

## Comparative performance evaluation of self-adaptive differential evolution with GA, SCE and DE algorithms for the automatic calibration of a computationally intensive distributed hydrological model

Saswata Nandi<sup>a</sup> and M. Janga Reddy<sup>a,b,\*</sup>

<sup>a</sup> Department of Civil Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India

<sup>b</sup> Interdisciplinary Programme in Climate Studies, Indian Institute of Technology Bombay, Mumbai, India

\*Corresponding author. E-mail: mjreddy@civil.iitb.ac.in

### Abstract

Recently, physically-based hydrological models have been gaining much popularity in various activities of water resources planning and management, such as assessment of basin water availability, floods, droughts, and reservoir operation. Every hydrological model contains some parameters that must be tuned to the catchment being studied to obtain reliable estimates from the model. This study evaluated the performance of different evolutionary algorithms, namely genetic algorithm (GA), shuffled complex evolution (SCE), differential evolution (DE), and self-adaptive differential evolution (SaDE) algorithm for the parameter calibration of a computationally intensive distributed hydrological model, variable infiltration capacity (VIC) model. The methodology applied and tested for a case study of the upper Tungabhadra River basin in India and the performance of the algorithms is evaluated in terms of reliability, variability, efficacy measures in a limited number of function evaluations, their ability for achieving global convergence, and also by their capability to produce a skilful simulation of streamflows. The results of the study indicated that SaDE facilitates an effective calibration of the VIC model with higher reliability and faster convergence to optimal solutions as compared to the other methods. Moreover, due to the simplicity of the SaDE, it provides easy implementation and flexibility for the automatic calibration of complex hydrological models.

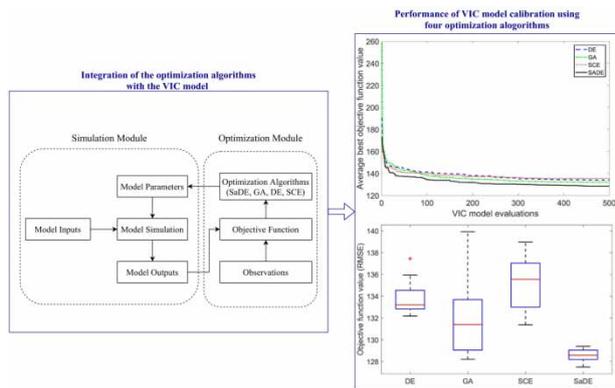
**Key words:** differential evolution, genetic algorithm, parameter estimation, self-adaptive differential evolution, shuffled complex evolution, VIC model

### Highlights

- Calibration of a computationally intensive distributed hydrologic model is performed.
- Evaluated the performance of different evolutionary algorithms for automatic calibration of VIC model.
- Self-adaptive scheme avoids tuning of the algorithm parameters.
- SaDE method is capable of finding the optimal parameters quickly and gives superior performance over GA and SCE algorithms.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

## Graphical Abstract



## INTRODUCTION

Hydrological models are prevalent tools for representing the hydrological processes of a catchment in mathematical form. Consequently, they are widely used in various water resources management studies such as flood forecasting (Refsgaard *et al.* 1988), drought risk assessment (Sheffield & Wood 2008), integrated river basin management (Cai *et al.* 2003), analyzing impacts of climate change and anthropogenic activities on water resources (Arnell 1999), etc. These hydrological models are broadly classified into conceptual and physically-based models with reference to the interpretations of the dominant physical processes at a catchment. According to the spatial representation of catchment, they are further categorized into lumped, semi-distributed, and distributed models. Physically-based distributed hydrological models (PBDHM) are considered to enhance the hydrological simulation of a watershed by incorporating spatial heterogeneity of catchment variables such as land uses, soil types, topographic features, and meteorological forcing (Chen *et al.* 2016). However, the operational use of PBDHMs is complex and also requires greater computational resources even for a smaller catchment, as it uses different model parameters for its various spatially distinct computational units (Chen *et al.* 2016).

However, regardless of a physical or conceptual model, practically all hydrological models include many parameters, and it cannot be possible to measure all of them in the field directly. Therefore, an effective application of these models requires proper model calibration, which is related to the adjustment of the model parameters to fit the simulated results with the observation data (Tolson & Shoemaker 2007). The calibration of hydrological models can be performed either manually or automatically (Kim *et al.* 2007). Manual calibration involves a trial and error procedure for adjustments of the model parameters, but such calibration is time-consuming, tedious and its efficacy relies on the subjective judgement of a modeller, which includes knowledge of the model and its interaction with the participating watershed (Kim *et al.* 2007). On the contrary, automatic calibration is quick, less subjective, and more efficient for finding the optimal parameter set than that of manual calibration (Kamali *et al.* 2013). With the advancement of computational resources, many studies have chosen automatic calibration over the manual calibration approach for hydrological model applications (Wang 1991; Duan *et al.* 1992). The automatic calibration approach uses an optimization algorithm, which may be either a local or global optimization method, to explore the parameter space according to some specified objective function, which refers to the disparity between the observation and model simulation (Madsen 2000). Local optimization methods were used in the initial studies of automatic calibration for hydrological modelling (Ibbitt & O'Donnell 1971; Sorooshian & Gupta 1983). However, the drawback of such local search methods is that they may get stuck in

the local optima and never reach the global optimal solution as there may be several local optima for the calibration problem of a hydrological model (Gan & Biftu 1996). Consequently, the overall performance of any local search method mostly depends on its initial solution.

To address such limitations, recent studies have inclined toward the utilization of global search methods for hydrological model calibration. In hydrological modelling, the most commonly used population-based evolutionary algorithms (EAs) are genetic algorithm (GA) (Wang 1991), shuffled complex evolution (SCE) algorithm (Duan *et al.* 1992), particle swarm optimization (PSO) (Scheerlinck *et al.* 2009), and differential evolution (DE) (Guo *et al.* 2014). Even though it is well-known that due to the issue of equifinality, i.e. multiple parameter values leading to the same model efficiency (Beven & Freer 2001), the obtained solution from these algorithms may strongly depend on the model representation, and chosen objective function (Scheerlinck *et al.* 2009; Osuch *et al.* 2015). The GA is one of the earliest evolutionary search based techniques that has been successfully applied for calibration of hydrological models (Wang 1991; Liong *et al.* 1995). Also, few studies on hydrological model calibration have highlighted the dominant performance of the SCE algorithm over other concurrent methods. For example, Arsenault *et al.* (2014) compared several optimization algorithms for three hydrological models in multiple basins over the United States and noted the SCE algorithm would be the most preferred choice, as it outperformed the GA in both convergence speed and computing power. However, the majority of the studies were conducted on simple conceptual hydrological models, which require hardly a few seconds for a model run. Consequently, those studies took advantage of utilizing a large number of function evaluations for model calibration with EAs. For example, Sorooshian *et al.* (1993) noted the convergence of 13 parameter calibration problem for the Sacramento Soil Moisture Accounting (SAC-SMA) model after 23,024 function evaluations, and Gan & Biftu (1996) reported that the EAs required an average of 10,818 function evaluations for estimating 13 parameters of SAC-SMA in eight catchments across the globe; and Qin *et al.* (2016) accomplished the calibration of seven parameters of the Simplified HYDROLOG (SIMHYD) model with an average of 11,153 function evaluations. In the case of modern complex hydrological models, such a large number of function evaluations demands much higher computational resources and time, which makes their calibration unreasonable. For instance, Tolson & Shoemaker (2007) noticed that the calibration of the SWAT model using SCE took a computational time of 14 and 140 days for 10,000 and 100,000 SWAT model evaluations, respectively.

To deal with such issues, there is a need to focus on improving the performance of the algorithms to achieve faster convergence to the optimal solution. In this direction, few optimization algorithms were applied to achieve faster convergence than the others, but the optimization algorithms require sensitivity analysis and fine-tuning of the several algorithm parameters to achieve their best performance. Furthermore, the performance of most of the EAs also depends on the complexity of the problem, which may require huge computational time for a large number of decision variables. Therefore, for the calibration of complex hydrological models, it is desirable to develop an algorithm that would demand nominal user configuration and yield optimal (or near-optimal) solutions with a lesser number of function evaluations (Vrugt *et al.* 2009; Zheng *et al.* 2012).

The primary focus of this study is to present an efficient methodology for automatic calibration of a physically-based complex distributed hydrological model, which will require minimal user configuration and a lesser number of function evaluations. This would facilitate efficient utilization of hydro-meteorological data, which in turn could be extracted by well-calibrated hydrological models. For this purpose, a self-adaptive differential evolution (SaDE) algorithm is presented. The DE algorithm is a global optimization technique proposed by Storn & Price (1997). DE is easy to implement and is reported to have a better (or comparable) performance with respect to other EAs (Reddy & Kumar 2007). However, like other EAs, the performance of the DE is also dependent on the setting of its control parameters, and this may result in a huge computational burden (Zheng *et al.* 2012). The SaDE algorithm is thus an improvement over the basic DE in this regard, where

two control parameters of the algorithm are continuously updated during the search process. A few recent studies have also found that self-adaptive variants of DE outperformed other EAs, including GAs and basic DEs, in terms of faster convergence and a higher success rate (Schardong & Simonovic 2015; Pulluri *et al.* 2017; Sirsant & Reddy 2018). However, only a few studies used the self-adaptive DE method for water resources problems, like Zheng *et al.* (2012) used for the optimal design of four water distribution networks; Sirsant & Reddy (2018) applied DE and SaDE methods for the reliability-based design of water distribution networks and noted that SaDE provided quick optimal solutions consistently and performed better than DE, particularly for problems with a large number of decision variables. With this view, the present study examines the potential of the SaDE algorithm for calibration of a complex distributed hydrological model, namely the variable infiltration capacity (VIC) model.

In this study, a three-layer VIC hydrological model (Liang *et al.* 1994) is chosen to simulate the rainfall-runoff process at a medium-size catchment located in the state of Karnataka, India. The VIC model is a physically-based hydrological model, which can simulate most of the important hydrological processes in a catchment. Previously, the VIC model was successfully applied for water resources assessment studies (Schumann *et al.* 2013). The VIC model has several parameters that require rigorous calibration before its application to a catchment for hydrological studies. The calibration of these parameters is a challenging task and requires much computational time and a large number of VIC model simulations. This study presents an effective methodology for automatic calibration of the VIC hydrological model using the SaDE algorithm, and its performance is evaluated by comparing it with the GA, SCE and DE algorithms to verify the competence of the methodology in terms of reliability, variability, and efficiency. The specific objectives of the present study include: (i) to present SaDE based methodology for automatic calibration of the VIC hydrological model; (ii) to evaluate the performance of the SaDE by comparing with the results of the GA, SCE and DE algorithms.

