction method first builds a refined archive, and our consensus-accelerated Shapley analysis then leverages this archive to ensure reliable importance quantification.

Proposed Method

The introduction of MTGP-BS and its two key innovations are presented here.

The MTGP-BS Framework: An Overview

We present MTGP-BS, which introduce a novel post-hoc refinement framework to synthesize superior symbolic expressions. The framework reconstructs an expression for each task by analyzing, evaluating, and combining the most impactful subexpressions from the entire population evolved across all tasks. The process, depicted in Figure 1, begins with **bidirectional subexpression cooperative extraction** to create a compact, high-quality subexpression archive $\mathbf{A}_t$. This archive is then subjected to a **consensus-accelerated Shapley analysis** to select the most consistently impactful building blocks $\overline{\mathbf{A}}_t$. Finally, the least squares method (See et al. 2017) is utilized to construct the final mathematical expression e_t from this refined set $\overline{\mathbf{A}}_t$.

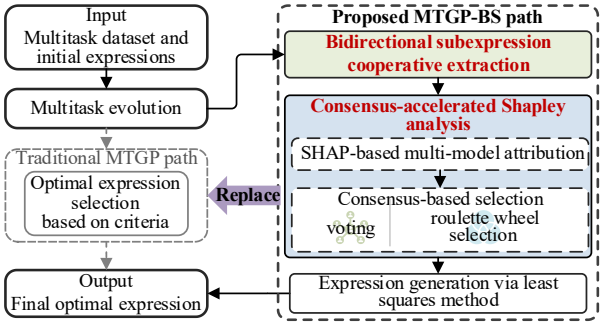


Figure 1: Overall flowchart of MTGP-BS.

Bidirectional Subexpression Cooperative Extraction Method

To address challenge that raw subexpressions from MTGP are too large and noisy for direct Shapley analysis (see Appendix A.2), we propose a bidirectional subexpression cooperative extraction method to construct a compact, high-quality subexpression archive, $\mathbf{A}_t$, for each task. This is achieved through a multi-stage selection process illustrated in Figure 2 and detailed in Appendices B.1, B.2, and B.4. The pseudocode **Algorithm S1** and complexity analysis are available in Appendix B.3.

Intra-Task Frequency-Guided Archiving

As shown in Figure 2, the process begins by filtering subexpressions $\mathbf{R}_t$ from the current task. We apply k-means clustering to the frequency distribution of all unique subexpressions to partition them into high- and low-frequency groups. The initial archive $\mathbf{A}_t^{(1)}$ is formed by selecting all subexpressions from the high-frequency cluster. Let $\text{cluster}_{\text{high}}(\cdot)$ be the function that performs this operation, then

$$\mathbf{A}_t^{(1)} = \text{cluster}_{\text{high}}(\mathbf{R}_t).$$

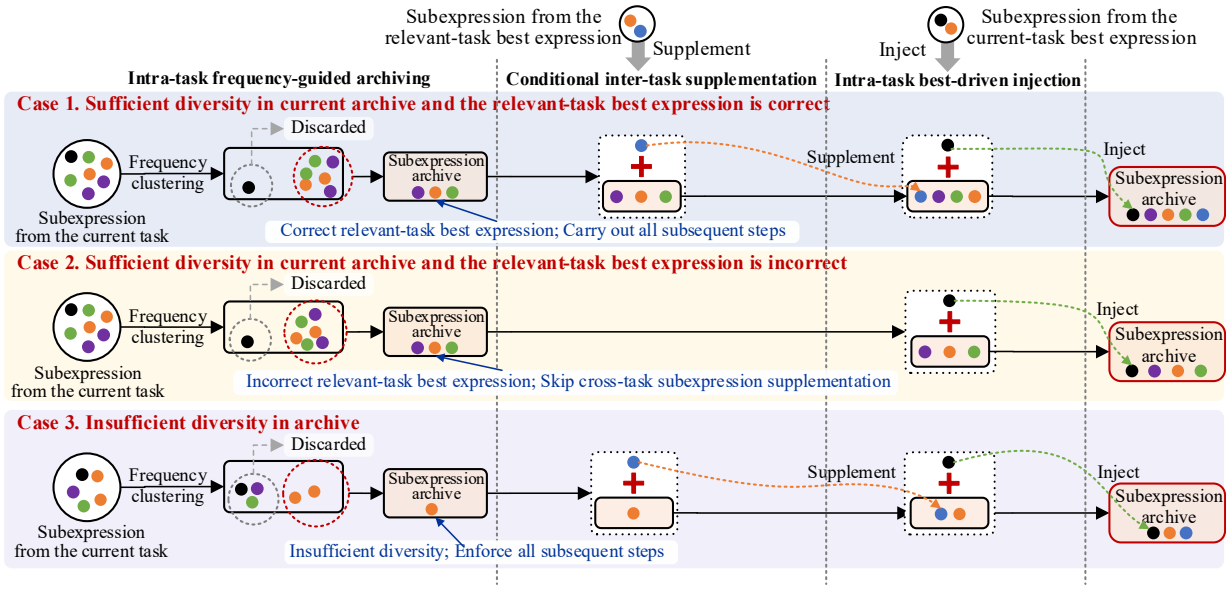


Figure 2: The processes of bidirectional subexpression cooperative selection method, illustrated under the three conditions.

