

Towards a hybrid algorithm for the robust calibration of rainfall–runoff models

Umut Okkan and Umut Kirdemir

ABSTRACT

In this study, the hybrid particle swarm optimization (HPSO) algorithm was proposed and practised for the calibration of two conceptual rainfall–runoff models (*dynamic water balance model* and *abcde*). The performance of the developed method was compared with those of several metaheuristics. The models were calibrated for three sub-basins, and multiple performance criteria were taken into consideration in comparison. The results indicated that HPSO was derived significantly better and more consistent results than other algorithms with respect to hydrological model errors and convergence speed. A variance decomposition-based method – analysis of variance (ANOVA) – was also used to quantify the dynamic sensitivity of HPSO parameters. Accordingly, the individual and interactive uncertainties of the parameters defined in the HPSO are relatively low.

Key words | ANOVA, hybrid particle swarm optimization, hydrological model calibration, metaheuristics, optimization

Umut Okkan (corresponding author)
Umut Kirdemir
Department of Civil Engineering,
Hydraulic Division,
Balikesir University,
Balikesir,
Turkey
E-mail: umutokkan@balikesir.edu.tr

INTRODUCTION

Conceptual rainfall–runoff (CRR) models are frequently exerted by hydrologists and water resources planners to interpret and manage both climate change and anthropogenic activities affecting the watershed system. Since the defined parameters in these models are often not directly measured on account of the physical constraints and scaling problems, the effective implementation of modeling depends primarily on how and to what extent the model is calibrated (Goswami & O'Connor 2007; Zhang *et al.* 2009; Qin *et al.* 2017). Early calibration studies were performed manually due to the lack of inadequate usage of optimization tools. In parallel with the developments in the field of optimization sciences, the model calibrations have been carried out with automatic search approaches for nearly 30 years (Piotrowski *et al.* 2017). The first notable applications of automatic calibration were performed through basic techniques such as simplex and pattern search algorithm in the category of local search algorithms. In the same category, gradient-type algorithms (such as Gauss–Newton) have been also preferred for the calibration of CRR models

owing to their rapid convergence features (e.g., Hendrickson *et al.* 1988). Then, the non-convex nature of the calibration problem in the CRR models was demonstrated, and it was stated that the multiple local minima presence in the models could made the exploring of global optimal point complicated. It was also detected that the obtained results were hypersensitive to the initial parameter estimations (Duan *et al.* 1992). These constraints and increases in degrees of freedom based upon model structures have led hydrologists to global search algorithms with stochastic characteristics rather than local search algorithms since the 1990s. The metaheuristics such as genetic algorithm (GA), shuffled complex evolution (SCE) algorithm and simulated annealing (SA) algorithm have been the first global search techniques adapted to hydrological model calibration (Duan *et al.* 1992; Franchini & Galeati 1997; Wang 1997; Thyer *et al.* 1999).

Even though the GAs can overcome a great deal of trouble encountered in local search methods, the relatively high number of variables controlling the crossover, mutation and

selection operators can confuse the user in the decision-making phase. In addition to this matter, the fact that it does not guarantee the same global solution for each run has triggered the development of more practical and stable metaheuristics (Li *et al.* 2016). The differential evolution algorithm (DEA) and the particle swarm optimization (PSO) algorithm, which are the elementary alternatives to GAs, have been used in the relevant literature (e.g., Goswami & O'Connor 2007; Zhang *et al.* 2009; Piotrowski *et al.* 2017). As the number of nature inspired metaheuristics is increasing day by day, it will be unavoidable to adapt them to the scope of hydrological model calibration. For instance, relatively new algorithms such as artificial bee colony (ABC), ant colony optimization (ACO), artificial immune system (AIS), cuckoo search (CS), harmony search (HS), invasive weed colonization (IWA) and successful history-based DEA with linear population size reduction (L-SHADE) have been found to be implemented to water resources issues in a limited number (e.g., Zhang *et al.* 2009; Arsenault *et al.* 2014; Chen *et al.* 2015; Asgari *et al.* 2016; Piotrowski *et al.* 2019).

In the above paragraphs, the effectuations of algorithms ranging from local search ones to population-based metaheuristics are mentioned within the scope of the calibration of CRR models. Apart from a few exceptions, it is widely accepted that the derived results from algorithms such as GA, SCE, DEA and PSO are more stable than those of local search algorithms. However, some contradictions have been experienced when questioning the superiority of them against each other. In other words, a study in the literature states that a certain algorithm is fairly prosperous, while another study argues that different methods may produce similar results in both statistical and hydrological terms. For instance, Wang *et al.* (2010) used two variants of GA and the SCE to calibrate a distributed rainfall-runoff model and demonstrated that the calibration performance obtained from three algorithms was quite similar for a catchment in Taiwan. In a study conducted by Goswami & O'Connor (2007), algorithms such as GA, SA, SCE and Nelder-Mead Simplex were tried to calibrate the soil moisture accounting and routing the model for various basins in Ireland and China. Though their results did not imply statistical significance, it was reported that SA exhibited slightly better performance. Piotrowski *et al.* (2017)

tested both the diverse variants of PSO and DEA and much more sophisticated metaheuristics in the calibration of two CRR models but emphasized that the responses from employed algorithms did not differ significantly. In contrast, some algorithms have been claimed to have obvious advantages over other ones in various studies. In contrast to the findings of Goswami & O'Connor (2007), Cooper *et al.* (2007) underlined that SA was found to be weaker than other adopted algorithms. In a study performed by Zhang *et al.* (2009), the Soil and Water Assessment Tool (SWAT) was calibrated by GA, SCE, PSO, DEA and AIS algorithms. According to their results, if the control parameters are well adjusted and the number of fitness functions calls is increased, the appropriate results can be achieved through GA. However, it has also been proved in the same study that the PSO can give reasonable results with low population size and few runs. In the study prepared by Arsenault *et al.* (2014), 10 algorithms were utilized to calibrate parameter sets for three models on several basins. Among them, the weakest performances were offered with GA, PSO, DEA, CS and HS. The above-mentioned inferences will seem to continue to be made.

So, what could be the reasons of these dissimilarities? According to Kavetski & Clark (2011), incorrect tuning of the variables controlling the algorithm, the variability of the tested hydrological models, the study regions and thus the data attributes/quantities may lead to the aforementioned troubles. Besides, despite the robust structures of the metaheuristics, in a comprehensive work prepared by Tolson & Shoemaker (2007), they were shown to be markedly influenced by the dimension problem. This situation has been broadly discussed by various researchers, and it is stated that the calculation process with metaheuristics can move away from hydrology practice (Qin *et al.* 2018). According to Qin *et al.* (2017, 2018), the rapid convergence feature of gradient-based algorithms should be utilized to effectively reduce the computational costs, and they should be made more resolute to get rid of local minima traps. One of the strategies to be applied for this purpose is to run consecutively the gradient algorithm by assigning multiple random parameter values to the CRR model initially. Qin *et al.* (2017) indicated that Gauss-Newton techniques with the multi-start version produced more convergent solutions than SCE. In terms of the frequencies

of convergence to the global optimum, it is expressed that there are model-based uncertainties and extra effort is required to obtain a robust case.

Another remedy is to integrate global search properties of metaheuristics with local search capabilities of gradient algorithms (Karahan *et al.* 2013). In this context, a hybrid algorithm proposed by Karahan *et al.* (2013), which consists of the combination of HS and the quasi-Newton, yielded rather stable results in their flood routing problem. However, this type of hybridization has not yet been operated for the calibration of CRR models except for a few exemplary applications (e.g., Qin *et al.* 2018). This approach has shed light on us to give a new perspective to the calibration step of CRR modeling. Instead of choosing the best one among many algorithms, the idea advocated in the study is to propose a hybrid algorithm, which is both fast convergent, and has an adaptable content at the point of relieving the uncertainties mentioned. On the other side, it is necessary not to ignore the issue discussed by Kavetski & Clark (2011) to avoid an inadequate selection of control parameters of a selected algorithm. In this respect, it was considered that coupling a computationally intensive algorithm like HS with a gradient-local search algorithm would not be a practical procedure. Therefore, it was decided that providing local search capability to an elementary algorithm such as PSO, which has a relatively minimal number of control constants, would be more consistent. In this respect, hybrid PSO (HPSO), which was based on the combined use of PSO and the Levenberg–Marquardt (LM) algorithm, was formulated and compared with several algorithms in terms of both robustness and convergence. To the best of our knowledge, the usage of HPSO has not yet been identified in the context of the CRR model calibration in the literature. Thus, an attempt was made to fill a gap in previous studies. The remainder of the presented study is arranged as follows. The details about methods employed are given under ‘Methods’. The study area and data are briefly explained under ‘Study area and data’, while the results derived from the adopted algorithms and the conclusions are presented under ‘Results’ and ‘Conclusions’, respectively.

METHODS

Selected CRR models

In this study, two lumped CRR models, which are known as *dynamic water balance model (dynwbm)* and *abcd* model, were considered. Models have been widely used to simulate runoff under stationary and changed climatic conditions (e.g., Tekleab *et al.* 2011; Okkan & Kirdemir 2018; Shahid *et al.* 2018; Okkan & Kiyamaz 2020). These models, which require only monthly total precipitation (P) and potential evapotranspiration (EPOT) as inputs, include a series of conceptual soil moisture and groundwater storage functions. Both models utilized in the study contain similar storage elements as seen in Figure 1. In the models, the sum of the two components obtained as direct runoff (Q_{direct}) and baseflow (Q_{base}) provides the modeled runoff (Q_{modeled}).

Among these two models, the *dynwbm* developed by Zhang *et al.* (2008) is an expanded analogous of the hypothesis introduced by Budyko (1958). Although there are four parameters in the original model, an additional parameter has been added to its groundwater storage function by Okkan & Kirdemir (2018) so as to improve runoff prediction performances. In the model, P is made up of two components which are Q_{direct} and catchment rainfall retention X , respectively. In this partition, the parameter α_1 plays an active role, and a larger α_1 value results in more rainfall retention and less direct runoff. The model has also a parameter termed as maximum soil moisture capacity (S_{max}) representing the soil and vegetation characteristics of the basin. Besides, the parameter α_2 controls evapotranspiration opportunity y , which is assumed to be composed of the sum of soil moisture content S and actual evapotranspiration E_{act} . During the month i , the available water content W_i can be expressed by the sum of the soil moisture remaining from the previous month (S_{i-1}) and X , as well as by the sum of S_i , E_{act} and the amount of recharge (Rec) draining into groundwater storage. After the Rec is taken from the budget calculation, the balance equation is made for the groundwater storage G , which is postulated to represent linear reservoir behavior, using the d and e parameters together, and the baseflow is then predicted (Okkan & Kirdemir 2018).

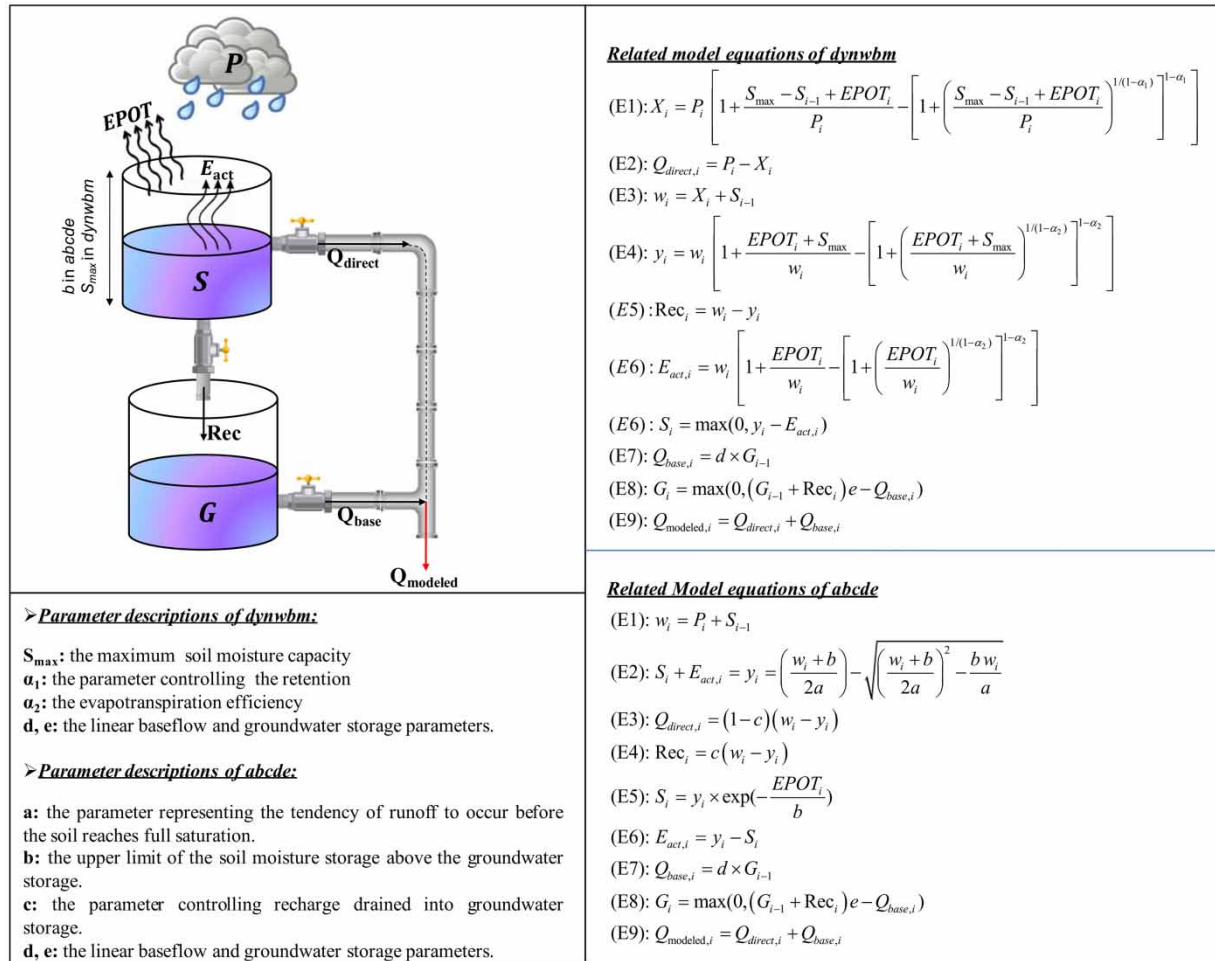


Figure 1 | Schematic representation of the utilized CRR models, the related calculation steps and the parameter descriptions.

In the *abcd* model developed by Thomas (1981), the available water content in the basin during the relevant month is considered to be equal to the sum of the S_{i-1} and P_i , with a coarser calculation compared with that of *dynwbm*. The evapotranspiration opportunity y is calculated depending on the parameters a and b , which represent the propensity of runoff occurrence and recharge before the soil being fully saturated and the soil saturation level, respectively (b in the *abcd* can be said to be the counterpart of S_{max} in *dynwbm*). While a larger a value results in less direct runoff and recharge amount, a larger b value brings about the contrary of this event. After a certain part of the water amount is reserved for y , the remaining portion is allotted to Q_{direct} and Rec components depending on the linear coefficient c . In the study, as in the previous model,

the modified groundwater storage function was regarded in the *abcd* model, and it was transformed into a five-parameter *abcde* (see Figure 1).

Metaheuristics used

Prior to proceeding with the proposed HPSO approach, the algorithms used in the comparison phase will be mentioned. Here, the selected GA, PSO, DEA, IWA and ABC algorithms are all population-based, and their parameter solutions refer to various cases which happened in nature such as propagation, hunting and selection. The population types for GA, PSO, DEA, IWA and ABC algorithms are represented by ‘chromosomes’, ‘particles’, ‘chromosomes or individuals’, ‘weed colonies’ and ‘bee colonies’, respectively.

In all applied algorithms, random solutions of parameters with regard to the number of population sizes defined are initially generated as follows:

$$x_{i,j} = x_j^{\min} + rand \times [x_j^{\max} - x_j^{\min}], \quad i = 1, 2, \dots, N_{\text{pop}}, \quad j = 1, 2, \dots, N_{\text{par}} \quad (1)$$

where x is the parameter vector, N_{pop} is the population size, N_{par} is the number of parameters to be calibrated, $rand$ is a random number that is uniformly distributed between 0 and 1, and x_{\min} and x_{\max} are the lower and upper limits of the parameter j , respectively.

The performance evaluation of probable solutions in the population is provided with the help of fitness (objective) function (Qin *et al.* 2017). In the study, traditional sum-of-squared errors (SSE) statistics were used as the fitness function as

$$\text{fitness}(x) = \text{SSE} = \sum_{i=1}^{T_{\text{cal}}} (Q_{\text{obs},i} - Q_{\text{modeled},i}(x))^2, \quad i = 1, 2, \dots, T_{\text{cal}} \quad (2)$$

where $Q_{\text{obs},i}$ is the observation at time i ; $Q_{\text{modeled},i}$ is the corresponding model prediction derived from model parameters x ; T_{cal} represents the data length used in the calibration period, in which model requires minimizing $\text{fitness}(x)$ with respect to x .

For the steps following the joint implementation of Equations (1) and (2), the operating mechanism of the algorithms are briefly described under the following sub-headings. Basic operators and control parameters of algorithms are also summarized in Table 1.

Genetic algorithm

GA is the oldest evolutionary technique that mimics the biological process in the computer environment by referring to Darwin's theory. The algorithm embodies the fundamental operators such as selection, crossover and mutation. These operations performed on chromosomes can be in the form of binary or real-coded (Tayfur 2017). In a work conducted by Chang & Chen (1998), it was emphasized that the real-coded version produces better quality results than the

binary-coded GA for the optimization of a flood control reservoir model. Similarly, in this study, the real-coded GA was also preferred. Following the initialization process, the two parental chromosomes that will be subjected to crossover is selected according to a specific rule. The tournament selection was chosen within this context. The crossover operator is then applied to obtain better offspring chromosomes through chromosomes determined by the selection process. The frequency of crossover is designated by the crossover probability (P_c). This process is repeated round of $0.5 \times P_c \times N_{\text{pop}}$ times in the iteration step. Although there are various crossover techniques in the real-coded GAs (Peltokangas & Sorsa 2008), the *arithmetic crossover method* based upon the principle of producing two offspring from two selected parents was considered in the study. In order to prevent new solutions from copying previous solutions and to increase diversity, the mutation operator is applied following the crossover. How many of the crossed individuals will be mutated depends on the mutation probability (P_m). The mutation formula used in this study can be adapted to any j gene on the chromosomes as well as to all genes (Michalewicz 1994).






$$\Delta x_j = \mu \times \phi_j \times [x_j^{\max} - x_j^{\min}] \times \tau_j, \quad j = 1, 2, \dots, N_{\text{par}} \quad (3)$$

$$x_{j,\text{mutated}} = x_{j,\text{crossed}} + \Delta x_j \quad (4)$$

where μ is a relative small factor that controls the Δx mutation quantity (μ was set to 0.05 as a result of various trials); ϕ is a random generated number having a range of $[-1, 1]$ for any parameter j ; τ represents a random integer taking only 0 or 1 (in case of $\tau = 0$, the related gene j is not exposed to any mutation process).

After the implementation of all operators in the relevant iteration, new individuals are added to the old population, and then they are reordered so as to eliminate the chromosomes having higher SSE values (*competitive exclusion*). For both GA and other used metaheuristics, the individual having the best fitness in the current population is stored as a global solution. The above-mentioned operations of GA are continued until the maximum generation number is reached (Tayfur 2017). For the details about the real-coded GAs, the report that was edited by Peltokangas & Sorsa (2008) can be viewed.

Table 1 | The brief descriptions about employed metaheuristic algorithms

Algorithm/population type	First proposer(s)	Operators or control parameters	Example reference	Employed technique, assigned value or formula
GA (real-coded)/ chromosomes	 Holland (1975), Goldberg (1989)	Selection Crossover Mutation	Peltokangas & Sorsa (2008)	Tournament arithmetical crossover ($P_c = 0.5$) Equations (3) and (4) ($\mu = 0.05$; $P_m = 0.1$)
PSO/particles	 Kennedy & Eberhart (1995)	Acceleration coefficient Velocity updating Inertia weight RW LDW NLDW CRW	Rathore & Sharma (2017)	$c_1 = c_2 = 2$ Equation (5) $\omega = 0.5 + 0.5 * rand$ $\omega = [(\text{iter}_{\max} - t) (\omega_{\max} - \omega_{\min}) / \text{iter}_{\max}] + \omega_{\min}$, $t = 1, 2, \dots, \text{iter}_{\max}$ $\omega_t = (\omega_{\max} - \omega_{\min}) (\text{iter}_{\max} - t)^3 / \text{iter}_{\max}^3 + \omega_{\min}$, $t = 1, 2, \dots, \text{iter}_{\max}$ $Z_i = 4z_{i-1} (1 - z_{i-1})$, $\omega_t = 0.5 * rand + 0.5 Z_i$, $t = 1, 2, \dots, \text{iter}_{\max}$
DEA/chromosomes	 Storn & Price (1997)	Mutation Crossover Selection	Xu <i>et al.</i> (2012)	Equation (6) in which $F = 0.5$ Non-uniform crossover in which $C_r = 0.5$ Greedy criterion
IWA/weed colony	 Mehrabian & Lucas (2006)	Initial population Production of seeds Spread of seeds Selection	Asgari <i>et al.</i> (2016)	$N_{\text{pop}}, 0 = 5$ Equation (7) in which $\text{Seed}_{\min} = 1$, and $\text{Seed}_{\max} = 5$ NLDW (same as in PSO) Competitive exclusion (weed population size $\leq N_{\text{pop}}$)
ABC/bee colony	 Karaboga (2005)	Function of employed bees Function of onlooker bees Function of scout bees	Karaboga & Basturk (2008)	$x_{\text{neighbor}} = x_{\text{old}} + \Delta x$ (similar to the use of Equations (3) and (4)) Equation (8) Limit = $0.5 * SN * N_{\text{par}}$

N_{pop} , population size; N_{par} , number of parameters to be calibrated; iter_{\max} , number of generation; rand, uniform random variable between [0, 1]; P_c , crossover probability; P_m , mutation probability; c , acceleration coefficient; ω , inertia weight; RW, random inertia weight; LDW, linear decreasing inertia weight; NLDW, nonlinear decreasing inertia weight; CRW, chaotic random inertia weight; CR, crossover constant in DEA; F, mutation factor in DEA; $N_{\text{pop}, 0}$, initial weed colony size in IWA; Seed_{\min} , minimum number of seeds produced; Seed_{\max} , maximum number of seeds produced; SN, number of food sources in ABC; limit, a limit value used in scout bee step of ABC.

Particle swarm optimization

The PSO is a metaheuristic algorithm proposed by Kennedy & Eberhart (1995), inspired by the hunting behavior of bird swarms. It begins with the random distribution of the swarm of birds into the solution space using Equation (1). The particles in the swarm try to iteratively determine the positions x pertaining to the food source in space by the velocity vector v expressed in Equation (5). Initially, $v(t=0)$ can be assumed as zero for all particles. In the iterations, the calculated velocity vector is added to the position vector stored in the previous iteration step to update the position of the particles.

$$v_{ij}(t+1) = \omega \times v_{ij}(t) + rand \times c_1 \times (pb_{ij}(t) - x_{ij}(t)) + rand \times c_2 \times (gb_j(t) - x_{ij}(t)) \quad (5)$$

In Equation (5), pb_i represents the best position that the i th particle has ever reached, while gb represents the global best solution among pb matrix with the $N_{pop} \times N_{par}$ dimension (where $t=0$ to $iter_{max}$; $i=1$ to N_{pop} ; $j=1$ to N_{par}). $rand$ is the same variable that is previously defined in Equation (1). ω is the inertia weight controlling particle's velocity. Also, the acceleration coefficients c_1 and c_2 are both usually set to 2.0. $pb(t=0)$ is considered equal to the randomly generated solution matrix that is formulated in Equation (1). If the fitness value of the solution obtained after the position update for any particle i is improved compared with that of the previous iteration, the corresponding row of the pb is replaced with the new position. A similar approach is repeated iteratively for the gb as well. The operations described above are preceded until the iterations are completed (Tayfur 2017). In the operation of Equation (5), four inertia weight strategies, which are mentioned in Table 1, are examined referring to Rathore & Sharma (2017). For example, the variation of inertia weights over 500 iterations according to the random weight (RW), the linear decreasing weight (LDW), the nonlinear decreasing weight (NLDW) and the chaotic random weight (CRW) is shown in Figure 2(a). In the study, the inertia weight functions were requested to produce their values in a common range, and ω_{max} and ω_{min} were taken as 1.0 and 0.001, respectively.

Differential evolution algorithm

The DEA is a metaheuristic algorithm introduced by Storn & Price (1997) and involves similar operators with GA. Contrary to GA, the process of producing new individuals in DEA is performed with far fewer chromosomes, and the mutation operator is applied to the whole individuals of population prior to crossover. Once the initialization is made through Equation (1), aside from the i th chromosome, three chromosomes with different row numbers (x_a , x_b and x_c) are randomly selected from the available population for the mutation process. This process is operated with the help of mutation factor F as follows (Tayfur 2017).

$$x_{mutated,i} = x_c + F \times (x_a - x_b), \quad (6)$$

$$a, b, c \in \{1, 2, \dots, N_{pop}\}, \quad a \neq b \neq c \neq i$$

Xu *et al.* (2012) have stated that F -values between 0.5 and 1.0 provide a similar contribution on the simulations. Besides, the effectiveness of the mutation is yielded by a non-uniform strategy applied during the crossover stage. In the crossover, following the random number ($rand$) derivation for each defined j gene in the present chromosomes, it is checked whether the $rand_j \leq C_r$ is ensured. In that case, the state of using the j gene of the vector obtained by Equation (6) occurs. Otherwise, the j gene of the corresponding chromosome is unchanged. Here, C_r is a crossover constant that gives plausible results with its values between 0.5 and 1.0 (Xu *et al.* 2012). Finally, a greedy criterion is implemented by means of Equation (2) to determine the chromosome that will be transmitted to the new generation. If the fitness function calculated from the candidate chromosome, which is consecutively subjected to mutation and crossover, improves compared with that of the old one, it is stored for the next generation, and the old solution is removed, or else the location of the old vector is preserved. The operations stated above are continued until the maximum generation number is reached (Tayfur 2017).

Invasive weed algorithm

The IWA is another metaheuristic proposed by Mehrabian & Lucas (2006), inspired by the invading colonization of weeds in field. In IWA, which has operators such as seed production, seed spreading and selection, a small amount of seed is

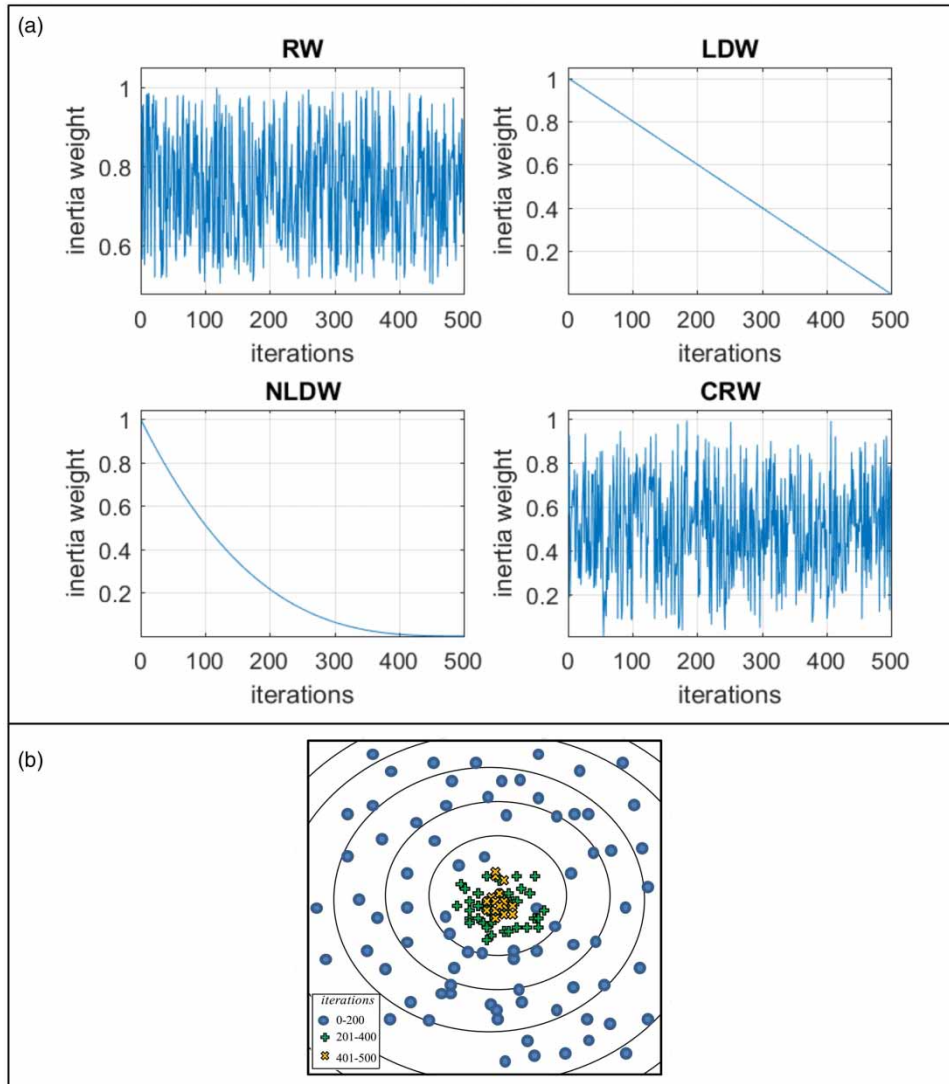


Figure 2 | (a) The change of inertia weights under the different approaches in the PSO and (b) the change in the distance of the seeds to the origin point during the iterations in the IWA ($iter_{max} = 500$).

initially dispersed to the field using Equation (1) (the initial seed population $N_{pop,0}$ was set to 5). To mimic the propagation process of seeds, the weed colony is allowed to produce seeds proportional to the fitness values stored (Asgari *et al.* 2016). However, this procedure is limited between the maximum and minimum seed numbers ($Seed_{max}$ and $Seed_{min}$) as follows:

$$Seed_i = \text{round} \left[Seed_{min} + (Seed_{max} - Seed_{min}) \cdot \left(\frac{fitness_i - fitness_{worst}}{fitness_{best} - fitness_{worst}} \right) \right] \quad (7)$$

where ‘round’ is the function that rounds the real value to closest integer; $fitness_{worst}$ and $fitness_{best}$ are the maximum and minimum fitness values in the colony, respectively; $fitness_i$ and $Seed_i$ are, respectively, the fitness function of i th weed and the number of seeds to be produced.

The seeds produced by Equation (7) are then randomly distributed to the field again. Depending on the number of $Seed_i$, a uniform random solution is generated in the neighborhood of the parental solution with the line number i . In this study, this process resembles the mutation operation defined in Equation (3), but is applied to the entire

parameter vector. The time-related reduction of the distance of the newly produced seeds to their parents is iteratively adjusted by the NLDW stated in Table 1 (see Figure 2(b)). After seed production and spreading, it is checked whether the total population size exceeds P_{\max} , the maximal population size of colony ($P_{\max} = N_{\text{pop}}$ in other algorithms). If such a situation occurs, only the weeds having better fitness values survive, depending on the competitive exclusion procedure. The operations summarized above are managed with regard to the maximum number of iterations. In the study, sensitivity analysis made by Asgari *et al.* (2016) is referred when assigning the control variables of IWA (see Table 1).

Artificial bee colony

The ABC proposed by Karaboga (2005) is based on the foraging behavior of honey bees. It uses three kinds of bees, which are namely employed bee, onlooker bee and scout bee, as operators. In the first step of the algorithm, depending on the number of food sources SN, the random solution matrix is generated by Equation (1), and the quality of the nectar obtained from sources is detected with Equation (2) (fitness is represented by the quality of the stored nectar). SN is the equivalent of N_{pop} in other algorithms, while bee population size is $2 \times \text{SN}$. Afterwards, employed bees produce neighbor solutions around existing food sources x_i (where $i = 1, 2, \dots, \text{SN}$). This process is analogous to the mutation in GA, expressed in Equations (3) and (4), but differs in terms of its replication for all food sources (Chen *et al.* 2015). If the new i source experienced in the x_i neighborhood is better in terms of nectar, this solution is stored and the failure counter is zeroized for the corresponding source. Otherwise, the failure counter is incremented by one for that source. Then, the onlooker bees observe the dancing of the employed bees that have arrived in the hive and tend to new sources based on a given probability value. According to Babayigit & Ozdemir (2012), Equation (8) can be preferred instead of the roulette wheel style probability calculation to give a chance for low-quality solutions as well.

$$p_i = (1 + zf_i) \exp(-zf_i), \quad i = 1, 2, \dots, \text{SN} \quad (8)$$

where zf is the normalized fitness value.

Once $p_i > \text{rand}$, the onlooker bee moves to this source and produces neighboring solutions, as performed in the phase of the employed bees. Similar to other phases, it is once again decided which solutions will be stored and the condition of the failure counter is. If the failure counter in a source exceeds a *limit* value set by the user, the employed bee using that source is assigned as a scout bee and search for another location in space by means of Equation (1). If the nectar quality after scout bee phase is improved compared with the old one, this solution derived is stored and the failure counter is zeroized again. All the above phases are iterated until the maximum generation number is achieved (Karaboga & Basturk 2008; Chen *et al.* 2015).

Hybrid particle swarm optimization

In this section, the integration process of the LM algorithm with PSO, which only requires c_1 and c_2 coefficients and inertial weight, is described. Essentially, the proposed HPSO is not very complicated. The hybrid approach is based on the introduction of the *gb* vector obtained by the standard operation of the PSO as an input to the LM in each iteration step. A similar procedure was previously operated by Zhang *et al.* (2007) and Nawi *et al.* (2014) in the training of artificial neural networks. Different hybrid versions of PSO were also proposed for job shop scheduling problem (Sha & Hsu 2006) and multi-modal functions (Kao & Zahara 2008). Both HPSO and other algorithms are coded in the MATLAB environment.

In the implemented HPSO, the LM addition needs the Jacobian matrix (J) of model errors e with respect to each j parameter in the vector *gb*.

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial x_1} & \frac{\partial e_1}{\partial x_2} & \dots & \frac{\partial e_1}{\partial x_j} \\ \frac{\partial e_2}{\partial x_1} & \frac{\partial e_2}{\partial x_2} & \dots & \frac{\partial e_2}{\partial x_j} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_{T_{\text{cal}}}}{\partial x_1} & \frac{\partial e_{T_{\text{cal}}}}{\partial x_2} & \dots & \frac{\partial e_{T_{\text{cal}}}}{\partial x_j} \end{bmatrix}_{T_{\text{cal}} \times N_{\text{par}}}, \quad (9)$$

$$j = 1, 2, \dots, N_{\text{par}}$$

where T_{cal} is data length in calibration.

After the J matrix is formed with a finite difference approach (preferably forward difference) in each iteration step, updating of gb is performed through Equation (10), which is based on an approximate solution of the Hessian matrix.

$$gb_{t+1} = gb_t - [J_t^T J_t + \lambda_t I]^{-1} J_t^T e_t, \quad t = 1, 2, \dots, iter_{max} \quad (10)$$

where the e_t represents the error vector with the $T_{cal} \times 1$ dimension, gb_t is the global best solution derived from PSO, while gb_{t+1} is the LM-based operated global best solution, λ denotes the Marquardt parameter that is adaptively adjusted.

In Equation (10), λ is multiplied by a certain decay rate β when fitness decreases in a certain iteration step, and it is

divided by β when fitness increases in another step. As suggested in the literature, λ_0 , β and the maximum limit value of λ (λ_{max}) were set to 0.01, 0.1 and 10^{10} , respectively. Since the formulations of RW, LDW, NLDW and CRW used as inertia functions in PSO are also performed for the hybrid approach, the hybrid ones are mentioned as HPSO1, HPSO2, HPSO3 and HPSO4, respectively. The flowchart of the proposed hybrid strategy is given in Figure 3.

Performance measures for algorithms

In order to test the stability and convergence performance of algorithms statistically, each of these algorithms should be run many times. Similar to the works of Tigkas *et al.* (2016) and Piotrowski *et al.* (2017), it was decided to run the algorithms 30 times. Following the operation of

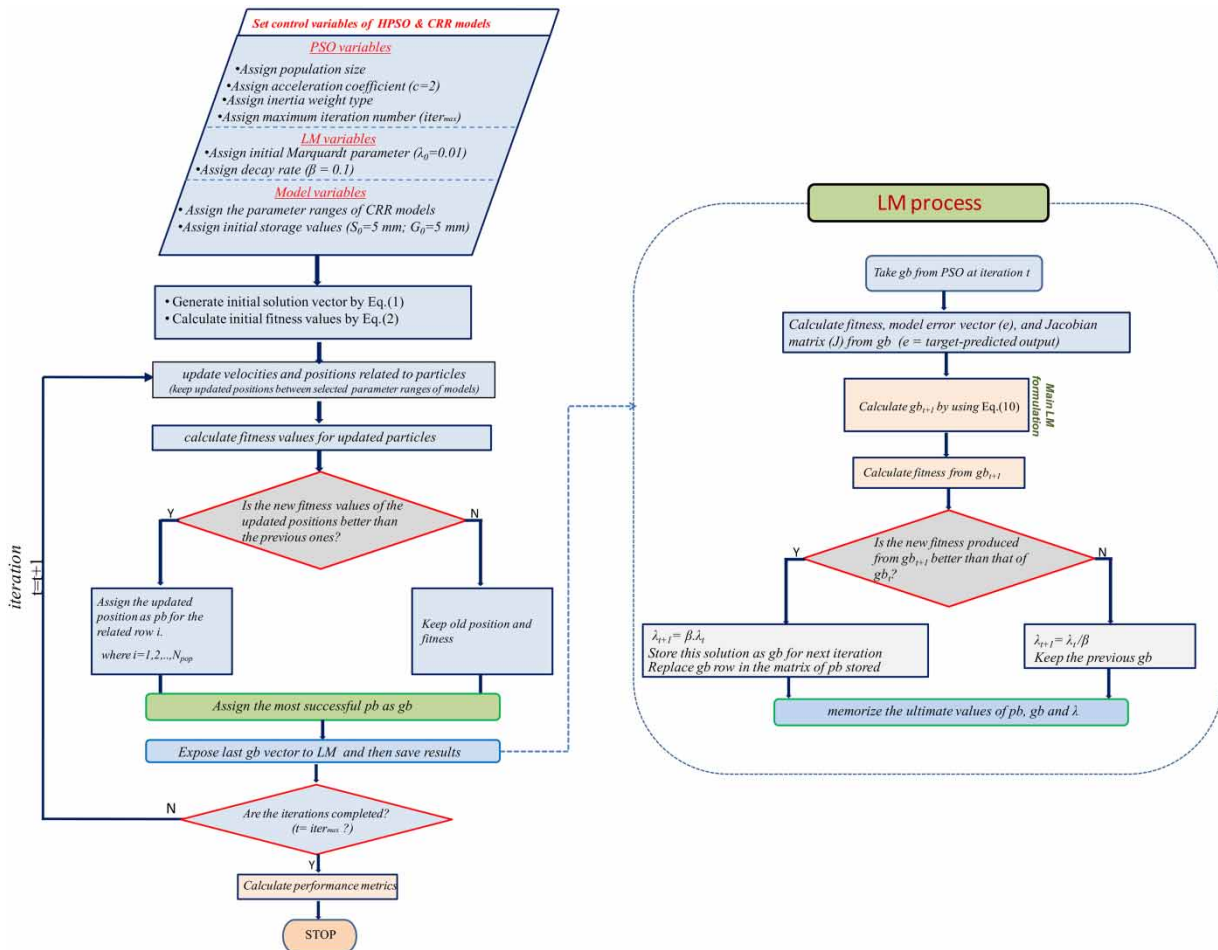


Figure 3 | Flowchart of the proposed HPSO algorithm.

algorithms with multiple runs, stored fitness values at last iteration were primarily used to explicate mean and standard deviation statistics. Thereafter, the geometric mean convergence rate formula recommended by He & Lin (2016) was implemented in the study as follows:

$$CR = 1 - \left[\frac{fitness_{desired} - fitness_1}{fitness_{desired} - fitness_0} \times \frac{fitness_{desired} - fitness_2}{fitness_{desired} - fitness_1} \times \dots \times \frac{fitness_{desired} - fitness_h}{fitness_{desired} - fitness_{h-1}} \right]^{1/h} \quad (11)$$

where $fitness_{desired}$ is equal to zero for the targeted SSE value, $fitness_0$ is the best fitness value among the produced ones through Equation (1), $fitness_h$ is the stable fitness value that is assumed to not be significantly changed after h th iteration.

When a threshold value ϵ is taken, the required iteration number h can be determined by using the following equations:

$$sign_t = \begin{cases} 0, & (fitness_t - fitness_{iter_{max}}) \leq \epsilon = 10^{-3} \\ 1, & (fitness_t - fitness_{iter_{max}}) > \epsilon = 10^{-3} \end{cases}, \quad (12a)$$

$$t = 1, 2, \dots, iter_{max}$$

$$h = \left(\sum_{t=1}^{iter_{max}} sign_t \right) + 1 \quad (12b)$$

STUDY AREA AND DATA

In the study, Gediz Basin, which represents a major part of agricultural activities of the Aegean region in Turkey, was selected as a study area. CRR models were applied on three sub-basins that are differing by locations and hydro-meteorological properties. These sub-basins, which are represented by Acisu (E05A23), Hacıhidir (D05A28) and Hacıhaliller (D05A38) flow gauging stations, are located on the main stream of Gediz, Gordes creek and Nif creek, respectively (Figure 4). Station DO5A28 is also located on the branch that feeds the Gordes Dam reservoir. While the drainage area of the station is 808.2 km², the drainage area of the dam is 1,045.4 km². When switching to the reservoir

inflow volumes, the flow volumes of the station DO5A28 are multiplied by 1,045.4/808.2. When topographically examined, it is seen that both Hacıhidir and Acisu sub-basins fall within the mountainous area with higher altitudes than that of Hacıhaliller sub-basin.

Since the 30-year window regarding the current climate normals is 1981–2010, which is also called a reference climate period, this period has been also considered in this study. Among the related sub-basins, the Hacıhaliller sub-basin received the highest annual precipitation with the value of 733 mm, while the annual precipitation values of 573 and 505 mm were observed in Hacıhidir and Acisu sub-basins, respectively. The highest mean annual temperature value was observed in Hacıhaliller sub-basin as 16.8°C, while it was observed as 13.2°C in the rest of the study area. According to the data obtained from flow gauging stations, the highest flow depth was observed in the Hacıhidir sub-basin (~129 mm), such that the lowest *EPOT* is predicted in the related sub-basin with the value of 964 mm. Even if the Hacıhaliller sub-basin stands for the wettest part among the sub-basins, its flow potential decreases on account of relatively higher temperature and *EPOT* values.

The all-flow measures obtained from The General Directorate of State Hydraulic Works (DSI) for the 1981–2010 water year period were then converted to millimeters to be used in modeling. Monthly time-scale Thiessen-weighted precipitation series (*P*) obtained from the data of Turkish State Meteorological Service (MGM) was prepared as a first model input. While monthly temperature and relative humidity data were compiled from the meteorological stations operated by MGM, ERA-Interim reanalysis data sets having 0.75° × 0.75° resolution were used for other variables (wind speed and solar radiation) needed for the Penman–Monteith equation to obtain areal mean *EPOT* estimations used as a second model input. These reanalysis data are served by the European Center for Medium-Range Weather Forecasts for several categories (<http://apps.ecmwf.int/datasets/>). When examined in terms of the drought index based on the *P/EPOT* ratio on the annual time scale, Acisu has semi-arid climate characteristics, while arid–semi-humid climate is prominent for the remaining sub-basins. For the land use/land cover classification of the region, readers can be directed to Droogers & Kite (1999). Further details about the study region and ERA-Interim grids that almost uniformly

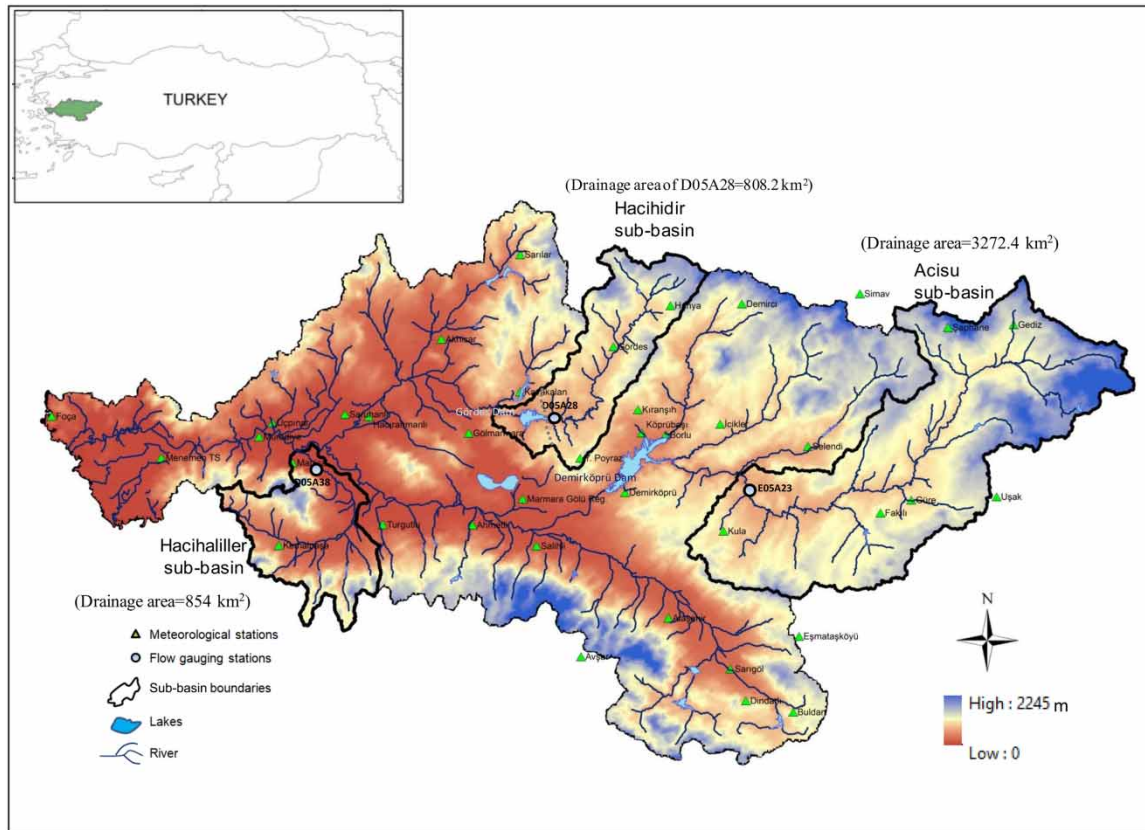


Figure 4 | Locations of sub-basins and hydro-meteorological stations.

encompass the basin can be found in Okkan & Kirdemir (2018) and Okkan & Kiyamaz (2020).

RESULTS

Performances of calibration algorithms

In the parameter optimization studies carried out with meta-heuristics, the population size (N_{pop}) and the maximum iteration number ($iter_{max}$) to be used in the cycle of algorithms assuredly influence the calibration. Since the convergent and stable features of the proposed HPSO technique were advocated within the scope of the study, it was checked over whether the computational intensity diminishes with this approach by using fewer iterations and population sizes. For this purpose, in all algorithm experiments, N_{pop} and $iter_{max}$ were fixed to 20 and 500,

respectively. Additionally, a wide range of parameters have been defined for CRR models to make algorithms further force in the process of finding the global solution. The range of parameters S_{max} (in *dynwbm*) and b (in *abcde*) is 10–1000 mm, while the range 0.01–0.99 is assigned for parameters α_1 , α_2 , d (in *dynwbm*) and a , c , d (in *abcde*). The range of 0.5–0.99 is chosen for the common parameter e in both models. These parameter ranges are also within the physically possible limits, and there is no need to define a comprehensive penalty function within the fitness function. Not only the initial solutions generated by Equation (1) but also the iterative solutions within the algorithm loops are kept between these closed ranges. In the study, the split-sample procedure, where the observed runoff series were divided into two equal parts for calibration and validation, was implemented (Refsgaard & Knudsen 1996; Zhang *et al.* 2008). According to this, the data covering the water period of 1981–1995 compiled for

sub-basins were evaluated during the parameter calibration stage of the CRR models, while data covering the water period of 1996–2010 were used in the validation.

In some daily hydrological model applications, initial soil moisture can also be defined as a parameter to be calibrated. One such application for the daily GR4J model (*modèle du Génie Rural à 4 paramètres Journalier*) was performed by Mostafaie *et al.* (2018). As the models used in the study are monthly time-scaled ones, they are not overly sensitive to the initial soil moisture and the initial groundwater store values. In the first month of the water year period (at the end of the dry period), due to the low rainfall, the initial storage values in the conceptual reservoirs of CRR models (S_0 and G_0) have been chosen close to almost zero (5 mm was fitted for these initial values).

As mentioned in the previous sections, the algorithms were run 30 times according to their control variables specified in Table 1, and the CRR models' parameter ranges specified, and the results were then stored individually. For each CRR model, a total of 13 different algorithms, which are four different inertia weight variants of PSO and HPSO, and GA, DEA, IWA, ABC and LM, were operated. As the implementation of them on two CRR models for three sub-basins produced 78 graphical results, only some convergence graphs produced with *dynwbm* exerted to the Acisu sub-basin have been given as an example in Figure 5. From Figure 5, it can be clearly seen that some algorithms have relatively different convergence and stabilization properties. In this example, when focusing on the last 50 iterations of 30-run average values, it can be monitored that especially GA and PSO variants do not provide stable results and are possibly sensitive to initial solutions. On the other hand, DEA, IWA and ABC produced more stable results than other metaheuristics, but they displayed late convergence. The same example also proves that the HPSO shows both stable and rapid convergent performances. Instead of graphically interpreting the completed analyses for two CRR models applied to three sub-basins, descriptive statistics of the fitness values realised at the last iteration (the statistics concerning 30 fitness values generated under each algorithm) are extracted in Table 2. In order to quantify the number of iterations required and the convergence performance of the algorithms against different initial solutions, the formulas h and CR expressed in the section 'Performance

measures for algorithms' were implemented. Following being stored of CR and h values obtained from the related cycle for each run, the mean statistics of these indices calculated from 30 runs are also stated in Table 2.

According to Table 2, there are noteworthy findings that are jointly determined from all cases. For example, though LM among all implemented algorithms is the most convergent kind, it has been the worst performer in terms of mean fitness statistics. The standard deviation statistics extracted from this application have proved that LM has not produced robust results. In fact, the solutions derived from this algorithm are not surprising as the gradient algorithms are known to be affected by the matter of local minima. On the other hand, it can be also seen that DEA, IWA and ABC algorithms averagely provide more appropriate fitness statistics than those of GA and PSO. What is remarkable is that HPSO variants are much more prominent in all sub-basins exposed to two different CRR models in the context of achieving the best stabilization. The standard deviation statistics of stored fitness values of the existing HPSO simulations, which are about 10^{-9} – 10^{-12} , emphasize that the variability between their global solutions is quite minimal. Moreover, as the study focused on the convergence performance of the calibration algorithms, an index based upon the mutual evaluation of mean fitness statistics and means CR ratios has been proposed. This index expressed in Equation (13) is based on the typical standardization of the results derived and has been exerted for all algorithms, except for LM, which is problematic alone. In Equation (13c), p_{geomean} is the geometric mean probability value calculated from the individuals probabilities p_1 and p_2 corresponding to mean CR and mean fitness, respectively.

$$p_{1,k} = p_{\text{norm}} \left[\frac{\text{CR}_{\text{mean},k} - \overline{\text{CR}_{\text{mean}}}}{\text{std}(\text{CR}_{\text{mean}})} \right] \quad (13a)$$

$$p_{2,k} = p_{\text{norm}} \left[\frac{\text{inv } f_{\text{mean},k} - \overline{\text{inv } f_{\text{mean}}}}{\text{std}(\text{inv } f_{\text{mean}})} \right] \quad (13b)$$

$$p_{\text{geomean},k} = \sqrt{p_{1,k} \times p_{2,k}} \quad (13c)$$

where k symbolizes the line number of a total of 12 algorithms except for LM ($k = 1, \dots, 12$, and the order of the

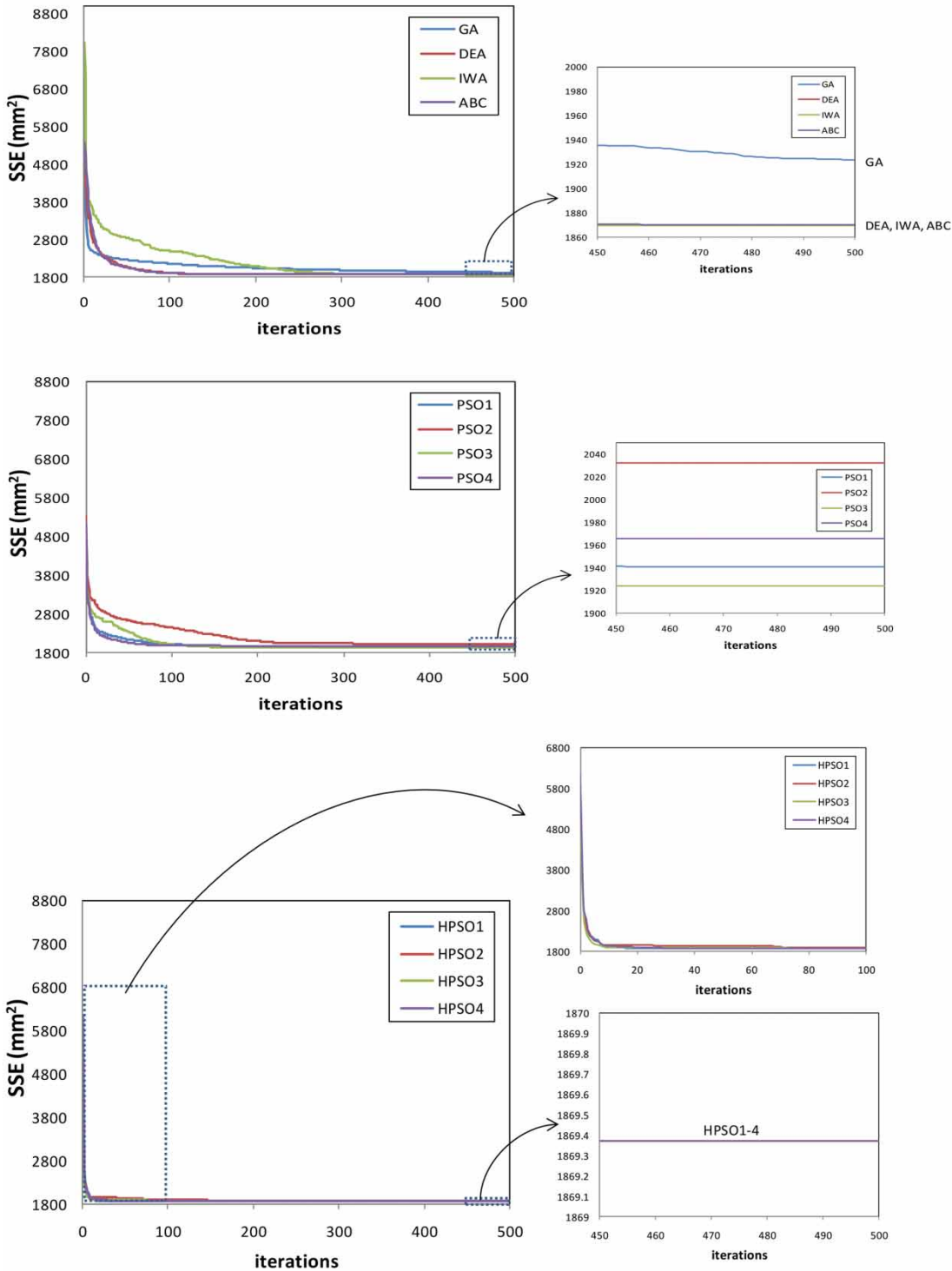


Figure 5 | Convergence plots compiled from employed algorithms for *dynwbn* application in the Acisu sub-basin case (each algorithm was repeated 30 times; 30-run average values during iterations are shown).

Table 2 | The convergence and stability indicators derived from the calibration processes of CRR models operated under different algorithms

CRR model	Algorithms	Acisu sub-basin				Hacihidir sub-basin				Hacihaliller sub-basin			
		CR _{mean}	h _{mean}	Fitness _{mean}	Fitness _{STD}	CR _{mean}	h _{mean}	Fitness _{mean}	Fitness _{STD}	CR _{mean}	h _{mean}	Fitness _{mean}	Fitness _{STD}
<i>dynzobm</i>	GA	0.0019	466	1923.5122	70.6603	0.0019	453	7031.2966	14.3403	0.0031	413	3140.6994	40.1867
	PSO1	0.0022	456	1940.9631	274.7085	0.0018	422	7073.4997	254.0512	0.0028	445	3195.2268	200.4081
	PSO2	0.0033	285	2032.1258	304.2534	0.0031	255	7030.2938	24.1281	0.0042	299	3207.5234	227.3971
	PSO3	0.0045	188	1924.1132	124.4991	0.0059	145	7086.7903	249.1943	0.0063	194	3232.2214	420.0500
	PSO4	0.0058	172	1965.7949	253.8102	0.0080	108	7101.7418	280.9920	0.0073	178	3132.2236	89.5593
	DEA	0.0046	269	1869.7976	1.4168	0.0054	167	7014.7642	0.0028	0.0054	238	3108.6902	1.122 × 10⁻⁸
	IWA	0.0028	481	1869.3754	0.0022	0.0026	487	7014.7735	0.0061	0.0040	482	3108.6953	0.0019
	ABC	0.0024	437	1870.1011	0.4775	0.0018	442	7015.4136	0.3684	0.0033	420	3109.5468	0.5010
	LM	0.1662	115	2759.3692	3756.0965	0.1575	180	8584.6290	2181.1152	0.1014	46	4317.7102	3689.1102
	HPSO1	0.1287	9	1869.3713	3.331 × 10⁻¹⁰	0.1446	6	7014.7637	1.208 × 10⁹	0.1720	8	3108.6902	1.569 × 10⁻¹⁰
	HPSO2	0.1242	17	1869.3713	5.802 × 10⁻¹²	0.1284	8	7014.7637	4.099 × 10⁻¹²	0.1924	8	3108.6902	4.580 × 10⁻¹¹
	HPSO3	0.1245	10	1869.3713	3.222 × 10⁻¹²	0.1125	8	7014.7637	4.851 × 10⁻¹²	0.1826	8	3108.6902	3.592 × 10⁻¹¹
	HPSO4	0.1364	10	1869.3713	2.885 × 10⁻¹²	0.1354	6	7014.7637	4.137 × 10⁻¹²	0.1855	7	3108.6902	2.464 × 10⁻¹¹
	<i>abcde</i>	GA	0.0019	458	2403.6905	50.2396	0.0024	445	8002.6599	267.2675	0.0028	484	3713.8467
PSO1		0.0025	366	2540.1567	288.7886	0.0043	305	8361.0920	1446.6445	0.0063	335	4549.0336	2395.7666
PSO2		0.0038	253	2488.8802	142.7987	0.0045	219	8133.0850	278.0582	0.0047	240	4102.7604	909.4121
PSO3		0.0060	154	2710.0375	893.5297	0.0090	124	8369.8447	1417.1795	0.0112	134	4010.9025	418.5354
PSO4		0.0092	121	2507.0071	313.3282	0.0119	102	8033.0716	153.9627	0.0139	128	3782.1565	443.0509
DEA		0.0035	279	2346.4884	24.7892	0.0056	218	7923.4904	0.0415	0.0061	236	3512.4101	0.2065
IWA		0.0033	484	2341.9125	0.0026	0.0029	495	7926.3310	3.4529	0.0043	485	3512.3752	0.0080
ABC		0.0021	454	2344.8903	2.5837	0.0026	443	7925.2631	0.9866	0.0033	449	3518.5462	4.6468
LM		0.1155	209	2947.3623	1230.1110	0.0327	159	9984.8427	5121.5852	0.0208	348	8811.6827	4761.7685
HPSO1		0.0548	26	2341.9071	8.431 × 10⁻¹¹	0.0251	46	7923.4795	7.090 × 10⁻⁹	0.0714	32	3512.3578	1.466 × 10⁻¹⁰
HPSO2		0.0559	26	2341.9071	5.401 × 10⁻¹²	0.0261	49	7923.4795	1.189 × 10⁻¹¹	0.0571	29	3512.3578	9.549 × 10⁻¹²
HPSO3		0.0491	28	2341.9071	3.817 × 10⁻¹²	0.0231	50	7923.4795	1.454 × 10⁻¹¹	0.0764	32	3512.3578	9.822 × 10⁻¹²
HPSO4		0.0546	19	2341.9071	2.856 × 10⁻¹²	0.0269	42	7923.4795	7.699 × 10⁻¹²	0.0732	21	3512.3578	7.069 × 10⁻¹²

CR_{mean} versus h_{mean} are the mean of CR and h values obtained over 30 runs. Fitness_{mean} and fitness_{STD} are the mean and standard deviation of fitness values in the 500th iteration row obtained over 30 runs (in mm²). The five best values in the table were denoted in bold character and the best value among them was underlined.

second column of Table 2); CR_{mean} is the mean convergence rate of k th algorithm; $\text{inv}f_{\text{mean}}$ is expressed by $1/\text{fitness}_{\text{mean}}$; std is the standard deviation function; top bar symbol represents the arithmetic mean for the related index; p_{norm} gives the cumulative distribution function of the standard normal distribution.

With the approach formulated as above, the performances of the algorithms in the corresponding model and sub-basin variations were rated from 0 to 1 and are stated in Figure 6. After the rank statistics of the p_{geomean} values were taken for each sub-basin and the CRR model, the general grading of utilized algorithms is introduced in Figure 7, in which p_{geomean} is inversely proportional to rank statistics. With the help of the mean statistics of the ranks compiled from the sub-basins, it is tested in Figure 7(a) whether the algorithms react distinctively depending on the CRR models. According to Figure 7(a), excluding a couple of PSO variants, there was no significant difference in the responses of the algorithms to the models. Here, the high determination coefficient detected in inter-model scattering has supported this remark ($R^2 = 0.901$). The detection made at this stage has also provided more comfortable comparison of the algorithms within themselves. Then, the mean statistics of the rank numbers arranged under all cases are indicated in Figure 7(b) for each algorithm.

Indeed, Figure 7 summarizes the indications presented in Table 2 and Figure 6. From Figure 7(b), it can be seen that the CRW inertia weight is more compatible with the PSO. Even though this inference is parallel to that advocated by Rathore & Sharma (2017), all PSO types have settled at the last ranks along with GA. The remaining metaheuristics (DEA, IWA and ABC) have produced relatively more reliable results than PSO and GA. At the same time, the most momentous diagnosis is that all HPSO variants appear in the first four and are conspicuously steady in all existing conditions. Also, HPSO4, operating with CRW, has notably come to the forefront compared with other ones since it has shown similar responses in both CRR models and has been treated as stable and well convergent. While LM and PSO algorithms individually give almost the worst results in all applications, it is an essential detection that the best solutions are achieved under each variation in case of their combining. This supports why we couple

derivative algorithms and metaheuristics for an enhanced calibration.

Surely, it would be inevitable that LM combined with other metaheuristics would have the same impression. In the context of being exemplary, to what extent the HDEA approach made by the hybridization of DEA with LM has a resemblance to HPSO4 has been questioned. As the mean fitness values obtained from both hybrid algorithms after 30 runs are almost equal, the mean convergence rates and the standard deviations of the fitness values at the last iteration of completed runs are given in Figure 8. It can be seen from Figure 8 that there are no marked variations between the convergence performances of HPSO4 and HDEA on model calibrations. The standard deviations of the fitness values are in the range of 10^{-10} – 10^{-12} in both hybrids as well. In those cases, HPSO4 with fewer control parameters has been proved to be a more reasonable method again. Furthermore, it will be preferred in various calibration practices since it has carried out its convergence with low population size and a moderate iteration number.

Quantification of dynamic sensitivity of HPSO parameters

The optimization algorithm with less parameter uncertainty would be more preferable. To confirm this idea, in this section, a variance decomposition-based method – analysis of variance (ANOVA) – was used to determine whether the dynamic sensitivity of the parameters of the HPSO meaningfully changed compared with that of the classical PSO. Qi *et al.* (2016) previously demonstrated that ANOVA effectively revealed the dynamic sensitivity of SCE parameters in the search processes, including individual impacts of parameters defined and their interactive contributions. The ANOVA was also used to decompose uncertainties in climate projections arising from global circulation models and emission scenarios (Yip *et al.* 2011). In the study, all sources of potential uncertainty in the parameters controlling algorithms were expressed in terms of variances. Accordingly, the total uncertainty (T) connected with PSO/HPSO sensitivity analysis can be divided into four partitions due to the internal variability and individual/interactive

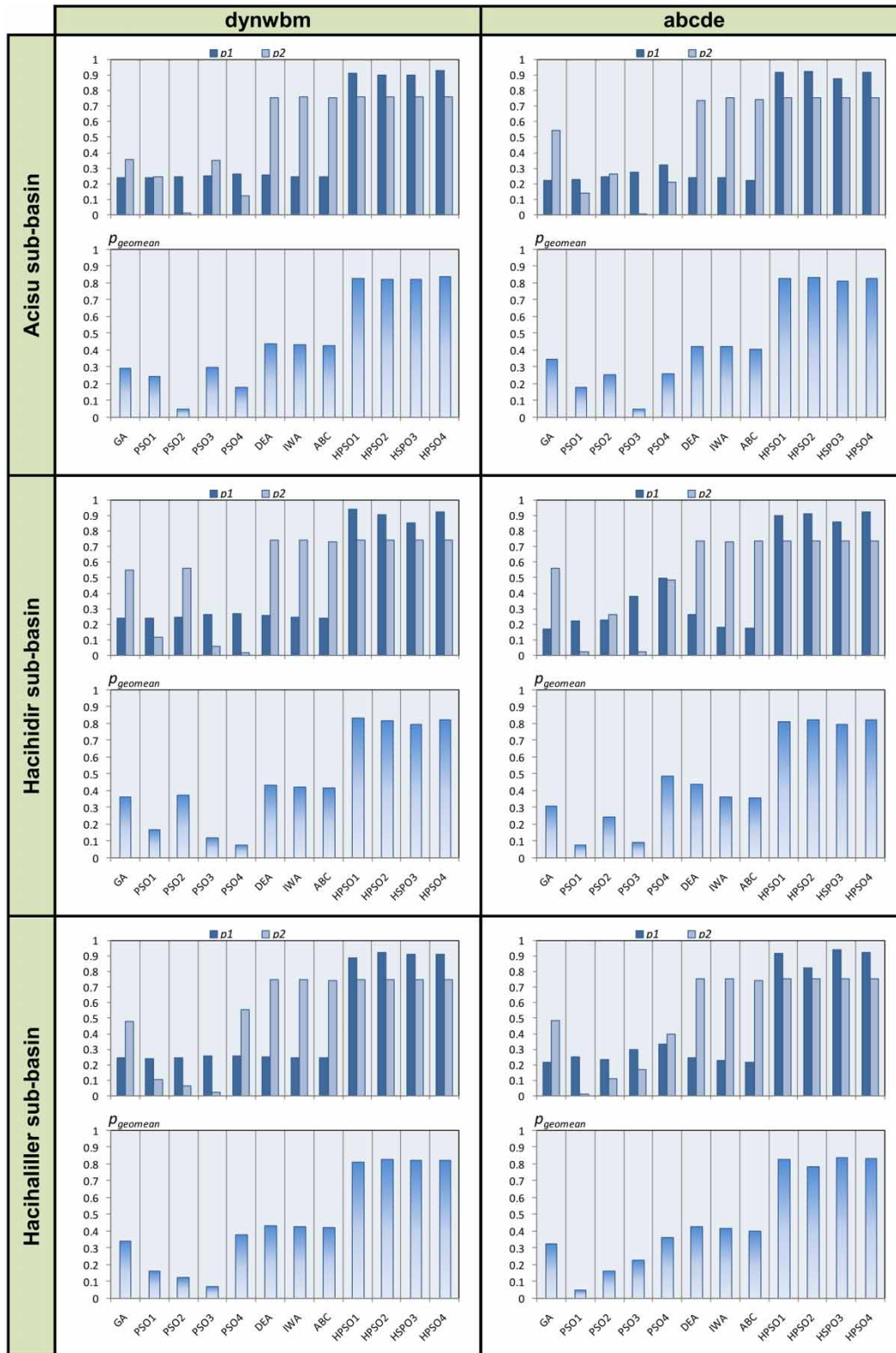


Figure 6 | Common assessment of algorithms by both fitness and convergence performances.

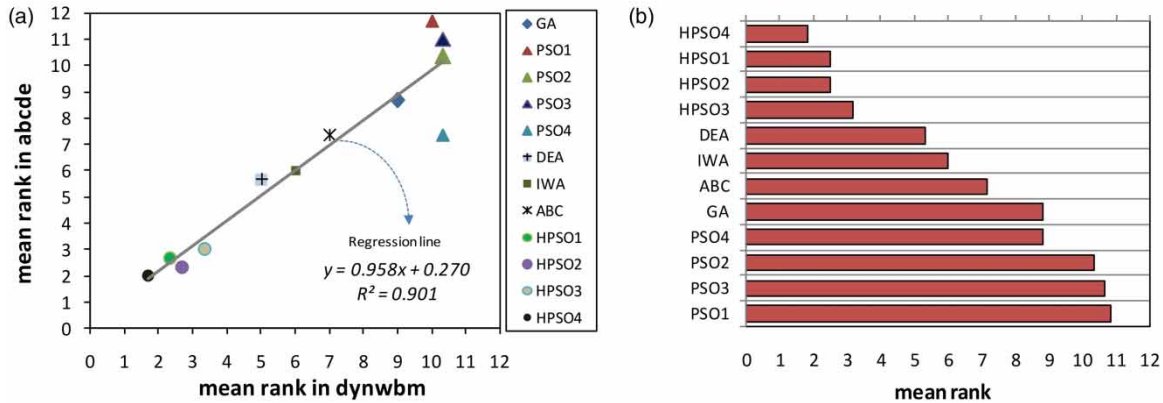


Figure 7 | (a) The change of mean rank numbers depending on CRR models and (b) overall mean ranks derived from all practices (the ranking was made through ρ_{geomean} values).

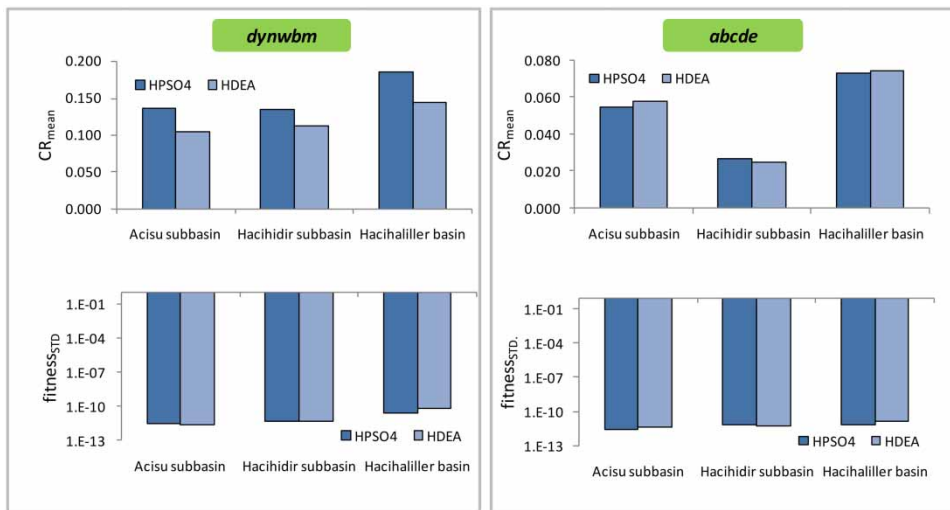


Figure 8 | Comparison of HPSO4 and HDEA algorithms.

parameter effects as follows:

$$T = V + \text{Var}c_1 + \text{Var}c_2 + I \quad (14)$$

where V is the internal variability uttered by variance around SSE means obtained from c_1 and c_2 parameter pair combinations (30 runs were considered here); $\text{Var}c_1$ is the contribution of c_1 ; $\text{Var}c_2$ is the contribution of c_2 ; and I is the contribution of their interactions.

For sub-sampling formulations related to the terms in Equation (14), readers can go through the studies performed by Qi *et al.* (2016) and Yip *et al.* (2011). Similar to the

procedure given by Qi *et al.* (2016), sub-combination groups belonging to the c_1 and c_2 coefficients ranging from 1.0 to 3.0 with an increment of 0.5 were formed. Together with the combinations generated between parameter value pairs, PSO4/HPSO4 algorithms were run $5 \times 5 \times 30 = 750$ times in total. Figure 9(a) outlines the results of SSE collected from the runs by means of contour plotting for the Hachidir sub-basin example. It can be clearly seen from Figure 9(a) that the coefficient combinations actuated in HPSO give more stable outputs and these are much closer to uniform, excluding the coefficients greater than 2.5. According to ANOVA analyses performed to analytically interpret Figure 9(a) and to decompose the sources of

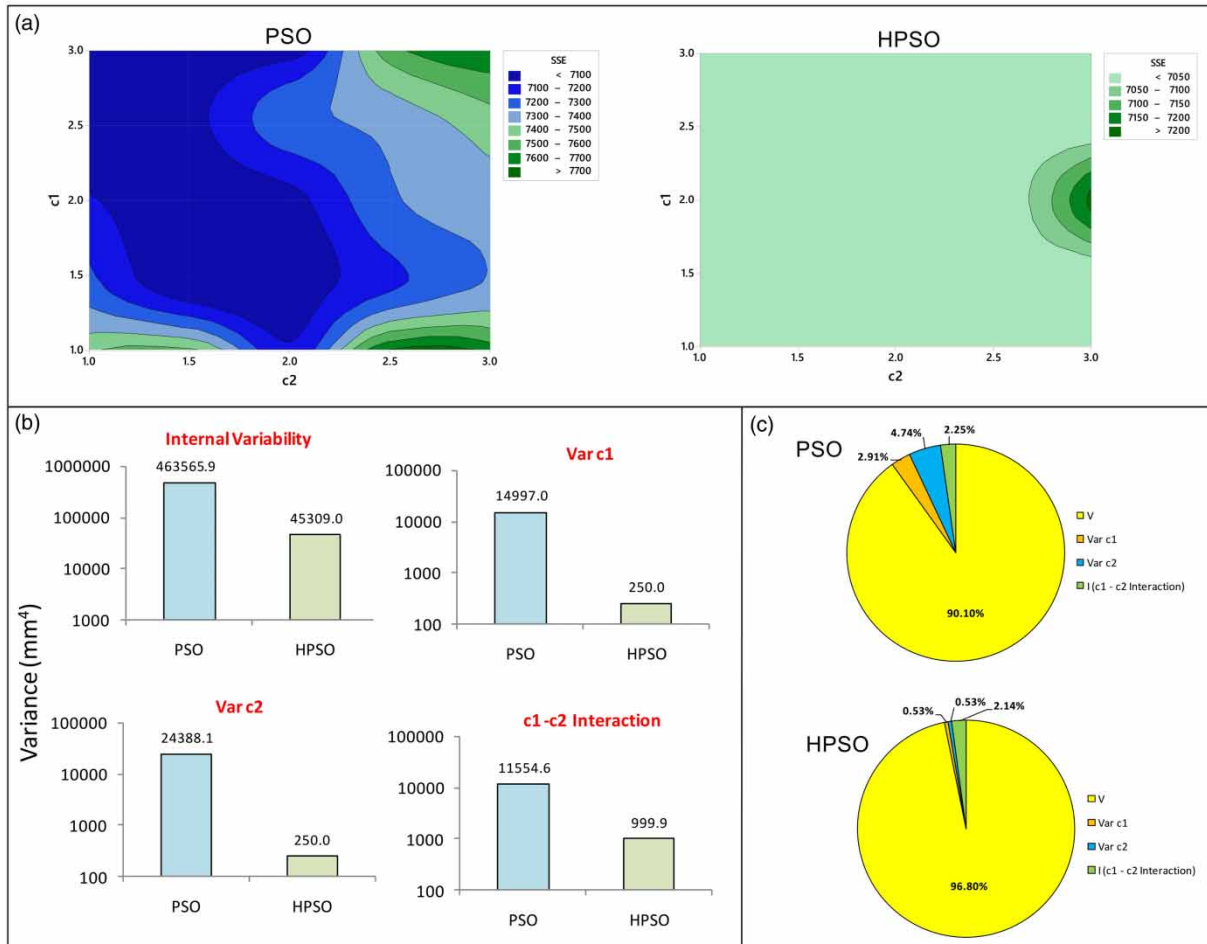


Figure 9 | (a) Contour plots of fitness values obtained from parameter combinations, (b) algorithm-dependent variation of the sources of uncertainty and (c) comparison of the fraction of variances (Hacihidir sub-basin example).

uncertainty, the contribution of internal variability over total variance was more dominant in both algorithms (see Figure 9(c)). This is, of course, a result of the stochastic structure in the algorithms. From Figure 9(b), it is noteworthy that the internal variability in HPSO is 90% less than that of PSO. (The vertical axis in Figure 9(b) is the logarithmic base.) Moreover, individual influences of parameters and their interactions do not have significant impacts on HPSO performance, while the same conditions add much more uncertainty to the PSO. It can be observed that individual parameter uncertainties (Varc1 and Varc2) and c_1 - c_2 interaction uncertainties in HPSO are 98.3%, 98.9% and 91.4% less than those of PSO, respectively. Similar inferences were obtained for other sub-basins but not shared due to word limit. As a consequence, the fact that

HPSO is not extremely sensitive to control parameters makes the algorithm more reliable again.

Performances of CRR models

Although the main goal of this study is to scrutinize the performance of the algorithms and to recommend the coexistence of gradient approaches and swarm intelligence optimizations, for drawing a hydrological perspective, it would be useful to interpret the validation outputs of the CRR models, whose calibrated parameters are denoted in Table 3(a). Acisu and Hacihidir basins have a long-term average annual precipitation of about 550 mm, while this value is nearly 200 mm higher in the Hacihaliller basin. This was reflected in the estimated S_{\max} and b parameters,

Table 3 | (a) Ultimate parameter estimations of CRR models run through HPSO4 algorithm and (b) summary of performance criteria for calibrated CRR models

Sub-basin	<i>dynwbm</i>					<i>abcde</i>				
	S_{max}	α_1	α_2	d	e	b	a	c	d	e
(a)										
Acisu	221.8456	0.6358	0.6612	0.4371	0.8196	180.6705	0.9716	0.6319	0.1006	0.8240
Hacihidir	298.0096	0.6208	0.7255	0.8527	0.7521	224.6173	0.9698	0.5031	0.0505	0.7107
Hacihaliller	652.8411	0.6052	0.5766	0.4402	0.6163	395.3262	0.9721	0.6918	0.0377	0.8628
Sub-basin	Period	<i>dynwbm</i>			<i>abcde</i>					
		NS	RSR	Pbias (%)	NS	RSR	Pbias (%)			
(b)										
Acisu	Calibration	0.8638	0.3691	6.8030	0.8293	0.4131	– 2.1489			
	Validation	0.7637	0.4861	– 13.1156	0.8108	0.4349	– 28.3318			
Hacihidir	Calibration	0.8901	0.3315	5.4428	0.8758	0.3524	– 5.2357			
	Validation	0.8496	0.3878	16.5671	0.8388	0.4015	0.6696			
Hacihaliller	Calibration	0.8896	0.3323	7.3128	0.8753	0.3532	– 2.0555			
	Validation	0.7780	0.4711	14.9649	0.8899	0.3318	0.1687			

The best value in the related performance criterion for the calibration/validation period was indicated in bold-shaded character.

and significant correlations were obtained between these soil moisture storage parameters and annual precipitation regimes. Additionally, since both models have similar evapotranspiration opportunity mechanisms, strong negative correlations were detected between the predicted α_2 and $[1 - c]$ values and long-term annual *EPOT* values. Nash–Sutcliffe coefficient (*NS*), *RSR*, which standardizes root mean square error (*RMSE*) using the standard deviation of observed runoff, and the percentage of bias (*Pbias*), were also assessed in the performance check of *abcde* and *dynwbm* (see Table 3(b)). The details about the model ratings (*very good*, *good*, *sufficient* and *insufficient*) can be accessed from Moriasi *et al.* (2007). In their criterion, the model is rated *very good* if the *NS* is greater than 0.75 and the *RSR* is less than 0.50. In this regard, these two indices in both the calibration and validation period of CRR models refer to *very good* modeling. However, in general, *dynwbm* provides more appropriate results in terms of the indices, whereas *abcde* comes into prominence during the validation period. On the other hand, the *Pbias* shows that the *dynwbm* falls into the *good model* category in the validation period, while inter-model variability pertaining to this index is found to be more pronounced. (According to

the same criteria, if the *Pbias* is less than $\pm 10\%$, the model is rated *very good*, while the model is rated as *good* at the values of *Pbias* in the range of $\pm 10\text{--}15\%$.) In this sense, there may be potential reasons why the model outputs display dissimilarities. For example, although the *EPOT* is based on the reference method Penman–Monteith, it has been determined by Okkan & Kiyamaz (2020) that *dynwbm* in the same basin shows different responses to several empirical *EPOT* methods, and it is emphasized that the most appropriate *EPOT* equations should be decided locally at this stage. On the other hand, predictions obtained with the reference to a single CRR model may contain a set of uncertainties. Therefore, it has been emphasized in Okkan & Inan (2015) and Okkan & Kirdemir (2018) that multi-model ensemble approaches that evaluate multiple model outputs together are more consistent in the stage of making hydro-meteorological projections.

CONCLUSIONS

In this study, a hybrid scheme was offered to overcome some calibration handicaps, which are met in metaheuristics, such

as slow convergence and failure to achieve global minimum solution in each individual run. In the application, algorithm experiments were carried out on two different CRR models by assessing three sub-basins having relatively different drought classes and runoff regimes, and it was questioned whether a common inference was derived from all practices made. First of all, the performances of exerted popular algorithms on the calibration process were stored. As a result of various criteria, the all variants of PSO and GA have not produced reasonable results, while DEA, IWA and ABC are more qualified among their companions. Although the DEA has been at the forefront of the implemented species, the main common matter in all these simulations is still characterized as late convergence. However, LM, which is often subjected to local minima drawback, has become surprisingly noteworthy when it is coupled with PSO algorithm. While the PSO having diverse inertia-weighted variants is the last in the overall ranking, the hybrid HPSO supplemented by LM is apparently much more prominent. Thus, the significance of the response obtained from the combination of PSO and LM algorithms has supported the original aspect of the study. Similar to the fact that the oxygen, which triggers the combustion, is transformed into the water with an extinguisher character after it is combined with hydrogen; the metaheuristics have become fairly robust structured when subjected to derivative algorithms. A similar deduction was also obtained with the HDEA and even approximate results were produced with the HPSO4 using CRW. Moreover, it is found more attractive to utilize the HPSO4 algorithm that is minimalist one in point of control parameters compared with the other hybrid approaches (e.g., Karahan *et al.* 2013; Liao *et al.* 2017) existed in the hydrological modeling literature. This outlook was supported by the implementation of ANOVA.

In summary, the advantages and possible limitation of the proposed hybrid algorithm are listed below:

- It provides fast convergence with less population and an iteration number.
- The variability in fitness values between runs is very minimal (its stability feature).
- It constitutes confidence in various modeling usages, as the uncertainty of the algorithm's control parameters is minimal.

- The hybridization style argued by HPSO can be easily accomplished with another metaheuristics like HDEA.
- However, due to the fact that use of the metaheuristic process and LM together in the same iteration step increase the number of function calls, and also the Jacobian matrix calculation is made, depending on the computer processor, the solution time may be relatively long.

On the other hand, as both examined models have almost similar structures, adaptation of this hybrid algorithm to other hydrology problems will be useful. The parameter estimation of Muskingum flood routing models, the calibration of rainfall-runoff models with intensive parameters (e.g., Zhang *et al.* 2009; Nourani *et al.* 2011; Nourani & Zanardo 2014) and geomorphological rainfall-runoff models (e.g., Nourani *et al.* 2009), reservoir operation optimization and training weights of artificial neural network-based streamflow prediction models are some of these applications.

Besides, since only the classic SSE objective function was used in the study, an alternative calibration scheme may also be needed to enclose the optimization of multiple objectives that quantify different aspects of the runoff series. For example, when the objective function was taken as $[1 - NS]$, although algorithm grading did not change, there were minor alterations in the performance indexes of the CRR models (these results were not shared due to space limitations). In this sense, in the future works, a prospective automatic scheme with HPSO is planned to be formulated that regards the CRR model calibration issue in a multi-objective framework consisting of additional numerical performance statistics such as NS, and the average RMSE of peak and low runoff events (see Madsen 2000).

Furthermore, the uncommon use of some modern variants of evolutionary algorithms such as modified differential evolution with *pb* crossover, successful history-based DEA with linear population size reduction and genetic learning PSO in CRR modeling is also remarkable (e.g. Piotrowski *et al.* 2019). A wide comparison study including such methods will be one of our topics in the future as well.

ACKNOWLEDGEMENTS

This study was funded by the scientific research projects Department of Balikesir University under Grant No.

BAP2017/145. Also, the authors are grateful to two anonymous reviewers for providing valuable comments and suggestions, which we considered to improve the quality of this paper.

REFERENCES

- Arsenault, R., Poulin, A., Côté, P. & Brissette, F. 2014 [Comparison of stochastic optimization algorithms in hydrological model calibration](#). *Journal of Hydrologic Engineering* **19** (7), 1374–1384. doi:10.1061/(asce)he.1943-5584.0000938.
- Asgari, H., Haddad, O. B., Pazoki, M. & Loáiciga, H. A. 2016 [Weed optimization algorithm for optimal reservoir operation](#). *Journal of Irrigation and Drainage Engineering* **142** (2), 04015055. doi:10.1061/(asce)ir.1943-4774.0000963.
- Babayigit, B. & Ozdemir, R. 2012 [A modified artificial bee colony algorithm for numerical function optimization](#). In: *2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 245–249. doi:10.1109/iscc.2012.6249302.
- Budyko, M. I. 1958 *The Heat Balance of the Earth's Surface*. U.S. Department of Commerce, Washington, DC.
- Chang, F. J. & Chen, L. 1998 [Real-coded genetic algorithm for rule-based flood control reservoir management](#). *Water Resources Management* **12**, 185–198. doi:10.1023/A:1007900110595.
- Chen, X., Chau, K. & Busari, A. 2015 [A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model](#). *Engineering Applications of Artificial Intelligence* **46**, 258–268. doi:10.1016/j.engappai.2015.09.010.
- Cooper, V., Nguyen, V. & Nicell, J. 2007 [Calibration of conceptual rainfall-runoff models using global optimisation methods with hydrologic process-based parameter constraints](#). *Journal of Hydrology* **334** (3–4), 455–466. doi:10.1016/j.jhydrol.2006.10.036.
- Doogers, P. & Kite, G. 1999 [Water productivity from integrated basin modeling](#). *Irrigation and Drainage Systems* **13** (3), 275–290. doi:10.1023/A:1006345724659.
- Duan, Q., Sorooshian, S. & Gupta, V. 1992 [Effective and efficient global optimization for conceptual rainfall-runoff models](#). *Water Resources Research* **28** (4), 1015–1031. doi:10.1029/91wr02985.
- Franchini, M. & Galeati, G. 1997 [Comparing several genetic algorithm schemes for the calibration of conceptual rainfall-runoff models](#). *Hydrological Sciences Journal* **42** (3), 357–379. doi:10.1080/02626669709492034.
- Goldberg, D. 1989 *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.
- Goswami, M. & O'Connor, K. M. 2007 [Comparative assessment of six automatic optimization techniques for calibration of a conceptual rainfall-runoff model](#). *Hydrological Sciences Journal* **52** (3), 432–449. doi:10.1623/hysj.52.3.432.
- He, J. & Lin, G. 2016 [Average convergence rate of evolutionary algorithms](#). *IEEE Transactions on Evolutionary Computation* **20** (2), 316–321. doi:10.1109/tevc.2015.2444793.
- Hendrickson, J. D., Sorooshian, S. & Brazil, L. E. 1988 [Comparison of Newton-type and direct search algorithms for calibration of conceptual rainfall-runoff models](#). *Water Resources Research* **24** (5), 691–700. doi:10.1029/wr024i005p00691.
- Holland, J. H. 1975 *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor.
- Kao, Y.-T. & Zahara, E. 2008 [A hybrid genetic algorithm and particle swarm optimization for multimodal functions](#). *Applied Soft Computing* **8** (2), 849–857. doi:10.1016/j.asoc.2007.07.002.
- Karaboga, D. 2005 *An Idea Based on Honey bee Swarm for Numerical Optimization*. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
- Karaboga, D. & Basturk, B. 2008 [On the performance of artificial bee colony \(ABC\) algorithm](#). *Applied Soft Computing* **8** (1), 687–697. doi:10.1016/j.asoc.2007.05.007.
- Karahan, H., Gurarlan, G. & Geem, Z. W. 2013 [Parameter estimation of the nonlinear Muskingum flood-routing model using a hybrid harmony search algorithm](#). *Journal of Hydrologic Engineering* **18** (3), 352–360. doi:10.1061/(asce)he.1943-5584.0000608.
- Kavetski, D. & Clark, M. P. 2011 [Numerical troubles in conceptual hydrology: approximations, absurdities and impact on hypothesis testing](#). *Hydrological Processes* **25** (4), 661–670. doi:10.1002/hyp.7899.
- Kennedy, J. & Eberhart, R. 1995 [Particle swarm optimization](#). In: *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.
- Li, Z., Liu, P., Deng, C., Guo, S., He, P. & Wang, C. 2016 [Evaluation of estimation of distribution algorithm to calibrate computationally intensive hydrologic model](#). *Journal of Hydrologic Engineering* **21** (6), 04016012. doi:10.1061/(asce)he.1943-5584.0001350.
- Liao, S., Sun, Q., Cheng, C., Zhong, R. & Su, H. 2017 [Multicore parallel genetic algorithm with tabu strategy for rainfall-runoff model calibration](#). *Journal of Hydrologic Engineering* **22** (8), 04017024. doi:10.1061/(asce)he.1943-5584.0001542.
- Madsen, H. 2000 [Automatic calibration of a conceptual rainfall-runoff model using multiple objectives](#). *Journal of Hydrology* **235** (3–4), 276–288. doi:10.1016/s0022-1694(00)00279-1.
- Mehrabian, A. & Lucas, C. 2006 [A novel numerical optimization algorithm inspired from weed colonization](#). *Ecological Informatics* **1** (4), 355–366. doi:10.1016/j.ecoinf.2006.07.003.
- Michalewicz, Z. 1994 *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York.
- Moriasi, D. N., Arnold, J. G., Liew, M. W., Bingner, R. L., Harmel, R. D. & Veith, T. L. 2007 [Model evaluation guidelines for systematic quantification of accuracy in watershed](#)

- simulations. *Transactions of the ASABE* **50** (3), 885–900. doi:10.13031/2013.23153.
- Mostafaie, A., Forootan, E., Safari, A. & Schumacher, M. 2018 Comparing multi-objective optimization techniques to calibrate a conceptual hydrological model using in situ runoff and daily GRACE data. *Computational Geosciences* **22** (3), 789–814. doi:10.1007/s10596-018-9726-8.
- Nawi, N. M., Khan, A., Rehman, M. Z., Aziz, M. A., Herawan, T. & Abawajy, J. H. 2014 An accelerated particle swarm optimization based Levenberg Marquardt back propagation algorithm. In: *Neural Information Processing* (C. K. Loo, K. S. Yap, K. W. Wong, A. Teoh & K. Huang eds). Springer. Lecture Notes in Computer Science, Special issue for International Conference on Neural Information Processing 2014, Kuching, Malaysia, Vol. 8835. pp. 245–253.
- Nourani, V. & Zanardo, S. 2014 Wavelet-based regularization of the extracted topographic index from high-resolution topography for hydro-geomorphic applications. *Hydrological Processes* **28** (3), 1345–1357. doi: 10.1002/hyp.9665.
- Nourani, V., Singh, V. P. & Delafrouz, H. 2009 Three geomorphological rainfall–runoff models based on the linear reservoir concept. *Catena* **76** (3), 206–214. doi:10.1016/j.catena.2008.11.008.
- Nourani, V., Roughani, A. & Gebremichael, M. 2011 Topmodel capability for rainfall-runoff modeling of the Ammameh watershed at different time scales using different terrain algorithms. *Journal of Urban and Environmental Engineering* **5** (1), 1–14. doi:10.4090/juee.2011.v5n1.001014.
- Okkan, U. & Inan, G. 2015 Statistical downscaling of monthly reservoir inflows for Kemer watershed in Turkey: use of machine learning methods, multiple GCMs and emission scenarios. *International Journal of Climatology* **35** (11), 3274–3295. doi:10.1002/joc.4206.
- Okkan, U. & Kirdemir, U. 2018 Investigation of the behavior of an agricultural-operated dam reservoir under RCP scenarios of AR5-IPCC. *Water Resources Management* **32** (8), 2847–2866. doi:10.1007/s11269-018-1962-0.
- Okkan, U. & Kiyamaz, H. 2020 Questioning of empirically-derived and locally calibrated potential evapotranspiration equations for a lumped water balance model. *Water Supply* **20** (3), 1141–1156. doi:10.2166/wcc.2019.292.
- Peltokangas, R. & Sorsa, A. 2008 *Real Coded Genetic Algorithm and Nonlinear Parameter Identification*. Control Engineering Laboratory University of Oulu, Report No. 34.
- Piotrowski, A. P., Napiorkowski, M. J., Napiorkowski, J. J., