Evolutionary Design of a PSO-Tuned Multigene Symbolic Regression Genetic Programming Model for River Flow Forecasting

Alaa Sheta[1], Amal Abdel-Raouf[2], Khalid M. Fraihat[3], Abdelkarim Baareh[4]
Computer Science Department, Southern Connecticut State University, New Haven CT, USA[1,2]
Applied Science Department, Al-Balqa Applied University, Ajlune College, Jordan[3,4]

***Abstract*—The earth's population is growing at a rapid rate, while the availability of water resources remains limited. Water is required for various purposes, including drinking, agriculture, industry, recreation, and development. Accurate forecasting of river flows can have a significant economic impact, particularly in agricultural water management and planning during water resource scarcity. Developing precise river flow forecasting models can greatly improve the management of water resources in many countries. In this study, we propose a two-phase model for predicting the flow of the Blackwater river located in the South Central United States. In the first phase, we use Multigene Symbolic Regression Genetic Programming (MG-GP) to develop a mathematical model. In the second phase, Particle Swarm Optimization (PSO) is employed to fine-tune the model parameters. Fine-tuning the MG-GP parameters improves the prediction accuracy of the model. The newly fine-tuned model exhibits 96% and 94% accuracy in training and testing cases, respectively.**

Keywords*—*River flow; forecasting; genetic programming; evolutionary computation; particle swarm optimization

I. Introduction

In recent decades, river flow forecasting has become a key research topic because it has substantial practical applications in various fields. Forecasting indicates predicting or estimating future events, conditions, or trends based on accessible data from previous events. Forecasting aims to offer a reliable guess about what could happen. River flow forecasting can help 1) the effective management of floods by delivering an early alert and permitting arrangements to be made to avoid damages [1] 2) assist in the supervision of water resources by offering data on the accessibility and timing of water supply to allow for better optimization of water allocation and guarantee that water resources are used effectively [2] 3) offer farmers with adequate data on the timing and amount of water accessibility, permitting them to plan their implanting and harvesting plans [3] 4) improve the supervision of hydropower generation by offering information on the likely flow of water, permitting power plants to be driven more economically [4], [5] and 5) the management of environmental matters, such as the safety of wetlands and fish habitats, so we may identify regions that require protection and plan healthy ecosystems [6].

A. Statistical Models and their Limitations

Developing time-series forecasting models for river flows were explored using statistical models [7], [8]. Forecasting models such as the regression and neural network were presented in [9], [10]. For example, in Equation 1, $y(k)$ is predicted based on the values of $y(k-1),\dots,y(k-n)$, n is the delay in time [11], [12].

$$y(k) = a_0 + a_1 y(k-1) + \cdots + a_n y(k-n) \tag{1}$$

Some forecasting tools are developed based on statistical models, especially if the seasonal prediction of the water flow is needed as in [7], which forecasts the availability of the water resources supplied by the mountains in central Asia. Another work was introduced in [8], which used the River Vouga Basin in Portugal as a case study utilizing a statistical time series model that analyzed and predicted the water quality. The study showed that for such complex database models, it is challenging to use statistical analysis.

Although statistical models have some advantages in river flow forecasting, there are also several potential drawbacks to consider, such as:

- Statistical models are likely developed utilizing historical data, which means they consider specific features for model design, such as precipitation patterns, land use changes, and climate variability. Thus, they might not accurately predict the flow with extreme weather events or environmental changes.
- Statistical models are susceptible to data outliers that can affect the accuracy of the forecast. The model may not accurately predict future river flow if the historical data includes outliers.
- Statistical models overfit the data; this can happen if the model fits the noise in the data rather than the underlying patterns. Overfitting can lead to poor model generalization ability and unsuccessful predictions.

Therefore, assessing the statistical model's limitations and probable weaknesses is essential while developing forecasting models.

B. Why Evolutionary Computation Models?

Recently, Evolutionary Computation (EC) models have been presented to handle modeling and optimization problems [13]. Some well-known EC models are genetic algorithms (GAs) and genetic programming (GP). EC-based models show many advantages in the field of forecasting. Some of these abilities are:

- They can handle nonlinear relationships between river flow and other factors that traditional statistical models may not easily capture. This is because they can manage various functions with multiple variables simultaneously.
- EC-based models can adapt to varying environmental circumstances, such as climate variability or land use changes, by adjusting the model's parameters over time.
- EC-based models are immune to noise and missing values

Many forecasting models were presented in the past based on Artificial Intelligent (AI) methods such as Artificial Neural Networks (ANN) in different areas such as in [14] and some models are specifically for river flow forecast as in [15], [16]. In [1], the author used data on flooding in the city of Jakarta and developed a model that will be used to predict the rainfall and prevent any possible future damage in the surrounding area using ANN. Another study shows that ANN can be used to predict the water flow of dams that have much flood data, while regression models are better for dams that have limited flood data [10]. The author in [17] introduced a forecasting method that combined both ANN and general regression. In [18]–[21], the authors presented several forecasting models for the Nile river flow in Egypt using GP, ANN, and FL. In [22], authors contributed a hybrid radial-basis function network with weight-tuning GAs for time-series forecasting. A comparison between Auto Regression (AR) modeling, gene expression programming (GEP), radial basis function network and FeedForward (FF) neural networks, and adaptive neural-based fuzzy inference system (ANFIS) methods to forecast the average monthly flow for a River in Turkey was introduced in [23].

EC-based models have shown better outcomes in river flow forecasting than traditional statistical models. In [24], the authors provided a comparison between support vector regression (SVR) and artificial neural network (ANN) models, which are both evolutionary-based models, with traditional statistical models for river flow forecasting in the southwestern United States. The results show that both SVR and ANN models outperform the statistical models. Another comparative study shows that a three-layer ANN model outperforms Multivariate Linear Regression Analysis (MLRA) model when predicting the water flow in the watershed of Tarim [9].

