A Task-Similarity-Based Multitask Genetic Programming Approach to Symbolic Regression

Ying Bi[1, 2], Yaxin Chang[1], Wenjing Li[1], Caitong Yue[1, 2], Jing Liang[1, 2, 3*]
[1] *School of Electrical and Information Engineering*, *Zhengzhou University*, Zhengzhou, China
[2] *State Key Laboratory of Intelligent Agricultural Power Equipment, Luoyang, China*
[3] *School of Electrical Engineering and Automation, Henan Institute of Technology, Xinxiang, China*
Email:{yingbi, yuecaitong, liangjing}@zzu.edu.cn, yaxinchang@gs.zzu.edu.cn, liwenjing11088@163.com

***Abstract*—Symbolic regression (SR) automatically derives interpretable mathematical expressions from data and has been widely applied in various fields. Many real-world SR tasks are intrinsically similar, such as fault prediction models for different engine types sharing common structures. However, most existing SR methods focus on independent single-task modeling and fail to exploit inter-task relationships to improve performance. To address this, a task-similarity-based multitask genetic programming (TSMTGP) approach is proposed for SR. TSMTGP adopts a novel two-tree individual representation, in which each solution consists of a shared tree across tasks and a task-specific tree tailored to each task, enabling effective knowledge transfer. Correspondingly, two fitness evaluation methods are designed for the shared tree and the two-tree representation. A task similarity measure strategy is developed by using Spearman's rank correlation coefficient based on the performance of the shared tree on different tasks. Based on this measure, a similarity-aware evolutionary mechanism is introduced, in which the evolutionary process adaptively switched between high-similarity and low-similarity modes to improve knowledge transfer. TSMTGP is consistently evaluated under varying task similarities on both benchmark and real-world problems. TSMTGP consistently outperforms comparative methods, particularly on real-world datasets, demonstrating its effectiveness in dynamic knowledge transfer across different tasks.**

***Index Terms*—symbolic regression, genetic programming, multitask learning, task similarity, cross-task knowledge sharing.**

I. INTRODUCTION

Symbolic regression (SR) aims to automatically derive explicit mathematical expressions from observed data to characterize the intrinsic relationships between input variables and response variables [1]. Unlike traditional regression methods that rely on predefined model structures, SR directly searches the space of mathematical expressions and simultaneously optimizes both model structure and parameters. SR provides interpretable analytical expressions while maintaining strong modeling capability, making it more suitable for modeling complex and nonlinear real-world data [2].

Genetic programming (GP) is an evolutionary computation (EC) algorithm that automatically generates solutions by simulating natural selection and genetic variation mechanisms [3]. GP typically represents mathematical expressions using tree structures and evolves candidate solutions through genetic operators such as crossover and mutation. Guided by fitness evaluation, GP converges toward promising models and structures [4]. Owing to its flexible representation and powerful search ability, GP has become a dominant approach to SR.

Many real-world SR tasks exhibit intrinsic similarity in data distributions, variable compositions, or underlying model structures. Effectively sharing knowledge across related tasks may improve performance. However, most existing GP-based SR (GPSR) studies are limited to independent single-task modeling and fail to effectively exploit inter-task relationships to improve performance. Multitask evolutionary algorithms (MTEAs) have emerged as an important research direction in EC in recent years [5]. By exploiting the implicit parallelism of population-based evolution, MTEAs enable the simultaneous optimization of multiple tasks through designed knowledge transfer or information-sharing mechanisms.

When GP is applied to SR, related tasks often share common mathematical structures or sub-expressions. By jointly solving multiple related tasks within a multitask GP framework, useful knowledge can be shared across tasks, thereby reducing redundant computations, accelerating convergence, and enhancing model generalization. Previous studies have demonstrated the effectiveness of multitask GP in SR [6]. However, existing research in this area remains limited. Most existing methods adopt unified or static knowledge-sharing strategies, which overlook the impact of task similarity differences on knowledge transfer effectiveness. It has been widely recognized that task similarity is critical for effective knowledge exchange in MTEAs.

To address this limitation, a task-similarity-based multitask genetic programming (TSMTGP) approach is proposed for SR. The main contributions of the paper are summarized as follows.

- A novel two-tree individual representation is proposed to enable effective knowledge exchange among tasks, in which each individual consists of a task-specific tree for modeling task-dependent features and shared tree for capturing common mathematical structures.
- Two fitness evaluation methods are designed for the proposed representation, where the shared tree is evaluated using regression error and model complexity across multiple tasks, and two-tree individuals are evaluated using task-specific regression error to guide the evolutionary search.

* Corresponding author.

- A task similarity measure strategy based on Spearman's rank correlation coefficient (SRCC) [7] is developed by analyzing the performance rankings of the shared tree population on different tasks.
- Based on the measured task similarity, a similarity-aware evolutionary mechanism is proposed that adaptively switches the evolutionary process between high-similarity and low-similarity modes to balance cross-task knowledge transfer and task-specific optimization.

II. Related Work

A. Symbolic Regression

There are various methods proposed for SR, such as exhaustive enumeration methods [8], Bayesian SR [9], and deep learning-based methods [10]. But they still suffer from limitations in scalability, efficiency, or interpretability. Exhaustive search is hindered by combinatorial explosion; probabilistic models rely heavily on prior distributions. Deep learning-based methods often compromise interpretability. Consequently, GPSR remains the dominant paradigm due to its flexibility and adaptability.

