water

MDPI

Article

Surrogate-Assisted Evolutionary Algorithm for the Calibration of Distributed Hydrological Models Based on Two-Dimensional Shallow Water Equations

Juan F. Farfán-Durán [1,*], Arash Heidari [2], Tom Dhaene [2], Ivo Couckuyt [2] and Luis Cea [1]

[1] Water and Environmental Engineering Group, Center for Technological Innovation in Construction and Civil Engineering (CITEEC), Universidade da Coruña, 15008 A Coruña, Spain; luis.cea@udc.es Spain

[2] Faculty of Engineering and Architecture, Ghent University—imec, 9000 Ghent, Belgium; arash.heidari@ugent.be (A.H.); tom.dhaene@ugent.be (T.D.); ivo.couckuyt@ugent.be (I.C.)

* Correspondence: j.farfan@udc.es

Abstract: Distributed hydrological models based on shallow water equations have gained popularity in recent years for the simulation of storm events, due to their robust and physically based routing of surface runoff through the whole catchment, including hill slopes and water streams. However, significant challenges arise in their calibration due to their relatively high computational cost and the extensive parameter space. This study presents a surrogate-assisted evolutionary algorithm (SA-EA) for the calibration of a distributed hydrological model based on 2D shallow water equations. A surrogate model is used to reduce the computational cost of the calibration process by creating a simulation of the solution space, while an evolutionary algorithm guides the search for suitable parameter sets within the simulated space. The proposed methodology is evaluated in four rainfall events located in the northwest of Spain: one synthetic storm and three real storms in the Mandeo River basin. The results show that the SA-EA accelerates convergence and obtains superior fit values when compared to a conventional global calibration technique, reducing the execution time by up to six times and achieving between 98% and 100% accuracy in identifying behavioral parameter sets after four generations of the SA-EA. The proposed methodology offers an efficient solution for the calibration of complex hydrological models, delivering improved computational efficiency and robust performance.

Keywords: surrogate model; evolutionary algorithm; hydrological model; optimization; shallow water equations

check for updates

Citation: Farfán-Durán, J.; Heidari, A.; Dhaene, T.; Couckuyt, I.; Cea, L. Surrogate-Assisted Evolutionary Algorithm for the Calibration of Distributed Hydrological Models Based on on Two-Dimensional Shallow Water Equations. *Water* **2024**, *16*, 652. https://doi.org/10.3390/w16050652

Academic Editor: Paolo Mignosa

Received: 29 January 2024
Revised: 13 February 2024
Accepted: 15 February 2024
Published: 22 February 2024

1. Introduction

Hydrological models are mathematical tools that can be employed to simulate the hydrological response of a watershed to precipitation. They play a crucial role in water resources engineering, addressing various purposes such as flood forecasting, streamflow simulation, and water resources management, among others [1,2]. Through the utilization of computational resources, scientists can execute advanced physics-based simulation codes to explore phenomena in a controlled environment [3] without the need of conducting experiments directly on a physical catchment.

Based on the spatial representation of the basin, hydrological models can be classified into three groups: lumped models, semi-distributed models, and distributed models [4–6]. Lumped models represent the entire watershed as a single unit, with basin-averaged parameters and inputs, providing a single output for the entire basin. Semi-distributed models divide the watershed into a limited number of sub-basins or hydrological response units (HRUs), considering spatial variability in certain parameters, but still providing aggregated outputs. Distributed models represent the watershed using a grid or mesh, accounting for spatial variability in both inputs and parameters, simulating the hydrological processes at each grid cell or mesh element, and generating outputs at multiple spatial locations.

According to the guidelines provided by [5], distributed hydrological models can be classified as physically based, analytical, or empirical, based on their representation of the movement of water through the basin. Physically based distributed models can be derived from equations describing the conservation of mass, momentum, and/or energy [4,7,8]. The use of physically based distributed models has become increasingly important due to their ability to represent the variability of hydrological responses to changes in land use and climate, as well as to investigate hydrological processes in basins with non-uniform rainfall and land use conditions [9]. These type of models have been employed in various studies for runoff forecasting [10,11], climate change impact assessment [12], and water resources management [13,14], among other applications. However, to obtain reliable results, their parameters need to be calibrated with observed data.

Parameter calibration involves adjusting the parameters of a numerical model to improve its representation of the observed variables, in a process that aims to find the combination of parameters that yields the most accurate predictions [1]. It is achieved by iteratively modifying the model parameters and evaluating one or more objective functions that quantify the fitness between the observed and the simulated variables, with the goal of minimizing the error between these two series. The search for the optimal parameters is conducted within the parameter space, which represents the range of values that each model parameter can take [15–17]. An efficient exploration of the parameter space is critical to the success of the optimization process, but it can be computationally expensive when it comes to distributed models due to the level of detail of its spatial representation and the need to solve complex partial differential Equations [18]. To overcome this computational challenge, many research studies have focused on developing methods to optimize the model parameters. These methods can be classified into two main categories: local search methods and global search methods [1].

Local search algorithms, such as gradient descent, are designed to identify local minima within a parameter space, but they lack the capability to find the global optimum [16,17,19]. On the other hand, global search methods, such as evolutionary algorithms (EAs), utilize stochastic search mechanisms to overcome this limitation [20,21]. The main disadvantage of these methods is that the number of model runs increases with the number of parameters, which leads to a slow convergence rate and a high computational cost [22]. The computational cost of both local and global search methods can be reduced by the assistance of surrogate model-based methods, or meta-modeling methods, which focus on replacing the original numerical model with a simpler cost-effective surrogate model, derived from statistical or data-driven approaches. These methods have proven to be highly effective in reducing the computational cost for tasks such as optimization and sensitivity analysis [1,23]. Several techniques have been utilized for constructing surrogate models, including radial basis (RB) [24–27], least squares support vector machines (LS-SVMs) [18,28], and artificial neural networks (ANNs) [9,15,29–34].

Nevertheless, a significant number of model runs are required to train the surrogate model, in order to ensure its accuracy in representing the actual output of the numerical model. Extracting sufficient data from a distributed model can be time-consuming, as it involves running the model with a substantial number of non-optimal parameter sets. Conversely, if the data are insufficient for training the surrogate, it may not achieve proper generalization, hindering the identification of optimal solutions. Thus, there is ongoing interest in investigating techniques that simplify the implementation of surrogate models while minimizing computational time and maximizing model accuracy [1,15,33,35]. In this regard, the development of new strategies and EAs that are well-suited for exploring solutions in simulated solution spaces generated with surrogate models could significantly accelerate the processes of calibrating complex distributed models, extending its range of application in water resources research and management.

The aim of this study is to develop an SA-EA for the calibration of a physically based distributed hydrological model based on 2D shallow water equations. By creating a simulation of the parameter space, the surrogate model aims to reduce the computational cost of the calibration, making it possible to perform multiple simulations within this space with a low computational cost. The EA guides the search for optimal parameter sets within the simulated space. The calibrated parameters are then used in the real hydrological model to reproduce observed discharge series in the study catchment. This proposed methodology seeks a more efficient exploration of the parameter space while maintaining accurate model performance.

2. Study Area and Data

The proposed methodology was applied in a case study within the Mandeo basin, situated in the northwest region of Spain (Figure 1). The whole basin spans an area of 353 km^2, while the area draining to a gauge station, which is the part of the basin modeled in this work, measures 248 km^2. The main watercourse extends for approximately 50 km up to the gauge station. The elevation within the catchment fluctuates between 328 m and 810 m above sea level, averaging at around 480 m.

The orientation of the catchment towards the northwest coincides with the most common direction of low-pressure front arrivals in this Spanish region. Due to its geographical positioning and steep orographic profile, these low-pressure fronts are uplifted as they travel through the catchment, thereby enhancing the spatial variability of precipitation [36,37].

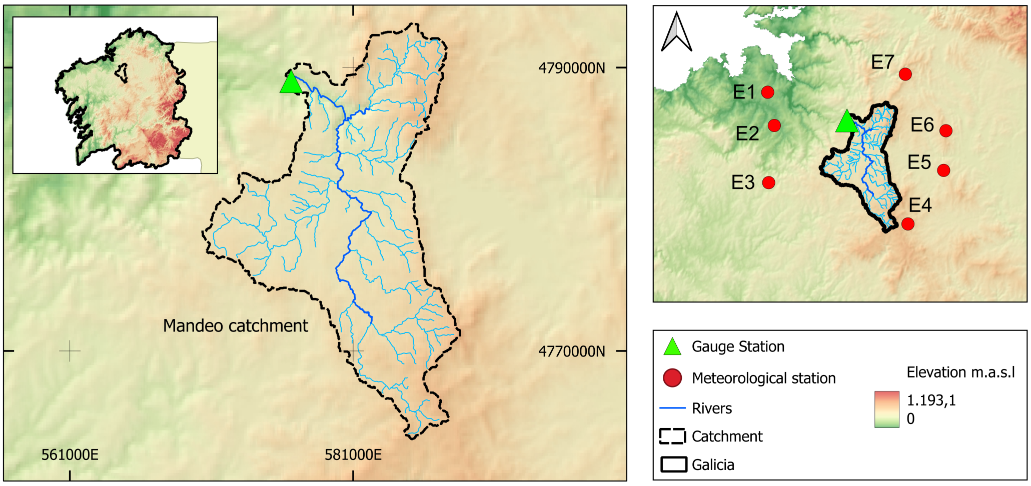

Figure 1. Mandeo catchment, including the location of the stream gauge and meteorological stations.

Historical rainfall data with a temporal resolution of 10 minutes are publicly available from the monitoring network of the regional meteorological agency, MeteoGalicia (https://www.meteogalicia.gal/). From this meteorological network, seven rain gauge stations were used to interpolate the precipitation fields over the entire catchment (Figure 1). The average annual precipitation (P_{annual}) at the rain gauge stations ranges from 1000 to 1500 mm (with the exception of one single station), while the average annual maximum daily precipitation (P_{24}) varies from 55 to 70 mm. The streamflow gauge station located at the outlet of the study basin (Figure 1) is managed by the regional water administration (Augas de Galicia) (https://augasdegalicia.xunta.gal/), with publicly available 10 min discharge data since 2008.

The proposed methodology was assessed in a synthetic rainfall event and in three real storm events (Table 1). In the synthetic event, the precipitation at each station was obtained by adding random perturbations to a synthetic 50-year return period hydrograph computed with the hydrological model using a random set of input parameters.

Table 1. Characteristics of the storm events.

Event	Start	Event Duration (h)	Max 1 h Intensity (mm/h)	Q Max (m^3/s)
E_0	-	42	11.3	201.1
E_1	05/01/2011 12:00	38	6.1	162.5
E_2	13/01/2016 00:00	38	10.7	171.3
E_3	15/02/2018 15:00	50	4.1	82.5

3. Distributed Hydrological Model

3.1. Iber

Iber is a numerical model based on the 2D depth-averaged shallow water equations for the simulation of free surface flow (https://www.iberaula.es/). It includes an hydrological module that enables its application to the simulation of rainfall–runoff transformation and overland flow at the catchment scale [38]. Even if the solver includes a GPU-enhanced implementation that reduces the computational time up to two orders of magnitude with respect to the sequential implementation [39], its computational demand is still high and thus, it is a prime candidate for the use of surrogate modeling techniques.

The mass and momentum conservation equations solved by Iber are given by:

$$\frac{\partial h}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = R - i \tag{1}$$

$$\frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h} + g\frac{h^2}{2}\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right) = -gh\frac{\partial z_b}{\partial_x} - g\frac{n^2}{h^{7/3}}|q|q_x \tag{2}$$

$$\frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_y^2}{h} + g\frac{h^2}{2}\right) = -gh\frac{\partial z_b}{\partial_y} - g\frac{n^2}{h^{7/3}}|q|q_y \tag{3}$$

where h denotes the water depth; q_x, q_y, and $|q|$ represent the two components of the unit discharge and its magnitude, respectively; z_b is the bed elevation; n is the Manning coefficient; g is the gravity acceleration; R is the rainfall intensity; i is the infiltration rate. The hydrodynamic equations are solved with an unstructured finite volume solver specifically designed for hydrological applications [38]. The model has been applied and validated for the simulation of rainfall–runoff at the basin scale in several previous works [40–42].

In order to generate the computational mesh for solving the hydrodynamic equations, we have established two distinct maximum element sizes corresponding to the different components of the terrain: slopes and rivers. The maximum element size for slopes is set at 100 m, while river channels have a maximum element size of 10 m. The resulting mesh comprises 200,757 elements in total.

A digital terrain model (DTM) with a resolution of 25 m, obtained from the Spanish National Geographic Institute (https://centrodedescargas.cnig.es/), was used to define the topography. Preliminary simulations shown that this DTM yields similar results to those generated with a 5 m resolution DTM.

The infiltration rate was calculated with an implementation of the Green-Ampt infiltration model [43] that has six parameters that are described in Section 3.2: saturated hydraulic conductivity, soil suction, soil porosity, initial soil saturation, initial loss, and soil depth.

3.2. Model Parameterization

To establish the parameter space for the calibration process, the study catchment was divided into distinct zones based on the existing land use types and a plausible range of variation for the Green-Ampt and roughness parameters was defined.

Land use information was obtained from the Spanish Land Occupation Information System (SIOSE) (https://www.siose.es/), and the land uses were assigned using the Hierarchical INSPIRE Land Use Classification System (HILUCS) developed by the Infrastructure for Spatial Information in Europe (INSPIRE) (https://inspire.ec.europa.eu/). Table 2 presents the distribution of the different land uses in the Mandeo watershed. Natural land areas and agriculture are the predominant land uses, occupying 58.82% and 36.52% of the watershed, respectively. To avoid over-parameterization the model was divided into two zones, with the remaining 4.66% of land use added to zone two, as shown in Figure 2.

Table 2. Distribution of land uses in the study catchment.

Land Use	%
Natural land areas	58.82
Agriculture	36.52
Forestry	1.92
Residential use	0.76
Transitional areas	0.59
Transport networks	0.45
Abandoned areas	0.16
Secondary production	0.15
Utilities	0.15
Use not known	0.15
Community services	0.14
Mining and quarrying	0.10
Water areas not in other economic use	0.08

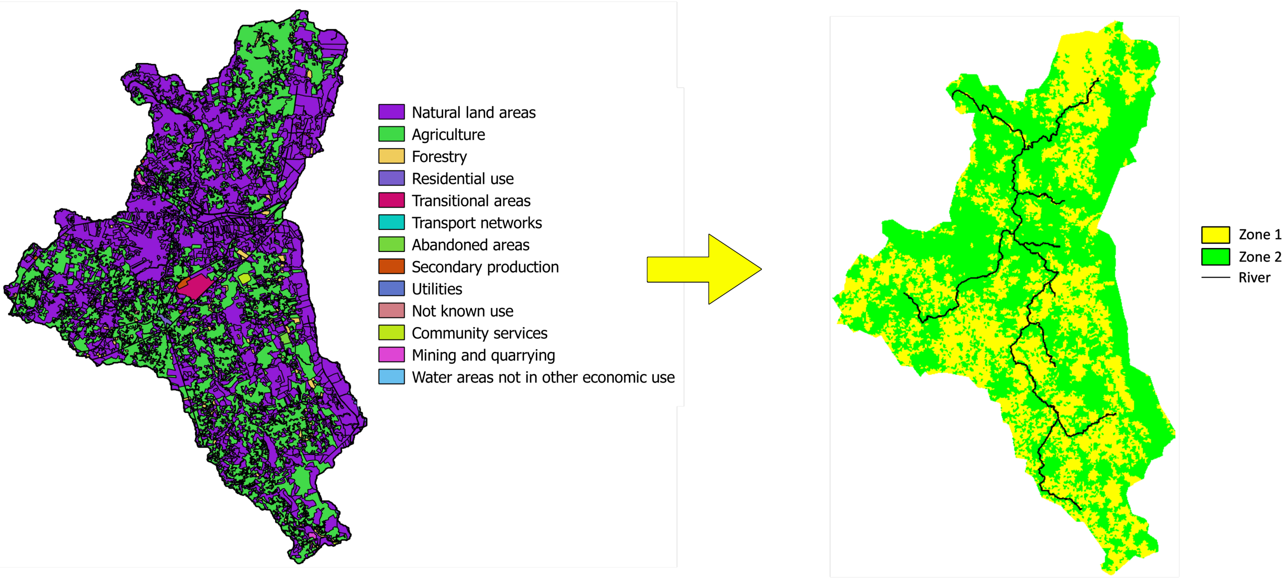

Figure 2. Zones considered for the assignation of model parameters.

Soil type information was retrieved from the SoilGrids (https://soilgrids.org/) platform, which provides global predictions of soil classes and properties with a spatial resolution of 200 m. The platform includes soil type information at various depths, and for our study, we integrate this information into the basin for the three texture components of the soils: clay, sand, and silt, as shown in Figure 3. To determine the soil type, we followed the classification guide provided by the United States Department of Agriculture in [44], which involves plotting the values in the texture diagram (Figure 3). In our study, area, on average the soil texture consists of 15% clay, 43% silt, and 42% sand, which classifies it as loam. This information was used to determine suitable values for the parameters of the Green-Ampt infiltration model. We used the values suggested for loam soil type by [45] to establish a preliminary estimation of the parameter space. We followed a similar approach to define a range of roughness values based on loam soil type, as provided by [46,47]. Additionally, we establish the roughness of the river as another parameter. In total, in this study the numerical model had 15 parameters to be optimized, which are detailed in Table 3.

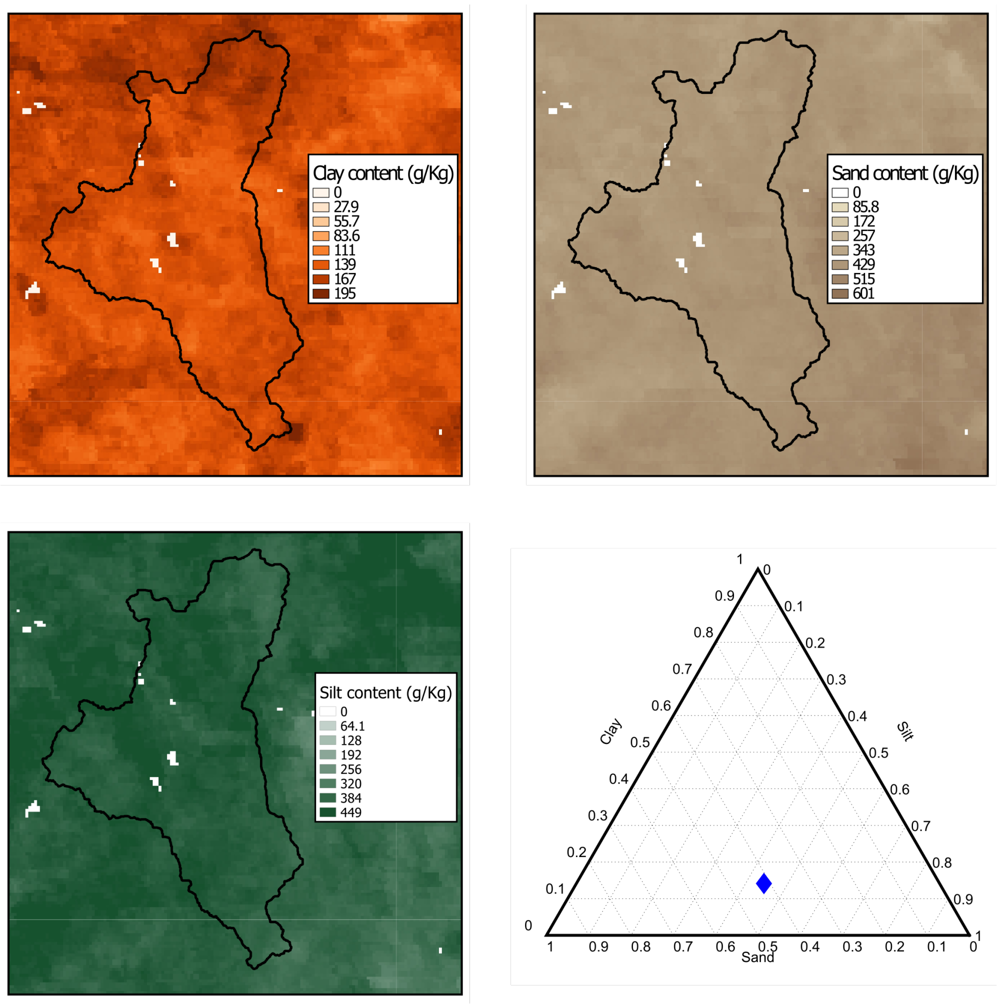

Figure 3. Soil type determination following the guidelines provided by [44].

Table 3. Parameter space considered for the calibration of the distributed hydrological model.

Zone	Description	Symbol	Units	Min Value	Max Value
Zone 1	Soil suction in zone 1.	Su_1	mm	88	273
	Soil porosity in zone 1.	Por_1	-	0.3	0.5
	Initial soil saturation in zone 1.	Sat_1	-	0.05	0.85
	Saturated hydraulic conductivity of the soil in zone 1.	Ks_1	mm/h	1	4
	Initial losses in zone 1.	$loss_1$	mm	0.1	10
	Depth of soil in zone 1.	$depth_1$	m	0.25	10
	Manning roughness coefficient for the zone 1.	n_1	-	0.012	0.18
Zone 2	Soil suction in zone 2.	Su_2	mm	88	273
	Soil porosity in zone 2.	Por_2	-	0.3	0.5
	Initial soil saturation in zone 2.	Sat_2	-	0.05	0.85
	Saturated hydraulic conductivity of the soil in zone 2.	Ks_2	mm/h	1	4
	Initial losses in zone 2.	$loss_2$	mm	0.1	10
	Depth of soil in zone 2.	$depth_2$	m	0.25	10
	Manning roughness coefficient for the zone 2.	n_2	-	0.012	0.18
River	Manning roughness coefficient for the river	n_{river}	-	0.012	0.18

3.3. Objective Functions

Two objective functions were used to quantify the fitness between the observed and simulated discharge series: the Nash–Sutcliffe Efficiency (NSE) [48] and the Weighted Nash–Sutcliffe Efficiency (WNSE) [2]. The NSE coefficient was computed as follows:

$$NSE = 1 - \frac{\sum_{i=1}^{n}(Q_{obs,i} - Q_{sim,i})^2}{\sum_{i=1}^{n}(Q_{obs,i} - \overline{Q_{obs}})^2} \tag{4}$$

where n represents the total number of data points; $Q_{obs,i}$ and $Q_{sim,i}$ are the observed and simulated discharge values at time i, respectively; $\overline{Q_{obs}}$ stands for the mean of the observed discharge series.