C. Goals

In this paper, we present a Multigene GP mathematical model that can be used for forecasting the flow of the Blackwater river. The model is optimized using the PSO algorithm to improve its accuracy. To train the model, we used flow measurements from 1975 to 1984 and tested them using different measures from 1984 to 1993. The structure of the paper is as follows. In Section I, we provided an introduction and motivation for solving the river flow forecasting problem. Section II discusses the importance of the Blackwater river in the USA. Steps for developing a forecasting model are shown in III. Section IV describes the newly developed forecasting model together with the evolutionary computational methods used to build the model. Section V lists our evaluation criteria, and we conclude our work with Section VII.

II. The Blackwater river in USA

The Blackwater river, which originates in Reynolds County, Missouri, in the Ozark Mountains, runs through southeastern Missouri and eastern Arkansas before eventually joining the White River near Newport, Arkansas, after covering a distance of 280 miles (450 km) with a southeasterly flow towards Poplar Bluff, Missouri. Due to different reasons, the Blackwater river holds significant value to the United States. Some are the following:

- The Blackwater river is a vital water source for irrigation, industrial use, and recreation in Missouri and Arkansas. The river also supports a prosperous fishing industry, donating to the local economy.
- The Blackwater river is the residence of many rare species, including the Missouri bladderpod, the eastern massasauga rattlesnake, and the Ozark cavefish. The river also delivers essential habitats for migratory birds and other wildlife.
- The Blackwater river is a famous terminus for recreational activities such as fishing, boating, and swimming. It draws visitors from throughout the region.
- The Blackwater river has played an essential role in the history and culture of the region. Native American tribes used the river for transportation and trade, later serving as a significant transportation route for steamboats and other vessels.

The Black Water River's flow data was recorded and gathered by the U.S. Geological Survey (USGS) at station number 02047500 whose location is shown in Fig. 1, as reported in [25]. The first 6 years of this dataset, spanning from October 1st, 1990, to September 30th, 1996, was used as the training data, and the final year spanning from October 1st, 1996, to September 30th, 1997, was used as the testing data.

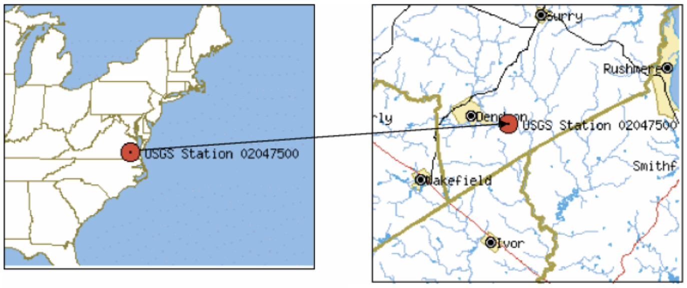

Fig. 1. The location of station no. 02047500 operated by the USGS.

III. Forecasting Model

Developing a forecasting model involves several steps. Here is a general framework to follow steps:

1) Identify the scope of the problem, including the data sources and any constraints.

2) Collect the data required to build the model. Clean the data as needed.
3) Choose a forecasting model suitable for the data under study. Many models in the literature can be used, such as regression models, time series models, neural networks, and evolutionary models. In our case, we are adopting the MG-GP model.
4) Use the historical data to train the model; this involves selecting an appropriate model structure and evaluation criteria that fulfill the error minimization to fit the data best.
5) Once the model is trained, use a testing data set to evaluate the development model quality.
6) Fine-tune the model parameters and make any necessary adjustments. This may involve tweaking the model parameters. In our case, we are adopting PSO for better tuning the MG-GP model.
7) Once the model has been trained and validated, it can be used to forecast new data.

Developing a forecasting model demands careful planning, data preprocessing, model selection and tuning, and ongoing monitoring and refinement (See Fig. 2). There are many forecasting models developed in the literature as the models in [9]–[12]. Moreover, the author in [26] includes a study comparing different preprocessing techniques and shows how to partition the complex forecasting problem into more minor sub-problems to solve.

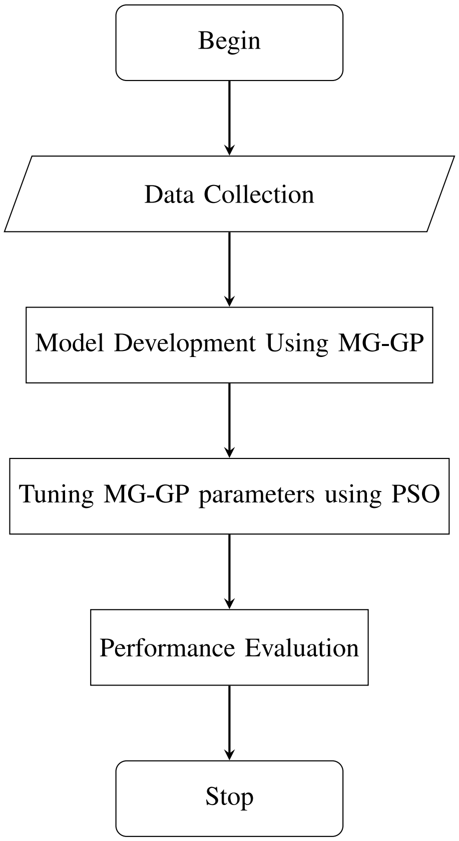
Begin
Data Collection
Model Development Using MG-GP
Tuning MG-GP parameters using PSO
Performance Evaluation
Stop

Fig. 2. Flowchart for the system identification process.

IV. Methodology

A. What is Genetic Programming?

Genetic programming (GP) is a kind of evolutionary computation that uses algorithms inspired by biological evolution to generate computer programs automatically. GP was introduced by J. Koza in 1992 [27] at Stanford University. GP is a population-based approach where computer programs are evolved over multiple generations using parameters inspired by nature, such as selection, reproduction, and mutation operators [27]–[29]. GP involves several evolutionary phases, such as:

- **Initialization:** A population of randomly generated programs is created.
- **Fitness Evaluation:** Each program in the population is evaluated based on a user-selected fitness function that calculates the performance of each solution to a given problem.
- **Selection:** The best-performing programs are selected for reproduction based on their fitness value.
- **Reproduction:** The selected programs are used as parents to generate new offspring programs using crossover and mutation operators.
- **Replacement:** The offspring programs replace the least-fit programs in the population, creating a new generation of programs.
- **Termination:** The GP process resumes until a stopping criterion is satisfied, such as a maximum number of generations or a satisfactory fitness level. In GP, the programs are represented as a tree structure, where each node denotes a function or operation, and the branches illustrate the operands or arguments. GP can develop favorably optimized programs by evolving the population of programs over multiple generations. The GP algorithm can be presented as given in Algorithm 1.