The rest of the paper is organized as follows. The following section presents details of methods used in the study viz., details of the optimization algorithms, and VIC hydrological model. Next, the application of the SaDE method for automatic calibration of the VIC model for the upper Tungabhadra River basin is discussed. The results of the VIC model for hydrological simulations and performance evaluation of the SaDE method by comparing them with the results of GA, SCE and DE methods are presented in the following section. Finally, the conclusions of the study are presented.

## METHODOLOGY

### VIC hydrological model

In this study, the variable infiltration capacity (VIC) model, a physically-based distributed hydrological model, is selected for simulation of hydrological variables from the basin. The VIC model considers the interaction between the biosphere, vegetation, atmosphere, and soil dynamics for the estimation of water and energy fluxes. In contrast with other comparable hydrological models, the exclusive features of the VIC model are the representation of subgrid variation in vegetation within a single grid, consideration of variable infiltration among multiple soil layers, and estimation of non-linear baseflow from bottom soil layer (Liang *et al.* 1994).

The Penman–Monteith method and the Horton infiltration curve are used by the model for estimation of evapotranspiration and subgrid variation of soil infiltration capacity, respectively (Liang *et al.* 1994). Total runoff estimation in the model comprises direct runoff (surface flow) and base flow (subsurface flow). Direct runoff is generated from the top two layers by assessing the saturation remainder flow and infiltration surplus flow from those two layers. Baseflow contributed from the bottom-most layer only (Franchini & Pacciani 1991) and is computed according to the ARNO model (ARNO refers to a semi-distributed conceptual rainfall-runoff model, which derives its name

from its first application to the Arno River, [Todini \(1996\)](#)). In the VIC model, direct runoff and baseflow are mathematically expressed as:

$$Q_d = \begin{cases} P + W - W^{\max}, & P + i_0 \geq i_m \\ P + W - W^{\max} \cdot \left[ 1 - \left( 1 - \frac{i_0 + P}{i_m} \right)^{1+b_i} \right], & P + i_0 \leq i_m \end{cases} \quad (1)$$

$$Q_b = \begin{cases} \frac{D_s D_m}{W_s W_3^{\max}} \cdot W_3, & 0 \leq W_3 \leq W_s W_3^{\max} \\ \frac{D_s D_m}{W_s W_3^{\max}} \cdot W_3 + \left( D_m - \frac{D_s D_m}{W_s} \right) \left( \frac{W_3 - W_s W_3^{\max}}{W_3^{\max} - W_s W_3^{\max}} \right)^2, & W_3 > W_s W_3^{\max} \end{cases} \quad (2)$$

where  $Q_d$  is the direct runoff (mm),  $P$  is the precipitation (mm),  $W$  and  $W^{\max}$  are the initial and maximum soil moisture for the upper soil layers (mm),  $i_0$  and  $i_m$  (mm) are the initial and maximum infiltration rate (mm),  $b_i$  is a parameter controlling the shape of the variable infiltration curve (unitless),  $Q_b$  denotes the base flow (mm),  $D_m$  represents maximum daily base flow (mm),  $D_s$  is a fraction (unitless) of  $D_m$  whereby nonlinear baseflow begins,  $W_3$  and  $W_3^{\max}$  are initial and maximum soil moisture at the bottom soil layer (mm), and  $W_s$  is the fraction (unitless) of  $W_3^{\max}$  where nonlinear baseflow begins.

The VIC model produces the runoff individually in model grid cells. Therefore, a separate routing model is selected to transfer simulated runoff from the individual model grids to the outlet of the study area. This study employed the routing method of [Nijssen \*et al.\* \(1997\)](#), which uses the unit hydrograph theory for transporting the flow inside a grid to the outlet of that grid and linearized Saint-Venant equations for the channel routing of the flow from the grid outlet to the basin outlet through the river network.

### Evolutionary algorithms

This study uses the GA, SCE, DE, and SaDE algorithms for calibration of the parameters of the VIC hydrological model. Brief details of these EAs are given below.

#### Genetic algorithm

Genetic algorithms (GA) are population-based heuristic search algorithms inspired by natural selection and survival of the fittest concept ([Goldberg 1989](#)). The GA algorithm starts by initializing a population of candidate solutions that have to be drawn randomly from a feasible parameter space. The individual members are chosen in each generation by fitness values, in which better fit chromosomes in the population are chosen to replicate new promising offspring. Next, the new population is produced by applying crossover and mutation operations to those selected members. The crossover operator selects parent solutions to generate new offspring solutions by sharing important characteristics of both parent solutions. In order to increase the diversity of the new population, the offspring solutions are randomly mutated. In this way, the generation of new candidate solutions is repeated until the stopping criterion is satisfied. More details of the algorithm can be found in [Goldberg \(1989\)](#) and [Reddy & Kumar \(2012\)](#). The pseudo-code of the GA algorithm is given in Appendix A.1.

#### Shuffled complex evolution algorithm

The shuffled complex evolution (SCE) algorithm was proposed as a global optimization technique for parameter estimation of hydrological models ([Duan \*et al.\* 1992](#)). The SCE algorithm begins by randomly initializing the population within its feasible range and then partitioning the population into different communities (or complexes). All such communities are allowed to evolve (generate

offspring) freely for guiding the search in a reformed direction. All the members of a particular community have the chance to take part in the offspring generation. However, one better member of the community has a higher possibility of getting engaged in the generation of the offspring, which ensures the competitiveness of evolution. After a few generations, the entire population is shuffled, and new communities are formed. This process helps in sharing information between various communities to explore the search space thoroughly. The pseudo-code of the SCE-UA algorithm is given in Appendix A.2. For more details of the method, one can refer to [Duan \*et al.\* \(1992\)](#) and [Vrugt \*et al.\* \(2003\)](#).

### Differential evolution

Differential evolution (DE) is one of the widely used population-based global optimization algorithms which is based on the process of natural evolution and has outperformed many other EAs ([Storn & Price 1997](#)). The unique trial vector generation strategy (such as mutation and crossover operation) distinguished DE from other EAs, although it has similar basic operations (mutation, crossover, and selection) to the other EAs. In DE, the mutation operator is applied first to produce the mutant vectors by modifying the initial vectors, and then the crossover operator is employed to those mutant vectors to generate the trial vectors. Finally, the selection operator is used to select the trial vectors which will survive to the next iteration. DE is an iterative procedure; after the initialization, it enters into an iterative loop of mutation, crossover, and selection operations and continues until the terminal condition is satisfied. Among the various operators, the mutation is the most important operator in DE and is highly dependent on the mutation factor (F). The crossover rate (CR) in the crossover operation greatly influences the generation of the trial vector and the convergence characteristics of the algorithm. Therefore, the performance of the DE algorithm is more reliant on the choice of F and CR values. The pseudo-code of the DE algorithm is given in Appendix A.3.

Moreover, like the GA and SCE algorithms, the performance of DE is also highly dependent on the chosen parameter settings, and their incorrect selection may lead to the poor performance of the algorithm ([Das \*et al.\* 2016](#)). So, proper sensitivity analysis is required before its application to a given problem.

### Self-adaptive differential evolution algorithm

The self-adaptive differential evolution (SaDE) algorithm abstains from the substantial computational costs of traditional DE for finding the appropriate control parameters of the algorithm. Here, the control parameters are progressively self-adapted by means of their past involvement in producing favourable solutions. This study adopts the SaDE algorithm with changes to the parameter selection of the crossover rate (CR) and the mutation factor (F) in contrast to the one presented in [Qin \*et al.\* \(2009\)](#). Here, the mean value of the F and CR parameters are updated after every ten generations based on the successful values of the parameters over the last ten iterations, and then a new set of mutation and crossover rates are generated for each individual using these updated mean values. More details about the steps involved in the SaDE methodology are explained below.

1. Generate initial population  $X_i^G = \{x_{i,1}^G, x_{i,2}^G, \dots, x_{i,D}^G\}$  by randomly choosing individuals within the search space. Here  $x_{i,j}^G$  represents the initial value of the  $j$ th parameter in the  $i$ th individual at the initial population,  $G$  represents the current generation number, and  $D$  denotes the dimension of the problem.
2. Initial values of mutation factor  $F$  and crossover rate  $CR$  are randomly generated using their specified mean and standard deviation values as follows:

$$\left. \begin{aligned} F_i &= \mu_F + \text{Rand}n_1 \times \sigma_F \\ CR_i &= \mu_{CR} + \text{Rand}n_2 \times \sigma_{CR} \end{aligned} \right\} \quad (3)$$

where  $F_i \in [0, 2]$  and  $CR_i \in [0, 1]$  are the initial values of mutation factor and crossover rate for the  $i$ th population;  $randn_1$  and  $randn_2$  are random numbers generated from the standard normal distribution;  $\mu_F$  and  $\sigma_F$  are the mean and standard deviation of the mutation factor;  $\mu_{CR}$  and  $\sigma_{CR}$  are mean and standard deviation of the crossover rate. The mean and standard deviation values are updated based on the following recommendation from Qin *et al.* (2009) and also after performing initial sensitivity analysis, the  $\mu_F$  and  $\sigma_F$  are initialized with values of 0.5 and 0.3, while  $\mu_{CR}$  and  $\sigma_{CR}$  are set at values of 0.5 and 0.1, respectively.

3. Calculate the objective function values for each member of the population in the current generation.
4. Mutation is performed to every individual ( $X_i^G$ ) in the population. These  $X_i^G$  are considered as the target vector, and their corresponding mutant vectors  $V_i^G = \{V_{i,1}^G, V_{i,2}^G, \dots, V_{i,D}^G\}$  are generated by adding the weighted difference between two randomly chosen individuals from the current population to a third randomly selected individual from the current population. Mathematically it can be written as:

$$V_i^G = X_{r_1}^G + F_i(X_{r_2}^G - X_{r_3}^G) \quad \forall i = 1, 2, \dots, NP \quad (4)$$

where  $V_i^G$  = mutant vector associated with the  $i$ th population at the  $G$ th generation.  $X_{r_1}^G$ ,  $X_{r_2}^G$  and  $X_{r_3}^G$  are randomly selected vectors from the current generation ( $r_1 \neq r_2 \neq r_3$ ),  $F_i$  is the mutation factor for target vector  $i$ , and  $NP$  represents the population size.