Recent research on GPSR has focused on improving search efficiency, mitigating overfitting, enhancing semantic diversity, and improving generalization performance. Nguyen et al. [11] introduced a semantic similarity measure to constrain subtree replacement, effectively avoiding unproductive search directions and accelerating convergence. Raymond et al. [12] incorporated Rademacher complexity into the fitness function to control model complexity, improving both generalization and interpretability. Chan et al. [13] integrated backward elimination into GP to remove insignificant polynomial terms and reduce overfitting, achieving superior predictive performance on real manufacturing processes.

However, most existing GPSR studies remain limited to single-task modeling. In many practical scenarios, SR tasks exhibit intrinsic similarity, such as shared physical laws or common influencing factors. Existing methods rarely exploit such inter-task relationships to enhance search efficiency, which becomes increasingly important in multitask settings.

B. Multitask Evolutionary Algorithms

MTEAs have emerged as a promising optimization paradigm in EC. The fundamental assumption of MTEAs is that related tasks share useful information, and knowledge discovered while solving one task can facilitate the optimization of other related tasks. The multifactorial evolutionary algorithm (MFEA) [14], one of the earliest and most influential MTEAs, enables implicit inter-task knowledge transfer through a unified population. Empirical studies have shown that MFEA can improve search efficiency compared to single-task evolutionary algorithms. Zhou et al. [15] proposed an adaptive knowledge transfer strategy for MFEA that improves optimization performance by dynamically adjusting the crossover operator. Qiao et al. [16] developed a multitask framework that enhances performance through knowledge transfer between constrained and unconstrained tasks. Hao et al. [17] introduced a graph-based evolutionary multitask hyper-heuristic framework that enables cross-domain optimization and effective knowledge exchange.

In recent years, multitask GP has been proposed for different tasks. Zhang et al. [18] introduced multitask GP-based hyper-heuristics for dynamic scheduling problems. Bi et al. [19] reformulated multitask feature learning as a multi-objective optimization problem and achieved promising results in GP for image classification. Other studies introduced task similarity measure and similarity-based task selection strategies into GP frameworks, demonstrating the potential of GP in recognizing multiple similarity tasks [20].

Existing studies demonstrate the effectiveness of multitask evolutionary algorithms. However, most MTEAs are designed for numerical optimization tasks and cannot be directly applied to SR. Although Zhong et al. [6] provided an initial exploration of multitask GP for SR within an MFEA framework, research in this direction remains limited. Therefore, further investigation of multitask GP specifically tailored for SR is necessary and promising.

III. THE PROPOSED APPROACH

A. Overall Framework

The framework of TSMTGP is illustrated in Algorithm 1. First, three populations (P^0_{shared}, P^0_1, P^0_2) are initialized using the proposed individual representation (detailed in section III-B). Next, their fitness values are evaluated, and the optimal shared tree is selected and combined with each task-specific population to form two-tree individual populations. Subsequently, task similarity is quantified using the proposed similarity measurement strategy (detailed in Section III-D), where the SRCC, ρ, of the shared-tree population across the two tasks is computed. The evolutionary strategy is adaptively adjusted based on the value of ρ. When tasks exhibit high similarity ($\rho > 0.6$), the approach operates in the high-similarity mode, in which the shared tree is shared to facilitate efficient cross-task knowledge transfer. Otherwise, in the low-similarity mode, shared-tree sharing is disabled, and the task populations evolve independently as separate two-tree individual populations. Iteration proceeds until the termination condition is fulfilled. It is worth noting that in each generation, only the single best shared tree is used to construct task-specific individuals in the high-similarity mode. Each task updates its current best individual, which consists of its task-specific tree and shared tree. The evolutionary strategies under high- and low-similarity modes are described in Algorithm 1.

B. New Individual Representation

TSMTGP aims to simultaneously optimize the mathematical models of two SR tasks to enhance generalization. To this end, a novel multi-tree individual representation is introduced, in which each individual consists of two trees, i.e., a task-specific tree and a shared tree. The shared tree is shared across tasks to enable knowledge transfer.

As illustrated in Fig. 1, the final model for each task is formed by combining its task-specific tree with the shared

Algorithm 1: Overall Procedure of TSMTGP

Input : D^1_{train}, D^2_{train}: training sets of tasks 1 and 2.
Output: Ind^1_{best}, Ind^2_{best}: best individuals for tasks 1 and 2.
begin
- $gen \leftarrow 0$;
- Initialize the shared-tree population P^0_{shared} and task-specific populations P^0_1, P^0_2;
- **while** $gen < MaxGen$ **do**
  - Evaluate the shared-tree population P^{gen}_{shared} (Algorithm 2);
  - Select the best shared tree st_{best} from P^{gen}_{shared};
  - Construct two-tree populations $(P^{gen}_1, st_{\text{best}})$ and $(P^{gen}_2, st_{\text{best}})$;
  - Evaluate the two-tree populations (Algorithm 3) and update Ind^1_{best} and Ind^2_{best};
  - Compute ρ by Eq. (3);
  - **if** $\rho > 0.6$ **then**
    - `// High-similarity mode`
    - Generate $P^{gen+1}_{\text{shared}}$ by selection, subtree mutation, and subtree crossover;
    - Generate P^{gen+1}_1 and P^{gen+1}_2 by elitism, selection, subtree mutation, and subtree crossover;
    - $gen \leftarrow gen + 1$;
  - **else**
    - `// Low-similarity mode`
    - **while** $gen < MaxGen$ **do**
      - Independently evolve P^{gen+1}_1 and P^{gen+1}_2 by elitism, selection, and two-tree crossover and mutation;
      - Evaluate populations and update Ind^1_{best} and Ind^2_{best};
      - $gen \leftarrow gen + 1$.

