

Article

A Hybrid Rank-Preserving and Evolutionary Algorithm for Multisite Daily Streamflow Simulation

Stefan Pitulić ^{1,*} , Dragana Radosavljević ¹ , Đurica Marković ²  and Siniša Ilić ¹ 

¹ Department of Computer Science and Informatics, Faculty of Technical Sciences, University of Priština in Kosovska Mitrovica, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia; dragana.radosavljevic@pr.ac.rs (D.R.); sinisa.ilic@pr.ac.rs (S.I.)

² Department of Civil Engineering, Faculty of Technical Sciences, University of Priština in Kosovska Mitrovica, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia; djurica.markovic@pr.ac.rs

* Correspondence: stefan.pitulic@pr.ac.rs; Tel.: +381-649001008

Abstract

This paper presents a hybrid algorithmic framework for nonparametric multisite daily streamflow simulation, evaluated on 52 years of observed data from three hydrological stations. The method generates streamflow data of 1000 synthetic years while preserving marginal distributions, daily rank structures, inter-station consistency, annual-sum behavior, and dependence across consecutive years. The workflow integrates Monte Carlo sampling, Schaaake-shuffle reordering, block-mosaic reconstruction with partial freezing, Hungarian assignment optimization, annual-sum matching, and an adaptive permutation genetic algorithm for year-order optimization. The results show that the proposed algorithm improves aggregate hydrological diagnostics, particularly annual-sum autocorrelation, hydrological indices, persistence, seasonality, and timing of extremes, while reducing runtime in the final optimization phase by 45.2% compared to the benchmark algorithm. The study therefore formulates daily streamflow simulation as a constrained time-series reconstruction and permutation-optimization problem, making the method suitable for further algorithmic development and other multisite environmental time-series applications.

Keywords: daily streamflow; genetic algorithm; Hungarian algorithm; multisite time series; nonparametric simulation; permutation optimization; Schaaake shuffle

1. Introduction

Long synthetic multisite time series are important for reliability analysis, planning, and stress testing of systems whose behavior cannot be fully evaluated from short observed records alone [1–5]. In daily streamflow simulation, the main algorithmic challenge is not only to reproduce marginal distributions, but also to preserve seasonal behavior, autocorrelation, cross-site dependence, local continuity, and transitions between consecutive years. These requirements make the generation problem a constrained-reconstruction and ordering task rather than a simple random-sampling problem.

From an algorithmic standpoint, this task can be formulated as a constrained stochastic time-series reconstruction problem. After values are generated from empirical or nonparametric distributions, they are arranged in a synthetic data matrix $G \in \mathbb{R}^{n_{\text{gen}} \times M}$, where n_{gen} is the number of generated synthetic years and $M = K \times T$ is the number of station–day variables, with K denoting the number of stations and T the number of time steps per year. Rows therefore represent generated synthetic years, while columns represent station–day



Academic Editor: José Manuel García-Nieto

Received: 20 May 2026

Revised: 26 June 2026

Accepted: 30 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

variables. The generated matrix must then be rearranged so that it simultaneously respects rank structure, inter-series coherence, annual-scale behavior, and local temporal continuity. The problem therefore belongs to the broader class of sequence-reordering, constrained-reconstruction, and permutation-optimization problems.

Nonparametric approaches are suitable for this setting because they avoid strong assumptions about distributional form and infer the relevant marginal and dependence structures directly from historical observations. Kernel methods, bootstrap-based approaches, k-nearest-neighbor resampling, and reordering procedures have therefore become important tools in stochastic hydrological simulation and related environmental time-series applications [5–9].

Streamflow simulation can be approached through process-based, data-driven, and stochastic-synthesis paradigms. Process-based and conceptual rainfall–runoff models represent water-balance, storage, routing, and transformation processes and generally require meteorological forcing, catchment descriptors, calibration data, and assumptions regarding model structure. Data-driven and AI-based models learn relationships between predictors—such as precipitation, temperature, antecedent discharge, and catchment attributes—and concurrent or future streamflow. Recent studies have demonstrated strong predictive performance, including in large-sample and ungauged-basin applications, but have also shown that performance depends on the coverage of the training data and that extrapolation beyond the training regime remains challenging [10–16].

These paradigms address a different primary objective from that of the present study. This distinction is also consistent with recent comparative analyses indicating that process-based and data-driven runoff models respond differently to data availability, predictor structure, and transfer conditions, whereas stochastic synthesis targets the statistical reproduction of observed records rather than conditional prediction [17]. Process-based and supervised AI models are mainly used to simulate or predict discharge conditional on meteorological and catchment inputs. The present work instead addresses unconditional stochastic synthesis: it generates many plausible multisite daily streamflow realizations from observed streamflow records alone. Recent stochastic-synthesis studies likewise emphasize multisite coherence and spatiotemporal dependence preservation as core requirements when constructing long synthetic streamflow ensembles [18,19]. Therefore, the proposed method is not presented as a universal alternative to process-based or AI-based models. Its specific advantage concerns applications in which the required output is a long ensemble of synthetic scenarios that explicitly preserves several empirical distributional and dependence properties simultaneously.

In practical terms, the generated values for each station and calendar day should reproduce the historical distribution across years; low, medium, and high flows should be arranged into plausible daily hydrographs; and flows at three analyzed streamflow-gauging stations should remain mutually coherent because the stations are located along the same river system. The generated series should also reproduce relatively dry and wet years, annual-sum persistence, and dependence at transitions between consecutive years. Standard supervised forecasting models do not guarantee the simultaneous preservation of all these properties unless they are explicitly incorporated into the model architecture, loss function, or post-processing procedure.

These requirements motivate the modular structure of the proposed workflow. Monte Carlo sampling controls the empirical value distribution for each station–day variable. The Schaake shuffle transfers historically observed rank templates without changing the sampled value set. Block-mosaic reconstruction with partial freezing introduces local memory between adjacent daily blocks. The Hungarian algorithm solves the resulting one-to-one reassignment problem under a cost function combining continuity and correlation-profile

similarity. Annual-sum matching corrects selected low-ranked annual scenarios through synchronous multisite swaps. Finally, the adaptive permutation genetic algorithm orders complete synthetic years so that annual-sum autocorrelation and cross-year-boundary dependence are reproduced as closely as possible.

The methodology introduced by Ilich & Despotovic [20] and further developed by Marković et al. [21,22] is used here only as a benchmark for controlled quantitative comparison. This benchmark was selected because its diagnostic indicators are comparable with those used in the present study and because it provides a reproducible nonparametric multisite generation framework. The present study addresses a daily-resolution case directly: each synthetic year contains 365 daily values per station, and the three-station configuration produces 1095 station–day variables per synthetic year. This high-dimensional daily structure substantially increases the size and complexity of the reordering problem and motivates the formulation of the proposed algorithm as a constrained-reconstruction and permutation-optimization workflow.

Several established algorithmic components are relevant to this reconstruction problem. The Schaake shuffle reconstructs space–time rank dependence by transferring historical rank patterns to generated values without changing their marginal distributions [23,24]. Assignment optimization, including the Hungarian algorithm, provides a formal mechanism for selecting locally coherent rearrangements under a cost function [25,26]. Genetic algorithms are suitable for the final year-ordering problem because a candidate solution can be naturally represented as a permutation of synthetic years [27–30].

Recent algorithmic studies also support this computational framing. Evolutionary and stochastic optimization methods are widely used for complex search problems [31], genetic algorithms have been applied to sequence ordering and periodicity recognition [32], and hybrid nonparametric–evolutionary procedures have been proposed for sampling and reconstruction tasks [33]. In hydrological and environmental applications, related ideas have also appeared in nonparametric streamflow disaggregation and representative time-series construction [34,35]. These works are not direct hydrological analogs of the present study, but they demonstrate the relevance of evolutionary search, sequence ordering, and hybrid nonparametric–evolutionary design in algorithmic research.

The aim of this paper is to present a modular hybrid algorithm for daily multisite streamflow simulation. The proposed workflow consists of nonparametric daily-value generation, rank-preserving reordering, block-wise reconstruction with partial freezing, assignment-based local optimization, annual-sum matching, and adaptive permutation-based year-order optimization.

The contribution of the paper is threefold. First, multisite daily streamflow simulation is formulated as a constrained time-series reconstruction and permutation-optimization problem. Second, the proposed algorithm integrates Schaake-shuffle reordering, block-mosaic reconstruction, Hungarian assignment, and annual-sum matching into a unified daily-resolution workflow. Third, the final year-ordering stage is solved using an adaptive permutation-based genetic algorithm, which reduces the runtime of the most demanding optimization phase while improving several aggregate diagnostic indicators. In this context, the proposed method is evaluated not only as a tool for hydrological simulation, but also as an algorithmic framework for constrained reconstruction and permutation-based optimization of multisite environmental time series.

Algorithmic Novelty and Contribution

The novelty of the proposed workflow does not lie in the isolated use of Schaake shuffle, Hungarian assignment, or a genetic algorithm, since these are established methods. The contribution lies in the way these components are combined into a single constrained-

reconstruction workflow for daily multisite time series. In the proposed formulation, each synthetic year is treated as a row object and each station–day combination as a variable, so the task becomes a structured reordering problem over a high-dimensional matrix rather than a conventional univariate simulation problem. This formulation is the basis for the algorithmic contribution of the paper.

The first algorithmic contribution is the use of rank-preserving reconstruction as an intermediate dependence-restoration layer. The Schaake-shuffle step transfers the observed daily rank structure to independently generated values while preserving the generated marginal distributions. This creates a dependence-aware starting point for later local optimization and avoids relying only on a target correlation matrix.

The second contribution is the block-mosaic procedure with a partial-freeze mechanism. Instead of applying one global reordering to the entire synthetic matrix, the daily sequence is processed through variable-length blocks. A fixed proportion of row identities is carried from the previous block, while the remaining moving positions are reassigned. This creates local memory between adjacent blocks and directly targets unrealistic day-to-day switching in the generated hydrograph.

The third contribution is the formulation of local rearrangement as an assignment-optimization problem. For each block, a cost matrix combines a continuity term and a correlation-profile term. The Hungarian algorithm is then used to select a feasible one-to-one assignment of moving row positions. In this way, the local reconstruction step is transformed from an informal heuristic into an explicit optimization subproblem.

The fourth contribution is the annual-sum matching step based on synchronous swaps across stations. This module adjusts selected low-ranked synthetic years while preserving cross-station coherence, because every accepted swap is applied to the same day at all stations. This step links daily reordering with annual-scale distributional behavior and is particularly important for low-flow scenarios.

The fifth contribution is the use of an adaptive permutation-based genetic algorithm for final year ordering. Each chromosome represents a complete ordering of synthetic years, and the fitness function jointly evaluates annual-sum autocorrelation and year-boundary dependence. This reformulation places the final stage within the class of permutation-optimization problems and improves scalability when the number of generated years is large.

Taken together, these components define a modular algorithmic workflow that can be adapted to other multisite environmental time series where values must be generated first and then rearranged under multiple statistical and structural constraints.

2. Materials and Methods

2.1. Data, Notation, and General Framework

The data analyzed in this study come from three hydrological stations located in Serbia—Devići, Mlanča, and Ušće—which are denoted in Section 3 as S1, S2, and S3, respectively, for brevity. The observed dataset consists of $n_{\text{obs}} = 52$ years of daily streamflow at $K = 3$ hydrological stations, with each year containing $T = 365$ daily values per station. Based on this sample, $n_{\text{gen}} = 1000$ synthetic years are generated, so that the total number of daily variables per year is $M = K \times T = 1095$.

In this matrix-based formulation, a marginal station–day distribution means the empirical distribution formed by one column of values; for example, all historical values recorded on the same calendar day at the same station across the observed years. A rank pattern describes only the relative order of values, i.e., which years are low, medium, or high for a given station–day variable, rather than their exact magnitudes. Preserving these properties means that the synthetic matrix should contain realistic values for each day

and station and should arrange them in a way that resembles observed dry, normal, and wet periods.

The intended application of the proposed algorithm is the generation of long synthetic multisite daily streamflow sequences when only observed streamflow records are available and when preservation of empirical distributional and dependence properties is required. The required input is an observed daily streamflow matrix containing complete annual records for all stations. The output is a synthetic daily streamflow matrix with a user-defined number of generated years, preserving the main marginal, temporal, interstation, annual-scale, and interannual characteristics of the observed record. The algorithm is most suitable for applications such as stochastic scenario generation, reliability analysis, stress testing, and water-resources planning, where long synthetic streamflow records are needed. Still, the objective is not event forecasting from meteorological predictors. The method is most appropriate when the observed record is sufficiently long to estimate seasonal ranks, dependence patterns, and annual-sum behavior, and when the generated series are intended to reproduce the statistical structure of the historical period.

The method assumes that the historical record adequately represents the hydroclimatic regime to be reproduced. It is therefore not intended to extrapolate directly to conditions outside the empirical support of the observed record unless additional nonstationary information or modified input scenarios are introduced. No universal minimum record length is prescribed, but shorter records provide fewer independent values for estimating station–day distributions, rank templates, annual-sum behavior, and interannual dependence.

The observed dataset is arranged in a matrix O of dimension $n_{\text{obs}} \times M$, whereas the synthetic dataset after the first step is denoted by G and has dimension $n_{\text{gen}} \times M$. The columns are grouped by station, so that the block of columns associated with a station contains days 1 to 365 for that station.

$$O = [o_{ij}], i = 1, \dots, n_{\text{obs}}, j = 1, \dots, M, M = K \times T = 3 \times 365 = 1095 \quad (1)$$

Linear dependence between any two variables is measured by the Pearson correlation coefficient, which is used in the dependence-reconstruction and evaluation steps.

$$\rho_{ij} = \frac{\sum_t (o_{i,t} - \bar{o}_i)(o_{j,t} - \bar{o}_j)}{\sqrt{\sum_t (o_{i,t} - \bar{o}_i)^2 \times \sum_t (o_{j,t} - \bar{o}_j)^2}} \quad (2)$$

In the present study, the notation is written for a general number of time steps T while $T = 365$ days is used in the numerical implementation. In the final year-ordering step, a four-day segment at the end and at the beginning of adjacent years is considered, and 13 lags are used for annual-sum autocorrelations.