5. Next, crossover is implemented for the generation of trial vectors  $U_i^G = \{U_{i,1}^G, U_{i,2}^G, \dots, U_{i,D}^G\}$  by choosing between either a mutant vector ( $V_i^G$ ) or the corresponding target vector ( $X_i^G$ ) using a crossover factor. Thus, the trial vector ( $U_i^G$ ) is generated as:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G, & \text{if } rand^j [0, 1] \leq CR_i \text{ or } (j = j^{rand}) \\ x_{i,j}^G, & \text{otherwise} \end{cases} \quad \forall i = 1, 2, \dots, NP \quad (5)$$

where  $u_{i,j}^G$ ,  $v_{i,j}^G$ ,  $x_{i,j}^G$  are the trial vector, mutant vector, and target vectors associated with the  $j$ th decision parameter of the  $i$ th member at generation  $G$ . The  $rand^j$  is a uniform random number in between 0 to 1, and  $j^{rand}$  can take any random value in the range 1 to  $D$ ,  $D$  is the dimension of the target vector.

6. Subsequently, the objective function value  $f(U_i^G)$  for the individual trial vector is computed and then compared with the objective function value of the corresponding target vector  $X_i^G$ . The vector which holds the lower value of the objective function continues for the next generation ( $X_i^{G+1}$ ) (for minimization type of objective function). In other words:

$$X_i^{G+1} = \begin{cases} U_i^G, & \text{if } f(U_i^G) \leq f(X_i^G) \\ X_i^G, & \text{otherwise} \end{cases} \quad \forall i = 1, 2, \dots, NP \quad (6)$$

The above steps of mutation, crossover, and selection operations in Equations (4)–(6) are performed for all members in the population ( $NP$ ).

7. To update the values of two control parameters (i.e.  $F$  and  $CR$ ) in the SaDE algorithm, the values of  $F$  and  $CR$  contributing to trial vectors effectively arriving for the next generation are noted after a certain number of generations (say, ten generations). Then, averaging those successful values of the  $F$  and  $CR$  provides the updated mean values of mutation factor ( $\mu_F$ ) and crossover rate ( $\mu_{CR}$ ), respectively. With these updated values of  $\mu_F$  and  $\mu_{CR}$ , a new set of  $F$  and  $CR$  values are generated as described previously in Equation (3) (i.e. assuming normal distributions for  $F$  and  $CR$  with the specified standard deviation values).

8. The above steps are repeated in a loop until the convergence criterion is satisfied. The convergence criteria employed in the present study are either there is no improvement in the objective function value for ten successive generations, or the algorithm has reached the maximum number of allowable generations ( $I_{\max}$ ).

More details of the SaDE method can be found in [Qin \*et al.\* \(2009\)](#) and [Sirsant & Reddy \(2018\)](#).

## APPLICATION

### Study area

The study area is situated in the upper Tungabhadra River basin in Karnataka, India. The location map of the study area is shown in [Figure 1](#). The spatial extent of the study area ranges from 75°7'E to 75°57'E and 13°7'N to 14°15'N, with the basin outlet at Honnali gauging station. The basin has a catchment area of 7,075 km<sup>2</sup> with elevations ranging from 542 to 1,890 m above mean sea level. The study area receives an average annual rainfall of 1,016 mm. The average annual temperature is 26.7 °C in the basin. A semi-arid climatic condition prevails in the study area, and the majority of rains occur from June to November during the south-west and north-east monsoons. It is predominantly covered by agricultural land with loamy to red sandy soil as the dominant soil type. Chikamagalur, Shimoga, and Devanagari are the three major districts falling under the study area and they are greatly dependent on water from the Tunga River. Accessibility of different observed hydro-meteorological datasets and nonexistence of large control structures are the main reasons for the selection of this study area.

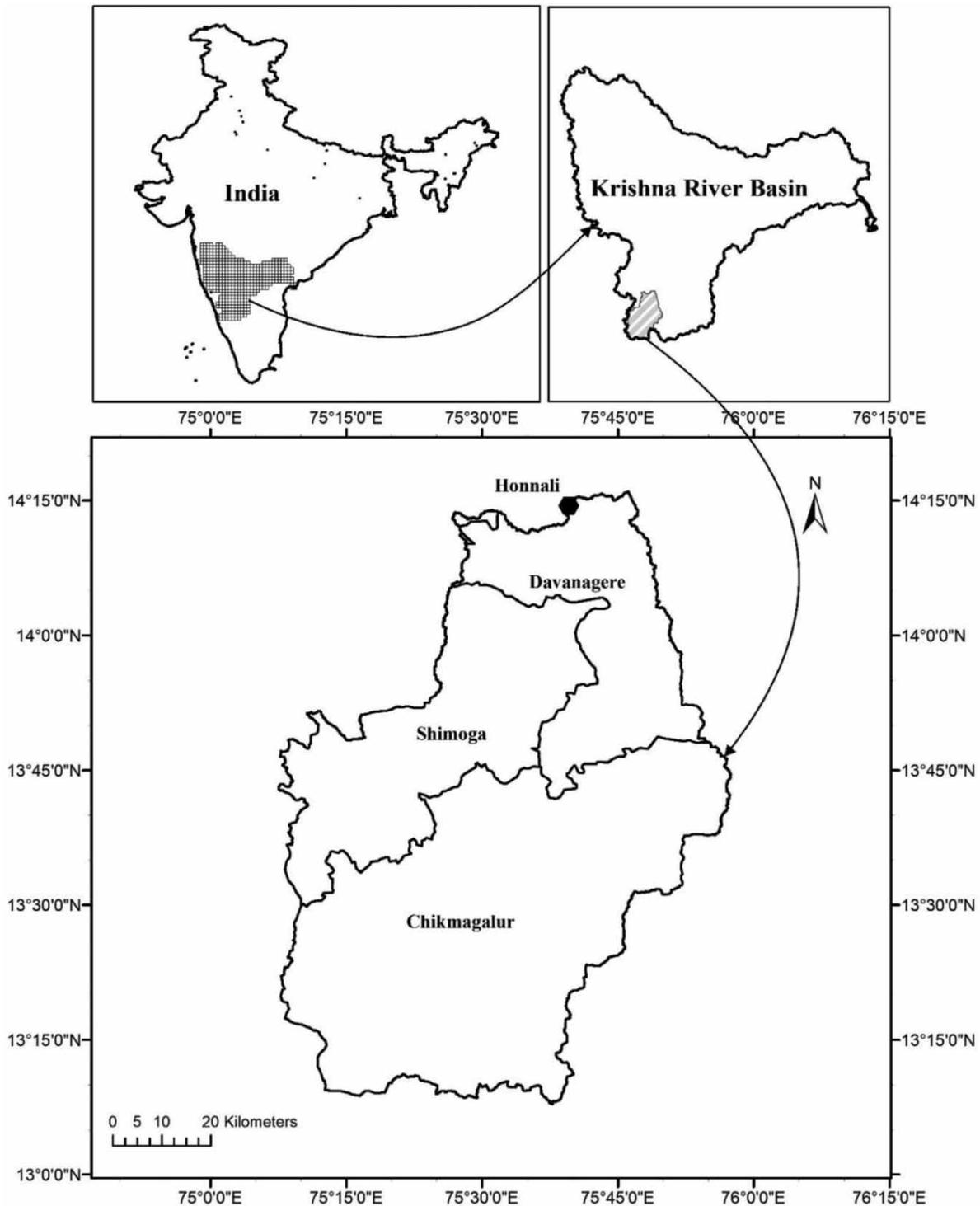
### Data used

Daily gridded precipitation data ( $0.25 \times 0.25^\circ$ ) collected from the Indian Meteorological Department (IMD) are used in this study as the primary forcing variable for all hydrological simulations. Daily temperature data was also taken from the IMD at a spatial resolution of  $1 \times 1^\circ$ . Wind speed data were collected from the National Center for Environmental Prediction (NCEP) reanalysis data at  $2.5 \times 2.5^\circ$  scale. This study used soil information from the Food and Agricultural Organization (FAO) harmonized world soil database at 30 arc-second ( $\sim 1$  km) spatial resolution. Land-use and land-cover data for the study were derived from MODIS MOD12Q1 Land Cover Products (1 km), which was obtained from NASA's data distribution portal. SRTM (90 m) DEM was collected from the National Remote Sensing Centre of India. The data of daily observed streamflow at the outlet of the catchment, Honnali Hydro-observation site, were obtained from WRIS-India.

### VIC model calibration

#### VIC model settings

Based on previous studies, 13 flow-related parameters from the VIC model were chosen for the automatic calibration ([Demaria \*et al.\* 2007](#); [Xie \*et al.\* 2007](#)). These 13 parameters, along with their descriptions, model response, ranges, and units, are presented in [Table 1](#). Among these 13 parameters, one governs direct runoff response, eight govern drainage responses, and four control the baseflow responses. During the calibration process, these selected 13 parameters shared the same value for the whole model domain. The other VIC model parameters are obtained from the field measurement, which is varied across the model domain. The calibration could have considered independent spatial values for all the calibration parameters, but that would have increased the complexity of the



**Figure 1** | Location map of the study area.

calibration to a great extent (i.e. more than 180 parameters to be estimated), which may have led to impractical longer optimization time with the available computational resources.

The VIC model and aforementioned routing model source code are first compiled in the Linux environment using the GNU Compiler Collection (GCC). The integration of the SaDE optimization algorithm with the VIC model is shown in Figure 2. The simulation-optimization framework is implemented in the MATLAB environment. For this integration it first requires creation of all the input files (i.e. meteorological forcing files, soil parameter file, vegetation parameter file, and vegetation

**Table 1** | Details of the VIC model parameters selected for calibration

Parameter	Description	Model response	Range	Units
$b_1$	Infiltration parameter	Direct flow	$1 \times 10^{-5}$ –0.4	–
$D_m$	Maximum baseflow velocity	Base flow	0–30	mm/day
$D_s$	Fraction of max baseflow velocity for nonlinear baseflow	Base flow	0–1	–
$W_s$	Fraction of max soil moisture for nonlinear baseflow	Base flow	0–1	–
$d_1$	Thickness of soil layer 1	Drainage	0.01–0.5	m
$d_2$	Thickness of soil layer 2	Drainage	0.1–2	m
$d_3$	Thickness of soil layer 3	Base flow	0.1–2	m
$K_{s1}$	Saturated hydraulic conductivity for first layer	Drainage	1–10,000	mm/day
$K_{s2}$	Saturated hydraulic conductivity for second layer	Drainage	1–10,000	mm/day
$K_{s3}$	Saturated hydraulic conductivity for third layer	Drainage	1–10,000	mm/day
$exp_1$	Exponent (Brooks-Corey drainage eq.) for first layer	Drainage	8–30	–
$exp_2$	Exponent (Brooks-Corey drainage eq.) for second layer	Drainage	8–30	–
$exp_3$	Exponent (Brooks-Corey drainage eq.) for third layer	Drainage	8–30	–