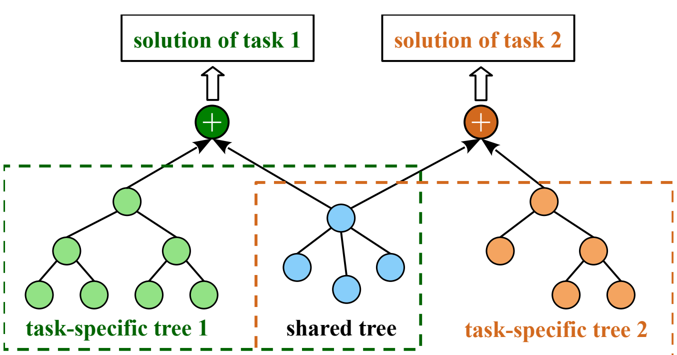

Fig. 1. An example of the new individual representation.

tree. The shared tree captures mathematical relationships shared across tasks, while the task-specific tree models task-dependent characteristics. Therefore, combining shared and task-specific representations improves both modeling accuracy and generalization for each task.

Specifically, the two trees are linearly combined to form the task model f_{task}. Let f_{shared} denote the expression represented by the shared tree, and let f_{specific} denote the expression of the task-specific tree. The task model is defined as Eq. (1).

$$f_{\text{task}}(\mathbf{x}) = f_{\text{shared}}(\mathbf{x}, \boldsymbol{\theta}_s) + f_{\text{specific}}(\mathbf{x}, \boldsymbol{\theta}_t) \tag{1}$$

where $\mathbf{x}$ denotes the input variables, $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_t$ are the parameter sets of the shared and task-specific trees, respectively.

C. Fitness Evaluation

During the evolution of TSMTGP, the optimal shared tree is selected from the shared-tree population based on fitness and combined with individuals from the task-specific tree population to form a two-tree individual population. Accordingly, different fitness evaluation measures are designed.

1) Fitness Evaluation of the Shared-Tree Population: The shared-tree population aims to capture knowledge shared between the two tasks. An effective shared tree should achieve strong performance on both tasks while maintaining low model complexity or size to enhance generalization. Prior studies indicate that simpler models often exhibit better generalization than overly complex ones [21]. Therefore, a composite fitness function is designed to jointly consider regression accuracy and tree size, as defined in Eq. (2).

$$f_{\text{shared}} = \text{rank1} + \text{rank2} + \alpha \cdot \text{rank3} \tag{2}$$

where rank1 and rank2 denote the ranks of individuals based on their RMSEs on tasks 1 and 2, respectively. rank3 denotes the rank based on tree size. This ranking-based strategy avoids scale discrepancies between RMSE and tree size, preventing biased fitness evaluation. It is worth noting that rank1 and rank2 are both assigned a weight of 1, while the weight of rank3, α, is set to 0.3 to penalize excessive complexity while preserving fitting accuracy.

The fitness evaluation procedure for the shared-tree population is detailed in Algorithm 2. First, each individual is evaluated by computing its RMSE on the training sets of two tasks, and its tree size is recorded. Second, individuals are ranked in ascending order according to the RMSEs and tree size to obtain rank1, rank2, and rank3. Finally, the fitness value of each individual is computed using Eq. (2), and the evaluated shared-tree population is returned.

2) Fitness Evaluation of the Two-Tree Individual Population: In TSMTGP, each individual consists of two trees, and the final regression model is formed by linearly combining the expressions of these trees. Accordingly, the objective of fitness evaluation is to assess how well the combined model fits the training data of a given task. Specifically, the prediction error of each individual on the task training set is computed to determine its quality.

RMSE is adopted as the fitness metric to quantify the discrepancy between the model predictions and the targets, thereby effectively reflecting regression performance. During evaluation, the mathematical expressions of the two trees are first extracted and linearly superimposed to construct the complete prediction model for the current task. The model is then evaluated on the task training data, and the RMSE over all training samples is calculated and assigned as the fitness value

Algorithm 2: Fitness Evaluation of the Shared-Tree Population

Input: D^1_{train}, D^2_{train}: training sets of tasks 1 and 2; P_{shared}: shared-tree population.
Output: Evaluated P_{shared}.
for each individual tree $t \in P_{\text{shared}}$ **do**
  Compute $\text{rmse}_{1,t} \leftarrow$ RMSE on D^1_{train};
  Compute $\text{rmse}_{2,t} \leftarrow$ RMSE on D^2_{train};
  Compute tree size $\leftarrow len(t)$;
end for
Rank all individuals by ascending rmse_1 to obtain rank1;
Rank all individuals by ascending rmse_2 to obtain rank2;
Rank all individuals by ascending tree size to obtain rank3;
for each individual tree $t \in P_{\text{shared}}$ **do**
  $f_{\text{shared},t} \leftarrow \text{rank1}_t + \text{rank2}_t + 0.3 \cdot \text{rank3}_t$
end for
return Evaluated P_{shared}.

of the individual. The detailed fitness evaluation procedure for the two-tree population is presented in Algorithm 3.