Algorithm 1 Genetic Programming Algorithm

```
Input: Training data $D$, population $P$, number of genes $G$,
       number of individuals $N$, maximum depth $d$, crossover
       rate $p_c$, mutation rate $p_m$, fitness $f$, $T$ terminal condition
Output: Optimal solution
initialization;
while $\neg T$ do
    1) Evaluate fitness of each individual in $P$;
    2) Select parents using $f$;
    3) Apply $p_c$ and $p_m$ to generate new offspring;
    4) Replace old population with new population;
end
return Best individual in $P$
```

GP has been successfully used in a variety of applications such as manufacture process modeling [30]–[32], fermentation process modeling [33], timetabling problem [34] and stock market prediction [35].

1) Crossover in GP: Let's consider two parent trees T_1 and T_2, and we want to perform a crossover operation to create two new offspring trees T_3 and T_4.

$$T_1 = \begin{bmatrix} + & \\ \times & 2 \\ x & 3 \end{bmatrix} \quad T_2 = \begin{bmatrix} - & \\ \div & 4 \\ y & 5 \end{bmatrix}$$

First, we randomly select a crossover point in each tree. Let's assume we chose the second node in T_1 and the third node in T_2:

$$T_1 = \begin{bmatrix} + & \\ \times & 2 \\ x & 3 \end{bmatrix} \quad T_2 = \begin{bmatrix} - & \\ \div & 4 \\ y & 5 \end{bmatrix}$$

We swap the subtrees rooted at the crossover points to obtain the offspring trees:

$$T_3 = \begin{bmatrix} + & \\ \div & 4 \\ x & 3 \end{bmatrix} \quad T_4 = \begin{bmatrix} - & \\ \times & 2 \\ y & 5 \end{bmatrix}$$

The resulting trees can then be evaluated and selected based on their fitness values.

2) Mutation in GP: Let's consider a parent tree T_1, and we want to perform a mutation operation to create a new offspring tree T_2.

$$T_1 = \begin{bmatrix} + & \\ \times & 2 \\ x & 3 \end{bmatrix}$$

First, we randomly select a node in the tree to mutate. Let's assume we selected the second node in T_1:

$$T_1 = \begin{bmatrix} + & \\ \times & 2 \\ x & 3 \end{bmatrix}$$

We randomly select a new function or terminal node to replace the selected node. Let's assume we selected the terminal node 4:

$$T_2 = \begin{bmatrix} + & \\ 4 & \\ x & 3 \end{bmatrix}$$

The resulting tree can then be evaluated and selected based on its fitness value.

B. What is Symbolic Regression?

Suppose we have a data set of input-output pairs (x_i, y_i), where x_i is the input variable, and y_i is the corresponding output variable. We want to find a function $f(x)$ that best approximates the relationship between the input and output variables. The symbolic regression problem J can be formulated as:

$$J = \min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda C(f) \tag{2}$$

Where $C(f)$ is a measure of the complexity of the function f, and λ is a regularization parameter that balances the trade-off between accuracy and complexity.

In symbolic regression, the function $f(x)$ is typically represented as a tree structure, where each node in the tree corresponds to a function or operator, and the leaves correspond to the input variables or constants. The tree structure is evolved using GP to find the best function that fits the data. Fig. 3 shows a symbolic regress tree. This expression can be presented using the following equation:

$$y = x\ sin(5x) + cos(7x) \tag{3}$$

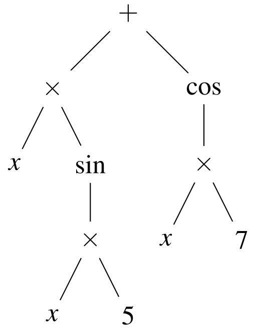

Fig. 3. Symbolic regression tree for expression.

C. What is Multigene Symbolic Regression GP?

Multigene Genetic Programming (MGGP) is an evolutionary algorithm used for symbolic regression to discover a mathematical expression that best fits a given dataset. MGGP boosts the basic GP algorithm by allowing multiple functions or genes to evolve simultaneously.

In MGGP, each individual in the population is represented by a set of genes, each of which can be an independent mathematical expression. The algorithm evolves these genes through genetic operations such as mutation, crossover, and selection, seeking to optimize the fitness function, which measures how well the set of genes fits a given dataset.

MG-GP has been known to be a powerful tool for solving complex regression problems, such as those found in modeling and optimization of manufacturing processes [36], [37], software effort estimation [38], image reconstruction [39], and many others [40], [41]. It can address problems with datasets that have a complex and noisy relationship. However, MGGP can be computationally expensive, especially when dealing with large datasets or complex models requiring significant computational resources and optimization methods.

The following equation can represent the multigene symbolic model:

$$y_t = \sum_{i=1}^{n} w_i f_i(x_t) + b \tag{4}$$

where y_t is the predicted value at time t, n is the number of genes, w_i is the weight of gene i, $f_i(x_t)$ is the expression level of gene i at time t, x_t is the input data at time t, and b is the bias term.

The expression level of gene i at time t can be further defined as:

$$f_i(x_t) = \phi_i(g_i(x_t)) \tag{5}$$

where ϕ_i is the activation function of gene i and $g_i(x_t)$ is the regulatory function of gene i at time t.