Conditional Inter-Task Supplementation

The archive is then potentially enriched with knowledge from the t'-th relevant task. Let $p_{t'}^*$ be the best expression of the t'-th task, $Div(\cdot)$ be archive diversity (number of unique subexpressions), Correct$(\cdot)$ be a Boolean function indicating if the performance of an expression exceeds a threshold θ_{perf}, and θ_{div} is a diversity threshold. The supplemented archive $\mathbf{A}_t^{(2)}$ is formed as

$$\mathbf{A}_t^{(2)} = \begin{cases} \mathbf{A}_t^{(1)} \cup \mathcal{E}(p_{t'}^*), & Div(\mathbf{A}_t^{(1)}) > \theta_{div} \\ & \wedge \text{Correct}(p_{t'}^*, \theta_{perf}) \quad \text{(Case 1)} \\ \mathbf{A}_t^{(1)}, & Div(\mathbf{A}_t^{(1)}) > \theta_{div} \\ & \wedge \neg\text{Correct}(p_{t'}^*, \theta_{perf}) \quad \text{(Case 2)}' \\ \mathbf{A}_t^{(1)} \cup \mathcal{E}(p_{t'}^*), & Div(\mathbf{A}_t^{(1)}) \le \theta_{div} \quad \text{(Case 3)} \end{cases}$$

using the function $\mathcal{E}(\cdot)$ for subexpression extraction.

Intra-Task Best-Driven Injection

Finally, to ensure knowledge from the best individual p_t^* is included, we perform a final injection,

$$\mathbf{A}_t = \mathbf{A}_t^{(2)} \cup \mathcal{E}(p_t^*).$$

This yields a set of refined archives $\{\mathbf{A}_t\}_{t=1}^T$ for analysis.

Consensus-Accelerated Shapley Analysis

The archive $\mathbf{A}_t$, while containing high-potential subexpressions selected based on static properties, requires an assessment of dynamic predictive contribution on each subexpression. Thus, we design the consensus-accelerated Shapley analysis. It overcomes the limitations of a single, stochastic SHAP analysis by integrating an ensemble approach (Zhou, Yu and Qian 2019) with a consensus-based subexpression selection scheme, as shown in Figure 3. A rationale for this design is in Appendix C.1.

Bootstrap-Based Data Preparation

As shown in Figure 3(a), we generate K distinct training datasets by bootstrap sampling (Ngo, Beard and Chandra 2022) on the raw dataset $\mathbf{D}$. For each sample $\mathbf{D}_k'$, out-of-bag samples $\mathbf{D}_k^{OOB}$ are reserved for unbiased evaluation.

SHAP-Based Multi-Model Attribution

This stage is illustrated in Figure 3(b) for each task. We train an ensemble of K models $\{f_k\}_{k=1}^K$. For each model, we compute the SHAP values $\{\phi_{k,m}\}_{m=1}^M$ for all M subexpression $\{a_{t,m}\}_{m=1}^M \in \mathbf{A}_t$ by treating them as features. These values are calculated using $\mathbf{D}_k^{OOB}$ from the raw SHAP values $\varphi_{k,m}$, which are derived from out-of-bag samples $x_{k,n}^{OOB}$,

$$\phi_{k,m} = \frac{1}{N_{OOB}} \sum_{n=1}^{N_{OOB}} \phi_{k,m}(f_k, x_{k,n}^{OOB}).$$

This yields two crucial outputs,

- An importance matrix $\mathbf{\Phi}_t \in \mathbb{R}^{K\times M}$, $\mathbf{\Phi}_{t,k,m} = |\phi_{k,m}|$.
- A rank matrix $\mathbf{\Gamma}_t \in \mathbb{Z}^{K\times M}$, where $\mathbf{\Gamma}_{t,k,:} = \{\Gamma_{t,k,m}\}_{m=1}^M$ is the importance rankings of $\{a_{t,m}\}_{m=1}^M$ with respect to model f_k, with the highest value receiving the top rank.

Consensus-Based Subexpression Selection

A consensus-based subexpression selection in Figure 3(c) synthesizes the multi-model attribution results by a theoretically-grounded consensus mechanism and a classic heuristic for exploration diversity. First, a voting-based selection identifies subexpressions with strong consensus. A subexpression $a_{t,m}$ is selected if a majority of models agree on its specific rank γ^* within the top half as,

$$\exists \gamma^* \in \left\{1, \cdots \left\lfloor \frac{M}{2} \right\rfloor\right\} \text{ s.t. } \sum_{k=1}^K \mathbb{I}(\Gamma_{t,k,m} = \gamma^*) > \frac{K}{2},$$

where $\mathbb{I}(\cdot)$ is an indicator function that equals to 1 if the condition is true, and 0 otherwise. Second, other consistently important subexpressions are selected via enhanced roulette wheel selection over K trials, one for each f_k. A multiset of

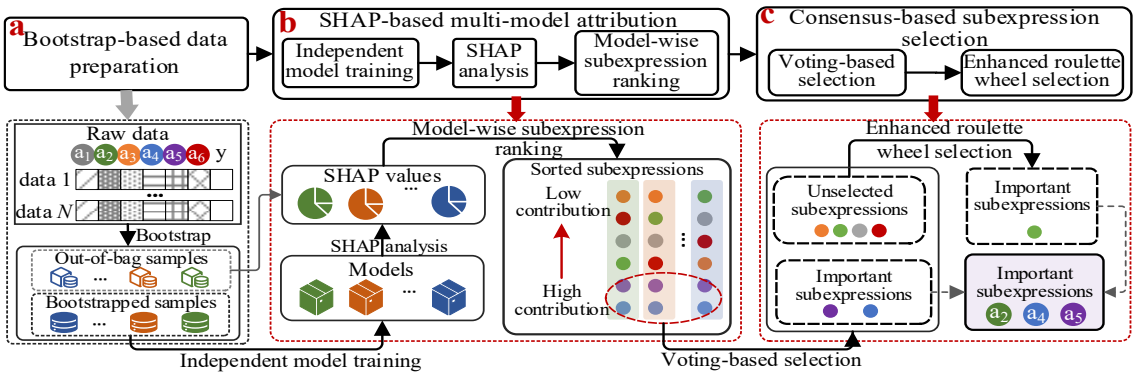


Figure 3: The process of consensus-accelerated Shapley analysis.