Algorithm 3: Fitness Evaluation of the Two-Tree Individual Population

Input: D_{train}: task training dataset; P: two-tree individual population.
Output: Evaluated P.
for each two-tree individual $(tree_1, tree_2) \in P$ **do**
  $f_{tree_1} \leftarrow$ mathematical expression represented by $tree_1$;
  $f_{tree_2} \leftarrow$ mathematical expression represented by $tree_2$;
  $f_{\text{task}} \leftarrow f_{tree_1} + f_{tree_2}$;
  Compute the RMSE of f_{task} on D_{train} and assign it as the fitness value;
end for
return Evaluated P.

D. The Task Similarity Measure Strategy

The proposed TSMTGP approach quantifies task similarity by examining the consistency of shared tree properties across different tasks. The measured task similarity directly controls the dynamic switching between two evolutionary modes, thereby achieving a balance between cross-task knowledge sharing and task-specific modeling.

Specifically, task similarity is quantified using SRCC. SRCC is a nonparametric, rank-based statistic that evaluates the monotonic relationship between two variables and remains effective even when the relationship is non-linear. The SRCC is computed as Eq. (3).

$$\rho = 1 - \frac{6 \sum_{i=1}^{N} (\text{s-rank1}_i - \text{s-rank2}_i)^2}{N(N^2 - 1)} \tag{3}$$

where s-rank1_i and s-rank2_i represent the ranks of the ith shared-tree individual on task 1 and task 2, respectively. The coefficient $\rho \in [-1, 1]$ represents the task similarity, where a larger absolute value $|\rho|$ indicates stronger consistency between the ranking trends of the two tasks and thus higher task similarity, while values close to zero imply weak or negligible task similarity.

Task similarity are measured using the absolute value of ρ and classified into different levels. Values in the ranges of 0.8–1.0 indicates very strong similarity, 0.6–0.8 strong similarity, 0.4–0.6 moderate similarity, 0.2–0.4 weak similarity, and 0.0–0.2 very weak or no similarity. When $|\rho| > 0.6$, tasks are considered strongly or very strongly similar; otherwise, they exhibit moderate, weak, or negligible similarity. Consequently, the proposed TSMTGP approach uses $\rho = 0.6$ as a decision threshold for adaptive strategy selection, enabling dynamic control of the evolutionary strategy between high- and low-similarity task scenarios.

IV. EXPERIMENT DESIGN

A. Datasets

The benchmark datasets follow the experimental settings in [6], as shown in Table I. $U[a, b, n]$ denotes n samples uniformly and randomly generated from the interval $[a, b]$. Five benchmark datasets are paired to form ten dual-task regression problems with different levels of task similarity.

TABLE I
BENCHMARK DATASETS

Task	Function	Dataset
$F1$	$x^3 + x^2 + x$	$U[-1, 1, 200]$
$F2$	$x^4 + x^3 + x^2 + x$	$U[-1, 1, 200]$
$F3$	$x^5 + x^4 + x^3 + x^2 + x$	$U[-1, 1, 200]$
$F4$	$x^5 - 2x^3 + x$	$U[-1, 1, 200]$
$F5$	$\sin(x^2)\cos(x) - 1$	$U[-1, 1, 200]$

The real-world datasets consist of two pairs of single-input datasets and two pairs of multi-input datasets. The single-input datasets are the carbon dioxide concentration dataset (CO2) and the diesel retail price dataset (DRP) [6]. According to [6], representative data points capable of capturing the overall trends in the original data were selected to construct the simplified datasets S_CO2 and S_DRP. This resulted in two dual-task problems: {CO2, S_CO2} and {DRP, S_DRP}. The multi-input datasets are two widely used regression datasets from the UCI Machine Learning Repository [22] were used. As shown in Table II, the red wine (Wine_R) and white wine (Wine_W) subsets within the wine quality dataset are treated as a dual-task SR problem. Female (Abalone_F) and male (Abalone_M) subsets of the abalone age dataset are used to construct another dual-task regression problem.

TABLE II
REAL-WORLD DATASETS

	Dataset	**Number of Features**	**Number of Instances**
1	CO2	1	434
	S_CO2	1	80
2	DRP	1	240
	S_DRP	1	80
3	Wine_R	10	1599
	Wine_W	10	4898
4	Abalone_F	7	1307
	Abalone_M	7	1528

B. Benchmark Methods and Parameter Settings

TSMTGP is compared with a single-task GP method and classical multitask GP method (MFGP) [6], [14]. MFGP is selected as a representative multitask method, which integrates tree-based GP with MFEA.