The regulatory function $g_i(x_t)$ can be modeled using a polynomial function:

$$g_i(x_t) = \sum_{j=0}^{d} c_{ij} x_t^j \tag{6}$$

where d is the degree of the polynomial and c_{ij} is the coefficient of the j-th term of gene i. Finally, the activation function ϕ_i can be defined as a sigmoid function:

$$\phi_i(z) = \frac{1}{1+e^{-\alpha_i z}} \tag{7}$$

where z is the input signal and α_i is the slope parameter of gene i. These equations can be combined to form a multigene symbolic model for predicting the flow of the Blackwater river.

D. Particle Swarm Algorithm

PSO is a metaheuristic search algorithm inspired by social organisms' collective behavior, particularly the flocking of birds and schooling of fish. Kennedy and Eberhart first introduced it in 1995 [42].

In PSO, a group of particles (representing candidate solutions to a problem) progress through the search space, adjusting their velocities according to their own best-known position and the best-known position of the swarm. Each particle holds its position and velocity and adapts by comparing its fitness value with the best fitness value found by the swarm. The algorithm gradually converges toward an optimal solution by iteratively modifying the velocities of the particles.

The equations that govern the PSO process of evolution to update both the velocity v and position x are given as follows:

$$v_{i,t} = wv_{i,t-1} + c_1 r_1 (p_{i,t-1} - x_{i,t-1}) + c_2 r_2 (g_{t-1} - x_{i,t-1}) \tag{8}$$

$$x_{i,t} = x_{i,t-1} + v_{i,t} \tag{9}$$

where w is the inertia weight, c_1 and c_2 are the cognitive and social learning coefficients, r_1 and r_2 are random values between 0 and 1, $p_{i,t}$ is the best position of particle i in dimension t, and g_t is the best position of the swarm in dimension t. The PSO algorithm is shown in Algorithm 2.

PSO has been successfully applied to various optimization problems, including computer network design [43] optimization of PID Controller [44]. It is beneficial when the search space is large and complex, and traditional optimization methods such as gradient descent and genetic algorithms may need to be more efficient.

Algorithm 2 PSO Algorithm

```
Input: Objective function $f(x)$, Swarm size $N$, Maximum number of iterations $T$, Initial particle positions $x_i$, and velocities $v_i$
Output: Optimal solution $x^*$
for $i = 1$ to $N$ do
    Initialize particle position $x_i$ and velocity $v_i$ within the search space;
    Evaluate particle fitness $f_i = f(x_i)$;
    Initialize personal best $p_i = x_i$ and best fitness $f_{p_i} = f_i$;
end
Find global best position $g = \arg\min_{f_i} f_i$;
for $t = 1$ to $T$ do
    for $i = 1$ to $N$ do
        Update velocity: $v_{i,t}$;
        Update position: $x_{i,t}$;
        Evaluate fitness: $f_{i,t} = f(x_{i,t})$;
        if $f_{i,t} < f_{p_i}$ then
            Update personal best: $p_i = x_{i,t}$ and $f_{p_i} = f_{i,t}$;
        end
        if $f_{i,t} < f_g$ then
            Update global best: $g = x_{i,t}$;
        end
    end
end
return $g$
```

V. Model Evaluation

Model evaluation is necessary for any forecasting process that evaluates how well a model predicts the interest results. It is essential to ensure that the model is correct and trustworthy before using it to make predictions and forecasting. Some of the criteria we are adopting in this research include the Variance-Accounted-For (VAF), the Mean Squares Error (MSE), and the Manhattan distance (MD). The following equations describe the proposed mathematical formulation of the adopted performance criteria.

1) Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{10}$$

2) Variance-Accounted-For (VAF):

$$VAF = [1 - \frac{var(y-\hat{y})}{var(y)}] \times 100\% \tag{11}$$

3) Mean Squares Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{12}$$

4) Euclidian distance (ED):

$$ED = \sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{13}$$

5) Manhattan distance (MD):

$$MD = (\sum_{i=1}^{n}|y_i - \hat{y}_i|) \tag{14}$$

y and $\hat{y}$ are the observed and the predicted river flow values based on the proposed model, and n is the number of measurements utilized in our experiments, respectively.

VI. Developed Multigene GP Model

We utilized the GPTIPS 2 Matlab toolbox to develop the proposed forecasting model based on MG-GP. GPTIPS 2 is an open-source software platform for symbolic data mining that delivers an easy-to-use code that can be customized framework for GP. GPTIPS 2 permits users to perform symbolic regression, classification, clustering, and feature selection on complex data sets using GP [45].

To develop the Multigene GP model, a user has to setup the following:

- The maximum number of genes, denoted as G_{max}, identifies the maximum number of genes allowed to be used in the model.
- The maximum tree depth, denoted as D_{max}, which controls the complexity of the model. Limiting the tree depth can result in a simpler model but may also reduce performance.
- When using GPTIPS, we obtain the optimal weights for the genes utilizing ordinary least squares to regress the genes against the output data.

The evolutionary process of the MG-GP algorithm was evolved using the parameters listed in Table I. The best generated Blackwater river Multigene GP model forecasting model is given in Equation 15. The adopted fitness function to evaluate the MG-GP model was selected as the Root Mean Squares Error (RMSE).

TABLE I. MULTIGENE GP TUNING PARAMETERS

Population size	100
Number of generations	200
Selection mechanism	Tournament
Tournament Size	5
Max. tree depth	5

$$\begin{aligned} y(t) &= 2.049\ y(t-1) - 1.524\ y(t-2) + 0.7478\ y(t-3) \\ &- 0.2646\ y(t-4) - 0.05178\ y(t-2)y(t-4) \\ &+ 0.04482\ y(t-2)y(t-5) - 0.05178\ y(t-1)^2 \\ &+ 0.04482\ y(t-2)^2 + 0.01711 \qquad (15) \end{aligned}$$

In Table II, we show the tuning parameters of MG-GP. The convergence of GP with a population size of 100 trees over 200 generations is depicted in Fig. 4. The upper section of the graph displays the log_{10} value of the population's best Root Mean Square Error (RMSE) achieved during each generation. Meanwhile, the lower section shows the population's mean RMSE achieved over time.

TABLE II. TUNING PARAMETERS OF GP

Number of generations	3000
Popultaion Size	100
Tournament Size	5
Maximum Genes	5
Functions Set	×, -, +
Acceleration factor c_1, c_2	2

Scatterplots have several advantages, including displaying the relationship between two variables, identifying outliers, evaluating patterns or trends, assessing the distribution of variables, and comparing groups. They provide a visual representation of data points and allow for easy interpretation of the data, making them a useful tool for data analysis and visualization. The scatterplots in both training and testing are given in Fig. 5. In training set the RMSE is calculated to be 0.19959 and the R^2 coefficient value is 0.96059 while in testing set, the RMSE is calculated to be 0.23312 and the R^2 coefficient value is 0.94296.

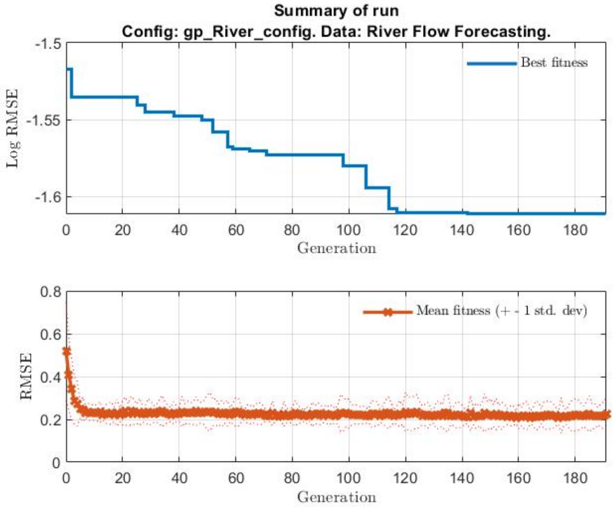

Fig. 4. MG-GP convergence curves.

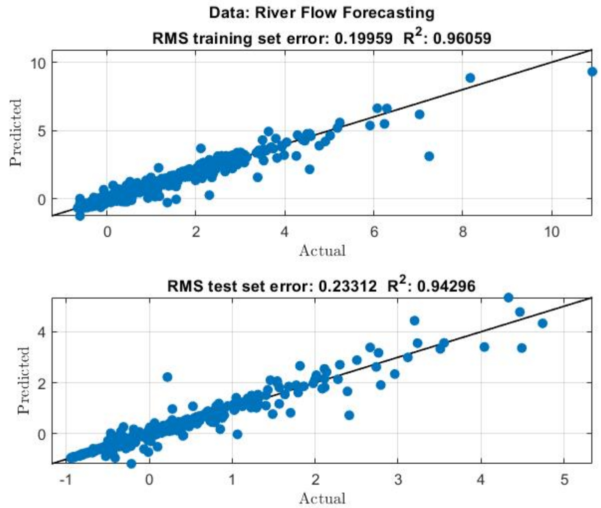

Fig. 5. Scatter plots in both training and testing cases.

The gene weights for symbolic regression are presented in Fig. 6, with gene three and four identified as the most significant for developing the forecasting model. The equations for these genes are provided below:

$$\begin{aligned} M_3 &= f(y(t-3) - y(t-2)) \\ M_4 &= f(y(t-1) - 9.785)) \qquad (16) \end{aligned}$$

As shown in Equation 16, the variables $y(t-1), y(t-2)$, and $y(t-3)$ are used in the equations for genes three and four. Fig. 7 shows the five symbolic GP models developed.

The simplicity and compactness of the final model make it easy to evaluate. The performance of the model was evaluated, and the results are presented in Table III.

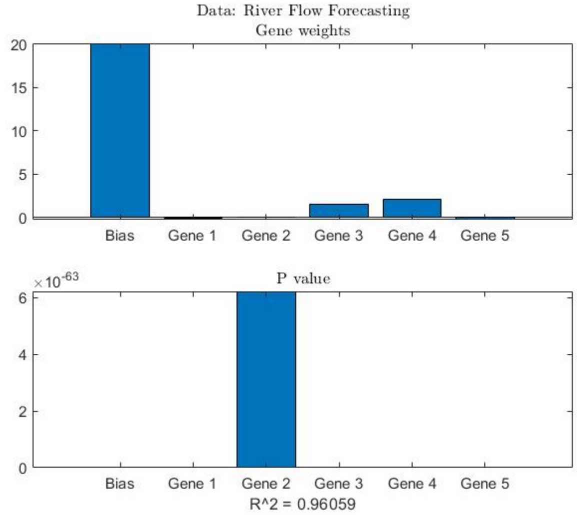

Fig. 6. Symbolic regression genes weights.

TABLE III. Calculated criteria for the GP and PSO Tuned MG-GP models

Technique	Case	VAF	MSE	ED	MD
GP	Training	96.059	0.039838	9.026	0.074631
	Testing	94.296	0.054344	5.2697	0.10413
PSO	Training	96.297	0.03786	8.7017	0.080066
	Testing	93.801	0.054755	5.2324	0.12237

A. Tuning GP Model Parameters Using PSO

In this section, we described the methodology we followed in tuning the parameters of the developed MG-GP model presented in Equation 15. We presented the structure of PSO as a δ value to update the nine parameters of the model. Thus, our particles are presented in Table IV. In Table V, we show the tuning parameters of PSO.

TABLE IV. PSO particles representation

$a_1 + \delta_1$	$a_2 + \delta_2$	$\ldots$	$a_9 + \delta_9$

In Table V, we show the tuning parameters of PSO. The developed MG-GP model parameters were optimized using the Euclidean distance (ED) as a fitness function, as expressed in Equation 13. The convergence of the PSO evolutionary process is demonstrated in Fig. 8.

TABLE V. Tuning Parameters of PSO

Maximum Iteration	150
Population Size	30
Maximum Inertia Weight	0.9
Minimum Inertia Weight	0.4
Acceleration factor c_1	2
Acceleration factor c_2	2

The Scatter plots between the actual and estimated river flow after tuning the MG-GP is depicted for both the training and testing cases in Fig. 9.

Furthermore, Fig. 10 exhibits the actual and predicted Blackwater river flow based on the optimized PSO MG-GP model for both the training and testing cases.

B. Comparison

We calculated the R-squared coefficient as the metric to use to compare the performance of the MG-GP before and after tuning its parameter using PSO. The closer the value of the R-squared coefficient to one, the better the model performs in forecasting the river flow values. The equation for R-squared is given in Equation 17.

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{17}$$

where n is the number of observations, y_i is the actual value of the response variable for observation i, $\bar{y}$ is the mean of the response variable, and $\hat{y}_i$ is the predicted value of the response variable for observation i. Table VI gives the calculated R-squared in training and testing cases.

TABLE VI. R-squared calculated before and after PSO fine tuning

Technique	Training	Testing
MG-GP model	0.96059	0.94296
PSO Tuned MG-GP model	0.98713	0.97759

VII. Conclusions

This study employed a two-phase evolutionary computation technique to forecast the Blackwater river flow. In the first phase, Multigene Symbolic Regression Genetic Programming was utilized to generate a mathematical model capable of predicting future river flow values. The model's parameters were fine-tuned in the second phase using the Particle Swarm Optimization algorithm. The data for our experiments was obtained from the US Geological Survey station 02047500 for the Black Water River near Dendron, Virginia. Various metrics, such as VAF, MSE, ED, and MD, were calculated to assess the techniques' performance. The experimental results confirm that the fine-tuned phase can produce significantly improved outcomes, as evidenced by the increase in the R^2 coefficient value in training and testing cases.

References

[1] W. Sardjono and W. G. Perdana, “The application of artificial neural network for flood systems mitigation at jakarta city,” in *2019 International Conference on Information Management and Technology (ICIMTech)*, vol. 1, 2019, pp. 137–140.

[2] B. Li, M. Wang, Y. Song, L. Li, and J. Zhang, “Coevolutionary particle swarm optimization algorithm for water resources problems and its application,” in *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, vol. 1, 2020, pp. 1231–1236.

[3] X. Lu, Y. Shuaipeng, and H. Hao, “Groundwater simulation of some farm nitrate pollution along the yellow river,” in *2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, 2022, pp. 259–263.

[4] Q. Fan, X. Wen, C. Lin, W. Peng, and Y. Zhang, “Research on influence factors analysis and countermeasures of improving prediction accuracy of run-of-river small hydropower,” in *2017 2nd International Conference on Power and Renewable Energy (ICPRE)*, 2017, pp. 548–552.

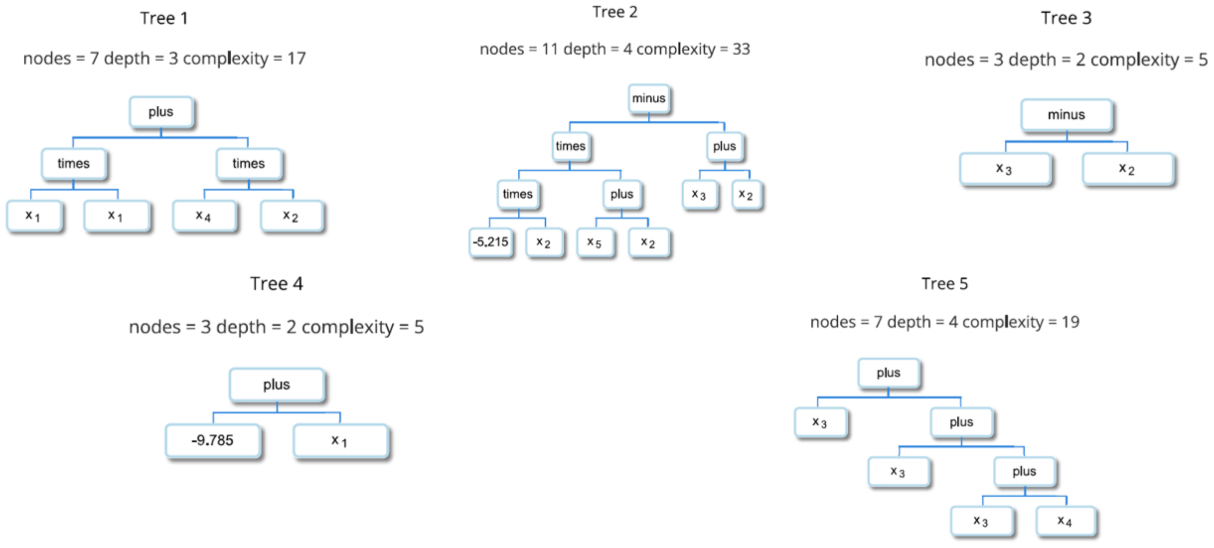
Tree 1
nodes = 7 depth = 3 complexity = 17
plus
times
times
x 1
x 1
x 4
x 2
Tree 2
nodes = 11 depth = 4 complexity = 33
minus
times
plus
times
plus
x 3
x 2
-5.215
x 2
x 5
x 2
Tree 3
nodes = 3 depth = 2 complexity = 5
minus
x 3
x 2
Tree 4
nodes = 3 depth = 2 complexity = 5
plus
-9.785
x 1
Tree 5
nodes = 7 depth = 4 complexity = 19
plus
x 3
plus
x 3
plus
x 3
x 4

Fig. 7. The developed five symbolic GP models.

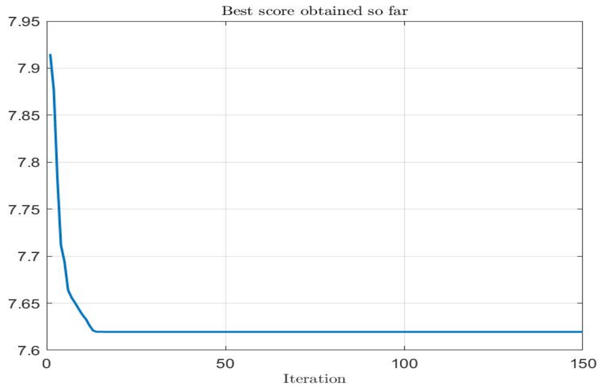

Fig. 8. PSO convergence curve with ED as a fitness function.

[5] E. A. Azrulhisham and M. A. Azri, “Application of lisst instrument for suspended sediment and erosive wear prediction in run-of-river hydropower plants,” in *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018, pp. 886–891.

[6] M. M. Billah, Z. M. Yusof, K. Kadir, A. M. M. Ali, and I. Ahmad, “Quality maintenance of fish farm: Development of real-time water quality monitoring system,” in *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, 2019, pp. 1–4.

[7] H. Apel, Z. Abdykerimova, M. Agalhanova, A. Baimaganbetov, N. Gavrilenko, L. Gerlitz, O. Kalashnikova, K. Unger-Shayesteh, S. Vorogushyn, and A. Gafurov, “Statistical forecast of seasonal discharge in central asia using observational records: development of a generic linear modelling tool for operational water resource management,” *Hydrology and Earth System Sciences*, vol. 22, no. 4, pp. 2225–2254, 2018.

[8] M. A. da Silva Costa and M. S. V. Monteiro, “Statistical modelling of water quality time series – the river vouga basin case study,” in *Research and Practices in Water Quality*, T. S. Lee, Ed. Rijeka: IntechOpen, 2015, ch. 6.

[9] R. Wang and J. Xia, “Comparative study on river flow forecasting methods of river networks,” in *2009 WRI World Congress on Software Engineering*, vol. 1, 2009, pp. 199–203.

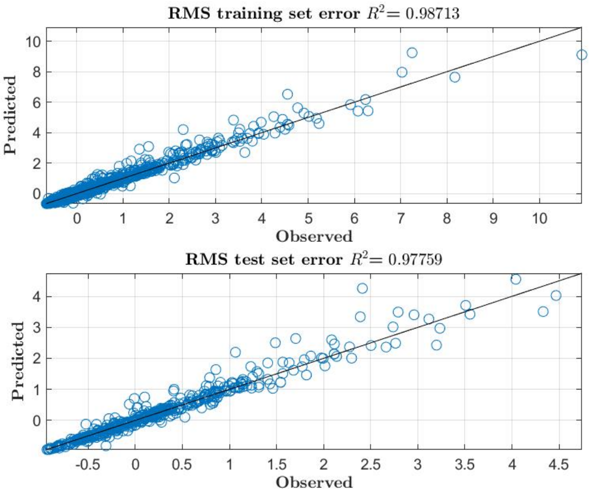

Fig. 9. Scatter plots between actual and predicted flow in both training and testing cases.

[10] T. Egawa, K. Suzuki, Y. Ichikawa, T. Iizaka, T. Matsui, and Y. Shikagawa, “A water flow forecasting for dam using neural networks and regression models,” in *2011 IEEE Power and Energy Society General Meeting*, 2011, pp. 1–6.

[11] P. J. Donnelly and O. Junkins, “Short-term river forecasting with a stacked ensemble of tributary models,” in *2022 7th International Conference on Frontiers of Signal Processing (ICFSP)*, 2022, pp. 189–193.

[12] C. Prakash, A. Barthwal, and D. Acharya, “Floodwall: A real-time flash flood monitoring and forecasting system using iot,” *IEEE Sensors Journal*, vol. 23, no. 1, pp. 787–799, 2023.

[13] Q. Zhang and H. Li, “Evolutionary computation in modeling and optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 1–3, 2015.

[14] V. Buyar and A. A. El-Raouf, “A convolutional neural network-based model for sales prediction,” in *the 2019 International Conference on Artificial Intelligence, Robotics and Control, AIRC 2019*. Association for Computing Machinery (ACM) New York NY United States, 2019,

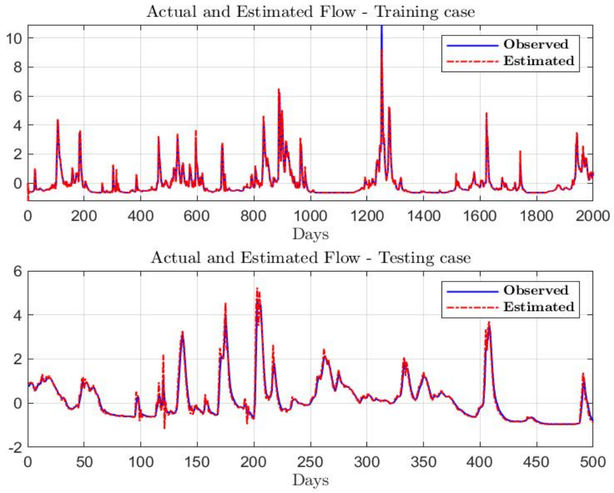

Fig. 10. Observed and computed (PSO-MGGP model) flows for the Blackwater River, training and validation period.

pp. 61–67.

[15] A. k. M. Baareh, A. Sheta, and K. A. Khnaifes, "Forecasting the daily flow of the black water river using soft-computing techniques," *WSEAS Transaction on Information Science and Applications*, vol. 1, no. 4, 2007.

[16] A. M. Baareh, A. Sheta, and K. A. Khnaifes, "Forecasting river flow in the usa: A comparison between auto-regression and neural network non-parametric models," *Journal of Computer Science, USA*, vol. 2, no. 10, 2006.

[17] S. Yin, D. Tang, X. Jin, W. Chen, and N. Pu, "A combined rotated general regression neural network method for river flow forecasting," *Hydrological sciences journal*, vol. 61, no. 4, pp. 669–682, 2016.

[18] A. Sheta and A. Mahmoud, "Forecasting using genetic programming," in *Proceedings of the 33 rd Southern Symposium on System Theory, March 19-20, Athens, Ohio, USA*, 2001, pp. 343–347.

[19] A. Sheta and M. El-Sherif, "Optimal prediction of the nile river flow using neural networks," in *Proceedings of the International Joint Conference on Neural Networks, Washington, D.C., July*, 1999.

[20] S. M. El-Shora, *Neural Networks in Forecasting Models: Nile River Application*. Master thesis, Cairo University, 1997.

[21] Y. Al-Zu'bi, A. Sheta, J. Al-Zu'bi *et al.*, "Nile river flow forecasting based takagi-sugeno fuzzy model." *Journal of Applied Sciences*, vol. 10, no. 4, pp. 284–290, 2010.

[22] A. Sheta and K. De Jong, "Time-series forecasting using GA-tuned radial basis functions," in *Information Science Journal*, 2001, pp. 221–228.

[23] O. Terzi, "A genetic programming approach to river flow modeling," *J. Intell. Fuzzy Syst.*, vol. 27, no. 5, pp. 2211–2219, sep 2014.

[24] K. T. Lee and K.-w. Chau, "Comparison of support vector regression and artificial neural network for river flow forecasting in the southwestern united states," *Journal of Hydrologic Engineering*, vol. 15, no. 9, pp. 729–744, 2010.

[25] Ö. Kisi, "Daily river flow forecasting using artificial neural networks and auto-regressive models," *Turkish Journal of Engineering and Environmental Sciences*, vol. 29, pp. 9–20, 2005.

[26] A. F. Atiya, S. M. El-Shoura, S. I. Shaheen, and M. S. El-Sherif, "A comparison between neural-network forecasting techniques-case study: river flow forecasting," *IEEE Transactions on neural networks*, vol. 10, no. 2, pp. 402–409, 1999.

[27] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, ser. A Bradford book. Bradford, 1992. [Online]. Available: https://books.google.com/books?id=Bhtxo60BV0EC

[28] J. Koza, "Evolving a computer program to generate random numbers using the genetic programming paradigm," in *Proceedings of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann, La Jolla,CA, 1991.

[29] J. R. Koza, *Genetic Programming II Automatic Discovery of Reusable Programs*. MIT Press, 1994.

[30] H. Faris and A. Sheta, "Identification of the tennessee eastman chemical process reactor using genetic programming," *International Journal of Advanced Science and Technology*, vol. 50, pp. 121–140, Jan. 2013.

[31] A. Sheta and H. Faris, "Improving production quality of a hot rolling industrial process via genetic programming model," *International Journal of Computer Applications in Technology*, vol. 49, no. 3/4, 2014, special Issue on: "Computational Optimisation and Engineering Applications".

[32] H. Faris and A. Sheta, "Identification of the tennessee eastman chemical process reactor using genetic programming," *International Journal of Advanced Science and Technology*, vol. 50, pp. 121–140, Jan. 2013.

[33] R. Hiary, A. Sheta, and H. Faris, "Fermentation process modeling using takagi-sugeno fuzzy model," *WSEAS Transaction on Systems*, vol. 11, pp. 375–384, Issue (8), 2012.

[34] H. Faris, A. Sheta, and A. Tobal, "A parallel genetic algorithm for solving time tabling problem," *ICGST International Journal on Artificial Intelligence and Machine Learning (AIML) Journal*, vol. 8, pp. 44–52, Issue (II), 2008.

[35] A. Sheta, H. Faris, and M. Alkasassbeh, "A genetic programming model for S&P 500 stock market prediction," *International Journal of Control and Automation*, vol. 6, no. 5, pp. 303–314, 2013.

[36] A. F. Sheta, H. Faris, and E. Oznergiz, "Improving production quality of a hot rolling industrial process via genetic programming model," *International Journal of Computer Applications in Technology*, vol. 49, no. 3/4, pp. 239–250, 6 Jun. 2014, special Issue on: Computational Optimisation and Engineering Applications.

[37] H. Faris, A. F. Sheta, and E. Oznergiz, "MGP-CC: a hybrid multigene GP-Cuckoo search method for hot rolling manufacture process modelling," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 39–49, 2016.

[38] S. Aljahdali and A. Sheta, "Evolving software effort estimation models using multigene symbolic regression genetic programming," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 12, pp. 52–57, 2013.

[39] A. Al-Afeef, A. Sheta, and A. Rabea, *Image Reconstruction of a Manufacturing Process: A Genetic Programming Approach*, 1st ed. Lambert Academic Publishing, Apr. 2011. [Online]. Available: https://www.morebooks.de/store/gb/book/image-reconstruction-of-a-manufacturing-process/isbn/978-3-8443-2569-0

[40] A. Sheta, R. Hiary, H. Faris, and N. Ghatasheh, "Optimizing thermostable enzymes production using multigene symbolic regression genetic programming," *World Applied Sciences Journal*, vol. 22, no. 4, pp. 485–493, 2013.

[41] H. Faris, A. Sheta, and R. Hiary, "On symbolic regression for optimizing thermostable lipase production," *International Journal of Advanced Science and Technology*, vol. 63, no. 11, pp. 23–33, 2014, special Issue on: Computational Optimisation and Engineering Applications. [Online]. Available: http://www.sersc.org/journals/IJAST/vol63/3.pdf

[42] J. Kennedy, "The behavior of particles," *Evolutionary Programming VII*, pp. 581–587, 1998.

[43] M. Yadav, B. Fathi, and A. Sheta, "Selection of wsns inter-cluster boundary nodes using pso algorithm," *J. Comput. Sci. Coll.*, vol. 34, no. 5, p. 47–53, apr 2019.

[44] A. Sheta, M. Braik, D. R. Maddi, A. Mahdy, S. Aljahdali, and H. Turabieh, "Optimization of pid controller to stabilize quadcopter movements using meta-heuristic search algorithms," *Applied Sciences*, vol. 11, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/14/6492

[45] D. P. Searson, D. E. Leahy, and M. J. Willis, "GPTIPS : An open source genetic programming toolbox for multigene symbolic regression," in *Proceedings of the International Multi-conference of Engineers and Computer Scientists 2010 (IMECS 2010)*, vol. 1, Hong Kong, 17-19 Mar. 2010, pp. 77–80.