The WNSE coefficient is a modification of NSE that weights the differences between observed and simulated discharges according to the magnitude of the observed discharges:

$$WNSE = 1 - \frac{\sum_{i=1}^{n} w_i(Q_{obs,i} - Q_{sim,i})^2}{\sum_{i=1}^{n} w_i(Q_{obs,i} - \overline{Q_{obs}})^2} \tag{5}$$

The vector of weights w_i for each time step i was calculated as:

$$w_i = \frac{(Q_{obs,i})^p}{\sum_{i=1}^{n}(Q_{obs,i})^p} \tag{6}$$

where p is an exponent that determines the weighting of high versus low discharge values. In this study, $p = 1$ was used to give a higher weight to high flows in the WNSE.

3.4. Uncertainty Assessment

The Generalized Likelihood Uncertainty Estimation (GLUE) approach [49] was applied to evaluate the predictive capacity of the model by comparing its predictions with the observed hydrographs for each storm event. The implementation of the GLUE approach requires to define threshold to identify those parameter sets that result in a good fit with the observations, the so-called *behavioral* parameter sets. The GLUE methodology assigns weights to each behavioral parameter set in order to account for modeling uncertainty. The weights are typically proportional to the fitness measure. In this study, weights are designated based on fitness, where a higher fitness results in a greater weight being attributed to the corresponding parameter set. This scheme ensures that solutions with a superior fit to the observed data have more influence on the predictions of the model.

4. Conceptual Framework

4.1. Artificial Neural Networks

Artificial neural networks (ANNs) are computational models that simulate the structure and function of the human brain through nodes or artificial neurons. These neurons perform simple computations on the input data and are interconnected in layers, facilitating complex information processing [50,51].

ANNs can consist of a single hidden layer or multiple hidden layers, with models build from the latter often referred to as deep learning models. Each layer in an ANN can be fully connected, meaning each neuron in a layer is connected to every neuron in the next layer. This comprehensive connectivity allows for the representation of more complex functions.

For an input matrix $\mathbf{X} \in \mathbb{R}^{N \times m_x}$ and an output matrix $\mathbf{Y} \in \mathbb{R}^{N \times m_y}$, where N indicates the number of data samples, m_x denotes the number of input features, and m_y is the number of output features, the ANN aims to define a mapping function $g(\mathbf{X}; \mathbf{W})$ to predict the output $\mathbf{Y}$ based on the input $\mathbf{X}$. Here, $\mathbf{W}$ is the weight matrix of the ANN [51]. The ANN is trained using a dataset composed of input–output pairs: $\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(i)}, \mathbf{y}^{(i)}), \ldots, (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^{m_x}$ and $\mathbf{y}^{(i)} \in \mathbb{R}^{m_y}$ denote the vector representation of the i^{th} sample from $\mathbf{X}$ and $\mathbf{Y}$, respectively.

The objective during the training phase is to find the optimal set of weights **W** that minimize a loss function $\mathcal{L}(\mathbf{W})$ [52] that quantifies the difference between the predicted and actual outputs. The training problem can thus be expressed as:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = \sum_{i=1}^{N} \mathcal{L}(g(\mathbf{x}^{(i)};\mathbf{W}), \mathbf{y}^{(i)}) \tag{7}$$

The training process starts with the random initialization of the weight matrix **W**. An optimizer, typically based on gradient descent, iteratively adjusts the weights towards the direction that minimizes the loss. The iterative process continues until the weights converge to an optimal configuration that yields the smallest loss [52].

For regression problems, a common choice for the loss function is the mean squared error (MSE). It computes the average of the squared differences between the predicted outputs, denoted by $g(\mathbf{x}^{(i)};\mathbf{W})$, and the actual outputs $\mathbf{y}^{(i)}$. The MSE loss function can be mathematically represented as:

$$\mathcal{L} = MSE = \frac{1}{N}\sum_{i=1}^{N}(g(\mathbf{x}^{(i)};\mathbf{W}) - \mathbf{y}^{(i)})^2 \tag{8}$$

The operation of the ANN can be conceptually illustrated as shown in Figure 4. The input data matrix **X** is fed through multiple interconnected layers of artificial neurons, ranging from the input layer to hidden layers, and finally to the output layer, thereby transforming it into the output matrix **Y** (Figure 4). During the training process, the weights **W** within these hidden layers are iteratively adjusted to minimize the mean squared error (MSE) loss function.

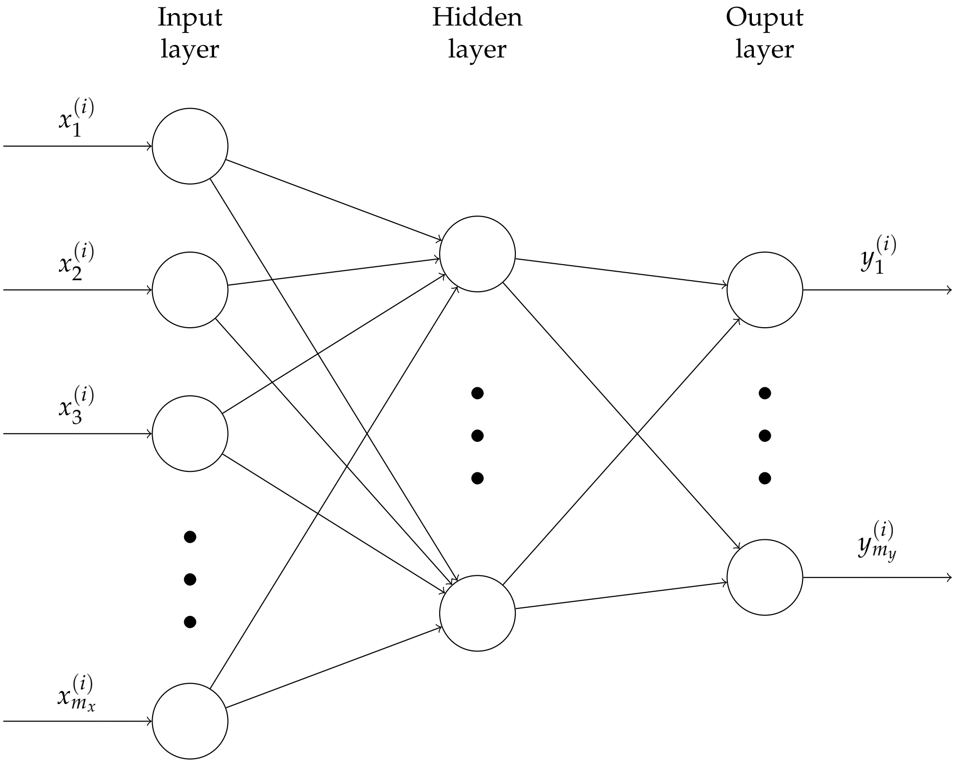

Figure 4. Structure of the ANN.

In the context of this work, i.e., the calibration of hydrological models assisted with surrogate modeling, the input matrix $\mathbf{X}$ is composed of parameters from the hydrological model, while the output matrix $\mathbf{Y}$ encapsulates their corresponding fitness values. Therefore, a properly trained ANN model can act as an efficient predictor of the fitness values $\mathbf{y}^{(i)}$ for a given set of parameters $\mathbf{x}^{(i)}$ with significantly less computational time than required by the original model.

4.2. Evolutionary Algorithms

Evolutionary algorithms (EAs) are a family of biologically inspired algorithms that are based on Darwinian evolution theory [21]. They find application across various problem domains, particularly in situations where traditional exploitative or stochastic algorithms face challenges due to resource limitations, high-dimensional spaces, or complex functionality. EAs can be effectively utilized for diverse problem types, including parameter optimization and the generation of new designs or improvements. Popular EAs include simulated annealing [53,54], genetic algorithms [55,56], and shuffled complex Evolution [57,58], among others [1]. EAs employ synthetic methods such as population management, selection, replication, and variation to achieve their objectives [20,21].

The population is a collection of potential candidate solutions. This population, initially generated randomly or by sampling from a feasible solution space, evolves with each generation. The fitness of a solution, measured using a fitness function, represents how close that solution is to achieving the desired goal. The selection process identifies and selects the best-performing members of the population for further progression. This could be likened to a natural setting, where the fittest individuals are chosen based on survival of the strongest. The replication process is the creation of a new population based on the best population members obtained in the selection process. This process usually entails assigning a probability of replication to the best population members based on their fitness. The variation process, crucial for introducing diversity within the population, can manifest in two forms: recombination and mutation. Recombination combines parts of different individuals to create new population members, akin to genetic crossover in nature. Mutation, conversely, introduces randomness by randomly altering features of each population member, reflecting genetic mutations in nature that can result in new and unique traits. Such processes can lead to the emergence of new individuals created randomly, thereby facilitating a more comprehensive exploration of the solution space. The evolution process refers to repeated cycles of selection, replication, and variation until a termination criterion is met, such as reaching a satisfactory solution or after a defined number of generations [20,21,59].

In the present work, a compact EA for optimization has been developed, in which the key components are population, fitness, selection, replication, mutation, and evolution. The initial state of the population is created using the Latin hypercube sampling (LHS) technique [60,61]. The population is given by $\mathbf{X} \in \mathbb{R}^{N \times m_x}$, where N is the number of individuals in the population and m_x is the number of features in the population member. Then, the fitness of each of the N population members is calculated by means of the fitness functions defined by the user. This fitness values are stored in the matrix $\mathbf{Y} \in \mathbb{R}^{N \times m_y}$, where m_y represents the number of fitness functions utilized.

The selection process is carried out by identifying the q best-performing members of the population, where $q \leq N$. Once these members are selected, they are assigned a selection probability based on a fitness measure, denoted as $fit(\cdot)$, which summarizes the information on their fitness contained in the matrix $\mathbf{Y}$. The specific fitness measure $fit(\cdot)$ can vary, depending on both the application of the EA and the number of fitness functions [15,59,62]. To this end, Equation (9) was applied:

$$p^{(i)} = \frac{fit(\mathbf{x}^{(i)})}{\sum_{i=1}^{q} fit(\mathbf{x}^{(i)})} \quad \text{for } i \in [1, 2, \ldots, q] \tag{9}$$

where $p^{(i)}$ is the probability of replication of the $i - th$ best population member. Equation (9) is an adaptation of the artificial bee colony algorithm proposed by [59]. Subsequently, replication consists of selecting one of the q best members and creating an initial replica $\mathbf{x}^* = \mathbf{x}^{(i)}$ for further mutation. Naturally, members with a higher $fit(\cdot)$ are attributed a higher replication probability. Then, the mutation process controlled by Equation (10) is applied:

$$x_j^* = \begin{cases} x_j^* - \alpha \cdot \text{rand}(0, x_j^*) & \text{if } \phi = -1, \\ x_j^* + \alpha \cdot \text{rand}(0, (1 - x_j^*)) & \text{if } \phi = 1 \end{cases} \tag{10}$$

where x_j^* represents a feature from the vector $\mathbf{x}^*$ that is randomly selected from the range $j \in rand[1 \ldots m_x]$. This mutation is controlled by two factors: the mutation rate, α, and the direction variable, ϕ. As illustrated in Figure 5, when the direction (ϕ) is -1, the selected feature experiences a reduction proportional to its current value. Conversely, when the direction is 1, the parameter undergoes an increase proportional to the remaining distance to 1. It is important to note that the features x_j^* are normalized between 0 and 1 within their range of variation, before applying the mutation given by Equation (10). The magnitude of the mutation is determined by α, which is a user defined parameter between 0 and 1. A high value of α promotes substantial mutations and encourages wider exploration of the solution space. Conversely, a lower value of α encourages small variations, which results in a more concentrated exploration around the most promising members [20,21,59].