Parameters for all methods are provided in Table III. TSMTGP adopts a multi-population framework and initializes k independent populations, where k denotes the number of tasks. In contrast, MFGP employs a single-population multitask framework with an initial population of $100k$ individuals. All methods are executed for 50 generations to ensure fair comparisons. The function set is consistent with [23], defined as $\{+, -, \times, \div \text{ (protected)}, \sin, \cos, e^x, \ln(|x|)\}$. In MFGP, the random mating probability (rmp) is set to 0.3 [6].

TABLE III
PARAMETER SETTINGS

Parameter	Value
Population size	100
Number of generations	50
Maximum number of fitness evaluations	$5000 \times k$ (k is the number of tasks)
Crossover rate and mutation rate	0.9 and 0.1
Selection method	Tournament (size = 3)
Initial tree depth	2–6
Maximum tree depth	10
Population initialization	Ramped half-and-half
Fitness function	Root mean square error (RMSE)
Function set	$+, -, \times, \div$ (protected), $\sin, \cos, e^x, \ln(\lvert x \rvert)$

V. RESULTS AND DISCUSSIONS

This section compares the test RMSE of TSMTGP with standard GP and MFGP on benchmark and real-world datasets. Results are reported as mean $\pm$ variance over 30 runs, with the best values in bold. The Wilcoxon rank-sum test at the 0.05 level is used for statistical analysis, where $(+)$, $(=)$, and $(-)$ denote significantly better, equivalent, and worse performance of TSMTGP, respectively.

A. Results and Analysis on Benchmark Datasets

Table IV compares the performance of TSMTGP with Standard GP and MFGP on ten two-task problem instances. In Table IV, the first two columns denote the two tasks, $task1$ and $task2$, that are solved simultaneously. The remaining columns report the RMSE results of different methods on $task1$, reflecting the generalization error of each algorithm on the target task. Overall, TSMTGP exhibits strong competitiveness across most task pairs. Among the 20 task combinations, TSMTGP significantly outperforms GP in 5 task pairs and significantly outperforms MFGP in 5 task pairs, while remaining comparable on all others.

Further analysis shows that TSMTGP frequently attains the lowest RMSE on task pairs involving $F1$, $F2$, and $F3$, and significantly outperforms GP and MFGP in multiple cases. As illustrated in Table I, these tasks share overlapping functional structures and thus exhibit high similarity. This indicates that TSMTGP can effectively facilitate knowledge exchange among highly similar tasks, thereby enhancing multitask generalization performance and validating the effectiveness of the proposed individual representation.

TABLE IV
TEST RMSE RESULTS OF GP, MFGP, AND TSMTGP ON BENCHMARK DATASETS

$task1$	$task2$	**GP**	**MFGP**	**TSMTGP**
F1	F2	3.81e-02 ± 3.84e-02 (=)	2.66e-02 ± 2.75e-02 (=)	**1.84e-02 ± 1.40e-02**
	F3	3.81e-02 ± 3.84e-02 (=)	2.82e-02 ± 2.50e-02 (=)	**1.73e-02 ± 1.78e-02**
	F4	3.81e-02 ± 3.84e-02 (=)	3.19e-02 ± 3.68e-02 (=)	**3.10e-02 ± 2.68e-02**
	F5	3.81e-02 ± 3.84e-02 (=)	3.08e-02 ± 3.15e-02 (=)	**2.73e-02 ± 2.39e-02**
F2	F1	5.17e-02 + 2.22e-02 (+)	3.85e-02 ± 3.04e-02 (+)	**2.47e-02 ± 2.20e-02**
	F3	5.17e-02 ± 2.22e-02 (+)	4.63e-02 ± 2.41e-02 (+)	**3.18e-02 ± 2.37e-02**
	F4	5.17e-02 ± 2.22e-02 (=)	**4.06e-02 ± 2.66e-02 (=)**	5.36e-02 ± 4.24e-02
	F5	5.17e-02 ± 2.22e-02 (+)	**3.82e-02 ± 2.61e-02 (=)**	4.33e-02 ± 3.74e-02
F3	F1	8.08e-02 + 7.96e-02 (+)	7.34e-02 ± 3.84e-02 (+)	**4.23e-02 ± 3.99e-02**
	F2	8.08e-02 ± 7.96e-02 (+)	6.48e-02 ± 3.58e-02 (+)	**3.63e-02 ± 2.43e-02**
	F4	8.08e-02 ± 7.96e-02 (=)	6.91e-02 ± 5.17e-02 (=)	**5.68e-02 ± 3.56e-02**
	F5	8.08e-02 ± 7.96e-02 (=)	**5.92e-02 ± 2.34e-02 (=)**	5.96e-02 ± 4.49e-02
F4	F1	2.77e-02 + 2.25e-02 (=)	**1.95e-02 ± 1.42e-02 (=)**	2.96e-02 ± 2.53e-02
	F2	2.77e-02 ± 2.25e-02 (=)	**2.39e-02 ± 1.79e-02 (=)**	2.54e-02 ± 2.79e-02
	F3	2.77e-02 ± 2.25e-02 (=)	**2.35e-02 ± 1.99e-02 (=)**	2.95e-02 ± 2.18e-02
	F5	2.77e-02 ± 2.25e-02 (=)	2.67e-02 ± 2.94e-02 (=)	**2.55e-02 ± 2.45e-02**
F5	F1	**2.48e-02 + 1.70e-02 (=)**	2.71e-02 ± 1.35e-02 (=)	2.70e-02 ± 2.08e-02
	F2	2.48e-02 ± 1.70e-02 (=)	2.68e-02 ± 1.98e-02 (=)	**2.09e-02 ± 1.34e-02**
	F3	2.48e-02 ± 1.70e-02 (=)	3.06e-02 ± 1.89e-02 (+)	**2.09e-02 ± 1.60e-02**
	F4	**2.48e-02 ± 1.70e-02 (=)**	2.82e-02 ± 1.58e-02 (=)	2.53e-02 ± 1.75e-02
+/=/-		5/15/0	5/15/0	