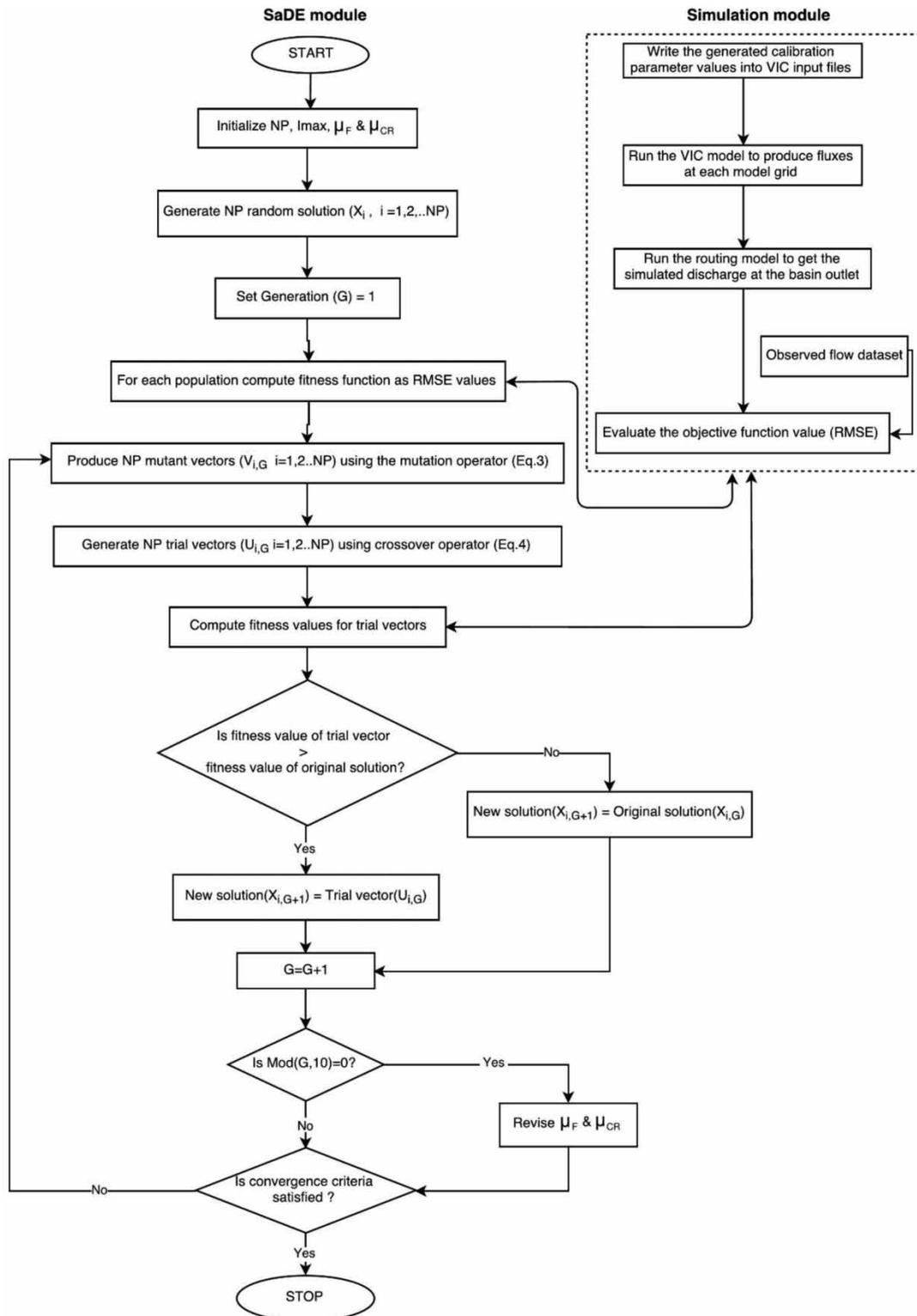
library file) and a global parameter file (i.e. model configuration file) for the simulation of the VIC model. We also need to define the required input files (i.e. flow direction file, area mask file, area fraction file, outlet station file, and a model configuration file) for the routing model. Next, the SaDE algorithm is implemented in the MATLAB, as discussed above under ‘Self-adaptive differential evolution algorithm’ (called the SaDE module). The SaDE module calls the simulation module whenever it needs to compute the fitness values of the member solutions of the initial population or the recent trial vectors. The simulation module first updates the appropriate VIC input files using the feedback received from the SaDE module and then runs the VIC model to produce the hydrological fluxes at each model grid. Next, the routing model is executed via the simulation module to obtain the simulated discharge at the basin outlet. Subsequently, the root-mean-square error (RMSE) is computed as the fitness measure by comparing the simulated discharge with the observed discharge data and sends this information back to the SaDE module. The optimization using the SaDE module progresses with its continuous interconnection with the simulation module until certain stopping conditions are met. Similarly, the GA, SCE and DE algorithms are also implemented and linked to the VIC model in the MATLAB environment. The study used four years of data (1 January 1995–31 December 1998) for calibration, and three years of data (1 January 1999–31 December 2001) for validation of the VIC model.

### Calibration criteria and performance measures

Evaluation of any hydrological model is usually conducted by pairwise comparisons of model-simulated flow values with the observed ones. The RMSE between observed and simulated flows is used as the calibration criteria (objective function) to determine optimal parameters of the VIC hydrological model (Madsen 2000; Kavetski & Clark 2010). RMSE calculates the average squared errors for all simulated flow values and is defined as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (S_i - O_i)^2} \quad (7)$$

where  $O_i$  and  $S_i$  are observed and simulated flow values for the  $i$ th time step, and  $n$  is the number of time steps.



**Figure 2** | Schematic layout of SaDE based automatic calibration of VIC hydrological model.

Then, the performance of the calibration algorithms is assessed by four commonly used evaluation criteria, i.e. Nash–Sutcliffe efficiency ( $E_{ns}$ ), determination coefficient ( $R^2$ ), percent bias ( $PBIAS$ ), and mean of simulated flow ( $\overline{Q_s}$ ). These  $E_{ns}$ ,  $R^2$ , and  $PBIAS$  are defined in Appendix B. The  $E_{ns}$  assesses the degree of fitness between simulated and observations hydrograph by a normalized statistic based on the ratio of the residual variance to observed data variance. Values of  $E_{ns}$  can range between

$-\infty$  to 1, while a value of  $E_{ns} = 1$  suggests a perfect match between simulated and observed data, and a value of  $E_{ns} < 0$  indicates that the observed mean would produce better accuracy than the model simulated streamflows. The  $R^2$  is the ratio of explained variation to total variation, which reveals the strength of association between the simulated and observed data. Values of  $R^2$  can vary between 0 and 1, while a higher value designates a better agreement. The *PBIAS* provides a view of the average tendency of under- or overprediction of the simulated discharge with respect to the observation data.

### Parameter settings of the algorithms

All the algorithms used in this study need some parameter settings to be defined by the user. For the SaDE algorithm, all the parameter settings except for the population size were discussed previously. With a limited number of function evaluations, a relatively lower population size is found to be more effective than the larger population size for calibration of hydrological models by EAs (Zhang *et al.* 2009). Therefore, a population size of 30 was selected for GA, DE and SaDE algorithms. The mutation factor and crossover rate for DE were chosen as 0.5 and 0.9, respectively, following suggestions from past studies (Storn & Price 1997; Brest *et al.* 2006) and after performing some preliminary sensitivity analysis for the parameters. For the implementation of GA, the MATLAB genetic algorithm toolbox is used with crossover fraction and probability rate of mutation values set at 0.8 and 0.01, respectively, which are selected after some preliminary sensitivity analysis. In the SCE algorithm, the number of complexes ( $p$ ) was the most important parameter. In this study, for the SCE algorithm the value of  $p$  is set equal to the number of calibration parameters (i.e. 13), and the other parameter values are selected following suggestions from past studies (Duan *et al.* 1992; Kuczera 1997) as well as after carrying out some preliminary sensitivity analysis.

## RESULTS AND DISCUSSION

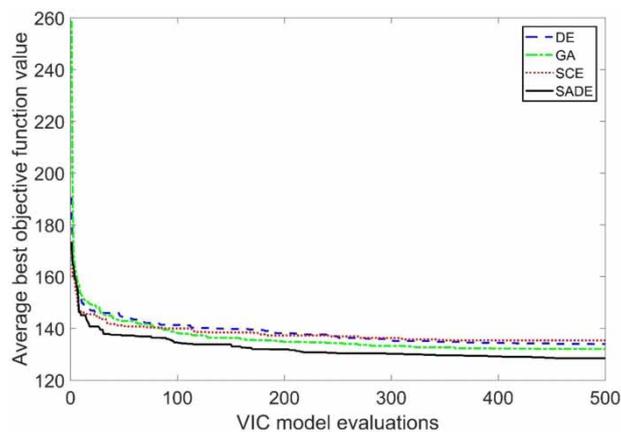
As one execution of the VIC hydrological model took an average of 65 seconds on a personal computer (PC) with a configuration of Intel Core i7-4790 3.60 GHz processor with 8 GB RAM, performing a large number of model evaluations during the calibration process is cumbersome. This is particularly true for EAs, which need to be executed for multiple trial runs. Therefore, the calibration of the VIC model is performed with a perspective of a limited number of model evaluations. The maximum number of model evaluations was fixed at 500 for each optimization run (except for checking the global convergence, where 5,000 function evaluations were permitted). Results from the proposed SaDE based calibration were compared with the results from GA, SCE and DE algorithms. This comparison consists of ten optimization runs for each optimization algorithm as all four algorithms are stochastic in nature, and therefore their comparative performance must be evaluated for several runs originating from the independent random population. The variation of the average best solution attained in ten optimization runs with the number of function evaluations was presented, which provided a comparison of average performance for the four algorithms. For further evaluation of the algorithm performance, additional tests such as stability (or variability in multiple trials), reliability, and statistical tests were also conducted in the study. It also examined the ability of the SaDE algorithm to achieve stable convergence to the global optimal solution with additional computational expenses.