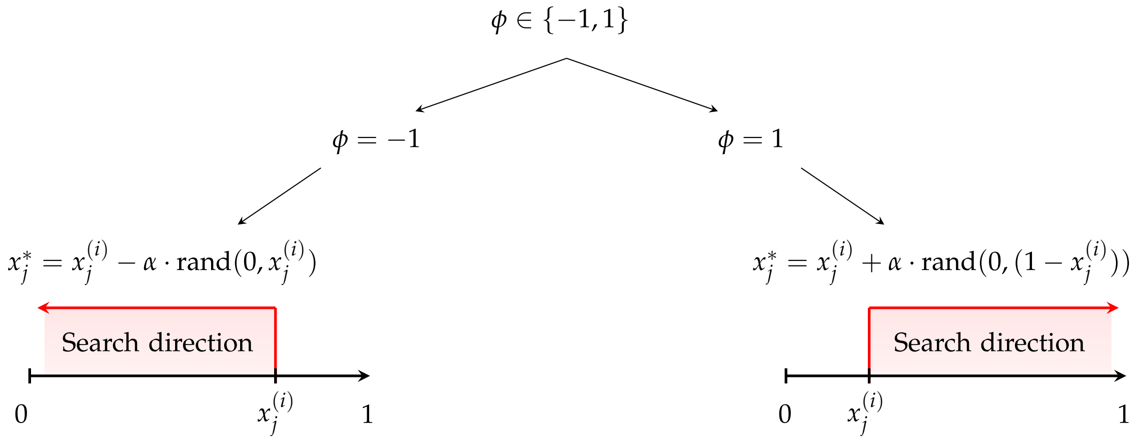
$\phi \in \{-1, 1\}$
$\phi = -1$
$\phi = 1$
$x_j^* = x_j^{(i)} - \alpha \cdot \text{rand}(0, x_j^{(i)})$
$x_j^* = x_j^{(i)} + \alpha \cdot \text{rand}(0, (1 - x_j^{(i)}))$
Search direction
Search direction
0
$x_j^{(i)}$
1
0
$x_j^{(i)}$
1

Figure 5. Schematic representation of the mutation process.

Each time $\mathbf{x}^*$ undergoes a mutation, its fitness measures ($\mathbf{y}^*$) are compared against those of its original state ($\mathbf{x}$) before mutating in terms of $fit(\cdot)$, with two counters updated accordingly: one for counting improvements (c) in relation to the previous state of each member, and another for counting the evaluations resulting in deterioration (d) (Algorithm 1). If $fit(\mathbf{x}^*)$ is found to be greater than $fit(\mathbf{x})$, the parameter $x_j^{(i)}$ is updated to its evolved state $x_j^{*(i)}$. On the other hand, if $fit(\mathbf{x}^*)$ is equal to or less than $fit(\mathbf{x})$, the population member remains unchanged. The evolution of the population members is finished once one of the stop criteria reaches its limit.

The use of these two counters aims to maintain a balance between exploration and exploitation during the optimization process, ensuring a more efficient search for optimal solutions [15,63].

Algorithm 1 Evolutionary algorithm

```
$\mathbf{X} = \{\mathbf{x_i} \,|\mathbf{x_i} \sim Uniform\}_{i=1}^{N}$
                                              ▷ Generation of the Initial Population
$\mathfrak{D} = \{(\mathbf{x_i}, \mathbf{y_i})|\mathbf{y_i} = fit(\mathbf{x_i})\}$          ▷ Fitness of each Candidate
while $Budget$ left do
    $\mathfrak{Q} = \{\mathbf{x_i}$ if $i < q$ in $sort(\mathfrak{D}|\mathbf{x_i} > \mathbf{x_j}$ if $\mathbf{y_i} > \mathbf{y_j})\}$   ▷ Selection of top $q$ best candidates
    $\mathbf{P} = \{p_i|p_i = \frac{y_i}{\sum_{i=1}^{q} y_i}$ and $x_i \in \mathfrak{Q}\}_{i=1}^{q}$          ▷ Assign probability
    for $i$ = 1 to $GN$ do
        $\mathbf{x}_i^* \sim \mathfrak{Q}, \mathbf{P}$                                  ▷ Replication
        $c \leftarrow 0, d \leftarrow 0$                                    ▷ Initialize counters
        while $c < c_{limit}$ or $d < d_{limit}$ do
            $\mathbf{x}_i^m \leftarrow Mutate(\mathbf{x}_i^*)$                       ▷ Use Equation (10)
            if $fit(\mathbf{x}_i^m) > fit(\mathbf{x}_i^*)$ then        ▷ Evaluate fit of the mutated members
                $\mathbf{x}_i^* \leftarrow \mathbf{x}_i^m$
                $c \leftarrow c + 1$
            else
                $d \leftarrow d + 1$
        $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{x}_i^*$                                  ▷ Update $\mathbf{X}$
        $\mathfrak{D} \leftarrow \mathfrak{D} \cup (\mathbf{x}_i^*, fit(\mathbf{x}_i^*))$                      ▷ Update $\mathfrak{D}$
```

5. Methodology

Traditional applications of EAs necessitate the calculation of the fitness function for each population member using the real model. This process can be computationally demanding, particularly for complex models with numerous input parameters, where a single model evaluation might take minutes or even hours. This computational load is mainly due to the sequential nature of the evaluation process, which becomes more challenging as the number of features and the size of the population increases [15].

To address this computational challenge, the present study incorporates an ANN to support the EA, aiming to reduce the associated computational cost. To this end, the ANN was employed as a surrogate model of the real hydrological model. The population members correspond to the inputs of the ANN, and their fitness values represent the outputs of the ANN. This strategy allows for the evaluation of a large number of solutions with a relatively small computational cost. The SA-EA primarily consists of the steps delineated in Figure 6. These steps are elaborated across the present section and ultimately summarized in Algorithm 2.

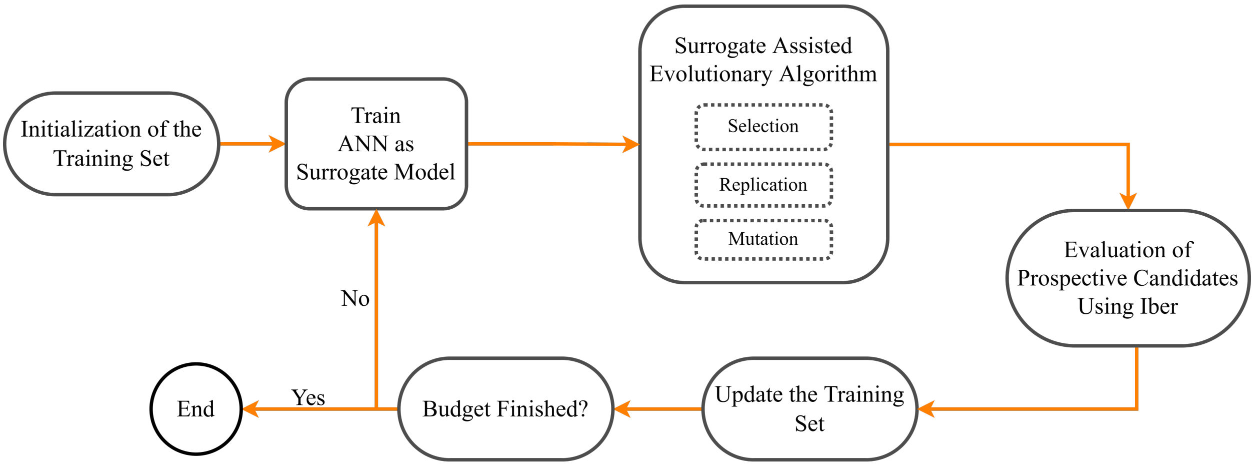
Initialization of the Training Set
Train ANN as Surrogate Model
Surrogate Assisted Evolutionary Algorithm
Selection
Replication
Mutation
Evaluation of Prospective Candidates Using Iber
Update the Training Set
Budget Finished?
No
Yes
End

Figure 6. Scheme of application of the SA-EA.

Algorithm 2 Surrogate-assisted evolutionary algorithm

```
$\mathbf{X} = \{\mathbf{x_i} \, | \mathbf{x_i} \sim Uniform\}_{i=1}^{N}$    ▷ Generation of the Initial Population
$\mathfrak{D} = \{(\mathbf{x_i}, \mathbf{y_i}) | \mathbf{y_i} = fit(\mathbf{x_i})\}$    ▷ Fitness of each Candidate
$G \leftarrow 0$    ▷ Initialization of the generation counter
while $G < NumGen$ do
    $\tilde{M} \leftarrow TrainANN(\mathfrak{D})$    ▷ Train a surrogate model
    $\mathfrak{Q} = \{\mathbf{x_i}$ if $i < q$ in $sort(\mathfrak{D} | \mathbf{x_i} > \mathbf{x_j}$ if $\mathbf{y_i} > \mathbf{y_j})\}$    ▷ Selection of top $q$ best candidates
    $\mathbf{P} = \{p_i | p_i = \frac{y_i}{\sum_{i=1}^{q} y_i}$ and $x_i \in \mathfrak{Q}\}_{i=1}^{q}$    ▷ Assign probability
    for $i$ = 1 to $GN$ do
        $\mathbf{x}_i^* \sim \mathfrak{Q}, \mathbf{P}$    ▷ Replication
        $c \leftarrow 0, d \leftarrow 0$    ▷ Initialize counters
        while $c < c_{limit}$ or $d < d_{limit}$ do
            $\mathbf{x}_i^m \leftarrow Mutate(\mathbf{x}_i^*)$    ▷ Use Equation (10)
            if $\tilde{M}(\mathbf{x}_i^m) > \tilde{M}(\mathbf{x}_i^*)$ then    ▷ Estimate fit of the mutated member using $\tilde{M}$
                $\mathbf{x}_i^* \leftarrow \mathbf{x}_i^m$
                $c \leftarrow c + 1$
            else
                $d \leftarrow d + 1$
        $\mathbf{X} \leftarrow \mathbf{X} \cup \mathbf{x}_i^*$    ▷ Update $\mathbf{X}$
        $\mathfrak{D} \leftarrow \mathfrak{D} \cup (\mathbf{x}_i^*, fit(\mathbf{x}_i^*))$    ▷ Update $\mathfrak{D}$ by evaluating $\mathbf{x}_i^*$ using Iber
    $G \leftarrow G + 1$
```

The initialization process requires the definition of an initial population $\mathbf{X} \in \mathbb{R}^{N \times m_x}$, where N, is the number of individuals in the population and m_x is the number of parameters of the hydrological model. In our case, the parameters used are those summarized in Table 3. The initial population was generated using the Latin hypercube sampling (LHS) technique, and it was employed to run the Iber model described in Section 3.1, resulting in a collection of N simulated discharge series. Then, the fitness of the each of the N population members was calculated using the objective functions described in Section 3.3. This fitness values were stored in the matrix $\mathbf{Y} \in \mathbb{R}^{N \times m_y}$, where m_y represents the number of objective functions utilized, which in the present case are two: NSE and WNSE.

The training of the ANN as a surrogate model of Iber was carried out using the model parameters in $\mathbf{X}$ and their corresponding fitness measures in $\mathbf{Y}$ as input and output data, respectively. To this end, a cross-validation technique was employed for performance evaluation. The training set was divided into k folds, with each k separate models trained on $k - 1$ folds as the training set and the remaining fold as the validation set. The surrogate model prediction was computed as the average prediction of the k models [64,65]. The ANN hyperparameters, such as the number of neurons, activation functions, number of epochs, and batch size were tuned through trial and error to ensure proper training.

Contrary to a standard application of the EA, where only one member replicates in each iteration, in the SA-EA version, GN replicas of the q best population members can be selected to take advantage of the high-throughput capabilities of the ANN. Once the replication is accomplished, the mutation of the replicated members could start using Equation (10) for the GN replicas of the best population members.

The fitness of the prospective members is predicted using the trained ANN, yielding $\mathbf{y}^{*(\mathbf{i})}$. If the fitness of the mutated member surpasses the current fitness, the original member is replaced by the mutated one, and the improvement counter c is incremented. Otherwise, the deterioration counter d is incremented. The evolution of a population member is stopped once one of the counters reaches a limit of c_{limit} for improvements or d_{limit} for deterioration. This iterative process is repeated until the stopping criterion is reached for all the GN-replicated members. Subsequently, the numerical model Iber is run to validate the fitness of the prospective population. Finally, the training set is updated for the next generation, and the generation counter is incremented ($G = G + 1$).

Since training an ANN requires a significant amount of data, which can pose a handicap for surrogate modeling applications [35,66], the proposed methodology uses a progressive approach to for update the training set that consists on increasing its size after each obtaining each new generation in the SA-EA. Initially, the ANN is trained with a population of size N and, by the end of each iteration, a new generation (G) of GN new members is added to the training set. By using this approach, the ANN can be updated with a population of GN members in each iteration of the SA-EA, leading to a more accurate ANN as the size of the training set grows, while still utilizing it from the beginning of the procedure to identify prospective population members. Once the training set is updated, the counter that tracks the number of generations (G) is increased by 1, and the SA-EA may start another iteration if the limit of generations ($NumGen$) has not been reached. The steps to deploy the proposed methodology are outlined in Algorithm 2.

Experimental Settings

In the following, we detail the experimental setting used in the implementation of the SA-EA for the specific application of the current study, which have shown the most stable performance after evaluation of different configurations. Therefore, it is important to note that they are tailored to this particular study and may not directly translate to other scenarios.

In the context of hydrological modeling, a population refers to a set of physically based parameter sets. The fitness refers to the similarity of the observed and simulated discharge series produced by running the hydrological model with a set of parameters and can be computed by applying the objective functions, NSE and WNSE, described in Section 3.3. Selection refers to choosing the parameters that best reproduce the observed hydrographs. Replication is the process of creating GN replicas of the best parameter sets. Mutation corresponds to the introduction of random changes to specific model parameters. Finally, evolution corresponds to the cyclical application of selection, replication, and mutation, until the objective functions converge to their optimal values.

Given that this study incorporates two fitness measures (NSE and WNSE), the selection criterion uses Equation (11) to define the $fit(\cdot)$ function in the selection step:

$$fit(\mathbf{x}^{(i)}) = 0.5 \cdot \text{NSE}^{(i)} + 0.5 \cdot \text{WNSE}^{(i)} \text{ for } i \in [1, 2, \ldots, N] \tag{11}$$

where $fit(\mathbf{x}^{(i)})$ is the average of the NSE and WNSE values of the discharge series generated by the $i - th$ population member. Subsequently, the best q members of the population, based on their fitness values, are selected.

The training of the ANN as surrogate model was carried out using TensorFlow [67] with a cross-validation technique employed for performance evaluation. The training set was divided into five folds, with five separate models each, trained on four folds and using the remaining fold as the validation set. In this way, the surrogate model prediction was computed as the average of the predictions of the five models [64,65]. The ANN hyperparameters were tuned through trial and error, considering between 8 and 32 neurons, activation functions such as hyperbolic tangent and swish [68], and the Adam [69] optimizer. The number of epochs ranged from 200 to 400. The batch size was selected between 1, 2, 4, and 8 to ensure proper training. In order to avoid a situation in which the parameters with large numeric ranges overshadow those with smaller ranges in our model, the input data **X** are scaled to [0, 1], using a min–max normalization method [70]. The exprimental settings are summarized in table 4.

Table 4. Experimental settings of the SA-EA for exploration and exploitation of the simulated parameter space

Parameters for SA-EA	Symbol	Value
Generations counter (initial value)	G	1
Number of generations of SA-EA	$NumGen$	4
Number of folds for ANN training	k	5
Members of the initial population	N	25
Replicated members per generation	GN	25
Model parameters	m_x	15
Objective functions	m_y	2
Best population members	q	5
Improvement counter (initial value)	c	0
Improvement counter limit	c_{limit}	25
Deterioration counter (initial value)	d	0
Deterioration counter limit	d_{limit}	10
Evolution parameter	α	0.25

6. Results and Discussion

In the following, we first examine the effectiveness of the ANN in simulating the solution space. Next, we evaluate the performance of the SA-EA and the quality of the solutions identified. We compare the computational cost of the SA-EA approach with a traditional global optimization method. Lastly, we discuss the outcomes of our calibrated model and the insights obtained from our investigation within the context of hydrological modeling.

6.1. Artificial-Neural-Network-Based Surrogate Model

In the present study, the ANN is applied to predict the NSE and WNSE associated with a particular set of parameters. The MSE is then computed by comparing the NSE and WNSE values provided by the ANN with those obtained from the hydrological model.

Table 5 presents the MSE on the NSE and WNSE obtained in the training and testing phases of the ANN for the four storm events after each of the generations (G1–G4) of the EA, while Figure 7 shows the scatter plots of NSE and WNSE during the ANN training across the different generations. There is a decrease in MSE for each event, indicating an enhancement in the performance of the model over generation iterations, while rainfall event E_3 exhibits the highest MSE values across all generations, events E_0, E_1, and E_2 demonstrate lower and comparable values, suggesting varied ANN performance across the events.

The surrogate model begins to provide acceptable training values (indicating good fitting) in generations G2–G3, depending on the event (Table 5 and Figure 7). This can be explained by the fact that, as described in Section 5, generation G1 is obtained using an ANN that is trained with insufficient data. This leads to a high degree of uncertainty in the predictive capabilities of the ANN, adding a significant randomness component to the EA. However, once the solutions of G1 are validated through Iber and added to the dataset, they contribute with valuable information for the next generations. This information is related to the zones of the feasible space that do and do not provide an adequate fitness. For instance, consider a scenario where the EA-ANN provides a plausible solution that is later validated through Iber and shows a poor fit. This wrong prediction becomes information for updating the ANN regarding areas of the feasible space where poor solutions are located. Consequently, the ANN can guide the EA to avoid these unfavorable areas and steer towards more promising regions of the solution space in each generation. In this context, the results obtained from each generation can be seen as a proxy for a sampling method that improves the generalization skills of the ANN in each training iteration (Table 5).

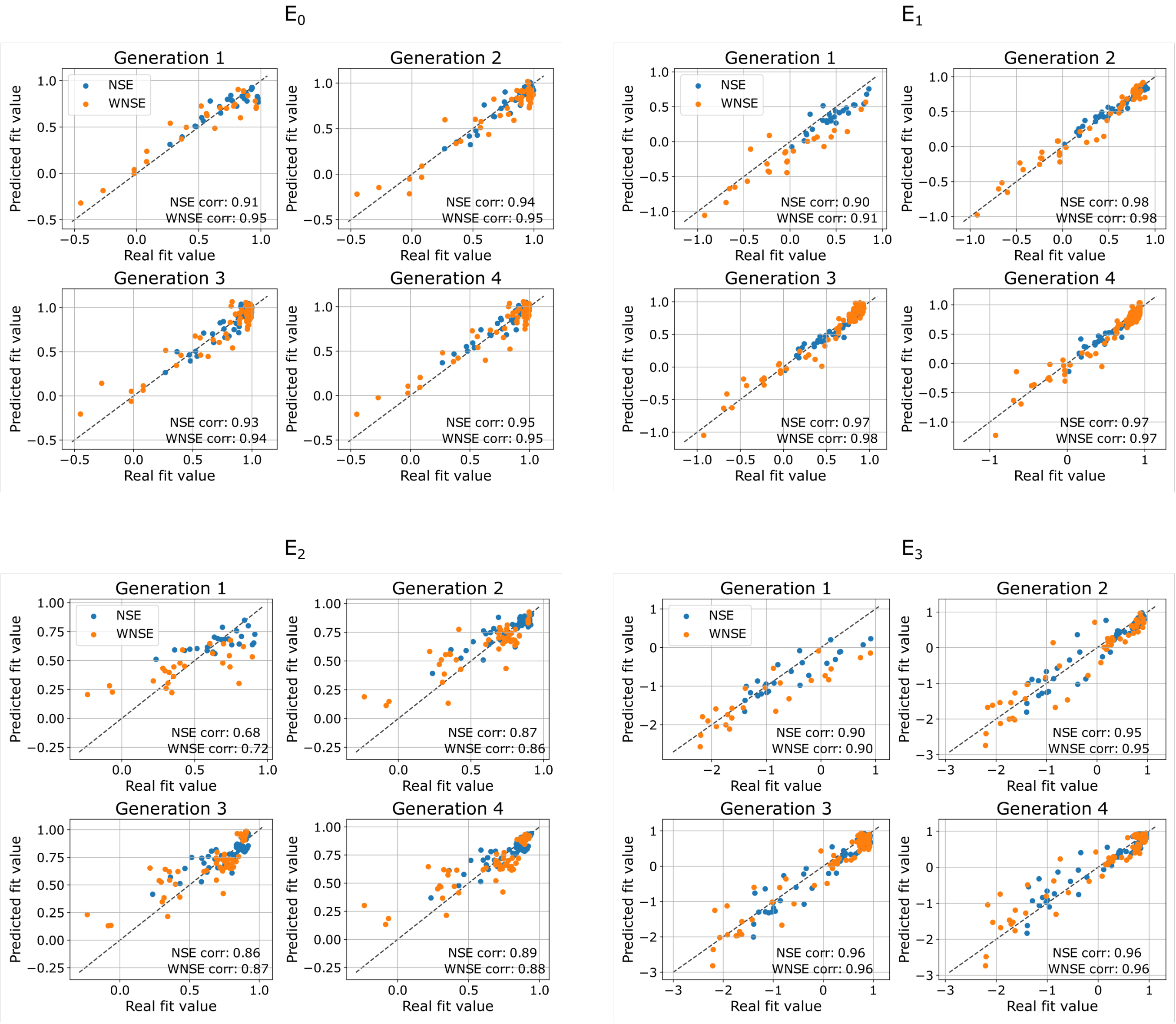

Figure 7. Scatter plots of the real vs. predicted fitness values in the training phase of the ANN for each generation of the SA-EA.

Furthermore, there is an improvement in correlation and fitting in relation to the identity line with each successive generation (Figure 7). In this regard, E$_2$ and E$_3$ exhibit a slightly worse correlation and fit than E$_0$ and E$_1$. Figure 7 reveals that the ANN is highly accurate in the zones where the highest values of NSE and WNSE (>0.90) are located. This accuracy can be attributed to the data fed to the model in each iteration, which is obtained by moving the search towards these zones through the EA. Thus, the ANN is highly accurate in the zones that can provide appropriate solutions for the calibration process. It is worth noting that a better fit for all feasible solutions could be achieved by training the model with a larger dataset, but it would significantly increase the computational time as the distributed model would need to run with parameter sets that do not provide essential information for the calibration process, as it is discussed later in Section 6.2. The primary objective of the proposed methodology is to optimize the distributed model while maintaining low the total computational time. Therefore, achieving a good fit for the high values through the ANN and the EA is the main concern.

Table 5. MSE values in the training phase of the ANN in each generation.