subexpressions is generated by performing a number of roulette wheel selections equivalent to the count of remaining subexpressions, $|\mathbf{U}|$. The probability of selecting an $a_{t,m} \in \mathbf{U}$ is proportional to its absolute SHAP value in that model,

$$Pr(a_m|f_k) = \frac{|\phi_{k,m}|}{\sum_{a_{t,m}\in\mathbf{U}}|\phi_{k,m}|}.$$

After K trials are complete, a subexpression is added to the final set $\overline{A}_t$ if it is selected by a majority of the trials, that is

$$\sum_{k=1}^{K}\mathbb{I}\big(Selected(a_{t,m},k)\big) > \frac{K}{2},$$

where $Selected(a_{t,m},k)$ is a function that is 1 if subexpression $a_{t,m}$ is selected in the k-th trail, and 0 otherwise. The voting-based selection is theoretically grounded by Proposition 1. The complete implementation details are provided in Appendixes C.2-C.4, and the pseudocode Algorithm S2 and its time complexity analysis are in Appendix C.5.

Proposition 1. *Under a null hypothesis of random rank assignment, the probability of a subexpression being selected by our voting-based selection is exceedingly low. (Proof in Appendix F).*

The Overall Algorithm

The complete MTGP-BS workflow is summarized in Algorithm 1. It begins with a standard MTGP evolution, using gene expression programming (GEP) (Li, Cheng and Yao 2016) to mitigate code bloat, followed by our two novel mechanisms for extraction and analysis, and concludes by synthesizing the final expressions using iterative least square. The mathematical details of the final stage in Appendix D and a time complexity analysis in Appendix E.

Experimental Studies

This section demonstrates the superiority of MTGP-BS which achieves significantly higher accuracy than state-of-the-art multitask algorithms across a comprehensive suite of problems (Table 1) while discovering more structurally correct expressions (Table 3). Ablation and sensitivity analyses confirm the crucial role of our proposed mechanisms and justify the choice of hyperparameter K (Figures 4, 5 and 9). The generality of our approach is established by showing it serves as a plug-and-play enhancer for various baselines, achieving significant performance gains at a modest computational cost that constitutes only a small fraction of runtime (Figures 6 and 7). In addition, MTGP-BS proves effective on new MTSR problems constructed from single-task benchmarks (Figure 8), and its practical utility is validated through a real-world case study on population and carbon emission prediction (Figure 10 and Table 4).

Experimental Settings

Benchmark Problems: The performance of all algorithms is evaluated on a benchmark suite of 16 MTSR problems, each composed of two distinct tasks (Zhang et al. 2024). These problems cover a range of complexities, operator requirements, and variable counts.

Baseline Algorithms: We compare MTGP-BS against five multitask algorithms, MO-MFEA (Gupta et al. 2017), MFEA-DGD (Liu et al. 2024), BLKT-DE (Jiang et al. 2024), MTDE-MKTA (Li and Gong 2025), and MTES-KG (Li, Gong and Li 2024). To test general applicability, our post-hoc refinement framework is also applied to these baselines.

Algorithm 1: MTGP-BS

Input: number of tasks T; number of models K
Output: expressions $\mathbf{E} = \{e_t\}_{t=1}^{T}$
// **Evolutionary process**
1. Initialize $\mathbf{P} = \{\mathbf{p}_t\}_{t=1}^{T}$ for T tasks;
2. Evolve $\mathbf{P}$ via MTGP;
// **Post-Hoc Refinement**
3. Extract raw subexpressions $\{\mathbf{R}_t\}_{t=1}^{T}$ from $\mathbf{P}$;
4. Contract $\{\mathbf{A}_t\}_{t=1}^{T}$ from $\{\mathbf{R}_t\}_{t=1}^{T}$ by Algorithm S1;
5. **For** $t = 1$ **to** T **do**
6. Select the most important subexpressions $\overline{\mathbf{A}}_t$ from $\mathbf{A}_t$ by Algorithm S2;
7. Construct e_t from $\overline{\mathbf{A}}_t$ by iterative least squares;
8. $\mathbf{E} = \mathbf{E} \cup e_t$;
9. **End For**
10. **Return E**.