### Comparison of algorithms' performance

To check the ability of the algorithms to reach the global optimal solution, ten separate optimization trial runs were performed (with a computational cost of 5,000 model evaluations in each trial run) for

the GA, SCE, DE, and SaDE algorithms. Each trial run of an EA required almost 4 days to execute for the current problem. Therefore, such comparisons were accomplished only after a massive computational cost, which can help in understanding the evolution skill of the SaDE and other algorithms in finding a global optimal solution, if possible, later in a restricted computational cost. The results showed that most of the time, SaDE and other EAs (i.e. GA, SCE, DE) attained optimal solutions at the end of 5,000 model evaluations. However, their performance at different stages of evolution was highly variable and dissimilar. For instance, the SaDE found its best solution after the 2,005 model evaluation, while GA and SCE took 2,982 and 4,474 model evaluations, respectively, to find the optimal solution. Even though the SaDE took 2,005 model evaluations for its complete convergence, it was noteworthy that more than 95% of its maximum fitness was achieved within 346 model evaluations. For the SCE algorithm, it took 2,632 model evaluations to attain the same 95% maximum fitness. Overall, it seems clear from these results that the SaDE algorithm can converge to a global optimal solution if an adequate computational budget allowed, and converges much faster than other EAs.

For computationally intensive complex hydrological models, the number of model evaluations required to achieve better objective values is a vital feature for choosing an optimization algorithm. With a detailed representation of a large catchment area, other physically-based hydrological models similar to the VIC model may also take a longer time for their executions. In such a case, running the model for a large number of times becomes very challenging. Therefore, preference should be given to the algorithm which is able to obtain superior objective function value with a limited computational budget (Duan *et al.* 1992; Tolson & Shoemaker 2007). In this direction, the performance of the SaDE was further assessed with limited model evaluations. To achieve this, the maximum number of model evaluations was fixed at 500 for each optimization run, which took about 8 hours of computational time. Ten such trial runs were performed for each of the GA, SCE, DE and SaDE algorithms and the plot of average best RMSE as the objective function value against the number of model evaluations is shown in Figure 3. From Figure 3, it is to be noted that the convergence speed of SaDE is faster than the other three algorithms. Initially, the SCE algorithm exhibited better convergence than the GA and DE algorithms, but finally, the GA algorithm reached lower objective function values than the SCE and DE algorithms. Overall, the SaDE algorithm consistently performed better than the other three algorithms. Even at half of the total model evaluations (250), the SaDE average objective function value (130.85) was still better than the final average objective function values (i.e. after 500 model evaluations) of GA (132.1), DE (138.25) and SCE (136.2). This shows the dominant performance of SaDE over the other algorithms, even with a smaller number of model evaluations.



**Figure 3** | Comparison of convergence characteristics of average best objective function values (RMSE) for the GA, SCE, DE and SaDE algorithms with a limited function evaluation over ten optimization trial runs.

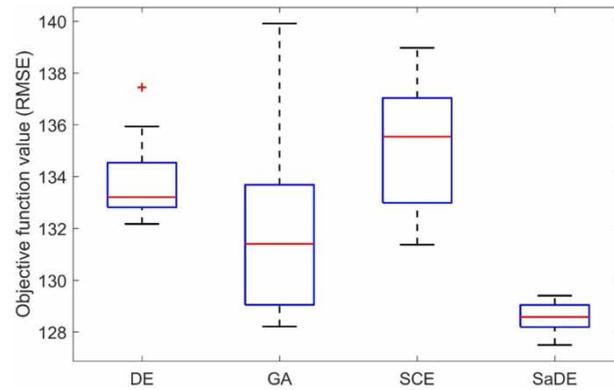
Nevertheless, the results reveal that four optimization algorithms showed diverse performance at a different number of model runs (Figure 3). Although the results confirmed that SaDE was the most efficient optimization algorithm for the calibration of the VIC model, as it had resulted in the lowest objective function value compared to the GA, SCE and DE algorithms, there was still a need to assess the significance of the results as well. Therefore, the statistically significant difference between the mean and median of the best objective function values from the four algorithms were examined using the well-known two-sample *t*-test and Wilcoxon rank-sum test, respectively, and the results are presented in Table 2. These two tests with the null hypothesis of identical mean were carried out at the 5% significance level. A *p*-value lower than the reference significance level suggested a rejection of the null hypothesis. The *p*-values obtained for the comparison of SaDE with those other algorithms were  $1.2 \times 10^{-5}$  (*t*-test for DE), 0.012 (*t*-test for GA),  $9.7 \times 10^{-5}$  (*t*-test for SCE),  $1.8 \times 10^{-4}$  (Wilcoxon Rank-Sum-test for DE), 0.017 (Wilcoxon Rank-Sum-test for GA), and  $1.8 \times 10^{-4}$  (Wilcoxon Rank-Sum-test for SCE). Therefore, results from both the two-sample *t*-test and the Wilcoxon rank-sum test were consistent, which suggested that the SaDE performed significantly better than the other three algorithms.

In the case of expensive model evaluation, it is of vital importance to select an optimization algorithm that can generate satisfactory solutions repeatedly, i.e. produce smaller variability in the objective function values for multiple trial runs. Smaller variability indicates greater stability of an algorithm. Therefore, a little variability with a smaller mean objective function value (RMSE) recommends a more robust algorithm for calibration. For practical calibration of the hydrological model with a lengthy simulation run, one will most likely prefer an optimization algorithm that needs only a single run for its calibration. Consequently, the algorithm that gives consistently good results is evidently a better choice than an algorithm which delivers a good solution with an equal probability to produce a poor solution. Conventionally, variability is measured by the standard deviation and the corresponding interquartile ranges. Therefore, a boxplot is used to represent the variability, which displays the lower quartile, median, upper quartile, whiskers (dashed lines spanning from top and bottom of the box for showing the stretch of the leftover data), and also the outliers with a '+' symbol. Figure 4 shows the boxplot of the best objective function values obtained from ten trial runs for each of the four algorithms. It can be noted from the figure that the SaDE algorithm is subjected to lowest variability (interquartile range) and best objective function value (lowest median) in the ten different trial runs, suggesting a robust performance of SaDE as compared to the other EAs. The GA was associated with a higher spread in the final objective function values. The DE algorithm is found to have a lower spread but higher objective function values (RMSE). The SCE algorithm was associated with higher variability and worst optimal values. Earlier successful applications of the SCE

**Table 2** | The results of *p*-values obtained for statistical significance tests to assess the comparative performance of SaDE with the GA, SCE and DE algorithms via the two-sample *t*-test and Wilcoxon rank-sum tests

Statistical test	DE	GA	SCE	SaDE
Two-sample <i>t</i> -test				
DE	–			
GA	0.221	–		
SCE	0.153	<i>0.005</i>	–	
SaDE	<i><math>1.2 \times 10^{-5}</math></i>	<i>0.012</i>	<i><math>9.7 \times 10^{-5}</math></i>	–
Wilcoxon rank-sum test				
DE	–			
GA	0.140	–		
SCE	0.212	<i>0.045</i>	–	
SaDE	<i><math>1.8 \times 10^{-4}</math></i>	<i>0.017</i>	<i><math>1.8 \times 10^{-4}</math></i>	–

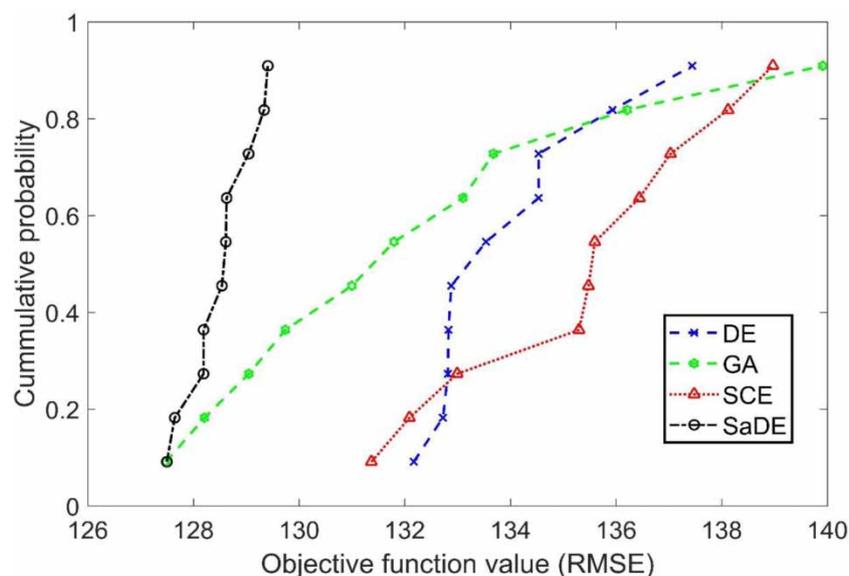
Note: the values in italics indicate statistical difference exists at the 5% significance level.



**Figure 4** | Boxplot of best objective function values (RMSE) obtained using GA, SCE, DE, and SaDE algorithms for ten trial runs.

algorithm for hydrological model calibration had used very large numbers (often in excess of 10,000) of model evaluations to obtain the final solution (Tolson & Shoemaker 2007). However, with a limited number of model evaluations, the SCE algorithm could not reach the global (or near-global) optimal solution, as reported by previous studies (Tolson & Shoemaker 2007; Arsenault *et al.* 2014; Qin *et al.* 2016). Thus, restrictions on the number of model evaluations may have led to the poor performance of the SCE algorithm in the current application.