Event	Training Iteration	Population Size	NSE		WNSE	
			Training	Testing	Training	Testing
			MSE	MSE	MSE	MSE
E0	G1	25	0.007	0.006	0.018	0.016
	G2	50	0.005	0.005	0.015	0.013
	G3	75	0.004	0.003	0.012	0.010
	G4	100	0.002	0.002	0.008	0.006
E1	G1	25	0.020	0.019	0.066	0.066
	G2	50	0.003	0.003	0.012	0.010
	G3	75	0.003	0.003	0.010	0.009
	G4	100	0.004	0.004	0.014	0.012
E2	G1	25	0.017	0.016	0.046	0.041
	G2	50	0.007	0.006	0.022	0.017
	G3	75	0.006	0.005	0.018	0.014
	G4	100	0.004	0.002	0.014	0.008
E3	G1	25	0.128	0.116	0.276	0.251
	G2	50	0.068	0.053	0.128	0.101
	G3	75	0.055	0.047	0.090	0.074
	G4	100	0.042	0.030	0.079	0.053

6.2. Parameter Identification with the SA-EA Method

In the initial population, with 25 randomly generated members, the number of members that exceed a fit value of 0.9 is lower than or equal to 4 in the four events. The synthetic event (E_0) yielded the highest number of population members with a fitness value above 0.90. In this event, after the first generation all the population members yielded fit values greater than 0.9, which is probably related to the absence of uncertainty on the observed data in this particular synthetic case, making the identification of parameters much easier for the SA-EA. It is interesting to note that, even if the first ANN was trained with a small-sized training set (just 25 randomly generated members), the EA effectively guided the search towards parameter values that produce high-fit values (Table 6). Furthermore, the fit values continue to increase as the number of EA generations increases, as shown in Figure 8. In this instance, the SA-EA discovered population members with fitness values greater than 0.90 in 96 % of the cases for E_1 and 100 % for the remaining events.

In the case of the real events (E_1-E_3), the identification of suitable population members remains low in the first generation and moderate in the second generations. However, in the third and fourth generations almost all the population members achieve fit values higher than 0.90. This improvement is better shown in Figure 8, which shows that after generation G1 the goodness of fit obtained with the identified parameters improves considerably. It is also worth noting that, in most of the cases, the convergence of the SA-EA is achieved already in generation G3, in which between 90% and 100% of the identified members provide fitness values over 0.90.

Table 6. Number of population members identified by the SA-EA that presented a fit value > 0.90 after validation with Iber.

	$\mathbf{E_0}$	$\mathbf{E_1}$	$\mathbf{E_2}$	$\mathbf{E_3}$
Initial population	4/25	0/25	1/25	0/25
G1	25/25	1/25	12/25	3/25
G2	23/25	10/25	25/25	7/25
G3	25/25	25/25	25/25	20/25
G4	25/25	24/25	25/25	25/25

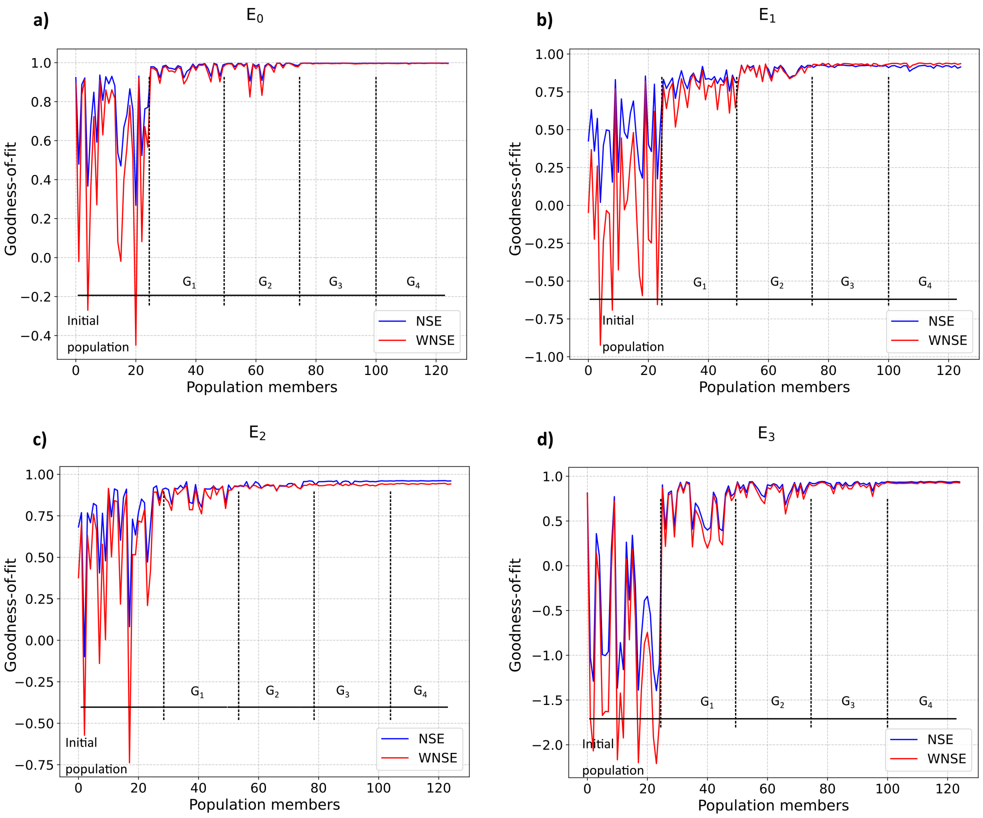

Figure 8. Evolution of the goodness-of-fit through generations (**a**) E_0, (**b**) E_1, (**c**) E_2, and (**d**) E_3.

6.3. CPU Time

The application of the SA-EA shows that, in order to obtain 25 population members at the end of each generation, the ANN performed the evaluation of 385 to 425 mutations to the replicated members depending on the event (Table 7). The time taken to complete all these evaluations was relatively low, with the duration ranging from 40 to 52 seconds. These evaluations are distributed among the 25 members of the initial state and are performed at a lower computational cost due to the parallel behavior of the ANN [15]. For instance, if the EA was applied without the assistance of the ANN, the real hydrological model would only be able to evaluate one mutated population member at a time, which includes the inherent evaluation of population members (i.e., parameter sets) with very poor fitness. When the number of model parameters is large and the computational cost of the numerical model is high (as in the present work), it would take several hours of computational time to identify only one suitable population member. Consequently, non-assisted evaluations of the EA become inefficient in the context of complex and computational demanding numerical models [15,22]. In those cases, a traditional Monte Carlo (MC) method might be a more appropriate alternative for performing comparisons with the proposed algorithm, as both methods perform global optimization.

The computation time needed by Iber to validate the 25 best population members ranged from 1.21 to 1.68 h, depending on the storm event and generation. The findings suggest that the computation time for evaluation of mutated members with the ANN was consistent across generations, while the duration of the Iber evaluation varied depending on the rainfall–runoff event.

Table 7. Performance of the SA-EA for each storm event: Number of evaluations performed with the ANN, time consumed by the ANN, and time consumed by Iber.

Generation	E_0			E_1			E_2			E_3		
	Evaluations with ANN	Time (s)	Evaluation with Iber (h)	Evaluations with ANN	Time (s)	Evaluation with Iber (h)	Evaluations with ANN	Time (s)	Evaluation with Iber (h)	Evaluations with ANN	Time (s)	Evaluation with Ibet (h)
G1	408	52	1.53	394	40	1.42	420	41	1.70	409	46	1.23
G2	425	50	1.53	400	50	1.43	412	41	1.65	424	54	1.22
G3	425	52	1.52	385	40	1.42	401	44	1.62	416	56	1.23
G4	403	50	1.50	347	47	1.41	416	38	1.62	412	49	1.28

Figure 9 shows, for each storm event, the total computation time required by the SA-EA and the MC approaches to calibrate the Iber model, together with the average fit value obtained with the 15 best population members (i.e., parameter sets) identified. The computation times and fit values are shown for the different generations (in the case of SA-EA) and for different number of MC simulations, in order to show the convergence rate of both methods. The results indicate that the SA-EA converges much faster than the Monte Carlo approach. This difference in computational time is explained by the fact that in the proposed approach the evaluation of poorly fitted members is carried out by the ANN. The ANN assists the EA in evaluating the quality of potential populations members, avoiding running the real model with lots of parameter sets that would generate poor fits to the observed data.

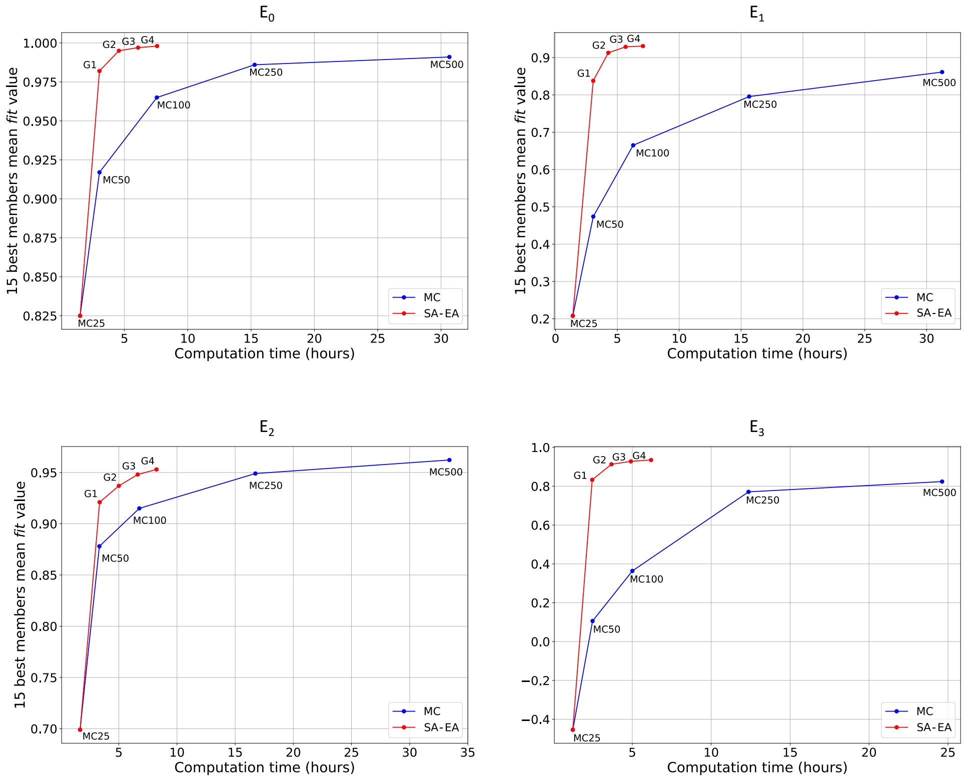

Figure 9. Cumulative computational times for convergence of the SA-EA and MC methods.

Moreover, in events E_0, E_1 and E_3, the convergence not only occurs faster but also reaches better fit values than the MC simulations. This demonstrates that the proposed approach can converge to optimal solutions in cases where global optimization methods have a slow convergence. In the case of the E_2 event, the Monte Carlo simulation with 500 parameter sets reached a slightly higher result than the SA-EA (0.96 vs. 0.95), but the computational time was 5 times higher. Although the SA-EA would probably outperform the Monte Carlo method in event E_2 by running an additional generation, the difference in results between the two methods can be considered negligible when compared to the cost of running another generation (which would take approximately 1.5 h).

6.4. Calibrated Model

The GLUE approach [49] was utilized in this study to evaluate the predictive capacity of the model by comparing its predictions with the observed hydrographs for each storm event. Therefore, the behavioral parameter sets were defined from those population members with $fit > 0.90$.

Figure 10 shows that the model was able to efficiently reproduce the synthetic and observed events. In the case of the synthetic event, the fit is nearly perfect, except for minor variations in certain areas of the hydrograph. This can be explained by the fact that simulations obtained using the parameter sets identified by the SA-EA method were averaged based on their fit values. Consequently, small deviations from the synthetic hydrograph may occur due to consideration of behavioral parameter sets that do not provide a perfect fit.