Set	Task	MTGP-BS	BLKT-DE	MFEA-DGD	MO-MFEA	MTDE-MKTA	MTES-KG
1	1	**1.80E-03**	4.0E-2+	6.3E-2+	3.9E-2+	3.5E-2+	4.2E-2+
	2	**1.30E-02**	2.4E-2+	4.9E-2+	2.4E-2+	1.9E-2+	3.7E-2+
2	1	**0.00E+00**	4.2E-3+	3.6E-2+	7.5E-3+	2.0E-3+	1.3E-2+
	2	2.90E-03	2.7E-5-	2.3E-2+	2.1E-3-	**2.3E-5-**	1.6E-2+
3	1	**6.50E-03**	4.2E-2+	5.3E-2+	4.1E-2+	3.5E-2+	4.2E-2+
	2	**3.00E-02**	5.4E-2+	7.4E-2+	5.2E-2+	5.4E-2+	5.9E-2+
4	1	**0.00E+00**	1.2E-2+	4.1E-2+	9.1E-3+	5.6E-4+	2.1E-2+
	2	**1.10E-01**	1.5E-1+	2.1E-1+	1.7E-1+	1.5E-1+	2.0E-1+
5	1	**0.00E+00**	1.4E-2+	3.1E-2+	1.0E-2+	3.4E-3+	1.2E-2+
	2	**5.10E-02**	8.6E-2+	1.0E-1+	9.4E-2+	8.4E-2+	9.6E-2+
6	1	**2.90E-17**	5.8E-3+	2.6E-2+	5.8E-3+	2.7E-3+	3.4E-3+
	2	1.50E-03	**2.2E-5≈**	2.3E-2+	1.0E-3-	1.1E-4≈	1.6E-2+
7	1	**1.30E-02**	2.9E-2+	5.3E-2+	2.1E-2+	2.1E-2+	2.0E-2+
	2	**3.90E-02**	5.4E-2+	7.1E-2+	5.4E-2+	5.3E-2+	5.5E-2+
8	1	**3.00E-17**	5.5E-3+	3.3E-2+	7.4E-3+	2.8E-3+	2.1E-3+
	2	**1.00E-01**	1.4E-1+	2.1E-1+	1.6E-1+	1.6E-1+	1.8E-1+
9	1	**2.90E-17**	6.6E-3+	2.7E-2+	4.2E-3+	1.3E-3+	3.8E-3+
	2	**5.20E-02**	8.6E-2+	1.0E-1+	8.8E-2+	7.8E-2+	1.0E-1+
10	1	7.40E-03	1.2E-3-	1.7E-2+	4.7E-4-	**3.1E-5≈**	1.8E-2+
	2	**1.80E-02**	3.7E-2+	4.5E-2+	3.8E-2+	3.5E-2+	4.1E-2+
11	1	2.10E-02	**8.2E-5≈**	1.3E-2-	1.3E-4≈	2.1E-4≈	8.3E-3-
	2	**9.10E-02**	1.4E-1+	2.1E-1+	1.7E-1+	1.4E-1+	1.8E-1+
12	1	3.70E-03	5.0E-4-	1.6E-2+	1.3E-3-	**5.2E-5-**	8.7E-3+
	2	**6.00E-02**	8.4E-2+	1.0E-1+	8.7E-2+	8.1E-2+	1.0E-1+
13	1	**1.60E-02**	3.9E-2+	4.8E-2+	3.8E-2+	3.3E-2+	4.0E-2+
	2	**1.00E-01**	1.3E-1+	2.0E-1+	1.8E-1+	1.4E-1+	1.8E-1+
14	1	**3.10E-02**	3.5E-2+	4.6E-2+	3.6E-2+	3.7E-2+	3.9E-2+
	2	**6.00E-02**	8.5E-2+	9.7E-2+	8.8E-2+	7.6E-2+	9.7E-2+
15	1	**1.40E-01**	1.4E-1≈	2.0E-1+	1.7E-1+	1.4E-1≈	1.8E-1+
	2	**7.40E-02**	7.9E-2+	1.0E-1+	8.2E-2+	8.3E-2+	9.6E-2+
16	1	**2.60E-03**	1.3E-2+	3.6E-2+	1.5E-2+	1.1E-2+	2.1E-2+
	2	**2.60E-03**	3.0E-3+	2.1E-2+	3.3E-3+	3.3E-3+	3.3E-3+
+/≈/-			26/3/3	31/0/1	27/1/4	25/5/2	31/0/1

Table 1: The *mse* for MTGP-BS and five competitors.

General Parameters: The population size is 50 for each task, with a maximum of 100,000 evaluations. We utilize GEP with a fixed head length of 10. Each experiment is repeated 30 times. Details are provided in Appendix G.

Experimental Results Analysis

Superiority of Learning Expressions

We compare the performance of MTGP-BS against five competing multitask algorithms. Table 1 summarizes the *mse*, with statistical significance determined by the Wilcoxon signed-rank test ($\alpha = 0.05$) (Derrac et al. 2011). MTGP-BS achieves statistically significant performance against all competitors on the vast majority of tasks, demonstrating its superior accuracy. A detailed analysis and numerical data are in Appendix H.

Qualitative Comparison of Learned Expressions

To inspect the structural integrity of solutions, we select three representative problems (Table 2). Table 3 compares the median-run expressions from each algorithm, revealing the superiority of MTGP-BS in discovering expressions that are both highly accurate and structurally sound (best results highlighted in bold). A detailed rationale and analysis are available in Appendix I.

Set	Expression
2	$x_1^2 + 2x_2^2$
	$100x_1^4 - 200x_1^2x_2 + x_1^2 - 2x_1 + 100x_2^2 + 1$
3	$x_1^2 + 2x_2^2 + 3x_3^2$
	$x_1^2 + x_2^2 + x_3^2 - 10cos(x_1)$ $-10cos(x_2) - 10cos(x_3) + 30$
16	$x_1^2 - 10cos(x_1) + 10$
	$x_1^2/4000 - cos(x_1) + 1$

Table 2 Test functions for qualitative analysis

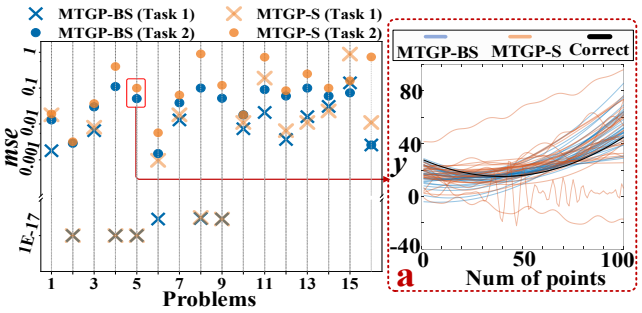


Figure 4: Performance comparison of MTGP-BS against its ablated variant, MTGP-S. The main plot shows the median *mse* on 16 MTSR problems (log scale). The subplot (a) shows the function mapping curves from 30 runs on Set 5, Task 2, compared to the correct expression (black line).

Ablation Study: Efficacy of Bidirectional Extraction

To verify our bidirectional extraction method, we compare MTGP-BS against a variant, MTGP-S, which omits this module. As shown in Figure 4, MTGP-BS consistently achieves significantly lower *mse*, demonstrating the crucial advantage of our strategic subexpression extraction. The inset plot (a) provides a qualitative example of this improved alignment with the ground truth. Details are in Appendix J.

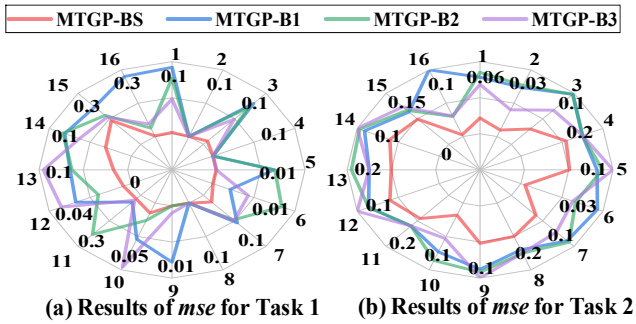


Figure 5: Performance comparison for the consensus analysis ablation study.

Set	Algorithm	Task 1	*mse*	Task 2	*mse*
2	MTGP-BS	$\boldsymbol{x_1^2 + 2x_2^2}$	**0.00E+00**	$0.26x_2 - 200x_1^2x_2 + 0.67x_1^2 + 100x_2^2 + 100x_1^4 - 1.7$	**3.74E-06**
	BLKT-DE	$x_1^2 + 2x_2^2 + 2.917\text{e} - 14$	1.05E-16	$100x_2^2 - 200x_1^2x_2 + 100x_1^4 + 8.21$	2.50E-05
	MFEA-DGD	$2.07x_1 + 0.52x_1^2 + 2.07x_2^2 + 4.5$	2.98E-02	$94.24x_1^4 - 11720$	2.71E-02
	MO-MFEA	$\boldsymbol{x_1^2 + 2x_2^2}$	**0.00E+00**	$100x_2^2 - 200x_1^2x_2 + 100x_1^4 + 7.58$	3.41E-05
	MTDE-MKTA	$\boldsymbol{x_1^2 + 2x_2^2}$	**0.00E+00**	$100x_2^2 - 200x_1^2x_2 + 100x_1^4 + 8.27$	2.55E-05
	MTES-KG	$1.86x_2 + 0.93x_1^2 + 1.86x_2^2 - 2.31$	9.39E-03	$91.09x_1^4 - 91.09x_1^2 - 91.09x_2^2 - 91.09x_2 - 5086$	1.92E-02
3	MTGP-BS	$\boldsymbol{x_1^2 + 2x_2^2 + 3x_3^2}$	**0.00E+00**	$x_1^2 + 0.99x_2^2 + x_3^2 - 0.91x_3\cos(x_3) - 1.4x_2\cos(x_3) + 31$	**3.43E-02**
	BLKT-DE	$1.74x_1x_3 - 1.74x_3 + 0.87x_2x_3 + 1.74x_2^2 + 1.74x_3^2 + 29.45$	3.43E-02	$0.65x_1x_2 + 0.65x_1x_3 - 0.65x_2x_3 + 0.33x_1^2 + 0.98x_2^2 + 0.98x_3^2 + 34.36$	5.97E-02
	MFEA-DGD	$2.9x_2^2 - 2.9x_3 - 5.79x_2 + 2.9x_3^2 + 50.01$	5.17E-02	$10x_1 + 10x_2 + 10x_3 + 17.55\cos(x_3^2) - 9.59\sin(x_3^2) - 15.48$	6.80E-02
	MO-MFEA	$5.18x_1 + 2.27x_2^2 + 2.59x_3^2 + 13.14$	4.11E-02	$1.03x_1 + 1.03x_2 + 1.03x_1^2 + 1.03x_2^2 + 1.03x_3^2 + 18.47$	5.17E-02
	MTDE-MKTA	$2.7x_1 + 2.7x_3 + 1.35x_1x_2 + 1.35x_2^2 + 2.7x_3^2 + 6.05$	3.10E-02	$9.39x_1 + 12.01x_2 + 9.39x_3 - 21.64$	**5.43E-02**
	MTES-KG	$1.89x_2 + 1.89x_1x_3 + 1.89x_2^2 + 1.89x_3^2 + 16.35$	3.56E-02	$0.33(x_1 + x_2 + x_3 + 0.5)^2 + 41.26$	6.62E-02
16	MTGP-BS	$\boldsymbol{x_1^2 - 10\cos(x_1) + 10}$	**0.00E+00**	$\boldsymbol{x_1^2/4000 - cos(x_1) + 1}$	**0.00E+00**
	BLKT-DE	$2.12x_1 - 2.12\cos(x_1) + 0.77\cos(x_1)^2 - 1.54x_1\cos(x_1) + 0.77x_1^2 + 6.13$	1.37E-02	$1.01 - 1.001\cos(x_1)$	2.94E-03
	MFEA-DGD	$1.2x_1^2 - 1.2\cos(x_1) - 1.2x_1 + 9.06$	3.95E-02	$1.01 - 1.004cos(x_1)$	2.91E-03
	MO-MFEA	$1.58x_1 - 1.58\cos(x_1) + 0.79\cos(x_1)^2 - 1.58x_1\cos(x_1) + 0.79x_1^2 + 8$	1.50E-02	$1.01 - 1.003cos(x_1)$	3.36E-03
	MTDE-MKTA	$3.29x_1 - 3.29\cos(x_1) + 0.65\cos(x_1)^2 - 1.32x_1\cos(x_1) + 0.66x_1^2 + 4.001$	1.14E-02	$0.82\cos(\cos(0.5x_1)) - 0.82\cos(x_1) + 0.39$	3.24E-03
	MTES-KG	$1.06x_1^2 - 1.06x_1\cos(x_1) - 0.53x_1 + 9.94$	2.12E-02	$1.01 - 1.003\cos(x_1)$	3.62E-03

Table 3: Expressions Obtained from the Median-Performing Run of each algorithm For Problem Sets 2, 3 and 16

Ablation Study: Efficacy of Consensus-Accelerated Shapley Analysis

To validate our consensus-accelerated Shapley analysis, we create three ablated variants (described in Appendix K), MTGP-B1 (single-model analysis), MTGP-B2 (simple SHAP averaging), and MTGP-B3 (voting-only consensus). Figure 5 shows that removing any core components leads to a significant degradation in performance.