In addition, for multitask optimization scenarios involving task $F5$, TSMTGP achieves a lower mean RMSE than MFGP in all four comparisons. As shown in Table I, task $F5$ has relatively low similarity with other tasks. This demonstrates that the similarity-aware adaptive strategy of TSMTGP can effectively regulate knowledge transfer across tasks, thereby maintaining stable performance even under low task similarity.

B. Results and Analysis on Real-World Datasets

To assess the generalization performance of TSMTGP, its RMSE results are compared with those of GP and MFGP on multiple real-world regression tasks, as shown in Table V.

Among the eight comparison groups, TSMTGP achieves significantly lower test RMSE on six tasks, demonstrating stronger generalization capability than GP. For instance, on the CO2 and SCO2 tasks, the mean test RMSE of TSMTGP is 7.36 and 8.92, respectively, compared with 10.7 and 11.8 obtained by GP, corresponding to reductions of approximately 31.2% and 24.4%. These results indicate that TSMTGP can more effectively capture data characteristics and alleviate overfitting. Similar performance gains are also observed on the DRP and SDRP tasks. On the Wine_R and Wine_W tasks, TSMTGP achieves test RMSE values comparable to those of GP, suggesting that the generalization gap between the two methods is relatively small on high-dimensional input problems. This phenomenon may be attributed to the increased complexity of high-dimensional data and the limited effectiveness of inter-task knowledge sharing under such conditions.

Compared with MFGP, TSMTGP achieves significantly lower test RMSE on two task pairs (SCO2 and SDRP) and comparable performance on the remaining six groups, with no cases of inferior performance. This indicates that TSMTGP is at least competitive with MFGP and exhibits superior generalization on selected tasks. In particular, the advantage of TSMTGP on the SCO2 task further confirms the effectiveness of the proposed TSMTGP approach, which can be attributed to

TABLE V

TEST RMSE RESULTS OF GP, MFGP, AND TSMTGP ON REAL-WORLD DATASETS

	Tasks	GP	MFGP	TSMTGP
1	CO2	1.07e+01 ± 4.90e+00 (+)	7.78e+00 ± 3.03e+00 (=)	**7.36e+00 ± 2.73e+00**
	SCO2	1.18e+01 ± 4.78e+00 (+)	9.31e+00 ± 2.76e+00(=)	**8.92e+00 ± 2.45e+00**
2	DRP	5.73e-01 ± 5.10e-02 (+)	5.25e-01 ± 5.43e-02(=)	**5.17e-01 ± 5.85e-02**
	SDRP	7.87e-01 ± 9.29e-02(+)	7.38e-01 ± 7.17e-02(=)	**7.14e-01 ± 7.55e-02**
3	Wine_R	3.16e-02 ± 5.74e-03(=)	3.18e-02 ± 3.37e-03(=)	**2.91e-02 ± 6.26e-03**
	Wine_W	1.13e-02 ± 1.79e-03(=)	1.16e-02 ± 1.17e-03(+)	**1.11e-02 ± 1.03e-03**
4	Abolone_F	3.34e+00 ± 2.61e+00(+)	2.66e+00 ± 1.86e-01(+)	**2.64e+00 ± 5.64e-01**
	Abolone_M	2.58e+00 ± 1.77e-01(+)	2.44e+00 ± 8.80e-02(=)	**2.42e+00 ± 1.30e-01**
	+/=/-	6/2/0	2/6/0	

its similarity-based evolutionary strategy and novel individual representation that enable more effective inter-task knowledge exchange.

Overall, the results on real-world datasets validate the superior generalization performance of TSMTGP. Compared with GP, it achieves lower test errors on most tasks, and compared with MFGP, it demonstrates consistent or improved performance without any degradation. These results confirm that TSMTGP effectively improves generalization, making it well-suited for practical multitask regression problems.

VI. FURTHER ANALYSIS

A. Validation of the novel individual representations

To verify the effectiveness of the proposed novel individual representation, a visual analysis of the evolved solutions was conducted. Fig. 2 presents a representative solution for the task pair $(F1, F2)$. Specifically, the solution for $F1$ consists of the task-specific tree $f_{\text{specific1}}$ and the shared tree f_{shared}, while the solution for $F2$ is composed of the task-specific tree $f_{\text{specific2}}$ and the same shared tree f_{shared}. The expressions of the three trees are given as follows:

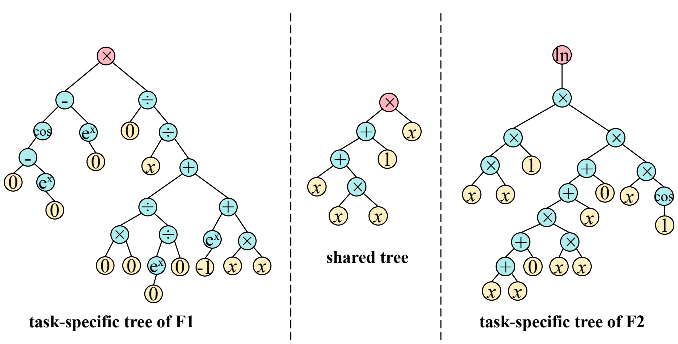

Fig. 2. An example solutions for F1 and F2.

$$f_{\text{specific1}} = 0 \tag{4}$$

$$f_{\text{shared}} = (x + x \cdot x + 1) \cdot x = x^3 + x^2 + x \tag{5}$$

$$\begin{aligned} f_{\text{specific2}} &= \ln\big(x^2 \cdot (2x \cdot x^2 + x) \cdot x \cdot \cos(1)\big) \\ &= \ln\big((2x^6 + x^4) \cdot \cos(1)\big) \end{aligned} \tag{6}$$

As illustrated by the example solutions for $F1$ and $F2$ in Fig. 2, the right subtree of the task-specific tree for $F1$ forms a division structure with a zero denominator, causing the subtree to evaluate to zero. Since the root node of this tree is a multiplication operator, the entire task-specific tree represents a value of zero, i.e., $f_{\text{specific1}} = 0$.

Moreover, the function learned by the shared tree f_{shared} accurately captures the shared structural components of $F1$ and $F2$. By adding $f_{\text{specific1}}$ to f_{shared}, the resulting expression corresponds to the solution of $F1$. For $F2$, the task-specific tree $f_{\text{specific2}}$ is adaptively adjusted based on the learned shared structure and combined with f_{shared} to form the final solution. This observation demonstrates that, in multitask SR, the shared tree in the proposed individual representation can effectively extract shared information across tasks, enabling efficient knowledge sharing. Meanwhile, task-specific trees provide flexible adaptations tailored to individual task requirements, thereby enhancing overall modeling performance and learning efficiency.

B. Validation of the Similarity-Based Evolutionary Strategy

To validate the effectiveness of the similarity-based evolutionary strategy, the task similarity measure strategy is removed from TSMTGP, and only the evolutionary strategy designed for high-similarity tasks is employed. The resulting variant is denoted as *TSMTGP-ns*. Fig. 3 compares the test RMSE performance of MFGP, TSMTGP, and TSMTGP-ns on the benchmark datasets. The bar chart reports the average test RMSE of the three algorithms. In the Fig. 3, T1–T20 denote the 20 individual tasks corresponding to the ten two-task problems listed in Table III, with the task order consistent with that in the table.

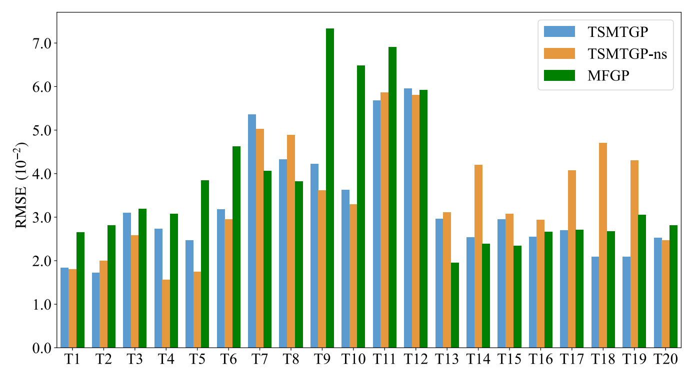

Fig. 3. The average test RMSE of TSMTGP, TSMTGP-ns, and MFGP on benchmark datasets.

As shown in Fig. 3, for tasks T1–T13, TSMTGP and TSMTGP-ns exhibit nearly identical average RMSE values, indicating no significant performance difference. In contrast, for tasks T14 and T15, where task similarity is low, TSMTGP-ns produces substantially higher average RMSE than TSMTGP. This demonstrates that TSMTGP can automatically monitor task similarity and adaptively switch to an evolutionary strategy tailored for low-similarity tasks, thereby effectively mitigating negative knowledge transfer under low-similarity conditions. These results validate the effectiveness of the proposed similarity-based evolutionary strategy.

However, for low-similarity tasks T17–T20, TSMTGP-ns exhibits significantly higher average RMSE than both TSMTGP and MFGP. This suggests that although the shared-tree-based knowledge sharing mechanism is effective for high-similarity tasks, it is less suitable for low-similarity scenarios. Notably, TSMTGP consistently achieves the lowest average RMSE on tasks T17–T20, highlighting the advantage of dynamically switching to a low-similarity evolutionary mode. Overall, these results demonstrate that the similarity-based evolutionary strategy significantly enhances both performance and robustness, particularly when handling low-similarity tasks.