Regarding the optimized fitness values (Table 8), the best model performance is achieved in the synthetic event E_0, with NSE and WNSE values of 0.99, aligning with the simulated hydrograph displayed in Figure 10. This high fitness level is anticipated, given that the synthetic hydrograph is generated using the hydrological model to evaluate the effectiveness of the SA-EA approach in a controlled case. In contrast, all the real events exhibit fitness values exceeding 0.92, indicating a very good model performance in line with the guidelines suggested by [71].

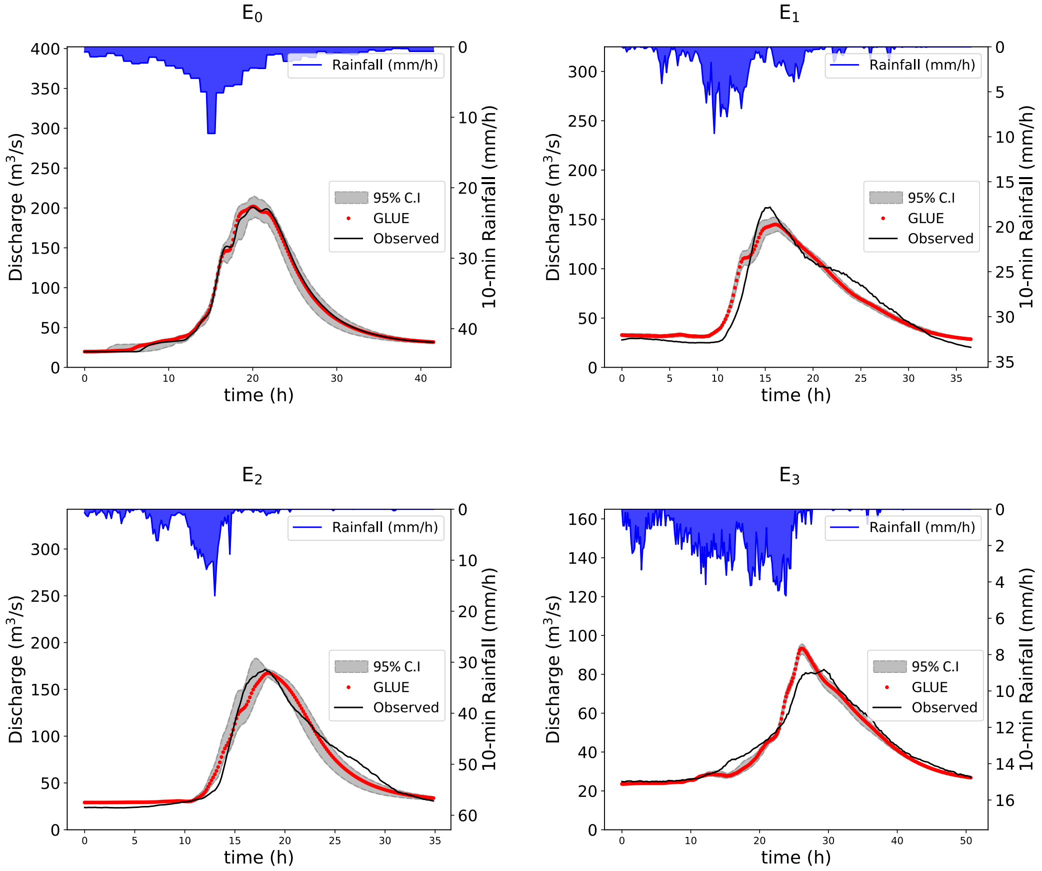

Figure 10. Observed and simulated hydrographs, including the 95% confidence interval estimated with GLUE.

Table 8. Average fitness obtained in the 4 storm events.

Event	NSE	WNSE	fit
E_0	0.99	0.99	0.99
E_1	0.92	0.93	0.93
E_2	0.94	0.93	0.93
E_3	0.93	0.92	0.93

The calibrated parameter values differ from one event to another, since antecedent soil moisture conditions significantly affect the potential infiltration and the peak discharge during a storm event, and consequently, the model parameters (table 9). Previous studies have reported that, in the study basin, the river discharges generated by storm events that take place over wet antecedent soil moisture conditions are, on average, three times larger than those occurring over normal antecedent moisture conditions. Taking into account the event-specific soil moisture conditions, or at least the potential interactions between soil moisture and rainfall intensity, is essential when estimating flood discharges.

It is noteworthy that a majority of the obtained roughness values are higher than those commonly used in river hydraulics. This is likely due to the presence of various types of vegetation and microtopography features that are not adequately resolved by the digital terrain model (DTM). Consequently, the effective roughness coefficient must account for all these unresolved features. Previous studies have reported Manning numbers significantly larger than those typically employed in river hydraulics applications, depending on factors such as vegetative cover, microtopography, rainfall intensity, and water depth [42,72]. These factors significantly influence calculations involving rainfall–runoff transformation over rough terrains.

Table 9. Mean and standard deviation of the behavioral parameter values obtained using the GLUE approach.

Event	Measure	Su_1	Por_1	Sat_1	Ks_1	$loss_1$	$depth_1$	Su_2	Por_2	Sat_2	Ks_2	$loss_2$	$depth_2$	n_1	n_2	n_{river}
E_0	mean	212.8	0.34	0.49	3.2	7.5	5.6	196.1	0.36	0.76	1.7	8.6	7.4	0.09	0.09	0.05
	std	33.6	0.05	0.07	0.4	2.8	1.3	31.6	0.03	0.22	0.3	2.0	0.9	0.04	0.02	0.03
E_1	mean	136.5	0.47	0.12	1.2	2.0	4.5	222.5	0.46	0.55	1.3	6.1	6.0	0.04	0.07	0.07
	std	9.1	0.01	0.02	0.0	1.0	0.9	19.9	0.01	0.08	0.0	0.3	0.6	0.00	0.00	0.01
E_2	mean	189.2	0.40	0.67	1.8	8.5	5.2	157.9	0.37	0.63	1.8	5.4	5.0	0.13	0.07	0.09
	std	18.3	0.05	0.14	0.2	1.1	1.2	27.1	0.02	0.10	0.4	1.6	1.6	0.01	0.03	0.01
E_3	mean	119.7	0.32	0.61	2.8	6.3	9.6	148.2	0.36	0.76	1.6	8.3	0.4	0.18	0.02	0.16
	std	8.5	0.00	0.02	0.2	0.5	0.0	7.7	0.00	0.01	0.1	1.3	0.0	0.00	0.00	0.00

7. Conclusions

This study aimed to develop an SA-EA for the calibration of distributed hydrological models based on the 2D shallow water equations. A surrogate model is used to simulate the solution space, drastically reducing the computational cost of running hundreds of simulations with the hydrological model, while the EA guides the search for optimal parameters. Training data for the surrogate model were incrementally collected in order to minimize data acquisition costs, ultimately enabling an accurate reproduction of the observed discharge series in the study catchment.

Our findings demonstrate that the SA-EA effectively calibrated the distributed hydrological model, while maintaining computational efficiency. The proposed algorithm was able to predict behavioral parameter sets, reducing execution times by five–six times compared to traditional global optimization techniques. This confirms its suitability as a calibration tool for distributed hydrological models.

The accuracy of the ANN in predicting NSE and WNSE improved with each generation of the EA, as more validated data were incorporated into the training set. The ANN achieves a high precision in regions with NSE and WNSE values higher than 0.90 for all the events, which proves its effectiveness in guiding the EA towards suitable regions of the solution space. For instance, the SA-EA identified parameter sets with fitness values higher than 0.90, achieving success rates between 90 % and 100 % by the third generation, demonstrating its efficiency and effectiveness in the calibration process. The SA-EA efficiently performs expensive simulations, achieving desired results more quickly and leveraging the exploration abilities of EAs for such optimization tasks. Without the SA-EA, these simulations would require solving the 2D-SWE thousands of times, making the application of the EA unfeasible.

Finally, while the present study has shown promising results, it is crucial to acknowledge other recent optimization methods, such as Bayesian optimization or gradient-based tools. Future research comparing these approaches with the SA-EA method could offer valuable insights into their respective strengths and limitations. These alternative methods require extensive development and evaluation before any meaningful comparison can be made. Therefore, they were not assessed in the current study, and future research should concentrate on their evaluation and comparison.

Author Contributions: Conceptualization, J.F.F.-D., A.H., T.D., I.C., and L.C.; methodology, J.F.F.-D.; software, J.F.F.-D. and A.H.; validation, J.F.F.-D.; investigation, J.F.F.-D.; data curation, J.F.F.-D.; writing—original draft preparation, J.F.F.-D.; writing—review and editing, J.F.F.-D., A.H., and L.C.; visualization, J.F.F.-D. and A.H.; supervision, L.C., T.D., and I.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the Galician Government (*Xunta de Galicia*) under its pre-doctoral fellowship program (*Axudas de apoio á etapa predoutoral 2019*), Register Nº ED481A-2019/014. Additional funding was provided by the INDITEX-UDC 2022 pre-doctoral stay aid program. The work also received support from the Flemish Government through the 'Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen' program and the 'Fonds Wetenschappelijk Onderzoek (FWO)'.

Data Availability Statement: The meteorological data were obtained from the agency MeteoGalicia. The discharge data have been provided by the regional water administration, *Augas de Galicia*. The scripts and specific events needed to recreate the results and figures of the present study are provided in the HydroShare repository [73].

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

References

1. Yang, T.; Hsu, K.; Duan, Q.; Sorooshian, S.; Wang, C. Method to estimate optimal parameters. In *Handbook of Hydrometeorological Ensemble Forecasting*; Springer: Berlin/Heidelberg, Germany, 2018.
2. Farfán, J.F.; Cea, L. Improving the predictive skills of hydrological models using a combinatorial optimization algorithm and artificial neural networks. *Model. Earth Syst. Environ.* **2022**, *9*, 1103–1118.
3. Wang, G.G.; Shan, S. Review of Metamodeling Techniques in Support of Engineering Design Optimization. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Philadelphia, PA, USA, 10–13 September 2006; Volume 4255, pp. 415–426.
4. Mediero Orduña, L.J. Pronóstico Probabilístico de Caudales de Avenida Mediante Redes Bayesianas Aplicadas Sobre un Modelo Hidrológico Distribuido. Ph.D. Thesis, Caminos, Universidad Politécnica de Madrid, Madrid, Spain, 2007.
5. Kampf, S.K.; Burges, S.J. A framework for classifying and comparing distributed hillslope and catchment hydrologic models. *Water Resour. Res.* **2007**, *43*.
6. Yoosefdoost, I.; Bozorg-Haddad, O.; Singh, V.P.; Chau, K.W. Hydrological Models. In *Climate Change in Sustainable Water Resources Management*; Springer: 2022; pp. 283–329.
7. Kavvas, M.; Chen, Z.; Dogrul, C.; Yoon, J.; Ohara, N.; Liang, L.; Aksoy, H.; Anderson, M.; Yoshitani, J.; Fukami, K.; et al. Watershed environmental hydrology (WEHY) model based on upscaled conservation equations: Hydrologic module. *J. Hydrol. Eng.* **2004**, *9*, 450–464.
8. Zanchetta, A.D.; Coulibaly, P. Hybrid Surrogate Model for Timely Prediction of Flash Flood Inundation Maps Caused by Rapid River Overflow. *Forecasting* **2022**, *4*, 126–148.
9. Gu, H.; Xu, Y.P.; Ma, D.; Xie, J.; Liu, L.; Bai, Z. A surrogate model for the Variable Infiltration Capacity model using deep learning artificial neural network. *J. Hydrol.* **2020**, *588*, 125019.
10. Mo, K.C.; Lettenmaier, D.P. Hydrologic prediction over the conterminous United States using the national multi-model ensemble. *J. Hydrometeorol.* **2014**, *15*, 1457–1472.