Tables 1 and 2 provide a compact summary of the workflow, but the following subsections describe the same modules at the data-operation level. In brief, Step 1 controls the distribution of values for each station and calendar day; Step 2a controls the rank ordering of low and high values; Step 2b controls day-to-day continuity through block memory and assignment; Step 2c controls annual water volume, especially for low-flow years; and Step 3 controls the order of complete synthetic years.

Table 1. A conceptual comparison of the benchmark and proposed algorithms.

Aspect	Benchmark Algorithm	Proposed Algorithm	Expected Effect
Step 1	Monte Carlo generation from nonparametric distributions	Nonparametric generation from empirical distributions	Preservation of marginal distributions
Step 2	Iman–Conover reordering for target correlations	2.a. Schaake shuffle + 2.b. block mosaic + 2.c. annual sums	Finer control of daily structure and annual sums
Step 3	Permutation of years using row-swap search and calculation of function D	Adaptive permutation using genetic algorithm	Reduced computation time and improved scalability
Daily continuity	Not explicitly optimized through local blocks	Explicitly introduced through blocks and partial-freeze	More natural transitions between adjacent days
Optimization strategy	Heuristic examination of row swaps	Metaheuristic search over a population of permutations	Faster identification of good solutions

Table 2. The algorithmic modules and expected diagnostic contribution of the proposed workflow.

Module	Algorithmic Role	Key Parameters	Expected Diagnostic Contribution
Step 1: Nonparametric Monte Carlo generation	Samples independent daily values from empirical/nonparametric distributions	station, day, n_{gen}	preserves marginal distributions
Step 2a: Schaake-shuffle reordering	Transfers observed rank structure to generated values	bootstrap analog years	improves rank-structure preservation
Step 2b: Block mosaic + partial freeze + Hungarian assignment	Reorders daily blocks with local memory and assignment optimization	$B_{\min}$, $B_{\max}$, keepFrac , w_{corr}	improves local continuity and inter-series coherence
Step 2c: Annual-sum matching	Adjusts low-ranked annual-sum scenarios through synchronous swaps	$GMIN$, minK	improves annual-sum and low-flow year representation
Step 3: Adaptive permutation GA	Optimizes final ordering of synthetic years	w_1 , w_2 , maxLag , W , population, mutation, crossover	improves interannual ordering and scalability

Each module was selected in a way to modify a different part of the synthetic matrix while preserving properties established by the preceding modules. Step 1 samples values independently for each station and calendar day, thereby establishing realistic station–day value distributions but not determining which values should be combined to belong to the same synthetic year. Step 2a then performs a rank-based permutation: it changes positions of the sorted values in synthetic years, while preserving the sampled values and their marginal distributions. Step 2b operates on consecutive daily blocks. Partial freezing carries selected synthetic-year identities from one block to the next, whereas the Hungarian algorithm assigns the remaining identities in one-to-one mapping so that local discontinuities and discrepancies in correlation profiles are jointly reduced.

Step 2c targets a property not explicitly controlled by the preceding daily reordering steps—the lower part of the annual-sum distribution—and applies every accepted daily swap synchronously at all stations in order to retain multisite coherence. Finally, Step 3 changes only the order of already set blocks of data values that belong to synthetic years. Consequently, it can modify annual-sum autocorrelation and year-boundary de-

pendence without changing the daily values within any synthetic year. The modules are therefore complementary rather than interchangeable, and each one addresses a structural property that is not fully controlled by the preceding operations.

2.2. Proposed Algorithm

The proposed algorithm is organized into three main steps. Step 1 generates daily synthetic values from nonparametric distributions for each station–day variable. Step 2 reconstructs the internal organization of the generated daily matrix through three sub-steps: rank-preserving Schaake-shuffle reordering, block-mosaic reconstruction with partial freezing and Hungarian assignment, and annual-sum matching by synchronous swaps. Step 3 determines the final ordering of synthetic years using an adaptive permutation-based genetic algorithm. The complete workflow, therefore, separates value generation, daily-structure reconstruction, annual-scale adjustment, and year-order optimization.

Each component of the workflow was selected to address a different structural requirement of multisite daily streamflow simulation. The nonparametric generation step preserves empirical marginal behavior without imposing a fixed parametric distribution. The Schaake-shuffle step is used because it transfers historical rank structures to generated values and therefore provides an initial dependence-aware ordering. The block-mosaic partial-freeze step is introduced to reduce unrealistic local switching between consecutive days and to preserve daily-scale continuity. The Hungarian assignment is used because the reassignment of moving rows within each block can be formulated as a one-to-one assignment problem under a cost function combining continuity and correlation-profile terms. Annual-sum matching is included to correct annual-scale behavior, particularly for low-flow years, while preserving spatial coherence through synchronous swaps across stations. Finally, the adaptive permutation genetic algorithm is used because the final year-ordering task is naturally represented as a permutation-optimization problem, and exhaustive search is infeasible for 1000 generated years. The complete structure of the proposed hybrid workflow is summarized in Figure 1.

2.2.1. Step 1: Nonparametric Daily-Value Generation

Step 1 generates the initial synthetic daily streamflow matrix G . For each hydrological station and each day of the year, synthetic values are sampled from the empirical/nonparametric distribution inferred from the observed record. This stage creates n_{gen} synthetic years and preserves the marginal behavior of daily streamflow at each station–day position. At this point, the generated values are not yet arranged so as to reproduce realistic temporal continuity, cross-station dependence, annual-sum behavior, or interannual ordering; these structural properties are addressed in the subsequent steps.

In simple terms, a station–day variable is one column of the data matrix, such as the values observed on day 1 at station S1 over all historical years. Step 1 samples new values from the empirical distribution of that column. Therefore, the generated values for a given day and station have the same type of distributional support as the observed values for that day and station. However, this operation alone does not group generated values by year, nor does it determine their chronological order across time.

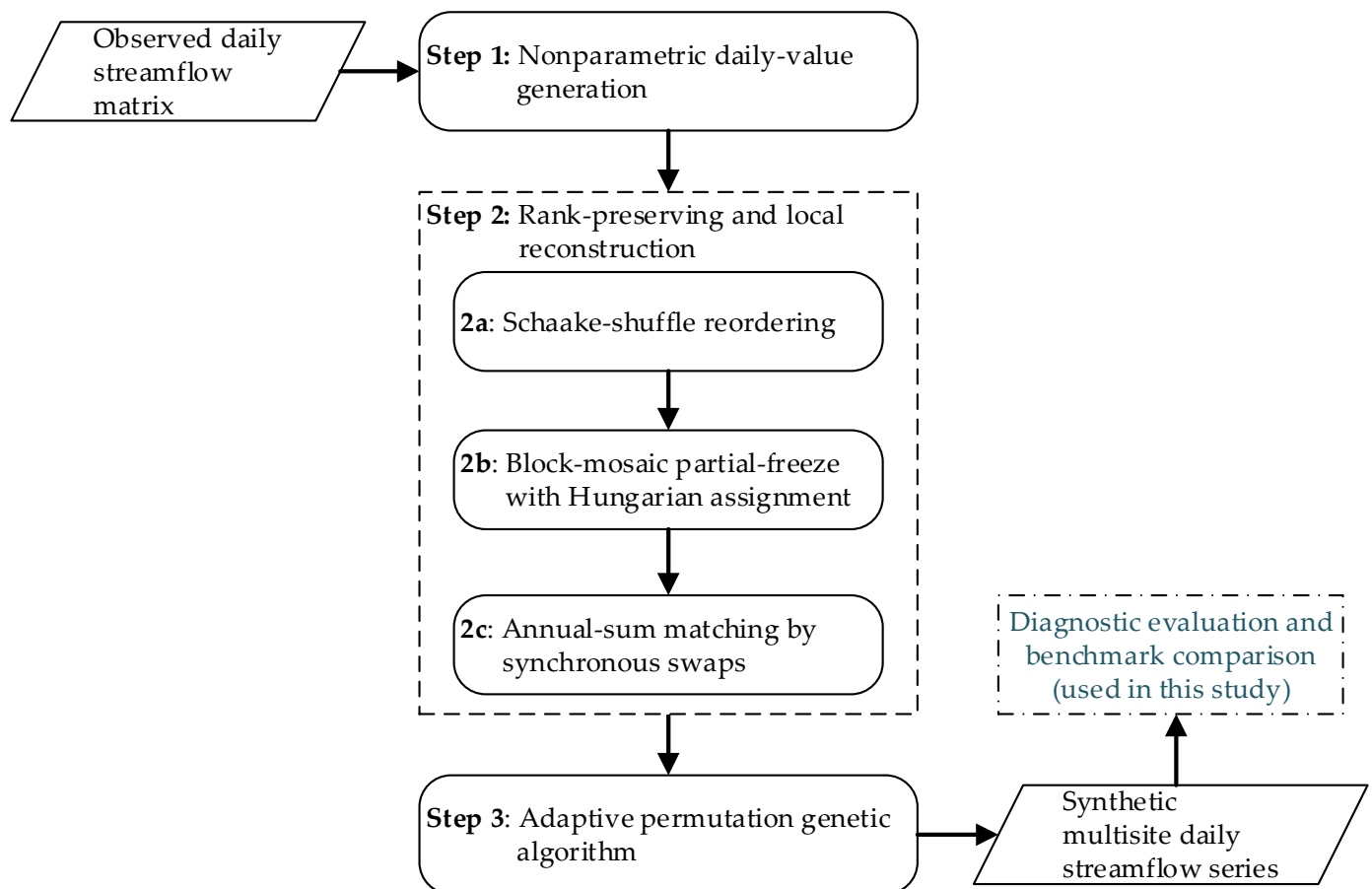


Figure 1. A flowchart of the proposed hybrid algorithmic workflow for multisite daily streamflow simulation. The dashed box denotes the Step 2 reconstruction module, which includes Schaake-shuffle reordering, block-mosaic partial-freeze reconstruction with Hungarian assignment, and annual-sum matching. The dash-dotted box indicates the diagnostic evaluation and benchmark comparison used solely for performance assessment in this study.

2.2.2. Step 2a: Schaake-Shuffle Reordering

Step 2a uses the Schaake shuffle as a rank-based procedure for reconstructing a realistic arrangement of high and low values. The procedure transfers the rank pattern from the historical record to the generated values without changing the generated value set itself [23,24]. In this way, the marginal distributions produced in Step 1 are preserved, while the generated values are rearranged according to historically observed rank structures.

Because the number of synthetic years ($n_{\text{gen}} = 1000$) is much larger than the number of observed years ($n_{\text{obs}} = 52$), the historical analog pattern is extended by bootstrap sampling with replacement. This produces a longer analog series H whose ranks serve as the template for reordering the sorted generated values [23,24].

$$i_m \sim \text{Unif}\{1, \dots, n_{\text{obs}}\}, H_m = O_{i_m}, m = 1, \dots, n_{\text{gen}} \quad (3)$$

$$r_m = \text{Rank}(H_m), C_m = G_{(r_m)}, m = 1, \dots, n_{\text{gen}} \quad (4)$$

Here n_{obs} denotes the number of observed years, n_{gen} is the number of generated synthetic years, and m is the index of the synthetic realization. The variable i_m is a bootstrap-selected index drawn uniformly from the set of observed years, so that O_{i_m} denotes the observed value selected from the historical record at draw m . Accordingly, H_m is the m -th element of the extended analog series H . In Equation (4), r_m denotes the rank of H_m within the analog series H , while $G_{(r_m)}$ denotes the r_m -th order statistic of the generated series G ,

i.e., the value occupying position r_m after sorting G in ascending order. Finally, C_m is the reordered synthetic value assigned to position m according to the historical rank template.

This procedure preserves the marginal distribution of the synthetic series G while modifying its order to follow the historical rank structure. A further advantage of the Schaaake-shuffle-based formulation adopted here is that it helps to preserve realistic daily hydrograph patterns, including during low-flow periods, because the rank structure is transferred directly from the historical record at the daily scale. In this sense, Step 2a provides a dependence-aware starting point for the subsequent local reconstruction stage, while preserving the generated marginal distributions. When ties occur, random tie-breaking is used to avoid rank degeneracy and preserve methodological flexibility [23]. The following pseudocode describes the one-dimensional Schaaake-shuffle procedure applied to a single station–day group. Repeating this procedure over all station–day groups yields the reordered output matrix of Step 2a.

Pseudocode of Step 2a:

Algorithm 1 One-dimensional Schaaake shuffle for a station–day group

Input: observed station–day vector $O \in \mathbb{R}^{n_{obs}}$ with $n_{obs} = 52$,

generated station–day vector $G \in \mathbb{R}^{n_{gen}}$ with $n_{gen} = 1000$

```

1  for  $m = 1 \dots n_{gen}$ :
2      draw analog index  $i_m$  from  $\{1, \dots, n_{obs}\}$  with replacement
3      set  $H_m = O[i_m]$ 
4  end for
5  compute rank positions  $r_m$  of the analog series  $H$ , with random tie-breaking if
   needed
6  sort  $G$  increasingly to obtain  $G_{sorted}$ 
7  for  $m = 1 \dots n_{gen}$  :
8      set  $C_m = G_{sorted}[r_m]$ 
9  end for
10 return reordered synthetic vector  $C$ 
```

Explanation of Algorithm 1. Lines 1–4 create an extended analog vector H with n_{gen} positions. For each generated position m , the algorithm draws one integer index i_m uniformly from the range $1, \dots, n_{obs}$. The expression “with replacement” means that the same observed year may be selected more than once, which is necessary because n_{gen} is larger than n_{obs} . The value $H_m = O[i_m]$ is therefore the observed value from the selected year for the same station–day variable; it is not a newly generated discharge value. Line 5 computes the rank order of the analog vector H . Lines 6–9 sort the generated values G and then place them into the generated years according to the rank order of H . In this way, the set of generated values remains unchanged, but their order across synthetic years follows a rank pattern sampled from the observed record.