VII. Conclusion

The TSMTGP approach was proposed to SR. A novel two-tree individual representation was designed, consisting of a shared tree and a task-specific tree, which enables shared structure learning while preserving task independence, supported by corresponding fitness evaluation strategies. Moreover, task similarity was quantified using SRCC, providing a principled basis for switching between shared and independent evolutionary modes. By dynamically evaluating task similarity, TSMTGP adopted different evolutionary strategies for high- and low-similarity modes, achieving an effective balance between cross-task knowledge sharing and task-specific optimization. Comprehensive evaluations on benchmark and real-world datasets showed that TSMTGP consistently outperformed comparative methods, particularly on low-similarity task pairs, highlighting its strong generalization ability in multitask SR.

Future work will explore more advanced adaptive evolutionary strategies, including dynamic population structures and operator selection mechanisms, to further enhance the performance. In addition, more accurate and robust task similarity measures will be investigated to support more effective similarity-aware knowledge exchange mechanisms.

Acknowledgment

This research was funded by the National Natural Science Foundation of China (62376253) and Key Research and Development Program of Henan (251111113900), and Henan Province Scientific Research Joint Fund Industrial Category Major Program (245101610001).

References

[1] D. Angelis, F. Sofos, and T. E. Karakasidis, "Artificial intelligence in physical sciences: Symbolic regression trends and perspectives," *Archives of Computational Methods in Engineering*, vol. 30, no. 6, pp. 3845–3865, 2023.

[2] H. Zhang, A. Zhou, H. Qian, and H. Zhang, "Ps-tree: A piecewise symbolic regression tree," *Swarm and Evolutionary Computation*, vol. 71, p. 101061, 2022.

[3] I. Azzali, N. D. Cilia, C. De Stefano, F. Fontanella, M. Giacobini, and L. Vanneschi, "Automatic feature extraction with vectorial genetic programming for alzheimer's disease prediction through handwriting analysis," *Swarm and Evolutionary Computation*, vol. 87, p. 101571, 2024.

[4] Z. Huang, Y. Mei, and J. Zhong, "Semantic linear genetic programming for symbolic regression," *IEEE Transactions on Cybernetics*, vol. 54, no. 3, pp. 1321–1334, 2024.

[5] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: a computer science view of cognitive multitasking," *Cognitive Computation*, vol. 8, pp. 125–142, 2016.

[6] J. Zhong, L. Feng, W. Cai, and Y.-S. Ong, "Multifactorial genetic programming for symbolic regression problems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4492–4505, 2018.

[7] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[8] S.-M. Udrescu and M. Tegmark, "Ai feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, p. eaay2631, 2020.

[9] Y. Jin, W. Fu, J. Kang, J. Guo, and J. Guo, "Bayesian symbolic regression," *arXiv preprint arXiv:1910.08892*, 2019.

[10] T. Mundhenk, M. Landajuela, R. Glatt, C. P. Santiago, B. K. Petersen *et al.*, "Symbolic regression via deep reinforcement learning enhanced genetic programming seeding," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 912–24 923, 2021.

[11] Q. U. Nguyen, T. H. Nguyen, X. H. Nguyen, and M. O'Neill, "Improving the generalisation ability of genetic programming with semantic similarity-based crossover," in *Genetic Programming: 13th European Conference, EuroGP 2010, Istanbul, Turkey, April 7–9, 2010, Proceedings*. Springer, 2010, pp. 184–195.

[12] C. Raymond, Q. Chen, B. Xue, and M. Zhang, "Genetic programming with rademacher complexity for symbolic regression," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 2657–2664.

[13] K. Y. Chan, C. K. Kwong, T. S. Dillon, and Y. Tsim, "Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming," *Applied Soft Computing*, vol. 11, no. 2, pp. 1648–1656, 2011.

[14] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[15] L. Zhou, L. Feng, K. C. Tan, J. Zhong, Z. Zhu, K. Liu, and C. Chen, "Toward adaptive knowledge transfer in multifactorial evolutionary computation," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2563–2576, 2020.

[16] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, and C. Yue, "An evolutionary multitasking optimization framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 263–277, 2022.

[17] X. Hao, R. Qu, and J. Liu, "A unified framework of graph-based evolutionary multitasking hyper-heuristic," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 35–47, 2020.

[18] F. Zhang, Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang, "Multitask genetic programming-based generative hyperheuristics: A case study in dynamic scheduling," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10 515–10 528, 2021.

[19] Y. Bi, B. Xue, and M. Zhang, "Multitask feature learning as multiobjective optimization: A new genetic programming approach to image classification," *IEEE Transactions on Cybernetics*, vol. 53, no. 5, pp. 3007–3020, 2022.

[20] R. Liu, H. Lv, P. Yang, and R. Wang, "A multi-task genetic programming approach for online multi-objective container placement in heterogeneous cluster," *Complex & Intelligent Systems*, vol. 11, no. 1, p. 29, 2025.

[21] Y. Bi, B. Xue, and M. Zhang, "Learning and sharing: A multitask genetic programming approach to image feature learning," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 218–232, 2021.

[22] A. Asuncion, D. Newman *et al.*, "Uci machine learning repository," 2007.

[23] J. McDermott, D. R. White, S. Luke, L. Manzoni, M. Castelli, L. Vanneschi, W. Jaskowski, K. Krawiec, R. Harper, K. D. Jong *et al.*, "Genetic programming needs better benchmarks," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, 2012, pp. 791–798.