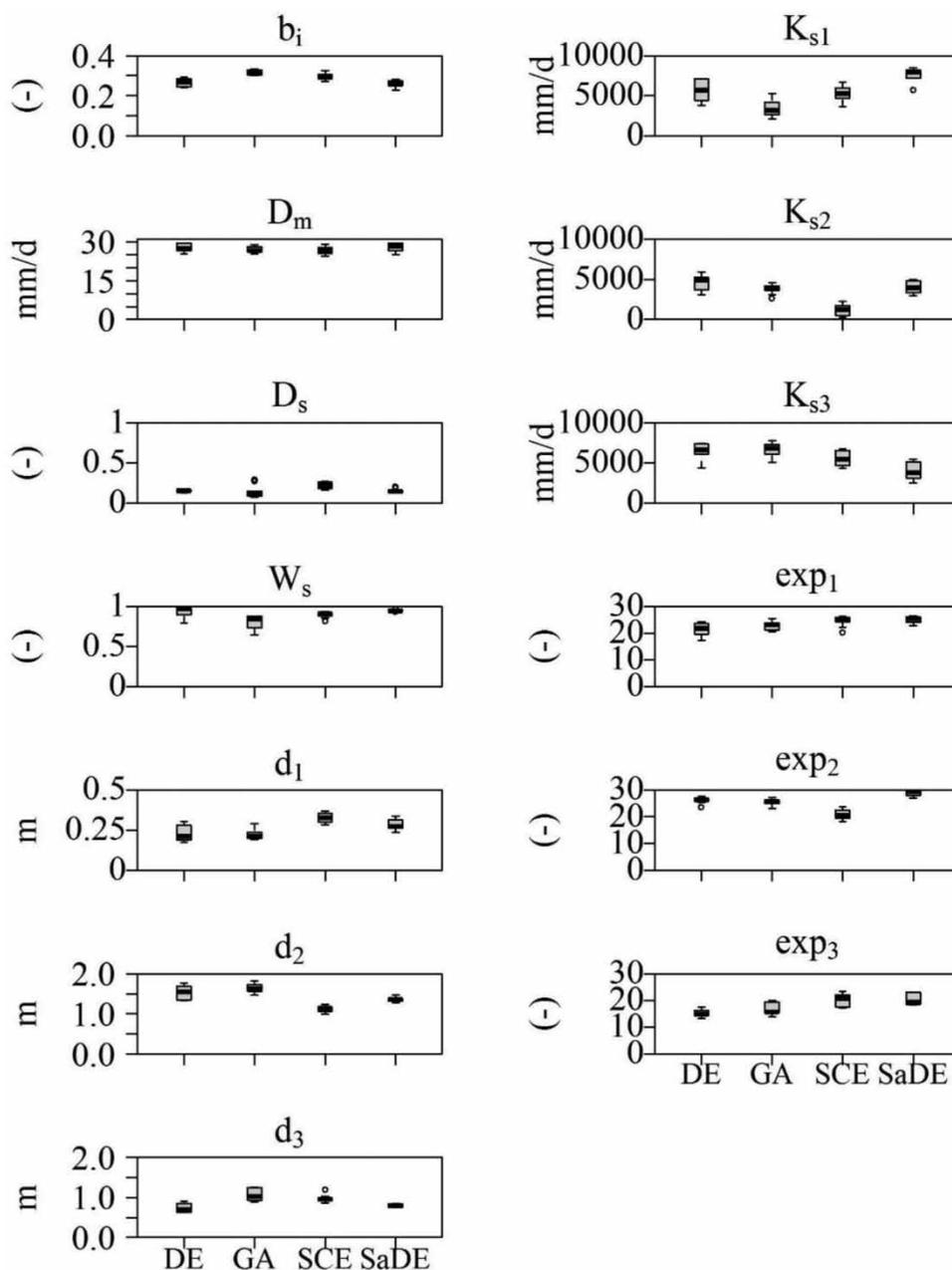
Figure 5 shows a comparison of the empirical cumulative distribution function plots between the four algorithms. The empirical cumulative distribution function (as used by Devore (2004)) for the different algorithms provides a useful comparison of the measure of the reliability of the algorithms. It is an approximation of the original cumulative distribution function and states the occurrence frequency of best objective function values below some reference value. In Figure 5, the empirical probabilities (or the occurrence frequencies) are the plotting positions attributed to the best objective function values of the four algorithms from their ten trial runs using  $r/(n + 1)$ , where  $r$  and  $n$  stand for the rank of ordered best objective value and the number of data points respectively. Therefore, for ten trial runs, there will be ten data points, and the empirical cumulative distribution will be in the range of 0.09–0.91. The algorithm with a higher probability (or approximately vertical empirical cumulative distribution plot) at lower objective function values would be preferred in the case of the minimization



**Figure 5** | Comparison of the empirical cumulative distribution for the best objective function values (RMSE) resulted from 10 trial runs using GA, SCE, DE, and SaDE algorithms.

problem. Based on such criteria, the SaDE algorithm is the best among the four algorithms, as it holds a nearly steep slope with a sense of smaller spread, and also as it occupies the position farthest left thus indicating its capability of producing smaller objective function values than other algorithms. Both the SCE and DE algorithms are positioned farther right to SaDE and have flattened out curves, suggesting that they are less reliable. It can be noted that GA starts at the farthest left but flattens out for higher cumulative probabilities, indicating that it is less reliable as it produces poor results in some trial runs.

This study also analysed the parameter equifinality (which concerns the principle that in open systems a given end state can be reached by several possible ways) of different EAs in their application to VIC model calibration. All the parameter sets within 1% of the optimal RMSE values are considered equifinal (Nemri & Kinnard 2019), see Figure 6. From the dispersion of the equifinal parameters in Figure 6, it can be noted that different parameters are associated with different degrees of equifinality.



**Figure 6** | Boxplots of equifinal parameters set for the VIC model calibration using GA, SCE, DE, and SaDE algorithms.

For example, parameters such as  $b_i$ ,  $D_m$  and  $D_s$  are more identifiable due to their smaller variability as compared to the parameters such as  $K_{s1}$ ,  $K_{s2}$ ,  $K_{s3}$ ,  $\exp_1$  and  $\exp_3$ . Different algorithms also have shown varying degrees of equifinality, as can be seen from Figure 6. Overall, the SaDE and GA methods have formed a more homogenous parameter distribution with reduced dispersion in their equifinal parameter sets.

### Runoff estimation

The earlier assessment of the algorithmic performance was carried out in terms of efficiency, variability, and reliability with a limited number of function evaluations after examining their ability to attain an optimal solution. In this section, an effort was made to evaluate the performance of different calibration algorithms for reliable runoff simulation. Four performance measures ( $E_{ns}$ ,  $R^2$ ,  $PBIAS$ , and mean of simulated flow) were computed by adopting the best parameter values obtained from the ten optimization trial runs with 500 function evaluations. It can be seen from Table 3 that the mean daily runoff was better estimated with the SaDE algorithm during both calibration and validation periods. The SaDE overestimated the mean daily flow by 1.55% during calibration, while GA, SCE and DE overestimated it by 2.07, 5.18, and 2.59%, respectively. For validation, the GA, SCE and DE underestimated the mean daily flow by 7.35, 5.39, and 6.37%, respectively, whereas SaDE overestimated it by 1.47%. The  $E_{ns}$ ,  $R^2$ , and  $PBIAS$  for SaDE during calibration were 0.87, 0.93, and -1.75, respectively, which were more favourable than the other three algorithms. For the validation period,  $E_{ns}$  and  $R^2$  were nearly the same for all the algorithms, but  $PBIAS$  pointed towards the marginally better performance of the SaDE. Earlier studies recommended that for a satisfactory model simulation its  $E_{ns}$  must be greater 0.5,  $R^2$  should be higher than 0.6, and  $PBIAS$  less than or equal to 25% (Santhi *et al.* 2001; Moriasi *et al.* 2007). Based on the recommendations from past studies, it can be inferred that all the algorithms performed satisfactorily in streamflow simulation, although the overall performance of the SaDE was comparatively better than the other three algorithms.

**Table 3** | Statistical performance measures for the simulated streamflow with the best DE, GA, SCE, and SaDE estimated parameters of the VIC model

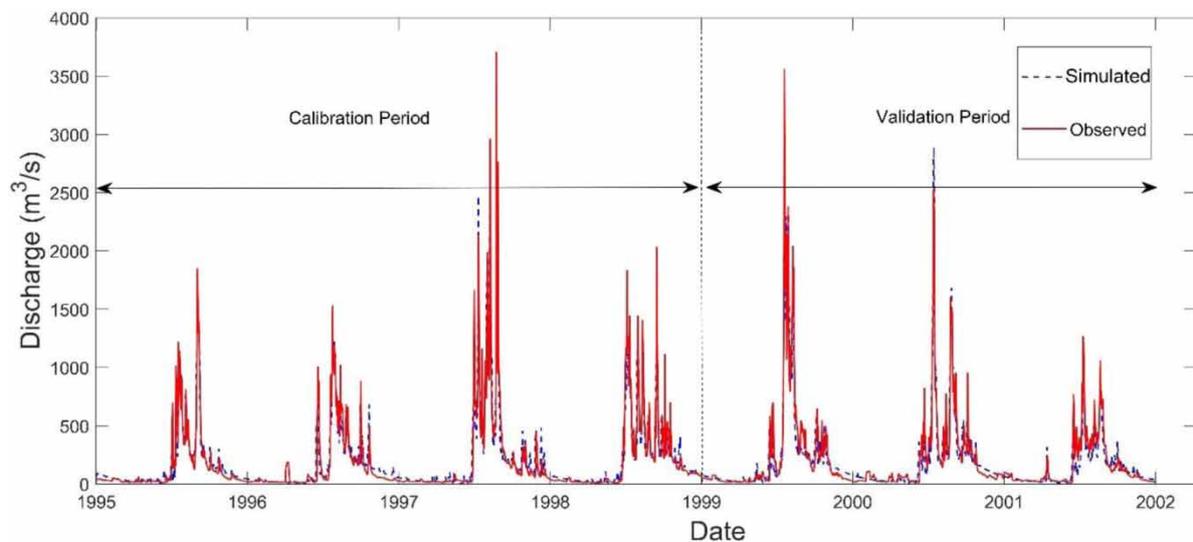
Simulation period	Algorithm	$\bar{Q}_s$ (m <sup>3</sup> /s)	$E_{ns}$	$R^2$	$PBIAS$
Calibration (mean observed flow = 193 m <sup>3</sup> /s)	GA	197	0.86	0.92	-2.10
	SCE	198	0.82	0.90	-2.63
	DE	203	0.82	0.91	-5.15
	SaDE	<i>196</i>	<i>0.87</i>	<i>0.93</i>	<i>-1.75</i>
Validation (mean observed flow = 204 m <sup>3</sup> /s)	GA	189	0.86	0.92	7.28
	SCE	191	0.86	0.93	2.46
	DE	193	0.85	0.93	5.32
	SaDE	<i>207</i>	0.86	0.93	<i>-1.51</i>

Note: Values in italics signify best among all the algorithms.

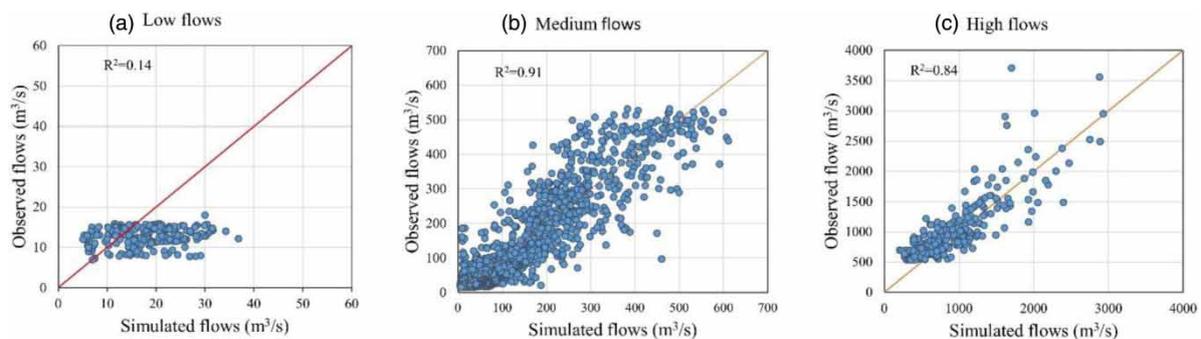
The calibrated values of the VIC model parameters (using the SaDE algorithm) are given in Table 4. These parameter values were obtained from the best solution of the SaDE algorithm at the end of 500 evaluations from ten optimization trial runs. The VIC model simulated flows using those calibrated parameter values is compared with the observed flows in Figure 7. It is clear from the figure that most of the time the simulated flows closely match the observed flows. Different streamflow characteristics are generally represented by the different flow signals (i.e. high flow, low flow, and medium flow) from the hydrograph. Therefore, to obtain a comprehensive idea about the different streamflow characteristics, the effect of SaDE based automatic calibration on different flow signals was analysed.

**Table 4** | Details of the VIC model parameters selected from calibration using SaDE

Parameter	Optimal value	Parameter	Optimal value
$b_1$	0.24	$Ks_1$	5,936.10
$D_m$	30.00	$Ks_2$	3,239.10
$D_s$	0.16	$Ks_3$	3,403.70
$W_s$	0.95	$exp_1$	21.58
$d_1$	0.28	$exp_2$	29.72
$d_2$	1.47	$exp_3$	16.46
$d_3$	0.84		

**Figure 7** | Comparison of SaDE calibrated VIC model simulated flows with observed flows during calibration and validation periods.

Based on the exceedance probability, the flow signals were categorised into the low flow (exceedance probability  $\geq 90\%$ ), medium flow ( $10\% < \text{exceedance probability} < 90\%$ ), and high flow (exceedance probability  $\leq 10\%$ ) signals. Figure 8 shows the comparison of VIC model simulated flows (with calibration parameters obtained using the SaDE) with the observed flows for different flow signals. It is clear from Figure 8(a) that the daily observed and simulated flows during low flow periods were very poorly associated ( $R^2 = 0.14$ ) as the model tends to overestimate the majority of low flow signals.

**Figure 8** | Scatter plots of observed vs simulated flows from the SaDE calibrated VIC model, for (a) low flows, (b) medium flows, and (c) high flows.

In contrast, for the medium flows, the agreement between the simulated and observed flows was high ( $R^2 = 0.91$ ), but both had some over- and underestimations (Figure 8(b)). The simulated high flows were also predicted with good accuracy ( $R^2 = 0.84$ ) along with a few underestimations (Figure 8(c)). Overall, the VIC model simulations for high and medium flows showed good agreement with the observed flows, while the low flows were poorly estimated by the VIC model.

To further investigate the behaviour of the modelled flow values at different flow levels, performance measures of the modelled streamflow were computed after partitioning the streamflow series into wet and dry season flows. The months of June–November, where the observed monthly flows had exceedance probability less than 50%, were considered as the wet season, and the rest of the months were considered as the dry season. The wet season contained more than 91% of the annual average streamflow in the study area. Table 5 represents the comparison of three performance measures ( $E_{ns}$ ,  $R^2$ , and  $PBIAS$ ) for wet and dry seasons. The wet season exhibited a similar performance to the whole time period (Table 3) in terms of  $E_{ns}$  and  $R^2$  during both calibration and validation periods. In Table 5, small positive biases were found for the wet season. In contrast, a large negative bias was found for calibration during the dry season. Overall, the performance measured during the dry season was very poor compared to those for the wet season and also for the entire period. Similar low performances of the estimated streamflows during dry periods (even after the rigorous model calibration) were reported in several past studies (Gan *et al.* 1997; Croke *et al.* 2004; Kim 2015). However, the limitation of accurate streamflow estimation in dry conditions may not be a consequence of the particular choice of a calibration algorithm, rather it may be linked to the inadequate model structure and parameterization of the hydrological model, selection of calibration data length, and in many instances due to the choice of the objective function (Kim 2015).

**Table 5** | Performance of the SaDE based calibrated streamflows for dry and wet periods during VIC model calibration and validation

	Calibration		Validation	
	Wet period	Dry period	Wet period	Dry period
$E_{ns}$	0.87	0.09	0.88	-0.21
$R^2$	0.93	0.63	0.94	0.43
$PBIAS$	3.12	-24.99	5.02	-1.84

## CONCLUSIONS

Hydrological models mimic the behaviour of hydrological processes in catchments, and their reliable results help in gaining enhanced knowledge of various environmental processes. Calibration of a physically-based hydrological model is a challenging task because of the large number of unknown parameters, equifinality of parameter sets, and the computational cost for their simulation that restricts the number of model runs that are required for the calibration. In this study, the SaDE algorithm is presented for the automatic calibration of a physically-based VIC hydrological model in the upper Tungabhadra river basin of India. The SaDE eliminates the need for fine-tuning of control parameters (i.e. mutation factor  $F$  and crossover rate  $CR$ ) of DE by implementing an automatic self-adapting process, which helps in saving a considerable amount of time and computational budget in finding optimal parameters. Further, apart from SaDE, three more evolutionary algorithms, namely GA, SCE, and DE algorithms, are applied for calibration of VIC model parameters, and the results of SaDE are compared with those from the GA, SCE and DE algorithms. The performance of algorithms was evaluated using standard performance measures subject to a limited computational budget (or a number of function evaluations) and assessed the usefulness of the SaDE approach.

Analysis from the results of the study revealed that the SaDE is a powerful tool for hydrological model calibration and it can produce significantly better performance than the other three algorithms. Here, the main advantage of the SaDE method is that it eliminates the tedious and computationally demanding task of parameter fine-tuning. The empirical CDF plots prepared for reliability assessment of the algorithm clearly indicated the advantage of the SaDE over the other algorithms during the calibration and validation of the VIC model. Moreover, the SaDE algorithm leads to the smallest standard deviation and interquartile range (evaluated based on different trial runs of fitness/objective values), which reveals its consistent precedence over the other three algorithms. Although the problem of 'equifinality' is found in the model calibration, it can be expected that the equifinality in parameter space will be less pronounced when using the SaDE method. The results of this study also demonstrate the superior performance of SaDE in the case of a limited number of model evaluations. If an adequate number of function evaluations are allowed, then SaDE is found to achieve global optimal solutions like other EAs, but it achieves much faster than the GA and SCE algorithms. A lesser number of function evaluations is a major advantage in hydrological modelling applications where time and computational constraints prevent higher model evaluations and recurrent runs of the algorithm. Results from this study also showed that SaDE produces a slightly better performance overall (in terms of  $E_{NS}$ ,  $R^2$ , and  $PBIAS$  statistics) during both model calibration and validation periods for streamflow simulation.

Thus, from the results of the study, it can be inferred that obtaining 'good' solutions with the SaDE is straightforward and efficient since it does not require a prior parametric study and it is also computationally inexpensive compared to the other EAs. Moreover, the SaDE algorithm is very simple and so it can be easily implemented in any programming language consistent with the operating environment of the model subjected to calibration. Therefore, the study recommends SaDE as a potential tool for calibration of complex hydrological models.

---

#### CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

---

#### ACKNOWLEDGEMENTS

The authors would like to thank the various organizations that provided support for the study: Department of Science and Technology (SPLICE–Climate Change Programme), Government of India, Project #DST/488/CCP/CoE/140/2018 for funding support; the Indian Meteorological Department (IMD), National Remote Sensing Centre (NRSC) of India, India-WRIS, Central Water Commission (CWC) of India, National Center for Environmental Prediction (NCEP), Food and Agricultural Organization (FAO), and the MODIS science and development team of NASA for providing various datasets used in this study.

---

#### DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

---

#### REFERENCES

- Arnell, N. W. 1999 *The effect of climate change on hydrological regimes in Europe: a continental perspective*. *Glob. Environ. Chang.* 9 (1), 5–23. [https://doi.org/10.1016/S0959-3780\(98\)00015-6](https://doi.org/10.1016/S0959-3780(98)00015-6).

- Arsenault, R., Poulin, A. & Brissette, F. 2014 Comparison of stochastic optimization algorithms in hydrological model calibration. *J. Hydrol. Eng.* **19** (7), 1374–1384. [https://doi.org/10.1061/\(ASCE\)HE.1943-5584.0000938](https://doi.org/10.1061/(ASCE)HE.1943-5584.0000938).
- Beven, K. & Freer, J. 2001 Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. *J. Hydrol.* **249** (1–4), 11–29. [https://doi.org/10.1016/S0022-1694\(01\)00421-8](https://doi.org/10.1016/S0022-1694(01)00421-8).
- Brest, J., Greiner, S., Boskovic, B., Mernik, M. & Zumer, V. 2006 Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput.* **10** (6), 646–657. <https://doi.org/10.1109/TEVC.2006.872133>.
- Cai, X., McKinney, D. C. & Lasdon, L. S. 2003 Integrated hydrologic-agronomic-economic model for river basin management. *J. Water Resour. Plan. Manag.* **129** (1), 4–17. [https://doi.org/10.1061/\(ASCE\)0733-9496\(2003\)129:1\(4\)](https://doi.org/10.1061/(ASCE)0733-9496(2003)129:1(4)).
- Chen, Y., Li, J. & Xu, H. 2016 Improving flood forecasting capability of physically based distributed hydrological models by parameter optimization. *Hydrol. Earth Syst. Sci.* **20** (1), 375–392. <https://doi.org/10.5194/hess-20-375-2016>.
- Croke, B. F., Merritt, W. S. & Jakeman, A. J. 2004 A dynamic model for predicting hydrologic response to land cover changes in gauged and ungauged catchments. *J. Hydrol.* **291** (1–2), 115–131. <https://doi.org/10.1016/j.jhydrol.2003.12.012>.
- Das, S., Mullick, S. S. & Suganthan, P. N. 2016 Recent advances in differential evolution—an updated survey. *Swarm. Evol. Comput.* **1** (27), 1–30. <https://doi.org/10.1016/j.swevo.2016.01.004>.
- Demaria, E. M., Nijssen, B. & Wagener, T. 2007 Monte Carlo sensitivity analysis of land surface parameters using the Variable Infiltration Capacity model. *J. Geophys. Res. Atmos.* **16** (112), D11113. <https://doi.org/10.1029/2006JD007534>.
- Devore, J. L. 2004 *Probability and Statistics for Engineering and the Sciences*, 6th edn. Cengage Learning, Ontario.
- Duan, Q., Sorooshian, S. & Gupta, V. 1992 Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.* **28** (4), 1015–1031. <https://doi.org/10.1029/91WR02985>.
- Franchini, M. & Pacciani, M. 1991 Comparative-analysis of several conceptual rainfall runoff models. *J. Hydrol.* **122** (1–4), 161–219. [https://doi.org/10.1016/0022-1694\(91\)90178-K](https://doi.org/10.1016/0022-1694(91)90178-K).
- Gan, T. Y. & Biftu, G. F. 1996 Automatic calibration of conceptual rainfall-runoff models: optimization algorithms, catchment conditions, and model structure. *Water Resour. Res.* **32** (12), 3513–3524. <https://doi.org/10.1029/95WR02195>.
- Gan, T. Y., Dlamini, E. M. & Biftu, G. F. 1997 Effects of model complexity and structure, data quality, and objective functions on hydrologic modelling. *J. Hydrol.* **192** (1–4), 81–103. [https://doi.org/10.1016/S0022-1694\(96\)03114-9](https://doi.org/10.1016/S0022-1694(96)03114-9).
- Goldberg, D. E. 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
- Guo, J., Zhou, J., Lu, J., Zou, Q., Zhang, H. & Bi, S. 2014 Multi-objective optimization of empirical hydrological model for streamflow prediction. *J. Hydrol.* **511**, 242–253. <https://doi.org/10.1016/j.jhydrol.2014.01.047>.
- Ibbitt, R. P. & O'Donnell, T. 1971 Fitting methods for conceptual catchment models. *J. Hydraul. Div.* **97** (9), 1331–1342.
- Kamali, B., Mousavi, S. J. & Abbaspour, K. C. 2013 Automatic calibration of HEC-HMS using single-objective and multi-objective PSO algorithms. *Hydrol. Process.* **27** (26), 4028–4042. <https://doi.org/10.1002/hyp.9510>.
- Kavetski, D. & Clark, M. P. 2010 Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction. *Water Resour. Res.* **46** (10), W10511. <https://doi.org/10.1029/2009WR008896>.
- Kim, H. S. 2015 Effectiveness of a regional model calibrated to different parts of a flow regime in regionalisation. *Hydrol. Earth Syst. Sci. Disc.* **12** (7), 7057–7098. <https://doi.org/10.5194/hessd-12-7057-2015>.
- Kim, S. M., Benham, B. L., Brannan, K. M., Zeckoski, R. W. & Doherty, J. 2007 Comparison of hydrologic calibration of HSPF using automatic and manual methods. *Water Resour. Res.* **43** (1), W01402. <https://doi.org/10.1029/2006WR004885>.
- Kuczera, G. 1997 Efficient subspace probabilistic parameter optimization for catchment models. *Water Resour. Res.* **33** (1), 177–185. <https://doi.org/10.1029/96WR02671>.
- Liang, X., Lettenmaier, D. P., Wood, E. F. & Burges, S. J. 1994 A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *J. Geophys. Res.* **99** (D7), 14415–14428. <https://doi.org/10.1029/94JD00483>.
- Liong, S. Y., Chan, W. T. & ShreeRam, J. 1995 Peak-flow forecasting with genetic algorithm and SWMM. *J. Hydraul. Eng.* **121** (8), 613–617. [https://doi.org/10.1061/\(ASCE\)0733-9429\(1995\)121:8\(613\)](https://doi.org/10.1061/(ASCE)0733-9429(1995)121:8(613)).
- Madsen, H. 2000 Automatic calibration of a conceptual rainfall-runoff model using multiple objectives. *J. Hydrol.* **235** (3), 276–288. [https://doi.org/10.1016/S0022-1694\(00\)00279-1](https://doi.org/10.1016/S0022-1694(00)00279-1).
- Moriasi, D. N., Arnold, J. G., Van-Liew, M. W., Bingner, R. L., Harmel, R. D. & Veith, T. L. 2007 Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Trans. ASABE* **50** (3), 885–900. <https://doi.org/10.13031/2013.23153>.
- Nemri, S. & Kinnard, C. 2019 Comparing calibration strategies of a conceptual snow hydrology model and their impact on model performance and parameter identifiability. *J. Hydrol.* **582**, 124474. <https://doi.org/10.1016/j.jhydrol.2019.124474>.
- Nijssen, B., Lettenmaier, D. P., Liang, X., Wetzel, S. W. & Wood, E. F. 1997 Streamflow simulation for continental-scale river basins. *Water Resour. Res.* **33** (4), 711–724. <https://doi.org/10.1029/96WR03517>.
- Osuch, M., Romanowicz, R. J. & Booi, M. J. 2015 The influence of parametric uncertainty on the relationships between HBV model parameters and climatic characteristics. *Hydrol. Sci.* **60** (7–8), 1165–1173. <https://doi.org/10.1080/02626667.2014.967694>.
- Pulluri, H., Naresh, H. R. & Sharma, V. 2017 An enhanced self-adaptive differential evolution based solution methodology for multiobjective optimal power flow. *Appl. Soft Comput.* **54**, 229–245. <https://doi.org/10.1016/j.asoc.2017.01.030>.
- Qin, A. K., Huang, V. L. & Suganthan, P. N. 2009 Differential evolution algorithm with strategy adaption for global numerical optimization. *IEEE Trans. Evol. Comp.* **13**, 398–417.

- Qin, Y., Kuczera, G. & Kavetski, D. 2016 Comparison of Newton-type and SCE optimisation algorithms for the calibration of conceptual hydrological models. *Australas. J. Water Resour.* **20** (2), 169–176. <https://doi.org/10.1080/13241583.2017.1298180>.
- Reddy, M. J. & Kumar, D. N. 2007 Multiobjective differential evolution with application to reservoir system optimization. *J. Comput. Civ. Eng.* **21** (2), 136–146. [https://doi.org/10.1061/\(ASCE\)0887-3801\(2007\)21:2\(136\)](https://doi.org/10.1061/(ASCE)0887-3801(2007)21:2(136)).
- Reddy, M. J. & Kumar, D. N. 2012 Computational algorithms inspired by biological processes and evolution. *Curr. Sci.* **103** (4), 1–11.
- Refsgaard, J. C., Havnø, K., Ammentorp, H. C. & Verwey, A. 1988 Application of hydrological models for flood forecasting and flood control in India and Bangladesh. *Adv. Water Resour.* **11** (2), 101–105. [https://doi.org/10.1016/0309-1708\(88\)90043-7](https://doi.org/10.1016/0309-1708(88)90043-7).
- Santhi, C., Arnold, J. G., Williams, J. R., Dugas, W. A., Srinivasan, R. & Hauck, L. M. 2001 Validation of the SWAT model on a large river basin with point and nonpoint sources. *J. Am. Water Resour. Assoc.* **37** (5), 1169–1188. <https://doi.org/10.1111/j.1752-1688.2001.tb03630.x>.
- Schardong, A. & Simonovic, S. P. 2015 Coupled self-adaptive multiobjective differential evolution and network flow algorithm approach for optimal reservoir operation. *J. Water Resour. Plann. Manag.* **141** (10), 4015015. [https://doi.org/10.1061/\(ASCE\)WR.1943-5452.0000525](https://doi.org/10.1061/(ASCE)WR.1943-5452.0000525).
- Scheerlinck, K., Pauwels, V., Vernieuwe, H. & Baets, B. D. 2009 Calibration of a water and energy balance model: recursive parameter estimation versus particle swarm optimization. *Water Resour. Res.* **45** (10), W10422. <https://doi.org/10.1029/2009WR008051>.
- Schumann, G. J. P., Neal, J. C., Voisin, N., Andreadis, K. M., Pappenberger, F., Phanthuwongpakdee, N., Hall, A. C. & Bates, P. D. 2013 A first large-scale flood inundation forecasting model. *Water Resour. Res.* **49** (10), 6248–6257. <https://doi.org/10.1002/wrcr.20521>.
- Sheffield, J. & Wood, E. F. 2008 Projected changes in drought occurrence under future global warming from multi-model, multi-scenario, IPCC AR4 simulations. *Clim. Dyn.* **31** (1), 79–105. <https://doi.org/10.1007/s00382-007-0340-z>.
- Sirsant, S. & Reddy, M. J. 2018 Reliability-based design of water distribution networks using self-adaptive differential evolution algorithm. *ISH J. Hydraul. Eng.* **24** (2), 198–212. <https://doi.org/10.1080/09715010.2017.1408038>.
- Sorooshian, S. & Gupta, V. K. 1983 Automatic calibration of conceptual rainfall-runoff models: the question of parameter observability and uniqueness. *Water Resour. Res.* **19** (1), 260–268. <https://doi.org/10.1029/WR019i001p00260>.
- Sorooshian, S., Duan, Q. & Gupta, V. K. 1993 Calibration of rainfall-runoff models: application of global optimization to the Sacramento Soil Moisture Accounting Model. *Water Resour. Res.* **29** (4), 1185–1194. <https://doi.org/10.1029/92WR02617>.
- Storn, R. & Price, K. 1997 Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11** (4), 341–359. <https://doi.org/10.1023/A:1008202821328>.
- Todini, E. 1996 The ARNO rainfall – runoff model. *J. Hydrol.* **175** (1–4), 339–382. [https://doi.org/10.1016/S0022-1694\(96\)80016-3](https://doi.org/10.1016/S0022-1694(96)80016-3).
- Tolson, B. A. & Shoemaker, C. A. 2007 Cannonville reservoir watershed SWAT2000 model development, calibration and validation. *J. Hydrol.* **337** (1), 68–86. <https://doi.org/10.1016/j.jhydrol.2007.01.017>.
- Vrugt, J. A., Gupta, H. V., Bouten, W. & Sorooshian, S. 2003 A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.* **39** (8), 1201.
- Vrugt, J. A., Robinson, B. A. & Hyman, J. M. 2009 Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Trans. Evol. Comput.* **13** (2), 243–259. <https://doi.org/10.1109/TEVC.2008.924428>.
- Wang, Q. J. 1991 The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Water Resour. Res.* **27** (9), 2467–2471. <https://doi.org/10.1029/91WR01305>.
- Xie, Z., Yuan, F., Duan, Q., Zheng, J., Liang, M. & Chen, F. 2007 Regional parameter estimation of the VIC land surface model: methodology and application to river basins in China. *J. Hydrometeorol.* **8** (3), 447–468. <https://doi.org/10.1175/JHM568.1>.
- Zhang, X., Srinivasan, R., Zhao, K. & Liew, M. V. 2009 Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrol. Process.* **23** (3), 430–441. <https://doi.org/10.1002/hyp.7152>.
- Zheng, F., Zecchin, A. C. & Simpson, A. R. 2012 Self-adaptive differential evolution algorithm applied to water distribution system optimization. *J. Comput. Civ. Eng.* **27** (2), 148–158. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000208](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000208).

First received 10 April 2020; accepted in revised form 21 June 2020. Available online 8 August 2020