Applying Algorithm 1 to all station–day groups (vectors C) produces the reordered matrix G^{in} , which serves as the input to Step 2b.

2.2.3. Step 2b: Block-Mosaic Reordering with Partial Freeze and Hungarian Assignment

Step 2b introduces block-wise permutation by day and is the main mechanism for preserving local daily continuity. The input matrix G^{in} , obtained as the output of Step 2a, is divided into successive blocks of variable length, and within each block a single row permutation is defined and applied identically to all stations. In this way, spatial coherence among stations is preserved at the block level.

The key idea is partial freeze: some positions retain the same year identifiers as in the previous block, whereas the remaining positions are optimized. This creates controlled memory between successive blocks, reduces discontinuities, and promotes more natural day-to-day transitions.

To optimize the moving positions, a cost matrix is constructed in which the first component measures differences in mean values across stations at the transition from the previous day to the current day, while the second component measures differences in the correlation profile relative to the “frozen” rows. The relative importance of the correlation component is controlled by the weight w_{corr} , whereas the continuity term retains unit weight. Accordingly, the total assignment cost is defined as $\text{Cost} = d\text{Val} + w_{\text{corr}} \times d\text{Corr}$, where $d\text{Val}$ denotes the continuity term and $d\text{Corr}$ denotes the correlation-profile term. In the adopted implementation, $w_{\text{corr}} = 0.8$, which gives slightly greater emphasis to continuity while still strongly penalizing deviations in the correlation profile.

The resulting assignment problem is then solved using the Hungarian (Munkres) algorithm [25,26].

Beyond the original Hungarian and Munkres formulations [25,26], the use of a rectangular two-dimensional assignment formulation is consistent with contemporary implementations of assignment algorithms [36]. In the present workflow, this formulation is used at a practical data level: movable positions in the current block are matched to candidate synthetic-year rows under a cost function that penalizes discontinuity and mismatch in correlation profiles.

$$\text{rowCorr}_{ij} = \text{corr}(G_i^{\text{in}}, G_j^{\text{in}}) \quad (5)$$

$$\text{Cost}(i, j) = |v_{\text{Prev}}(i) - v_{\text{New}}(j)| + w_{\text{corr}} \times \frac{1}{|F|} \sum_{f \in F} |\rho(\text{idPrev}, f) - \rho(\text{idNew}, f)| \quad (6)$$

Here, $v_{\text{Prev}}(i)$ denotes the streamflow value averaged over all stations for the series assigned to position i on the previous day, whereas $v_{\text{New}}(j)$ denotes the analogous mean value for candidate series j on the current day. Furthermore, F denotes the set of frozen rows, $\rho(\text{idPrev}, f)$ and $\rho(\text{idNew}, f)$ denote an element of the rowCorr matrix, i.e., the Pearson correlation coefficient between the corresponding rows. Because $w_{\text{corr}} > 0$ in the adopted implementation, the cost function simultaneously accounts for the continuity of mean values across adjacent blocks and the similarity of the correlation profile relative to the frozen rows [25, 26].

The advantage of this sub-step lies in its explicit block structure containing lengths in the range from B_{min} to B_{max} . By processing the daily sequence through variable-length blocks and retaining part of the previous ordering controlled by the parameter *keepFrac*, the algorithm introduces local memory between adjacent blocks. This reduces unrealistic day-to-day switching in the generated hydrograph and provides direct control over local continuity while maintaining cross-station coherence.

Pseudocode of Step 2b:

Algorithm 2 Block-mosaic partial-freeze reordering with Hungarian assignment

Input: reordered synthetic matrix $G^{\text{in}} \in \mathbb{R}^{n_{\text{gen}} \times M}$ from Step 2a, block-length limits $B_{\text{min}} = 7$ and $B_{\text{max}} = 120$, partial-freeze fraction *keepFrac* = 0.3, and correlation-profile weight $w_{\text{corr}} = 0.8$

- 1 initialize $\text{perm}_{\text{prev}} = [1, 2, \dots, n_{\text{gen}}]$, $\text{day} = 1$
 - 2 while $\text{day} \leq 365$:
 - 3 choose random block length L in $[B_{\text{min}}, B_{\text{max}}]$
 - 4 set $L = \min(L, 365 - \text{day} + 1)$
-

```

5   choose  $keep_{idx}$  of size  $round(keepFrac \times n_{gen})$ 
6   copy frozen IDs from  $perm_{prev}$  into  $perm_{new}$  at  $keep_{idx}$ 
7   randomly fill remaining moving positions with remaining IDs
8   if  $day > 1$  and moving positions exist:
9       compute the correlation-profile term  $dCorr$  for all position-candidate pairs
10      compute the continuity term  $dVal$  for all position-candidate pairs
11      form the cost matrix as  $Cost = dVal + w_{corr} \times dCorr$ 
12      solve assignment with Hungarian algorithm
13      update  $perm_{new}$  on moving positions
14  end if
15  apply  $perm_{new}$  to all stations over days  $[day, day + L - 1]$ 
16  set  $perm_{prev} = perm_{new}$ 
17   $day = day + L$ 
18 end while
19 return  $G^{out}$ 

```

Explanation of Algorithm 2. Lines 1–4 define the current block of consecutive days. Lines 5–7 keep a selected fraction of row identities from the previous block, meaning that some synthetic years continue into the next block without being reassigned. This partial-freeze mechanism creates local memory between blocks. Lines 8–14 are applied only when a previous block exists and at least one position remains movable. For every possible assignment of a movable position to a candidate row, the algorithm computes a continuity term and a correlation-profile term, combines them into a cost matrix, and solves the resulting one-to-one assignment problem with the Hungarian algorithm. Lines 15–17 apply the selected permutation to all stations over the current block, and then the algorithm continues to the next block. The output is the block-mosaic reordered synthetic matrix $G^{out} \in \mathbb{R}^{n_{gen} \times M}$.

2.2.4. Step 2c: Annual-Sum Matching by Synchronous Swaps

Step 2c targets the distribution of annual sums' values and links daily reordering with annual-scale behavior. Starting from the reference station $r = 1$, annual sums are computed for each year, after which the synthetic and observed years are ranked in ascending order. The *GMIN* synthetic years with the lowest annual sums are then processed sequentially and matched to the correspondingly ranked observed years.

This correction is motivated by the need to better represent annual scenarios, especially low-flow years, which are important in stochastic streamflow synthesis and water-management applications [21,22,37].

$$S_G(y) = \sum_{d=1}^{365} g_{y,r,d}, S_O(y) = \sum_{d=1}^{365} o_{y,r,d} \quad (7)$$

In Equation (7), $S_G(y)$ and $S_O(y)$ denote the annual sums of the synthetic and observed series, respectively, for year y at the reference station r . The variable $g_{y,r,d}$ represents the daily synthetic streamflow value from the Step 2b output matrix G^{out} , while $o_{y,r,d}$ represents the daily observed streamflow value from the observed matrix O ; both are expressed in year–station–day form for year y , station r , and day d , with $d = 1, \dots, 365$. Thus, the annual sum is defined as the total of all daily values within a given year.

After ranking, the *GMIN* synthetic years with the lowest annual sums are iteratively adjusted toward the correspondingly ranked observed years, where *GMIN* denotes the number of low-ranked synthetic years selected for correction. In the adopted implementation, this number is defined as $GMIN = round\left(2 \frac{n_{gen}}{n_{obs}}\right)$ where n_{gen} is the number of

synthetic years and n_{obs} is the number of observed years. Here, y_g denotes the selected low-ranked synthetic year, y_o denotes the correspondingly ranked observed year, and y_f denotes the donor synthetic year used for the synchronous exchange.

For each selected synthetic year, the days are first ordered in descending order according to their contribution to the annual sum. Candidate synchronous swaps are then examined sequentially, starting from the largest contributors. At each step, the currently selected day of the low-ranked synthetic year is exchanged with the same day from an admissible donor year y_f , where $y_f \neq y_g$, chosen as the synthetic year having the minimum value at that day among the admissible donor years. In this way, the annual total of the corrected year is gradually reduced.

A key feature of this step is that any exchange applied to a selected day is performed synchronously across all stations. In other words, if a given day is exchanged for the reference station, the values corresponding to that same day are simultaneously exchanged for all other stations as well. This preserves the spatial coherence of the synthetic scenarios, even though the ranking and correction criterion is defined only at the reference station. The correction process is regulated by an additional tolerance factor $tol(gg)$, where gg denotes the rank index of the low-ranked synthetic year being adjusted. For the gg -th selected synthetic year, synchronous swaps continue while $tol(gg)S_G(y_g) \geq S_O(y_o)$. Thus, $tol(gg)$ controls the intensity of the correction and determines how strongly each low-ranked synthetic year is shifted toward the corresponding observed annual sum before the adjustment is terminated.

Although this step is heuristic, it is useful because it explicitly targets the lower tail of the annual-sum distribution and complements the effects of Steps 2a and 2b. In this way, Step 2 as a whole targets not only rank structure and local continuity, but also the distribution of annual scenarios, which is particularly important for low-flow years and water-management applications.

Pseudocode of Step 2c:

Algorithm 3 Annual-sum matching by synchronous multisite swaps

Input: block-mosaic reordered synthetic matrix $G^{out} \in \mathbb{R}^{n_{gen} \times M}$ from Step 2b, observed matrix $O \in \mathbb{R}^{n_{obs} \times M}$, reference station $r = 1$, minimum tolerance $minK = 1.25$, and number of corrected low-ranked years $GMIN$

```

1  set  $G^{corr} = G^{out}$ 
2  determine  $N_{GEN}$  from  $G^{corr}$  and  $N_{OBS}$  from  $O$ 
3  compute  $tol = linspace(minK, 1, GMIN)$ 
4  compute annual sums  $S_G$  and  $S_O$  at station  $r$ 
5  sort synthetic and observed years increasingly by annual sums
6  for  $gg = 1 \dots GMIN$ :
7      choose synthetic year  $y_g$  and observed year  $y_o$  of the same rank
8      sort days within  $y_g$  by decreasing contribution to  $S_G(y_g)$ 
9      for each ranked day  $d$ :
10         if  $tol(gg) \times S_G(y_g) < S_O(y_o)$ 
11             break
12         find donor year  $y_f \neq y_g$  with minimum value on day  $d$  among admissible
synthetic years
13         swap day  $d$  between years  $y_g$  and  $y_f$  on all stations simultaneously
14         update annual sums
15     end for
16 end for
17 return corrected matrix  $G^{corr}$ 

```

Explanation of Algorithm 3. Lines 1–5 prepare the annual-sum matching step by copying the matrix obtained after Step 2b, defining the number of synthetic and observed years, constructing the tolerance sequence, computing annual sums at the reference station, and ranking both synthetic and observed years from the driest to the wettest. Lines 6–16 then process the selected low-ranked synthetic years. For each selected rank, the algorithm pairs a synthetic year with the observed year of the same annual-sum rank and examines the days of the synthetic year in decreasing order of their contribution to its annual sum. The correction stops when the tolerance condition in line 10 indicates that no further reduction is required for the selected synthetic year. Otherwise, the algorithm searches for an admissible donor year with the smallest value on the selected day and swaps that calendar day between the selected synthetic year and the donor year. Each accepted swap is applied synchronously across all stations, meaning that the same calendar day is exchanged at S1, S2, and S3. This adjusts the annual total of the selected synthetic year without correcting each station independently, thereby preserving the multisite structure and the coherence of the generated scenario along the river system. The output of this algorithm is annual-sum-adjusted synthetic matrix $G^{corr} \in \mathbb{R}^{n_{gen} \times M}$.

2.2.5. Step 3: Adaptive Permutation Genetic Algorithm

Step 3 determines the final ordering of synthetic years. Each candidate solution is represented as a complete permutation vector of the n_{gen} generated years. The aim is to identify an ordering that reproduces the target interannual dependence structure, including annual-sum autocorrelation and short-range dependence across the boundary between consecutive years. This formulation places the final year-ordering task within the class of permutation-optimization problems, similar in structure to other ordering problems such as the traveling salesman problem, and makes it suitable for an adaptive genetic-algorithm search [27–30].

For each station, target autocorrelations of annual sums up to lag 13 are first computed from the observed data. Autocorrelation analysis is a standard time-series diagnostic for assessing serial dependence across successive observations [3]. Let $A_{s,y}$ denote the annual sum at station s in year y ; then the annual autocorrelation at lag $\uparrow$ is computed over the series of annual sums rather than over daily values. This distinction is important because the third step permutes whole years, thereby changing the interannual ordering of annual sums.

The maximum lag of 13 follows the stability principle that the largest autocorrelation lag should remain substantially smaller than the available record length. In this study, the maximum lag is therefore treated as a record-length-based design choice rather than as a calibrated hydrological parameter. With 52 observed years, one quarter of the record length corresponds to 13 years. This choice evaluates medium-range interannual dependence while retaining enough paired annual observations for stable correlation estimation at the largest lag; at lag 13, 39 paired annual observations remain available. Thus, the selected value provides a compromise between the temporal range of the analysis and the stability of the estimated autocorrelation coefficients.

$$A_{s,y} = \sum_{d=1}^{365} Q_{s,y,d} \quad (8)$$

$$\rho_{s,\uparrow}^{\text{ann}} = \text{corr}(A_{s,1:n-\uparrow}, A_{s,1+\uparrow:n}), \uparrow = 1, \dots, 13$$

In this formulation, $Q_{s,y,d}$ denotes the daily streamflow at station s , in year y , on day d , where $d = 1, \dots, 365$. Accordingly, $A_{s,y}$ represents the annual sum of streamflow for station s in year y , obtained by summing all daily values within that year. The coefficient $\rho_{s,\uparrow}^{\text{ann}}$ denotes the annual autocorrelation at lag $\uparrow$, calculated from the sequence of annual

sums. The notation $A_{s,1:n-\downarrow}$ refers to the subsequence from year 1 to year $n - \downarrow$, whereas $A_{s,1+\downarrow:n}$ denotes the corresponding subsequence shifted forward by $\downarrow$ years; n is the total number of years in the observed record.