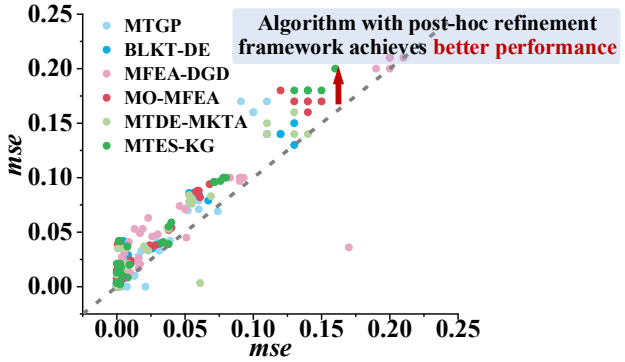


Figure 6: Paired comparison of algorithms with and without the post-hoc refinement. The scatter plot compares *mse* of each baseline (y-axis) against its enhanced version (x-axis).

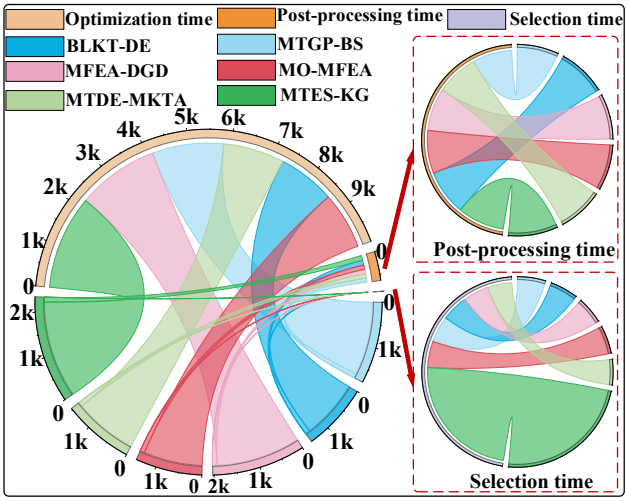


Figure 7: Total running time of optimization, post-hoc refinement, and direct selection.

Generality and Effectiveness of the Post-Hoc Refinement

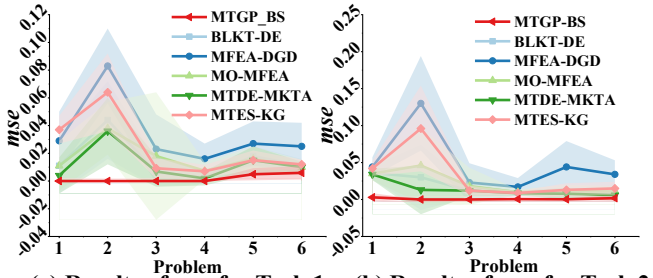
Figure 6 demonstrates the general effectiveness of our post-hoc module, which consistently and substantially improves performance across six multitask algorithms. Detailed analysis is discussed in Appendix L.

Algorithm	Task 1	*mse*	Task 2	*mse*
MTGP-BS	$43.62D\varepsilon - 24.92D - 12.46\varepsilon^2 + 7.28^7$	**0.0296**	$3.9D - 62.0\varepsilon + 0.25D\varepsilon + 0.096\varepsilon^2 - 0.2$	**0.0182**
BLKT-DE	$11.54D + 46.17D\varepsilon - 11.54\varepsilon^2 + 7.18^7$	0.0309	$0.21D\varepsilon^2 - 3.36D + 8921$	0.0491
MFEA-DGD	$32.11D\varepsilon - 16.06\varepsilon^2 + 7.69^7$	0.0354	$0.57\varepsilon + 0.19D\varepsilon + 12070$	0.0522
MO-MFEA	$46.19D\varepsilon - 11.55\varepsilon - 11.55\varepsilon^2 + 7.184^7$	0.0309	$2.42\varepsilon - 0.8052D + 0.2013D\varepsilon + 8580$	0.0490
MTDE-MKTA	$46.21D\varepsilon - 11.55\varepsilon^2 + 7.18^7$	0.0309	$0.07D + 0.28D\varepsilon + 0.04\varepsilon^2 - 22280$	0.0256
MTES-KG	$46.04D\varepsilon - 92.08\varepsilon - 11.51\varepsilon^2 + 7.19^7$	0.0309	$0.7641\varepsilon + 0.191D\varepsilon + 11800$	0.0519

Table 4: Expressions with median *mse* learned by MTGP-BS and all competing algorithms

Running Time Analysis

Figure 7 reveals that the initial evolution overwhelmingly dominates the runtime. The overhead of our framework is a small fraction of this cost, demonstrating that its significant performance gains are achieved with modest computational overhead. Numerical results are in Appendix M.