11. Schumann, G.P.; Neal, J.C.; Voisin, N.; Andreadis, K.M.; Pappenberger, F.; Phanthuwongpakdee, N.; Hall, A.C.; Bates, P.D. A first large-scale flood inundation forecasting model. *Water Resour. Res.* **2013**, *49*, 6248–6257.
12. Wang, Z.; Zhong, R.; Lai, C.; Zeng, Z.; Lian, Y.; Bai, X. Climate change enhances the severity and variability of drought in the Pearl River Basin in South China in the 21st century. *Agric. For. Meteorol.* **2018**, *249*, 149–162.
13. Sridhar, V.; Ali, S.A.; Lakshmi, V. Assessment and validation of total water storage in the Chesapeake Bay watershed using GRACE. *J. Hydrol. Reg. Stud.* **2019**, *24*, 100607.
14. Wang, K.; Shi, H.; Chen, J.; Li, T. An improved operation-based reservoir scheme integrated with Variable Infiltration Capacity model for multiyear and multipurpose reservoirs. *J. Hydrol.* **2019**, *571*, 365–375.
15. Farfán, J.F.; Cea, L. Coupling artificial neural networks with the artificial bee colony algorithm for global calibration of hydrological models. *Neural Comput. Appl.* **2021**, *33*, 8479–8494.
16. Kavetski, D. Parameter estimation and predictive uncertainty quantification in hydrological modelling. In *Handbook of Hydrometeorological Ensemble Forecasting*; Springer: Berlin/Heidelberg, Germany, 2018.
17. Li, Z.; Liu, P.; Deng, C.; Guo, S.; He, P.; Wang, C. Evaluation of estimation of distribution algorithm to calibrate computationally intensive hydrologic model. *J. Hydrol. Eng.* **2016**, *21*, 04016012.
18. Bermúdez, M.; Cea, L.; Puertas, J. A rapid flood inundation model for hazard mapping based on least squares support vector machine regression. *J. Flood Risk Manag.* **2019**, *12*, e12522.
19. Madsen, H. Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. *J. Hydrol.* **2000**, *235*, 276–288.
20. Pétrowski, A.; Ben-Hamida, S. *Evolutionary Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
21. Sloss, A.N.; Gustafson, S. 2019 evolutionary algorithms review. In *Genetic Programming Theory and Practice XVII*; Springer: Singapore, 2020; pp. 307–344.
22. Zhao, J.; Lv, L.; Sun, H. Artificial bee colony using opposition-based learning. In *Genetic and Evolutionary Computing*; Springer: 2015; pp. 3–10.
23. Razavi, S.; Tolson, B.A.; Burn, D.H. Review of surrogate modeling in water resources. *Water Resour. Res.* **2012**, *48*.
24. Mugunthan, P.; Shoemaker, C.A. Assessing the impacts of parameter uncertainty for computationally expensive groundwater models. *Water Resour. Res.* **2006**, *42*.
25. Khu, S. A fast evolutionary-based meta-modelling approach for the calibration of a rainfall–runoff model. In Proceedings of the 1st Biennial Meeting of the International Environmental Modelling and Software Society, iEMSs, Lugano, Switzerland, 24–27 June 2002; Volume 1, pp. 147–152.
26. Shoemaker, C.A.; Regis, R.G.; Fleming, R.C. Watershed calibration using multistart local optimization and evolutionary optimization with radial basis function approximation. *Hydrol. Sci. J.* **2007**, *52*, 450–465.
27. Sopelana, J.; Cea, L.; Ruano, S. A continuous simulation approach for the estimation of extreme flood inundation in coastal river reaches affected by meso-and macrotides. *Nat. Hazards* **2018**, *93*, 1337–1358.
28. Zhang, X.; Srinivasan, R.; Van Liew, M. Approximating SWAT model using artificial neural network and support vector machine 1. *JAWRA J. Am. Water Resour. Assoc.* **2009**, *45*, 460–474.
29. Chu, H.; Wu, W.; Wang, Q.; Nathan, R.; Wei, J. An ANN-based emulation modelling framework for flood inundation modelling: Application, challenges and future directions. *Environ. Model. Softw.* **2020**, *124*, 104587.
30. Shaw, A.R.; Smith Sawyer, H.; LeBoeuf, E.J.; McDonald, M.P.; Hadjerioua, B. Hydropower Optimization Using Artificial Neural Network Surrogate Models of a High-Fidelity Hydrodynamics and Water Quality Model. *Water Resour. Res.* **2017**, *53*, 9444–9461.
31. Shrestha, D.; Kayastha, N.; Solomatine, D. A novel approach to parameter uncertainty analysis of hydrological models using neural networks. *Hydrol. Earth Syst. Sci.* **2009**, *13*, 1235–1248.
32. Liong, S.Y.; Khu, S.T.; Chan, W.T. Derivation of Pareto front with genetic algorithm and neural network. *J. Hydrol. Eng.* **2001**, *6*, 52–61.
33. Yan, X.; Mohammadian, A.; Ao, R.; Liu, J.; Yang, N. Two-dimensional convolutional neural network outperforms other machine learning architectures for water depth surrogate modeling. *J. Hydrol.* **2023**, *616*, 128812.
34. Wu, M.; Wang, L.; Xu, J.; Wang, Z.; Hu, P.; Tang, H. Multiobjective ensemble surrogate-based optimization algorithm for groundwater optimization designs. *J. Hydrol.* **2022**, *612*, 128159.
35. Gorissen, D.; Couckuyt, I.; Demeester, P.; Dhaene, T.; Crombecq, K. A surrogate modeling and adaptive sampling toolbox for computer based design. *J. Mach. Learn. Res.* **2010**, *11*, 2051–2055.
36. Liang, J.; Melching, C.S. Experimental evaluation of the effect of storm movement on peak discharge. *Int. J. Sediment Res.* **2015**, *30*, 167–177.
37. Cabalar Fuentes, M. Los temporales de lluvia y viento en Galicia. Propuesta de clasificación y análisis de tendencias (1961-2001). *Investig. Geogr.* **2005**, 103–118.
38. Cea, L.; Bladé, E. A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water Resour. Res.* **2015**, *51*, 5464–5486.
39. García-Feal, O.; González-Cao, J.; Gómez-Gesteira, M.; Cea, L.; Domínguez, J.M.; Formella, A. An accelerated tool for flood modelling based on Iber. *Water* **2018**, *10*, 1459.

40. Cea, L.; Álvarez, M.; Puertas, J. Estimation of flood-exposed population in data-scarce regions combining satellite imagery and high resolution hydrological-hydraulic modelling: A case study in the Licungo basin (Mozambique). *J. Hydrol. Reg. Stud.* **2022**, *44*, 101247.
41. García-Alén, G.; Hostache, R.; Cea, L.; Puertas, J. Joint assimilation of satellite soil moisture and streamflow data for the hydrological application of a two-dimensional shallow water model. *J. Hydrol.* **2023**, *621*, 129667.
42. Sanz-Ramos, M.; Bladé, E.; González-Escalona, F.; Olivares, G.; Aragón-Hernández, J.L. Interpreting the manning roughness coefficient in overland flow simulations with coupled hydrological-hydraulic distributed models. *Water* **2021**, *13*, 3433.
43. Green, W.H.; Ampt, G. Studies on Soil Phyics. *J. Agric. Sci.* **1911**, *4*, 1–24.
44. USDA, N.R.C.S. *Soil Taxonomy: A Basic System of Soil Classification for Making and Interpreting Soil Surveys*; Number 436 in 1; US Department of Agriculture: Washington, DC, USA, 1975.
45. Maidment, D. *Handbook of Hydrology*; McGraw-Hill Education: New York, NY, USA, 1993.
46. Te Chow, V.; Maidment, D.R.; Mays, L.W. Applied hydrology. *J. Eng. Educ.* **1962**, *308*, 1959.
47. Samuels, P.; Bramley, M.; Evans, E. Reducing uncertainty in conveyance estimation. In Proceedings of the International Conference on Fluvial Hydraulics (River Flow 2002), Louvain-la-Neuve, Belgium, 4–6 September 2002.
48. Nash, J.E.; Sutcliffe, J.V. River flow forecasting through conceptual models part I—A discussion of principles. *J. Hydrol.* **1970**, *10*, 282–290.
49. Beven, K.; Binley, A. The future of distributed models: Model calibration and uncertainty prediction. *Hydrol. Process.* **1992**, *6*, 279–298.
50. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133.
51. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep learning*; MIT Press: Cambridge, MA, USA, 2016.
52. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
53. Kirkpatrick, S. Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.* **1984**, *34*, 975–986.
54. Granville, V.; Krivánek, M.; Rasson, J.P. Simulated annealing: A proof of convergence. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 652–656.
55. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
56. Wang, Q. The genetic algorithm and its application to calibrating conceptual rainfall–runoff models. *Water Resour. Res.* **1991**, *27*, 2467–2471.
57. Duan, Q.; Sorooshian, S.; Gupta, V. Effective and efficient global optimization for conceptual rainfall–runoff models. *Water Resour. Res.* **1992**, *28*, 1015–1031.
58. Duan, Q.; Sorooshian, S.; Gupta, V.K. Optimal use of the SCE-UA global optimization method for calibrating watershed models. *J. Hydrol.* **1994**, *158*, 265–284.
59. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471.
60. Audze, P. New approach to planning out of experiments. *Probl. Dyn. Strengths* **1977**, *35*, 104–107.
61. McKay, M.D.; Beckman, R.J.; Conover, W.J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **2000**, *42*, 55–61.
62. Karaboga, D.; Akay, B. A comparative study of artificial bee colony algorithm. *Appl. Math. Comput.* **2009**, *214*, 108–132.
63. Zhao, L.; Hu, Y.; Wang, B.; Jiang, X.; Liu, C.; Zheng, C. A surrogate-assisted evolutionary algorithm based on multi-population clustering and prediction for solving computationally expensive dynamic optimization problems. *Expert Syst. Appl.* **2023**, *223*, 119815.
64. Espinosa, R.; Jiménez, F.; Palma, J. Multi-surrogate assisted multi-objective evolutionary algorithms for feature selection in regression and classification problems with time series data. *Inf. Sci.* **2023**, *622*, 1064–1091.
65. Wang, H.; Jin, Y.; Sun, C.; Doherty, J. Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Trans. Evol. Comput.* **2018**, *23*, 203–216.
66. Kim, S.H.; Boukouvala, F. Machine learning-based surrogate modeling for data-driven optimization: A comparison of subset selection for regression techniques. *Optim. Lett.* **2020**, *14*, 989–1010.
67. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
68. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
69. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
70. Patro, S.; Sahu, K.K. Normalization: A preprocessing stage. *arXiv* **2015**, arXiv:1503.06462.
71. Ritter, A.; Munoz-Carpena, R. Performance evaluation of hydrological models: Statistical significance for reducing subjectivity in goodness-of-fit assessments. *J. Hydrol.* **2013**, *480*, 33–45.
72. Fraga, I.; Cea, L.; Puertas, J. Experimental study of the water depth and rainfall intensity effects on the bed roughness coefficient used in distributed urban drainage models. *J. Hydrol.* **2013**, *505*, 266–275.
73. Farfán-Durán, J.F.; Cea, L. *Visualizing the Results of the Calibration of the Distributed Hydrological Model Iber+ with the Surrogate-Assisted Evolutionary Algorithm*; HydroShare: 2023.

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.