In addition, a correlation matrix is computed from the observed data for a four-day interval at the year boundaries. For each station, the last four days of year y and the first four days of year $y + 1$ are considered, forming a 4×4 matrix of transition correlations.

For the year-boundary component, a four-day transition window was used to represent short-range daily dependence across the artificial boundary between consecutive synthetic years. The four-day length was adopted after preliminary numerical testing, where it yielded stable local year-boundary behavior while keeping the boundary-correlation term compact. The purpose of this window is not to describe longer seasonal or interannual dependence, but to evaluate whether the end of one generated year and the beginning of the next year remain locally coherent. For each station, the four-day window produces a compact 4×4 boundary-correlation matrix, which provides 16 local transition coefficients while preventing the year-boundary term from dominating the global objective function. This daily-scale choice is conceptually independent of the maximum annual-autocorrelation lag used for the annual-sum component.

$$C_{s,a,b}^{yb} = \text{corr}(Q_{s,y,365-4+a}, Q_{s,y+1,b}), \quad a, b = 1, \dots, 4 \quad (9)$$

The quantity $C_{s,a,b}^{yb}$ denotes an element of the year-boundary correlation matrix for station s , where the superscript yb refers to the year boundary. The indices a and b identify positions within the four-day interval at the end of year y and the beginning of year $y + 1$, respectively. The correlation is computed over all consecutive year pairs, $y = 1, \dots, n - 1$. Equation (9) therefore measures the correlation between day $365 - 4 + a$ of year y and day b of year $y + 1$, thereby quantifying the dependence structure across the transition between consecutive years.

The objective function of Step 3 contains two components: the deviation in annual-sum autocorrelation and the deviation in correlations across the year boundaries.

$$J(\pi) = \sum_{s=1}^3 \left[w_1 \sum_{\downarrow=1}^{13} \lambda_{\downarrow} (\rho_{s,\downarrow}^{\text{ann},\pi} - \rho_{s,\downarrow}^{\text{ann,obs}})^2 + w_2 \sum_{a=1}^4 \sum_{b=1}^4 (C_{s,a,b}^{yb,\pi} - C_{s,a,b}^{yb,\text{obs}})^2 \right] \quad (10)$$

The objective function $J(\pi)$ is defined for a candidate permutation π of synthetic years. The outer summation is taken over all stations s , while w_1 and w_2 are weighting coefficients controlling the relative importance of the two components. In the first term, $\lambda_{\downarrow}$ is the weight assigned to lag $\downarrow$, and $\rho_{s,\downarrow}^{\text{ann},\pi}$ and $\rho_{s,\downarrow}^{\text{ann,obs}}$ denote the annual autocorrelation coefficients of the permuted synthetic series and the observed series, respectively. In the second term, $C_{s,a,b}^{yb,\pi}$ and $C_{s,a,b}^{yb,\text{obs}}$ denote the year-boundary correlation coefficients of the permuted synthetic series and the observed series. Minimization of $J(\pi)$ therefore corresponds to finding the ordering of synthetic years that best reproduces both the interannual autocorrelation structure of annual sums and the short-range dependence across the year boundary.

The optimization uses an adaptive permutation-based genetic algorithm throughout the experiments. The search starts from a population of random permutations, retains elite individuals, applies a feasibility-preserving partially mapped crossover (PMX), and uses a two-position swap mutation. This combination is suitable for permutation problems because it preserves chromosome feasibility and avoids duplicate or missing synthetic-year indices [27–30]. The main GA elements are summarized in Table 3.

Table 3. The genetic-algorithm elements used in the year-ordering step.

GA Element	Implementation in This Study
Search mode	Adaptive permutation-based genetic algorithm used throughout the experiments.
Chromosome	A complete permutation vector of generated synthetic years; each candidate solution defines the final ordering of years.
Gene	One synthetic-year index in the permutation.
Initial population	Random permutations. In each adaptive generation block after the first one, the best permutation found so far is inserted into the new initial population together with newly sampled random permutations.
Fitness function	Station-aware minimization objective combining annual-sum autocorrelation error and a four-day year-boundary correlation error. In the implemented configuration, $w_1 = 1$ for annual autocorrelation error and $w_2 = 0.05$ for year-boundary correlations.
Lag weighting	Annual autocorrelation is evaluated up to lag 13. The first four lags receive higher weights (1.3), while the remaining lags receive unit weights.
Boundary-dependence term	For each station, a 4×4 correlation matrix between the last four days of one year and the first four days of the next year is compared with the corresponding observed matrix.
Selection and elitism	Selection is combined with elitism; the best individuals are retained between generations. The implemented elite count is 5.
Crossover	Feasibility-preserving partially mapped crossover (PMX), preserving valid permutations and avoiding duplicate years. The implemented crossover fraction is 0.8.
Mutation	Custom two-position swap mutation: two random positions in the permutation are selected and their values are exchanged. This preserves permutation feasibility. The swap mutation is particularly suitable for this problem because it changes the ordering locally while preserving a valid year permutation.
Adaptive control	The search is run in generation blocks. Population size starts at 100 and may increase by 5% up to 500 when improvement over a sliding error window becomes slow. The maximum allowed number of cumulative generations is 10,000.
Stopping criteria	The search stops when the target error tolerance, function-tolerance condition, stall-generation criterion, or the maximum cumulative generation budget is reached.

The main advantage of Step 3 is that it explores the space of possible year permutations through an adaptive evolutionary search [27–30]. Each generation evaluates a finite population of candidate orderings, and the best solutions are progressively retained and recombined. The objective function J evaluates the quality of a candidate ordering with respect to annual-sum autocorrelation and year-boundary dependence. Since the genetic algorithm is formulated as a minimization procedure, lower values of J correspond to better agreement with the target dependence structure and therefore to a better optimized ordering of synthetic years.

Pseudocode of Step 3:

Algorithm 4 Adaptive permutation-based genetic algorithm for year ordering

Input: observed matrix $O \in \mathbb{R}^{n_{\text{obs}} \times M}$, matrix G^{corr} after Step 2c, $\text{maxLag} = 13$, $\text{window} = 4$ days

- 1 compute target annual ACFs from annual sums of O
- 2 compute target 4×4 year-boundary correlation matrices from O
- 3 initialize population with random permutations of row indices of G^{corr}
- 4 while GA stopping criterion is not met:
 - 5 evaluate fitness $J(\pi^{(k)})$ for each candidate permutation $\pi^{(k)}$
 - 6 keep elite individuals
 - 7 select parents and generate offspring by feasibility-preserving PMX
 - 8 apply two-position swap mutation
 - 9 form the next population
- 10 end while
- 11 return best permutation π_{best} and the reordered matrix G_{final}

Explanation of Algorithm 4. Lines 1–2 define the target interannual structure that the final synthetic sequence should reproduce: annual-sum autocorrelations and 4×4 year-boundary correlation matrices computed from the observed matrix O . Line 3 initializes the genetic-algorithm population by generating random permutations of the row indices of G^{corr} . Each candidate permutation represents one complete possible ordering of all synthetic years, not a pairwise swap between two rows. Applying such a permutation changes only the order in which generated years appear; it does not change the daily values inside any generated year. Lines 4–10 describe the evolutionary search. At each generation, the fitness value $J(\pi^{(k)})$ is computed for every candidate permutation $\pi^{(k)}$, where lower values indicate better agreement with the observed annual-sum autocorrelation and year-boundary dependence. The best orderings are retained through elitism, while the remaining part of the next population is created by selecting parent permutations, recombining them using feasibility-preserving PMX, and applying two-position swap mutation. PMX preserves valid permutations by avoiding duplicated or missing year indices, whereas swap mutation introduces small random changes in the ordering. Line 11 returns the best permutation π_{best} found by the search and the corresponding reordered matrix G_{final} .

2.3. Implementation, Software Settings, and Reproducibility

The proposed workflow was implemented as a modular computational procedure in MATLAB R2024a. The numerical experiments were performed under Microsoft Windows 10 (Version 10.0.19045.6456) on a workstation equipped with an AMD Ryzen 9 5950X 16-Core Processor and 128 GB DDR4 RAM. The genetic-algorithm configuration used in the final year-ordering step included a maximum annual-autocorrelation lag of 13, a four-day year-boundary window, an initial population size of 100, a maximum population size of 500, an elite count of 5, a crossover fraction of 0.8, and a maximum cumulative generation budget of 10,000. The search was first run for 500 generations. If the required objective tolerance was not reached, the optimization was continued in additional blocks of up to 500 generations, with the best solution from the previous block retained, until convergence or until the maximum cumulative generation budget was reached. The stopping criteria combined a maximum stall limit of 30 consecutive generations, a target objective tolerance of (10^{-5}) , and the maximum cumulative generation budget. The adaptive framework also allowed the population size to increase by 5%, subject to the upper limit of 500 candidate solutions, when the optimization progress became insufficient.

The manuscript supports algorithmic reproducibility by providing pseudocode for the main computational modules, the relevant parameter settings, the objective-function structure, and the evaluation criteria used in the comparison. More specifically, pseudocode is provided for the Schaake-shuffle reordering step, the block-mosaic partial-freeze procedure with Hungarian assignment, the annual-sum matching procedure, and the adaptive permutation-based genetic algorithm. This enables independent implementation of the proposed workflow, although exact numerical replication with the same observed station data is limited by the availability of the third-party streamflow records used in the case study.

2.4. Comparison Criteria and Computational Complexity

The comparison between the benchmark and proposed algorithms is based on five groups of criteria: (i) errors in mean values, standard deviations, and skewness by day and station; (ii) reproduction of serial and inter-station correlations; (iii) alignment with the distribution of annual sums; (iv) reproduction of transition correlations at the year boundary; and (v) execution time. This set of criteria follows naturally from the structure of both algorithms and from the objectives defined in the benchmark studies [20–22]. For a fair comparison, both workflows were evaluated using the same observed record, the same number of generated years and the same evaluation criteria.

The proposed workflow consists of a sequence of sampling, reordering, assignment, matching, and permutation-optimization modules. Although the first generation step dominates the total runtime in the reported implementation, the most important algorithmic difference between the benchmark and proposed methods occurs in the final year-ordering stage. The benchmark evaluates repeated row-swap candidates, whereas the proposed method searches the space of year permutations using an adaptive genetic algorithm. The complexity analysis therefore focuses on the main computational drivers of each workflow component. The notation used in the analysis is as follows: n_{gen} denotes the number of generated synthetic years, n_{obs} the number of observed years, K the number of stations, T the number of days per year, $M = K \times T$ the number of station–day variables per year, B_{blk} the number of blocks, m_b the number of moving rows in block b , a the average number of sampling attempts required by the acceptance/tolerance rule, G_{min} the number of low-ranked synthetic years selected for annual-sum correction, P the genetic-algorithm population size, G_{GA} the number of genetic-algorithm generations, L_{aa} the maximum annual autocorrelation lag, W the year-boundary window length, and R the number of examined row-swap candidates in the benchmark year-ordering step.

As shown in Table 4, the main algorithmic difference between the benchmark and proposed workflows appears in the final year-ordering step. From the standpoint of complexity, Step 3 of the benchmark algorithm is the least favorable component because it requires examining a large number of row-swap combinations, with each modification entailing a new evaluation of the composite function D . By contrast, Step 3 of the proposed workflow uses a directed heuristic search: each generation evaluates a finite number of candidates, so its complexity is governed by population size, number of generations, and the cost of evaluating the objective function J , rather than by the explosive growth of the full permutation space [20–22,28,29]. Since the genetic algorithm is formulated as a minimization procedure, lower values of J correspond to better agreement with the target dependence structure and therefore to a better optimized ordering of synthetic years.

The proposed algorithm is also heuristic and therefore does not guarantee a global optimum in the strict mathematical sense. Nevertheless, this is a justified compromise, because in hydrological applications it is more important to obtain a fast, stable, and sufficiently accurate solution than to perform exhaustive search over a permutation space that becomes far more time and resource-intensive for 1000 synthetic years [20–22,28,29].

Table 4. The main computational drivers of the benchmark and proposed workflows.

Algorithmic Module	Dominant Operation	Approximate Computational Complexity	Comment
Step 1: nonparametric Monte Carlo generation	Sampling generated values for each station–day variable	$Mn_{gen}\mathcal{O}(a) = \mathcal{O}(aMn_{gen})$	a denotes the average number of sampling attempts required by the acceptance/tolerance rule; this generation step is shared by both workflows.
Step 2a: Schaake-shuffle reordering	Bootstrap analog selection, ranking, and sorting within each station–day group	$\mathcal{O}(M(n_{gen}\log n_{gen} + n_{obs}\log n_{obs}))$	The observed rank templates are obtained from n_{obs} observed years, while the generated values are sorted for n_{gen} synthetic years. If observed ranks are precomputed, the dominant term is $\mathcal{O}(M \times n_{gen}\log n_{gen})$.
Step 2b: block mosaic with partial freeze and Hungarian assignment	Precomputing row correlations and solving assignment problems for moving rows within blocks	$\mathcal{O}(n_{gen}^2M) + \sum_{b=1}^{B_{blk}} \mathcal{O}(m_b^3)$	The cubic term corresponds to the Hungarian assignment. Usually $m_b < n_{gen}$, because part of the rows is frozen in each block.
Step 2c: annual-sum matching	Annual-sum computation, ranking, and candidate synchronous swaps across stations	$\mathcal{O}(KTn_{gen} + Kn_{gen}\log n_{gen} + G_{min}Tn_{gen})$	G_{min} is the number of low-ranked synthetic years selected for correction.
Step 3: adaptive permutation genetic algorithm	Fitness evaluation over a population of year permutations	$\mathcal{O}(G_{CAP}Kn_{gen}(L_{aa} + W^2))$	Each chromosome is a permutation of synthetic years. Fitness uses annual-sum autocorrelation up to lag L_{aa} and year-boundary correlations within window W .
Benchmark Step 3	Repeated row-swap examination with recalculation of the composite function D	$\mathcal{O}(RKn_{gen}(L_{aa} + W^2))$	R denotes the number of examined row-swap candidates. Practical cost increases rapidly with R , n_{gen} and repeated recalculation of correlation diagnostics.