(a) Results of *mse* for Task 1 **(b) Results of *mse* for Task 2**

Figure 8: Performance comparison on six MTSR problems constructed from single-task SR benchmarks. The plots show the mean *mse* (solid lines) and standard deviation (shaded areas) across 30 runs

Performance on Constructed Multitask Benchmarks

We test MTGP-BS on six new MTSR problems derived from established SR benchmarks (Jin et al. 2019; Uy et al. 2011). As shown in Figure 8, its performance line (in red) is the lowermost, demonstrating its superior performance on these problems. Detailed numerical results is in Appendix N.

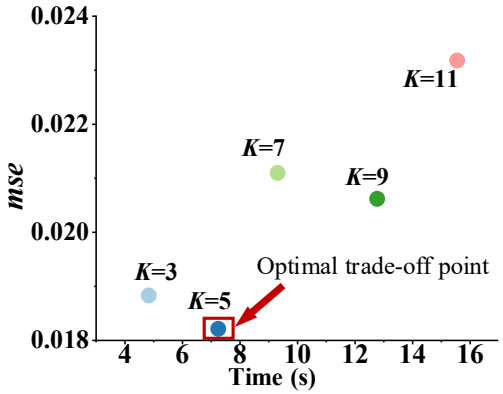


Figure 9: Sensitivity analysis for the number of models used in problem Set 3.

Sensitivity Analysis

Figure 9 shows our sensitivity analysis on the number of models K. We select $K = 5$ as it provides the optimal trade-off between performance and runtime. A complete analysis and additional figures are in Appendix O.

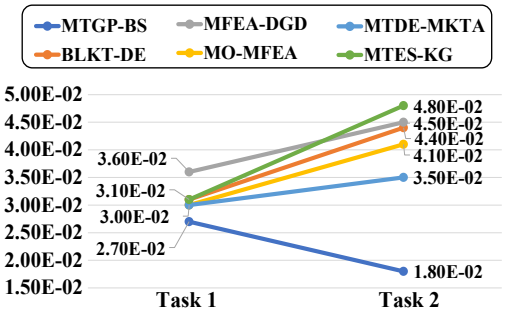


Figure 10: The *mse* for expressions learned for case study

Case Study on Population and Carbon Emission Prediction

To validate practical utility, we modeled human population (H) and carbon emissions (Q) as functions of gross domestic product (D) and energy consumption (ϵ). As shown in Figure 10 and Table 4, our framework achieves the highest predictive accuracy while discovering more interpretable expressions. A detailed analysis is in Appendix P.

Conclusion

We propose MTGP-BS with a post-hoc refinement framework that mitigates knowledge loss in traditional MTGP by synthesizing expressions from valuable subcomponents. It combines a bidirectional extraction strategy with a consensus-accelerated Shapley analysis to build more accurate and interpretable models. Experiments confirm that MTGP-BS significantly outperforms state-of-the-art algorithms on diverse benchmarks and a real-world case study. Future work could integrate this analysis directly into the evolutionary process for dynamic knowledge guidance.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China [grant number 61976046].

References

Al-Helali, B.; Chen, Q.; Xue, B.; and Zhang, M. 2024. Genetic Programming-based Feature Selection for Symbolic Regression on Incomplete Data, *Evolutionary computation,* 1-27. 10.1162/evco_a_00362.

Anthes, P.; Sobania, D.; and Rothlauf, F., "Transformer Semantic Genetic Programming for Symbolic Regression," presented at the Proceedings of the Genetic and Evolutionary Computation Conference, NH Malaga Hotel, Malaga, Spain, 2025. [Online]. Available: 10.1145/3712256.3726412.

Bi, Y.; Xue, B.; and Zhang, M. 2022. Learning and Sharing: A Multitask Genetic Programming Approach to Image Feature Learning, *IEEE Transactions on Evolutionary Computation,* 26(2): 218-232. 10.1109/TEVC.2021.3097043.

Choi, J. E.; Shin, J. W.; and Shin, D. W. 2025. Vector SHAP Values for Machine Learning Time Series Forecasting, *Journal of Forecasting,* 44(2): 635-645. 10.1002/for.3220.

Derrac, J.; García, S.; Molina, D.; and Herrera, F. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation,* 1(1): 3-18. doi.org/10.1016/j.swevo.2011.02.002.

Durasevic, M.; Dumic, M.; and Gala, F. J. G. 2025. Multitask genetic programming for automated design of heuristics for the container relocation problem, *Engineering Applications of Artificial Intelligence,* 144. 10.1016/j.engappai.2025.110001.

Fang, W.-Z.; Chang, C.-H.; Liu, J.-C.; and Yu, T.-L. 2023. GP with Ranging-Binding Technique for Symbolic Regression. In proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Lisbon, Portugal. 10.1145/3583133.3590605

França, F. O. d., and Kronberger, G. 2025. Improving Genetic Programming for Symbolic Regression with Equality Graphs. In proceedings of the Proceedings of the Genetic and Evolutionary Computation Conference, New York, NY, USA: Association for Computing Machinery. 10.48550/ARXIV.2501.17848

Gligorovski, N., and Zhong, J. 2023. LGP-VEC: A Vectorial Linear Genetic Programming for Symbolic Regression. In proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Lisbon, Portugal. 10.1145/3583133.3590695

Greenwood, G. W.; Abbass, H.; and Hussein, A. 2023. Interpretation of Neural Network Players for a Generalized Divide the Dollar Game Using SHAP Values. In proceedings of the 2023 IEEE Symposium Series on Computational Intelligence (SSCI). 10.1109/SSCI52147.2023.10371984

Gupta, A.; Ong, Y. S.; Feng, L.; and Tan, K. C. 2017. Multiobjective Multifactorial Optimization in Evolutionary Multitasking, *IEEE Transactions on Cybernetics,* 47(7): 1652-1665. 10.1109/TCYB.2016.2554622.

Huang, Z.; Zhang, F.; Mei, Y.; and Zhang, M. 2022. An Investigation of Multitask Linear Genetic Programming for Dynamic Job Shop Scheduling. In proceedings of the 25th European Conference on Genetic Programming (EuroGP) Held as Part of EvoStar Conference, Complutense Univ Madrid, Madrid, Spain. 10.1007/978-3-031-02056-8_11

Jaśkowski, W.; Krawiec, K.; and Wieloch, B. 2008. Multitask Visual Learning Using Genetic Programming, *Evolutionary Computation,* 16(4): 439-459. 10.1162/evco.2008.16.4.439.

Jiang, Y.; Zhan, Z. H.; Tan, K. C.; and Zhang, J. 2024. Block-Level Knowledge Transfer for Evolutionary Multitask Optimization, *IEEE Transactions on Cybernetics,* 54(1): 558-571. 10.1109/TCYB.2023.3273625.

Jin, Y.; Fu, W.; Kang, J.; Guo, J.; and Guo, J. 2019. Bayesian Symbolic Regression. arXiv:1910.08892.

Kumar, C. S.; Choudary, M. N. S.; Bommineni, V. B.; Tarun, G.; and Anjali, T. 2020. Dimensionality Reduction based on SHAP Analysis: A Simple and Trustworthy Approach. In proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP). 10.1109/ICCSP48568.2020.9182109

Li, B.; Yang, Y.; Liu, D.; Zhang, Y.; Zhou, A.; and Yao, X. 2024. Accelerating surrogate assisted evolutionary algorithms for expensive multi-objective optimization via explainable machine learning, *Swarm and Evolutionary Computation,* 88101610. doi.org/10.1016/j.swevo.2024.101610.

Li, Q.; Cheng, H.; and Yao, M. 2016. Adaptive Multi-phenotype Based Gene Expression Programming Algorithm, *Chinese Journal of Electronics,* 25(5): 807-816. doi.org/10.1049/cje.2016.08.041.

Li, Y., and Gong, W. 2025. Multiobjective Multitask Optimization With Multiple Knowledge Types and Transfer Adaptation, *IEEE Transactions on Evolutionary Computation,* 29(1): 205-216. 10.1109/TEVC.2024.3353319.

Li, Y.; Gong, W.; and Li, S. 2024. Multitask Evolution Strategy With Knowledge-Guided External Sampling, *IEEE Transactions on Evolutionary Computation,* 28(6): 1733-1745. 10.1109/TEVC.2023.3330265.

Liu, R.; Lv, H.; Yang, P.; and Wang, R. 2025. A multi-task genetic programming approach for online multi-objective container placement in heterogeneous cluster, *Complex & Intelligent Systems,* 11(1). 10.1007/s40747-024-01605-x.

Liu, Z.; Li, G.; Zhang, H.; Liang, Z.; and Zhu, Z. 2024. Multifactorial Evolutionary Algorithm Based on Diffusion Gradient Descent, *IEEE Transactions on Cybernetics,* 54(7): 4267-4279. 10.1109/TCYB.2023.3270904.

Ngo, G.; Beard, R.; and Chandra, R. 2022. Evolutionary bagging for ensemble learning, *Neurocomputing,* 5101-14. 10.1016/j.neucom.2022.08.055.

Oltean, M.; Grosan, C.; Diosan, L.; and Mihaila, C. 2009. Genetic Programming with Linear Representation: A Survey, *International Journal on Artificial Intelligence Tools,* 18(2): 197-238. 10.1142/s0218213009000111.

Peijin, G.; Wengiang, H.; Qinmiao, Z.; and Yuhui, W. 2023. Prediction algorithm of nugget diameter in resistance spot welding based on cascade forest, *Computer Integrated Manufacturing Systems,* 29(07): 2267-2276. 10.13196/j.cims.2023.07.012.

See, J. J.; Jamaian, S. S.; Salleh, R. M.; Nor, M. E.; and Aman, F. 2017. Parameter estimation of Monod model by the Least-Squares method for microalgae Botryococcus Braunii sp. In proceedings of the International Seminar on Mathematics and Physics in Sciences and Technology (ISMAP), Malaysia. 10.1088/1742-6596/995/1/012026

Uy, N. Q.; Hoai, N. X.; O'Neill, M.; McKay, R. I.; and Galván-López, E. 2011. Semantically-based crossover in genetic programming: application to real-valued symbolic regression, *Genetic Programming and Evolvable Machines,* 12(2): 91-119. 10.1007/s10710-010-9121-2.

Wang, C.; Chen, Q.; Xue, B.; and Zhang, M. 2025. Improving Generalization of Genetic Programming for High-Dimensional Symbolic Regression with Shapley Value Based Feature Selection, *Data Science and Engineering,* 10(2): 196-211. 10.1007/s41019-024-00270-x.

Zhang, F.; Mei, Y.; Su, N.; Tan, K. C.; and Zhang, M. 2022. Multitask Genetic Programming-Based Generative

Hyperheuristics: A Case Study in Dynamic Scheduling, *IEEE Transactions on Cybernetics,* 52(10): 10515-10528. 10.1109/tcyb.2021.3065340.

Zhang, Y.; Li, X.; Hu, W.; and Yen, G. G. 2024. Multiexpression Symbolic Regression and Its Circuit Design Case, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 1-14. 10.1109/TSMC.2024.3519675.

Zhong, J.; Feng, L.; Cai, W.; and Ong, Y. S. 2020. Multifactorial Genetic Programming for Symbolic Regression Problems, *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* 50(11): 4492-4505. 10.1109/TSMC.2018.2853719.

Zhou, Z.; Yu, Y.; and Qian, C. 2019. *Evolutionary Learning: Advances in Theories and Algorithms*. Singapore : Springer Singapore. doi.org/10.1007/978-981-13-5956-9