3. Results

The comparison between the benchmark and proposed algorithms is presented through four complementary perspectives: reproduction of annual-sum autocorrelation, preservation of dependence across adjacent years, aggregate diagnostic scores, and computational efficiency. This structure allows both approaches to be evaluated simultaneously in terms of statistical accuracy, hydrological realism, and practical applicability.

For ease of presentation, the three analyzed hydrological stations are referred to in the following as S1, S2, and S3, corresponding to locations Devíci, Mlanča, and Ušće, respectively. Before discussing aggregate statistical indicators, it is useful to examine whether the generated daily series reproduces the observed hydrograph pattern at the event scale.

Figure 2a–c compares a representative segment of the observed daily hydrograph with the corresponding synthetic series generated by the proposed and benchmark algorithms at the three stations. It can be seen that the proposed workflow provides a substantially more realistic reconstruction of the observed hydrograph pattern, particularly in the organization of the recession limb after the main peak and in the gradual decrease in streamflow before the next pronounced rise. By contrast, the benchmark algorithm produces a less coherent sequence, with more abrupt local oscillations and weaker alignment with the observed hydrograph shape. This confirms that the main advantage of the proposed workflow is not only improved aggregate statistical performance, but also a better daily-scale arrangement of generated values in accordance with the hydrograph characteristics of the observed record.

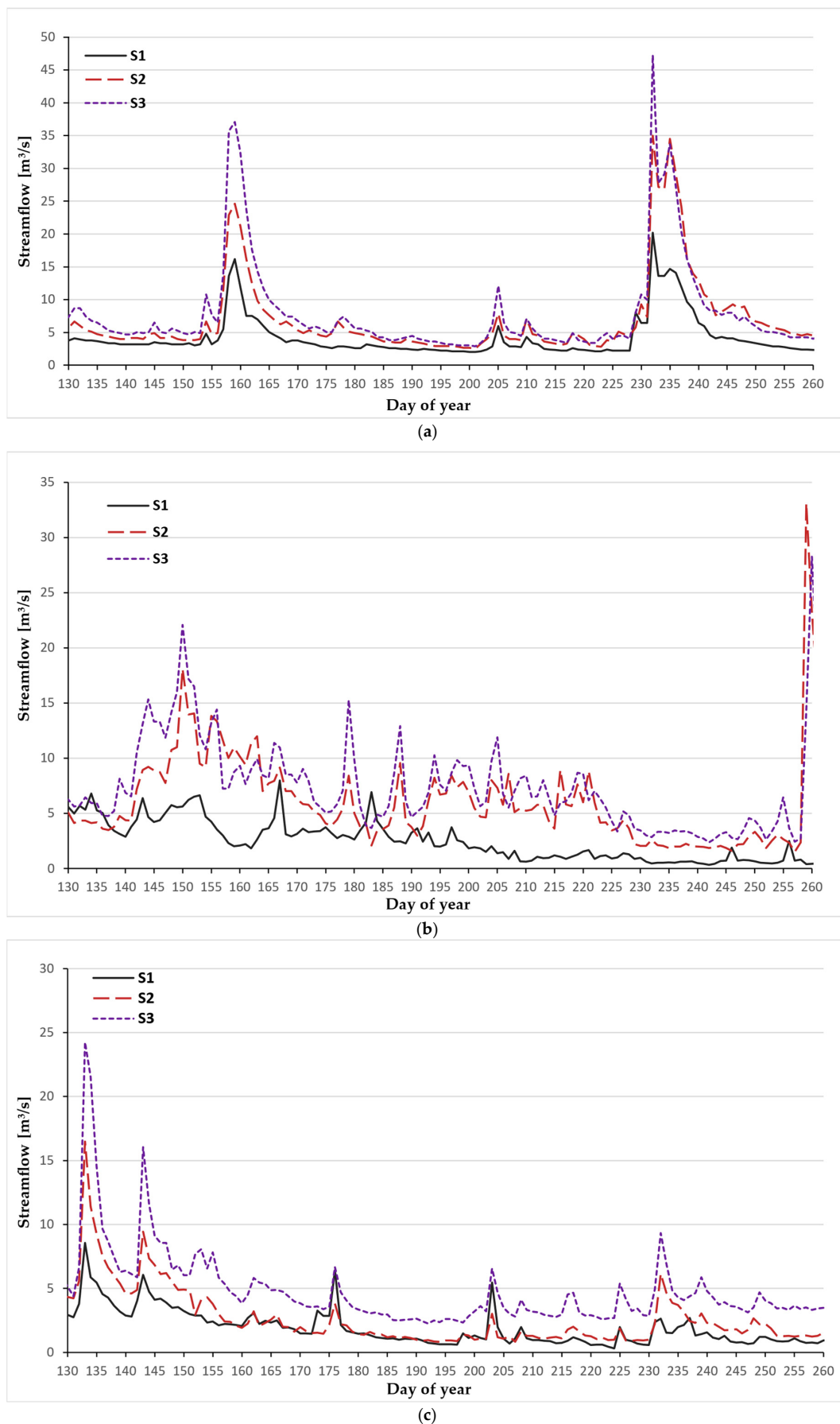


Figure 2. (a) The observed daily streamflow hydrograph for the selected segment of randomly selected observed year 13, days 130–260, at stations S1, S2, and S3. (b) The synthetic daily streamflow

hydrograph for the selected segment of randomly selected synthetic year 566, days 130–260, generated by the benchmark algorithm at stations S1, S2, and S3. (c) The synthetic daily streamflow hydrograph for the selected segment of randomly selected synthetic year 566, days 130–260, generated by the proposed workflow at stations S1, S2, and S3.

3.1. Reproduction of Daily Serial Correlation

Figure 3 presents errors in daily serial correlation coefficients at lags 1 and 2 by day of the year for all analyzed hydrological stations, comparing the synthetic series generated by the benchmark and proposed algorithms with the observed values. These results provide a detailed assessment of short-term temporal persistence in daily streamflow, which cannot be fully inferred from annual aggregates alone. Both algorithms capture the seasonal variation in serial dependence over most of the year, whereas the proposed algorithm shows more stable agreement with the observed values, particularly at lag 2, which is more sensitive to deviations in daily dynamics. This result is consistent with the aggregate indicators for the temporal-dependence group, in which the proposed algorithm achieved a lower error than the benchmark algorithm.

3.2. Reproduction of Annual-Sum Autocorrelation

Table 5 presents the autocorrelation coefficients of annual sums for lags 1–13 at all three stations. Both algorithms reproduce interannual dependence very well, as the deviations from the observed values are small and stable across the full lag range. This alignment is further confirmed in Figure 4a, where the simulated and observed coefficients are nearly coincident for all stations and almost all lags. At the same time, the proposed algorithm exhibits a slight but consistent advantage, with a mean absolute deviation across all 39 comparisons of 0.000846, compared with 0.001359 for the benchmark algorithm. The contrast is particularly clear in Figure 4b, where a compact station-wise comparison for lags 1–4 again shows the proposed algorithm to be slightly closer to the observed annual autocorrelation structure.

Table 5. Comparison of annual autocorrelation coefficients by lag for observed (Obs.) series, benchmark algorithm (BA), and proposed algorithm (PA).

Lag	Obs.S1	BA S1	PA S1	Obs.S2	BA S2	PA S2	Obs.S3	BA S3	PA S3
1	0.038	0.037	0.038	0.064	0.066	0.064	0.172	0.172	0.172
2	0.136	0.136	0.136	0.101	0.100	0.102	0.068	0.069	0.068
3	0.069	0.067	0.069	0.108	0.112	0.107	0.058	0.057	0.059
4	−0.069	−0.069	−0.069	−0.154	−0.153	−0.153	−0.231	−0.230	−0.232
5	0.142	0.142	0.141	0.066	0.065	0.068	−0.038	−0.037	−0.039
6	−0.154	−0.154	−0.153	−0.147	−0.147	−0.146	−0.225	−0.223	−0.223
7	−0.123	−0.124	−0.123	−0.270	−0.266	−0.268	−0.145	−0.144	−0.143
8	0.096	0.094	0.095	0.048	0.048	0.050	0.110	0.110	0.111
9	−0.077	−0.078	−0.076	−0.154	−0.153	−0.154	−0.064	−0.066	−0.065
10	−0.015	−0.017	−0.016	−0.065	−0.062	−0.064	0.102	0.098	0.101
11	−0.040	−0.042	−0.039	−0.006	−0.006	−0.006	−0.024	−0.021	−0.025
12	−0.285	−0.286	−0.284	−0.311	−0.310	−0.311	−0.370	−0.369	−0.370
13	−0.113	−0.113	−0.115	−0.123	−0.119	−0.123	−0.156	−0.158	−0.158

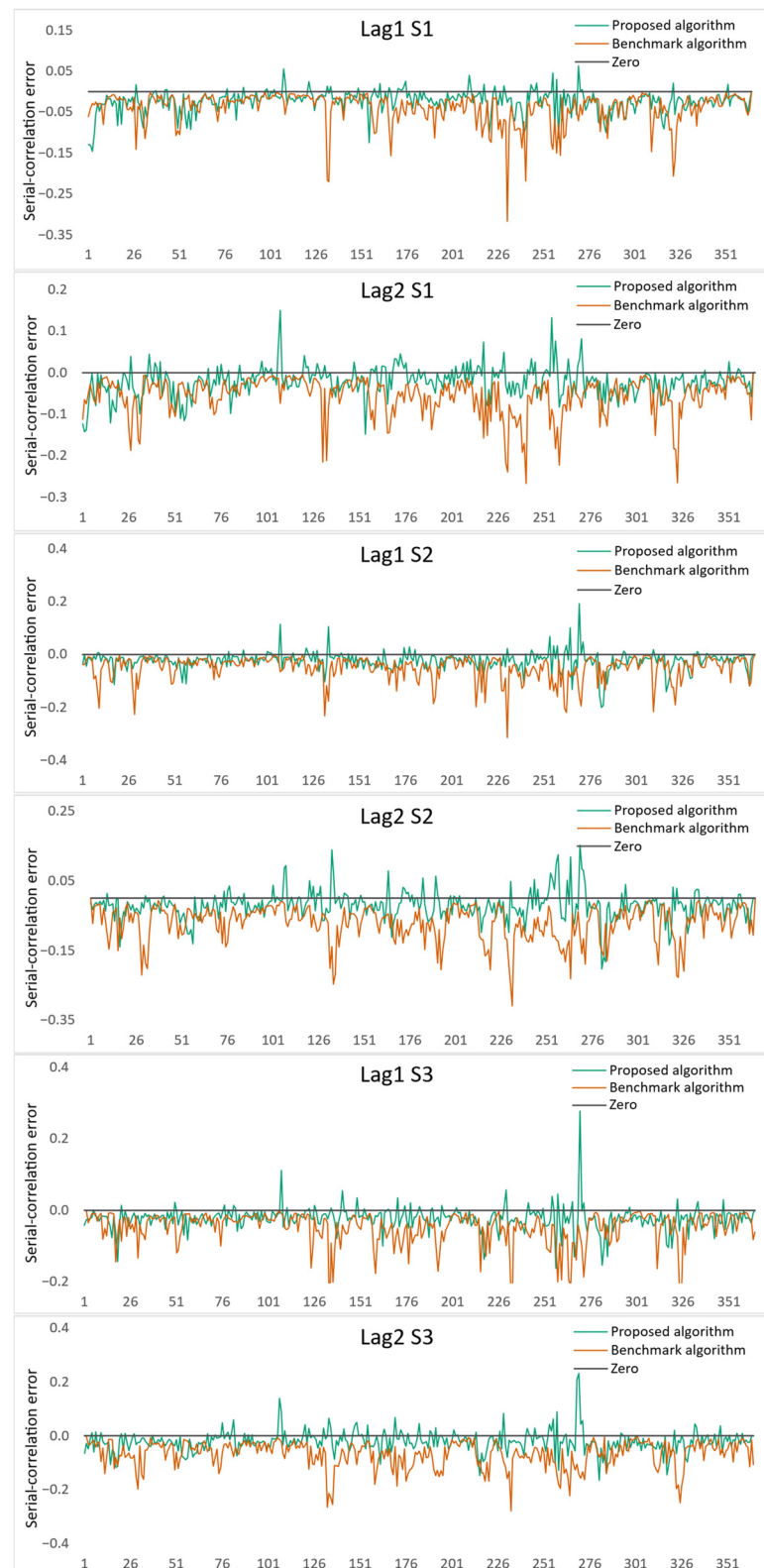


Figure 3. Errors in daily serial correlation coefficients for lags 1 and 2 at the analyzed stations: a comparison between the synthetic series generated by the benchmark and proposed algorithms and the observed values.

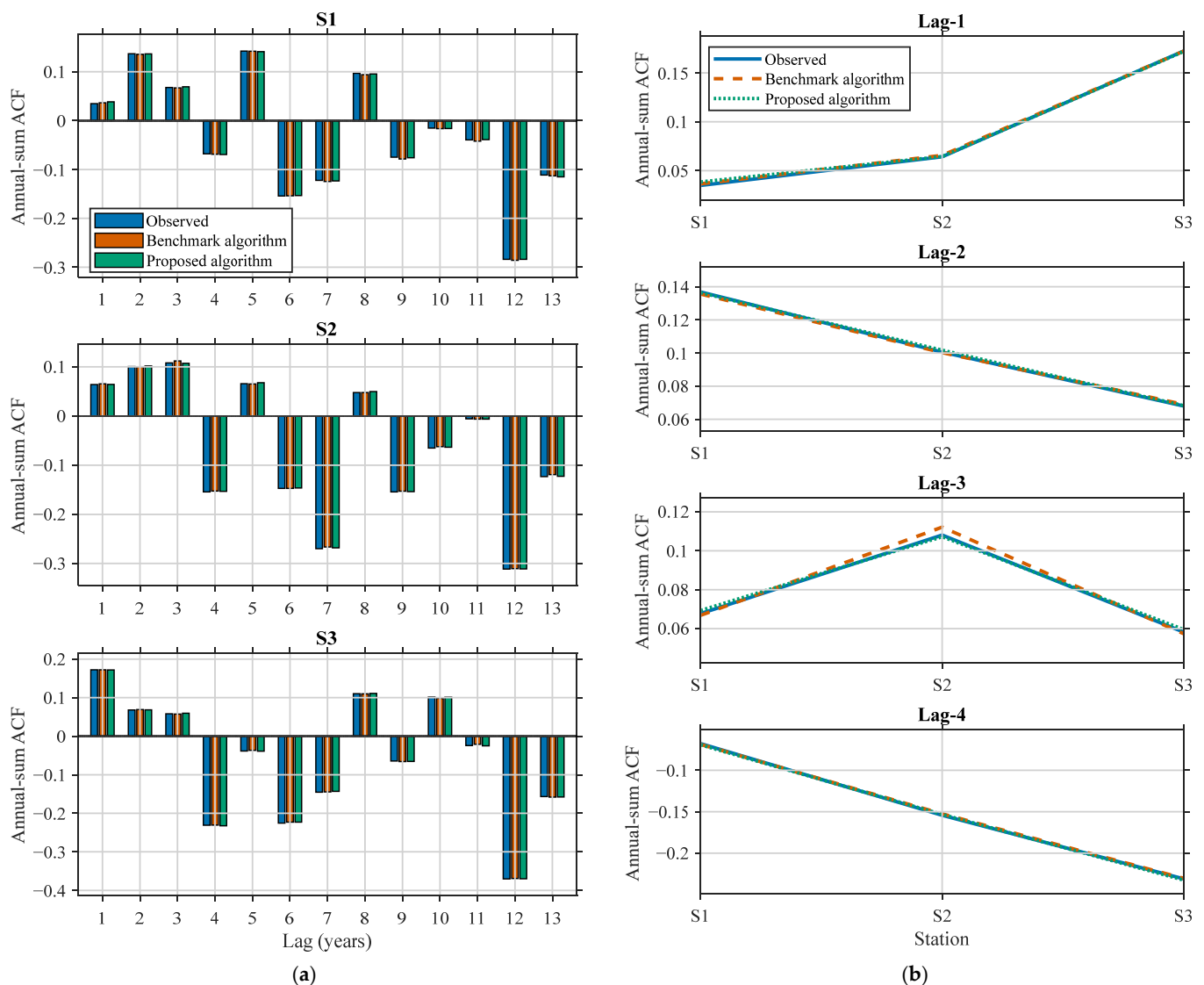


Figure 4. (a) The autocorrelation coefficients of annual sums for lags 1–13 at stations 1–3; (b) a comparison of the autocorrelation coefficients of annual sums across the stations for lags 1–4 for the observed series and the synthetic series generated by the benchmark and proposed algorithms.

3.3. Correlations at the Transition Between Consecutive Years

Table 6 reports Pearson correlation coefficients between the last four days of one year and the first four days of the next one. These indicators are particularly important for daily series because they directly characterize the hydrological continuity of the transition across the year boundary. The results show that both algorithms preserve the positive and increasing pattern of correlations well, especially for pairs involving the last day of the year (X365).

For the displayed set of 48 station-specific coefficients, the benchmark algorithm yields a slightly smaller mean absolute deviation from the observed values (0.008375 compared to 0.017667) and is closer to the observed values in 35 cases, whereas the proposed algorithm is closer in 12 cases, with one tie. This indicates that the proposed approach does not improve every individual local coefficient at the year boundary, but it also does not degrade the underlying dependence structure and remains within the same order of magnitude as the benchmark algorithm. Overall, the proposed algorithm improves aggregate diagnostics, while the benchmark remains competitive in selected local year-boundary correlations.

Table 6. Comparison of selected Pearson correlation coefficients between days X362–X365 and X001–X004 for observed and generated series at stations 1–3.

Lag	Obs.S1	BA S1	PA S1	Obs.S2	BA S2	PA S2	Obs.S3	BA S3	PA S3
X362–X001	0.683	0.689	0.721	0.357	0.355	0.368	0.573	0.582	0.590
X362–X002	0.715	0.699	0.718	0.438	0.440	0.430	0.581	0.582	0.582
X362–X003	0.727	0.656	0.711	0.392	0.401	0.393	0.608	0.587	0.600
X362–X004	0.718	0.691	0.707	0.392	0.402	0.388	0.607	0.582	0.589
X363–X001	0.661	0.664	0.662	0.460	0.463	0.481	0.579	0.569	0.573
X363–X002	0.672	0.671	0.644	0.518	0.513	0.506	0.540	0.535	0.524
X363–X003	0.673	0.677	0.632	0.444	0.450	0.465	0.532	0.538	0.532
X363–X004	0.662	0.699	0.638	0.437	0.453	0.456	0.514	0.521	0.524
X364–X001	0.800	0.806	0.803	0.580	0.583	0.612	0.679	0.679	0.686
X364–X002	0.813	0.819	0.793	0.687	0.688	0.677	0.663	0.662	0.674
X364–X003	0.821	0.807	0.791	0.619	0.620	0.641	0.675	0.675	0.691
X364–X004	0.817	0.825	0.796	0.612	0.614	0.630	0.672	0.671	0.681
X365–X001	0.830	0.828	0.815	0.708	0.703	0.689	0.741	0.735	0.712
X365–X002	0.856	0.849	0.825	0.840	0.833	0.792	0.719	0.723	0.703
X365–X003	0.867	0.860	0.822	0.780	0.774	0.766	0.737	0.739	0.725
X365–X004	0.863	0.869	0.829	0.775	0.774	0.741	0.725	0.729	0.708

3.4. Aggregate Diagnostic Scores and Key Differences Between the Algorithms

The clearest picture of overall performance is provided by the aggregate diagnostic scores presented in Table 7. The proposed algorithm achieves a lower mean error in 10 of the 14 indicator groups, while the remaining four groups are practically identical to the benchmark algorithm. Importantly, at the level of aggregate group scores, the proposed algorithm does not produce a worse score in any group. This pattern is also evident in Figure 5a, which shows that the proposed algorithm performs at least as well as the benchmark algorithm across all diagnostic groups and displays a clear advantage in several of them. The most pronounced improvements occur in groups that describe more complex and hydrologically meaningful properties of the series rather than only its basic moments. In particular, hydrologic indices (0.539 vs. 0.161); spells and persistence (17.820 vs. 11.122); timing of extremes (17.995 vs. 11.884); seasonality: day-of-year (DOY) statistics and correlations (0.413 vs. 0.244); and distribution distances (0.119 vs. 0.063) stand out. By contrast, daily distribution statistics, the flow duration curve, distribution quantiles, and overall goodness-of-fit indices remain essentially unchanged, showing that the proposed algorithm preserves the strengths of the benchmark procedure while improving the more demanding aspects of hydrological behavior. The same conclusion is supported by Figure 5b, where the radar chart shows a more favorable overall profile for the proposed algorithm; values closer to the center correspond to smaller normalized errors and therefore better agreement with the observed data.

Table 7. Group score (mean error).

Group	Benchmark Algorithm	Proposed Algorithm
Annual statistics	0.151	0.093
Distribution distances	0.119	0.063
Daily distribution statistics	0.063	0.063
Flow duration curve (FDC)	0.035	0.035
Distribution quantiles	0.014	0.014
Flow dynamics	0.490	0.417
Hydrologic indices	0.539	0.161
Overall goodness-of-fit indices	0.006	0.006
Seasonality: DOY statistics and correlations	0.413	0.244
Seasonality: Weekly aggregates	0.128	0.076
Spatial structure	0.028	0.016
Spells and persistence	17.820	11.122
Temporal dependence	0.136	0.084
Timing of extremes (DOY)	17.995	11.884

Table 8 shows that the largest discrepancies between the algorithms occur in indicators describing the duration of characteristic hydrological states and the timing of extreme events. This pattern is also clearly illustrated in Figure 6, where the metrics with the largest absolute difference Δ are highlighted. Positive Δ values indicate an advantage of the proposed algorithm, whereas negative values indicate an advantage of the benchmark algorithm. The greatest advantage of the proposed algorithm is observed for the number of low-flow periods, the timing of the annual minimum, the number of high-flow periods, and dynamic-response indicators such as the Richards–Baker index and fall rate. The only indicator in this group for which the benchmark algorithm retains an advantage is the timing of the annual maximum, confirming that the proposed method does not dominate every individual metric, but that its overall value arises from a larger number of consistent improvements in hydrologically relevant indicators.

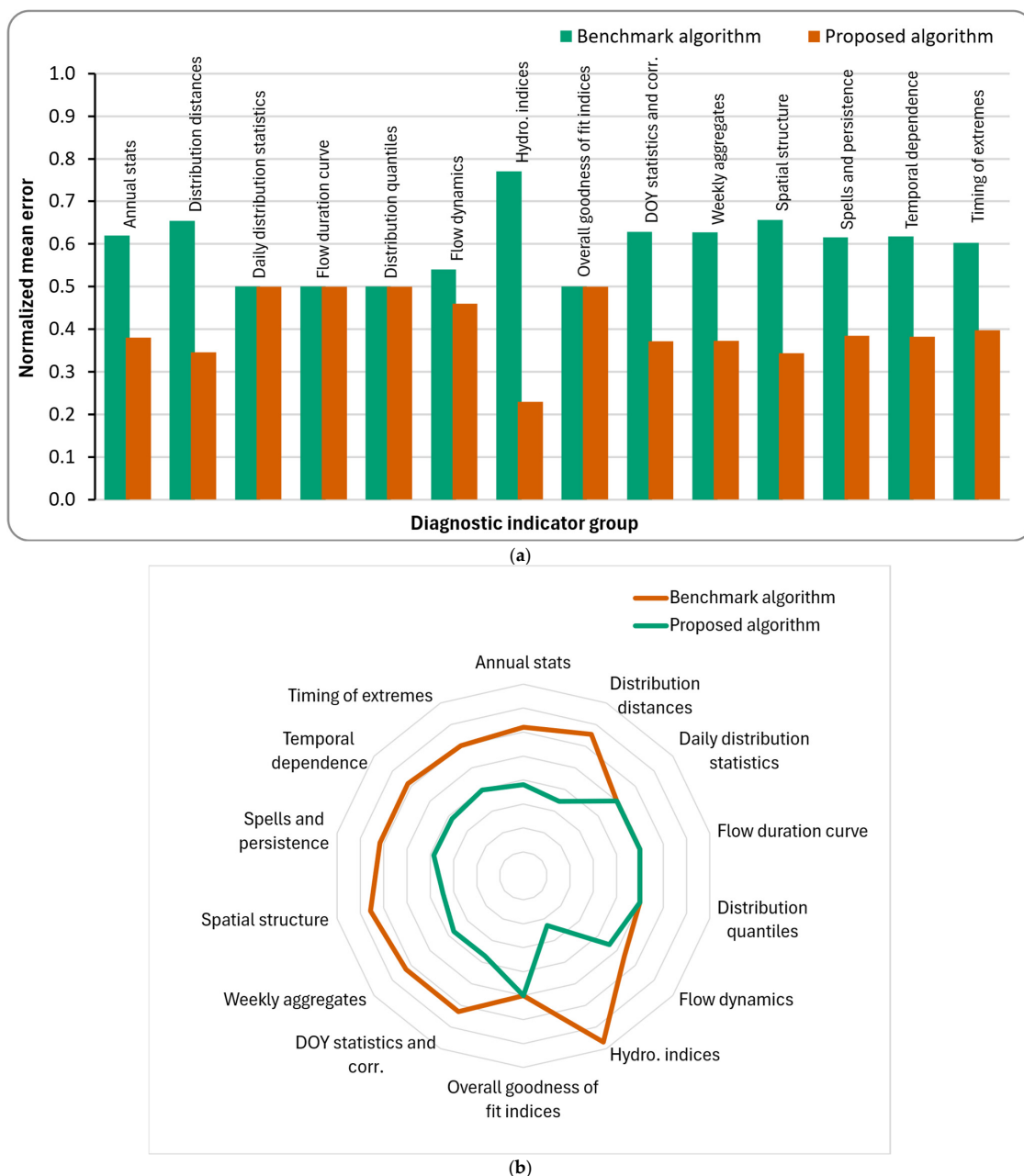
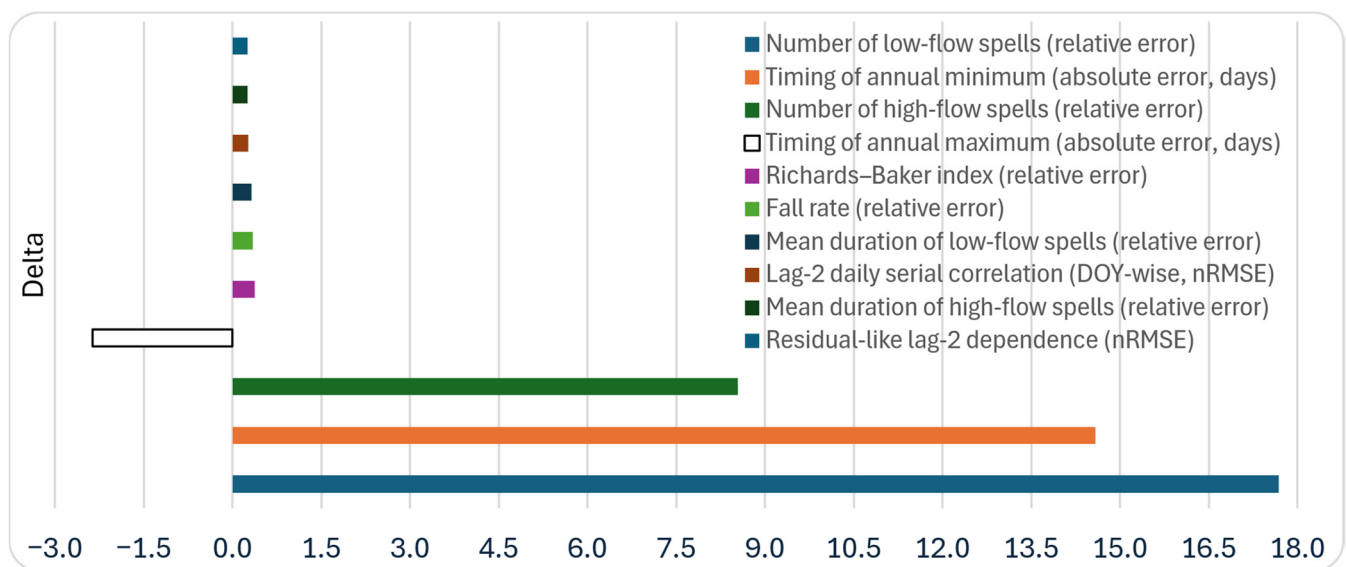


Figure 5. (a) Normalized group scores by diagnostic group for the benchmark and proposed algorithms. (b) Radar-chart comparison of normalized group scores for the benchmark and proposed algorithms; values closer to the center indicate better agreement with the observed series.

Table 8. Top 10 metrics by absolute difference between algorithms Δ , averaged across stations.

Metric	$\Delta = (BA - PA)$
Number of low-flow spells (relative error)	17.680
Timing of annual minimum (absolute error, days)	14.585
Number of high-flow spells (relative error)	8.536
Timing of annual maximum (absolute error, days)	−2.363
Richards–Baker index (relative error)	0.379
Fall rate (relative error)	0.338
Mean duration of low-flow spells (relative error)	0.319
Lag-2 daily serial correlation (DOY-wise, nRMSE)	0.264
Mean duration of high-flow spells (relative error)	0.259
Residual-like lag-2 dependence (nRMSE)	0.258

**Figure 6.** The top 10 diagnostic metrics ranked by the absolute difference between the benchmark and proposed algorithms, averaged across stations. Positive Δ values indicate an advantage of the proposed algorithm, whereas negative values indicate an advantage of the benchmark algorithm.

Figures 7 and 8 further confirm that the proposed algorithm reproduces annual-scale characteristics successfully and, in several displays, shows a slight advantage over the benchmark algorithm. The exceedance-probability curves in Figure 7 indicate that the differences between the algorithms are most noticeable for annual minima, for which the proposed algorithm more often tracks the observed values, whereas annual means and annual maxima are reproduced with similarly high alignment by both algorithms. Figure 8 likewise confirms very good alignment between the observed and simulated annual sums in terms of the empirical distribution function, boxplots, and selected quantiles, with the proposed algorithm yielding slightly more uniform and closer alignment in several station-specific characteristics. Taken together, these results show that the proposed algorithm not only preserves the quality of annual-scale distributional reproduction, but also improves agreement for several hydrologically relevant annual characteristics.

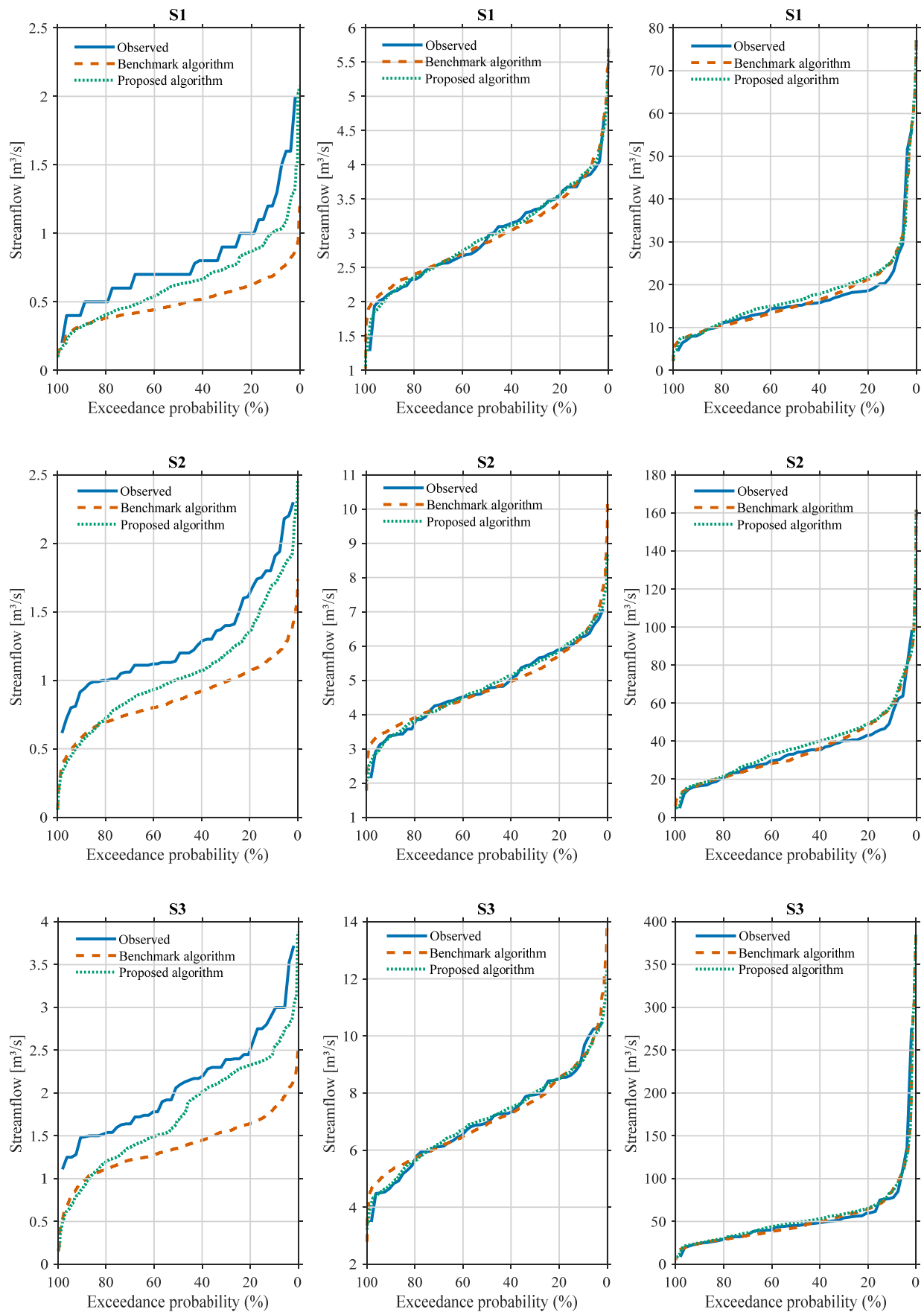


Figure 7. The exceedance-probability curves of the annual minimum, mean, and maximum flows for stations 1–3, comparing the observed series with the synthetic series generated by the benchmark and proposed algorithms.

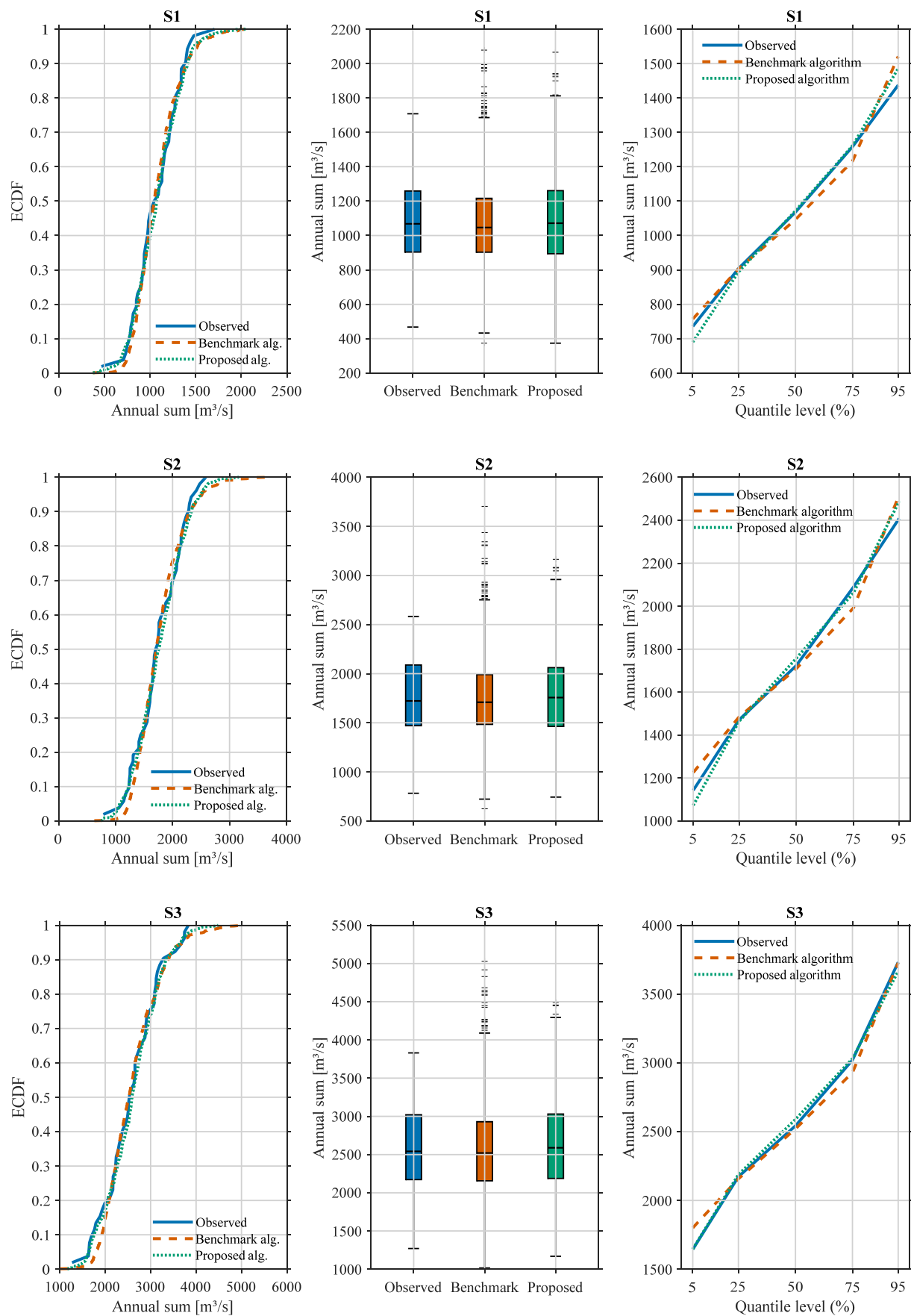


Figure 8. ECDF, boxplots of annual sums, and selected annual-sum quantiles for stations 1–3, comparing the observed series with the synthetic series generated by the benchmark and proposed algorithms.

3.5. Computational Efficiency

The execution times presented in Table 9 clearly show that the proposed workflow achieves its most noticeable advantage in the final optimization phase. Because Step 1 is shared by both workflows, its execution time is the same. The second step is somewhat longer in the proposed workflow, but this difference is negligible relative to the total execution time. The key difference arises in the third step, where the time is reduced from 15.67 h to 8.59 h, i.e., by 45.2%.

Table 9. A comparison of the execution times of the two algorithms in seconds.

Algorithm	Step 1 [s]	Step 2 [s]	Step 3 [s]	Total Time [s]
Benchmark algorithm	216,433.31	1.22	56,406.17	272,840.70
Proposed algorithm	216,433.31	8.22	30,937.40	247,378.94

Because the first step still accounts for the dominant share of the total runtime, the overall speed-up is more moderate but remains clearly quantifiable: the total runtime is reduced from 75.79 h to 68.72 h, i.e., by 9.3%. From a practical standpoint, this is important because it shows that a higher-quality synthetic series can be obtained without additional computational burden and with a simultaneous gain in efficiency in the final optimization step.

4. Discussion

The results show that the proposed method can be interpreted as a modular algorithmic workflow for daily multisite time-series reconstruction. Its main effect is the combined improvement of daily rank structure, local continuity, annual-sum behavior, and final year ordering. The evaluation indicates that these components work together to improve aggregate diagnostic performance while keeping the computational burden manageable.

A numerical leave-one-module-out experiment was not performed, and isolated numerical effect sizes are therefore not claimed. Nevertheless, the structural role of each module can be identified from the transformation it is permitted to make. Step 1 determines the sampled value set for each station–day variable. Step 2a only permutes positions of individual values within that set and therefore cannot change its marginal distribution. Step 2b is the only stage that explicitly rearranges consecutive within-year blocks under a local continuity and correlation-profile cost. Step 2c is the only stage that directly targets ranked annual sums and applies synchronous swaps of values that belong to the same column across stations. Step 3 only permutes complete synthetic years and therefore leaves all within-year values unchanged while modifying annual-sum autocorrelation and year-boundary dependence. The observed aggregate improvement should thus be interpreted as the combined outcome of complementary transformations rather than as a quantified causal contribution of any single module.

The most consistent advantage of the proposed workflow is observed in the reproduction of annual autocorrelation and in the aggregate diagnostic groups encompassing hydrologic indices, duration of characteristic hydrological states, seasonality, and timing of extremes. This suggests that the combination of Schaake shuffle, block mosaic with partial freeze, annual-sum correction, and genetic optimization yields synthetic series that reproduce complex hydrological patterns more faithfully, especially when simulation quality is assessed not only through local distribution moments but also through broader dynamic characteristics. In addition, the proposed workflow provides a closer alignment of the generated daily streamflow series with the hydrograph patterns present in the observed record, including during low-flow periods.

At the same time, the results in Table 6 show that the proposed algorithm does not achieve an absolute advantage for every individual correlation coefficient at the year boundary. Within the set of local Pearson coefficients, the benchmark algorithm is somewhat closer to the observed values. This is methodologically important because it shows that an overall improvement in the quality of generated series does not necessarily imply the best possible fit for every individual local indicator. In other words, the proposed workflow shows its main advantage in the overall preservation of hydrologically relevant series characteristics and in aggregate quality measures, whereas the benchmark algorithm remains competitive in the narrow domain of highly local dependence around the year boundary.

The better performance of the benchmark algorithm for some year-boundary correlation indicators can be explained by its final year-ordering strategy. The benchmark workflow applies a greedy pairwise-swap search: all admissible pairs of generated years within the considered segment are tested by temporarily exchanging their positions, re-computing the objective function, and accepting the swap only if it reduces the total error. This strategy can slightly improve selected year-boundary correlation coefficients because each candidate pairwise interchange is explicitly evaluated. However, it also leads to a substantially higher computational cost, since the objective function must be recomputed for a large number of candidate swaps. In contrast, the proposed workflow formulates the final year-ordering task as an adaptive permutation-optimization problem and uses a genetic algorithm to search the permutation space without exhaustively evaluating all pairwise swaps. In addition, the Step 3 objective gives greater weight to annual-sum autocorrelation than to year-boundary correlations ($w_1 = 1$, $w_2 = 0.05$), so the optimization seeks a global compromise rather than the best fit for each local boundary coefficient. Consequently, the proposed algorithm achieves a more favorable balance between accuracy and computational efficiency: it is less accurate for this narrow set of highly local year-boundary indicators, but improves broader aggregate diagnostics and considerably reduces the runtime of the final optimization stage.

From the standpoint of computational performance, the main advantage of the proposed workflow is the reduction in the third-step runtime from 56,406.17 s to 30,937.40 s, corresponding to a 45.2% decrease. Although Step 1 still dominates the total runtime and limits the overall speed-up, replacing repeated row-swap testing in the benchmark algorithm with an adaptive permutation genetic algorithm makes the final year-ordering phase considerably more efficient. Consequently, the total runtime is reduced from 75.79 h to 68.72 h, corresponding to a 9.3% overall reduction. This result is important because the proposed workflow improves aggregate diagnostic performance without increasing the total computational burden, while also improving scalability in the most algorithmically demanding optimization step.

The present study was evaluated on one three-station river system, which provides a controlled setting for comparing the benchmark and proposed workflows. Although the algorithm is formulated in a general way and can be applied to other multisite environmental time series, future work should test its transferability across additional basins, variables, and hydroclimatic regimes. Further work may also include a full numerical ablation study to quantify the isolated effect of each workflow module under controlled experimental settings.

5. Conclusions

This paper introduced a hybrid rank-preserving and evolutionary algorithm for non-parametric multisite daily streamflow simulation. A previously published nonparametric generator was used as a benchmark for controlled quantitative comparison, while the main contribution was formulated as a modular algorithmic workflow combining rank-

preserving reordering, block-wise continuity reconstruction, annual-sum matching, assignment optimization, and adaptive evolutionary year-order optimization. The results show that the proposed algorithm provides better aggregate performance than the benchmark, although its advantage should be interpreted primarily at the aggregate level rather than as absolute dominance in every individual indicator.

The proposed algorithm reproduces annual-sum autocorrelation more successfully, reducing the mean absolute deviation from 0.001359 to 0.000846. In addition, the aggregate group scores show lower error in 10 of the 14 analyzed indicator groups, while the remaining four groups are identical to the benchmark algorithm. The most noticeable advantages occur in the groups describing hydrologic indices, persistence, seasonality, and the timing of extremes, indicating that the proposed algorithm better captures more complex patterns of hydrological behavior.

More specifically, the largest aggregate improvements were obtained for hydrologic indices (0.539 to 0.161), spells and persistence (17.820 to 11.122), timing of extremes (17.995 to 11.884), temporal dependence (0.136 to 0.084), and distribution distances (0.119 to 0.063). These additional numerical results support the conclusion that the improvement is not restricted to runtime or to one correlation metric, but is visible across several hydrologically relevant diagnostic groups.

At the same time, the analysis of selected correlations at the transition between years shows that the benchmark algorithm remains highly competitive and is even somewhat closer to the observed values in this limited set of local coefficients (0.008375 versus 0.017667 mean absolute deviation). Accordingly, the most accurate conclusion is that the proposed workflow provides better aggregate performance, greater hydrological plausibility of the generated series, and closer alignment of generated daily streamflow with the hydrograph patterns observed in the historical record, while the benchmark algorithm still retains high quality in certain local aspects. These results indicate that the proposed workflow is most effective as an aggregate structure-preserving algorithm rather than as a method designed to optimize one isolated diagnostic criterion.

The study also has several limitations. The numerical evaluation was performed on one three-station river system and only for daily streamflow, so additional tests on other basins, longer records, and other hydrometeorological variables are needed to confirm broader transferability. The present implementation uses a 365-day annual matrix and therefore does not explicitly model 29 February; a more general leap-year treatment is left for future work. The method also assumes complete observed daily records for all stations, because missing values would directly affect the station-day distributions, rank patterns, and annual sums used by the algorithm. Finally, the study evaluates the proposed workflow as an integrated algorithm; a full numerical ablation study and extensions to other hydrological or meteorological variables remain subjects for future research.

In addition to the improvements in aggregate diagnostic performance, the proposed workflow also provides a practical computational advantage. The improvement in the preservation of hydrological structure is not achieved at the expense of longer computation time. Step 3 runtime decreased from 56,406.17 s to 30,937.40 s, corresponding to a reduction of 45.2%, while the total runtime decreased from 272,840.70 s to 247,378.94 s, corresponding to a reduction of 9.3%. This makes the proposed workflow suitable for long multisite stochastic simulation tasks in which both structural preservation and computational feasibility are required.

Author Contributions: Conceptualization, S.P., D.R., D.M. and S.I.; methodology, S.P., D.R., D.M. and S.I.; software, S.P., D.R. and S.I.; data curation, S.P.; formal analysis, S.P.; investigation, S.P.; writing—original draft preparation, S.P.; writing—review and editing, D.R., D.M. and S.I.; supervision, D.R., D.M. and S.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Science, Technological Development and Innovation of the Republic of Serbia, contract no. 451-03-34/2026-03/200155, realized by the Faculty of Technical Sciences in Kosovska Mitrovica, University of Priština.

Data Availability Statement: Observed streamflow data were provided by the Republic Hydrometeorological Service of Serbia and are not publicly redistributed with this submission because they are third-party data. The manuscript provides complete pseudocode for the main algorithmic modules, parameter settings, objective-function definitions, and evaluation criteria, enabling independent re-implementation of the proposed method. Additional implementation clarifications may be obtained from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the Republic Hydrometeorological Service of Serbia for providing the streamflow data.

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

References

1. Thomas, T.H., Jr.; Fiering, M.B. Mathematical synthesis of streamflow sequences for analyses of river basins by simulation. In *The Design of Water Resource Systems*; Maass, A., Ed.; Harvard University Press: Cambridge, MA, USA, 1962; pp. 459–493.
2. Matalas, N.C. Mathematical assessment of synthetic hydrology. *Water Resour. Res.* **1967**, *3*, 937–945. [[CrossRef](#)]
3. Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1970.
4. Sharma, A.; Tarboton, D.G.; Lall, U. Streamflow simulation: A nonparametric approach. *Water Resour. Res.* **1997**, *33*, 291–308. [[CrossRef](#)]
5. Srinivas, V.V.; Srinivasan, K. Hybrid moving block bootstrap for stochastic simulation of multisite multi-season streamflows. *J. Hydrol.* **2005**, *302*, 307–330. [[CrossRef](#)]
6. Künsch, H.R. The jackknife and the bootstrap for general stationary observations. *Ann. Stat.* **1989**, *17*, 1217–1241. [[CrossRef](#)]
7. Politis, D.N.; Romano, J.P. The stationary bootstrap. *J. Am. Stat. Assoc.* **1994**, *89*, 1303–1313. [[CrossRef](#)]
8. Lall, U.; Sharma, A. A nearest neighbor bootstrap for resampling hydrologic time series. *Water Resour. Res.* **1996**, *32*, 679–693. [[CrossRef](#)]
9. Lahiri, S.N. Theoretical comparisons of block bootstrap methods. *Ann. Stat.* **1999**, *27*, 386–404. [[CrossRef](#)]
10. Perrin, C.; Michel, C.; Andréassian, V. Improvement of a parsimonious model for streamflow simulation. *J. Hydrol.* **2003**, *279*, 275–289. [[CrossRef](#)]
11. Gassman, P.W.; Reyes, M.R.; Green, C.H.; Arnold, J.G. The Soil and Water Assessment Tool: Historical development, applications, and future research directions. *Trans. ASABE* **2007**, *50*, 1211–1250. [[CrossRef](#)]
12. Jahanbani, H.; Ahmed, K.; Gu, B. Data-driven artificial intelligence-based streamflow forecasting: A review of methods, applications, and tools. *J. Am. Water Resour. Assoc.* **2024**, *60*, 1095–1119. [[CrossRef](#)]
13. Kratzert, F.; Klotz, D.; Shalev, G.; Klambauer, G.; Hochreiter, S.; Nearing, G. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrol. Earth Syst. Sci.* **2019**, *23*, 5089–5110. [[CrossRef](#)]
14. Gauch, M.; Kratzert, F.; Klotz, D.; Nearing, G.; Lin, J.; Hochreiter, S. Rainfall–runoff prediction at multiple timescales with a single long short-term memory network. *Hydrol. Earth Syst. Sci.* **2021**, *25*, 2045–2062. [[CrossRef](#)]
15. Nearing, G.; Cohen, D.; Dube, V.; Gauch, M.; Gilon, O.; Harrigan, S.; Hassidim, A.; Klotz, D.; Kratzert, F.; Metzger, A.; et al. Global prediction of extreme floods in ungauged watersheds. *Nature* **2024**, *627*, 559–563. [[CrossRef](#)] [[PubMed](#)]
16. Baste, S.; Klotz, D.; Acuña Espinoza, E.; Bardossy, A.; Loritz, R. Unveiling the limits of deep learning models in hydrological extrapolation tasks. *Hydrol. Earth Syst. Sci.* **2025**, *29*, 5871–5891. [[CrossRef](#)]
17. Staudinger, M.; Herzog, A.; Loritz, R.; Houska, T.; Pool, S.; Spieler, D.; Wagner, P.D.; Mai, J.; Kiesel, J.; Thober, S.; et al. How well do process-based and data-driven hydrological models learn from limited discharge data? *Hydrol. Earth Syst. Sci.* **2025**, *29*, 5005–5029. [[CrossRef](#)]
18. Chadwick, C.; Babonneau, F.; Homem-de-Mello, T.; Letelier, A. Synthetic simulation of spatially-correlated streamflows: Weighted-modified fractional Gaussian noise. *Water Resour. Res.* **2024**, *60*, e2023WR035371. [[CrossRef](#)]
19. Wang, X.; Shen, Y.-M. R-statistic based predictor variables selection and vine structure determination approach for stochastic streamflow generation considering temporal and spatial dependence. *J. Hydrol.* **2023**, *617*, 129093. [[CrossRef](#)]
20. Ilich, N.; Despotovic, J. A simple method for effective multisite generation of stochastic hydrologic time series. *Stoch. Environ. Res. Risk Assess.* **2008**, *22*, 265–279. [[CrossRef](#)]
21. Marković, Đ.; Plavšić, J.; Ilich, N.; Ilić, S. Non-parametric stochastic generation of streamflow series at multiple locations. *Water Resour. Manag.* **2015**, *29*, 4787–4801. [[CrossRef](#)]

22. Marković, Đ.; Ilić, S.; Pavlović, D.; Plavšić, J.; Ilich, N. Multivariate and multi-scale generator based on non-parametric stochastic algorithms. *J. Hydroinform.* **2019**, *21*, 1102–1117. [[CrossRef](#)]
23. Clark, M.P.; Gangopadhyay, S.; Hay, L.E.; Rajagopalan, B.; Wilby, R.L. The Schaake shuffle: A method for reconstructing space-time variability in forecasted precipitation and temperature fields. *J. Hydrometeorol.* **2004**, *5*, 243–262. [[CrossRef](#)]
24. Scheuerer, M.; Hamill, T.M.; Whitin, B.; He, M.; Henkel, A. A method for preferential selection of dates in the Schaake shuffle approach to constructing spatiotemporal forecast fields of temperature and precipitation. *Water Resour. Res.* **2017**, *53*, 3029–3046. [[CrossRef](#)]
25. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [[CrossRef](#)]
26. Munkres, J. Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.* **1957**, *5*, 32–38. [[CrossRef](#)]
27. Goldberg, D.E.; Lingle, R., Jr. Alleles, loci, and the traveling salesman problem. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, USA, 24–26 July 1985*; Grefenstette, J.J., Ed.; Lawrence Erlbaum Associates: Hillsdale, NJ, USA, 1985; pp. 154–159.
28. Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989.
29. Whitley, D. A genetic algorithm tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [[CrossRef](#)]
30. Kramer, O.; Koch, P. Self-adaptive partially mapped crossover. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), London, UK, 7–11 July 2007*; Association for Computing Machinery: New York, NY, USA, 2007; p. 1523. [[CrossRef](#)]
31. Rashed, N.A.; Ali, Y.H.; Rashid, T.A. Advancements in optimization: Critical analysis of evolutionary, swarm, and behavior-based algorithms. *Algorithms* **2024**, *17*, 416. [[CrossRef](#)]
32. Zhassuzak, M.; Akhmet, M.; Amirgaliyev, Y.; Buribayev, Z. Application of genetic algorithms for periodicity recognition and finite sequences sorting. *Algorithms* **2024**, *17*, 101. [[CrossRef](#)]
33. Pendharkar, P.C. A Markov chain genetic algorithm approach for non-parametric posterior distribution sampling of regression parameters. *Algorithms* **2024**, *17*, 111. [[CrossRef](#)]
34. Lee, T.; Salas, J.D.; Prairie, J. An enhanced nonparametric streamflow disaggregation model with genetic algorithm. *Water Resour. Res.* **2010**, *46*, W08545. [[CrossRef](#)]
35. Lázaro, R.; Melero, J.J.; Arregui, S. Wind reference year: A new approach. *Appl. Sci.* **2025**, *15*, 13147. [[CrossRef](#)]
36. Crouse, D.F. On implementing 2D rectangular assignment algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 1679–1696. [[CrossRef](#)]
37. Studnicka, S.; Panu, U.S. Techniques and developments in stochastic streamflow synthesis—A comprehensive review. *Encyclopedia* **2025**, *5*, 198. [[CrossRef](#)]

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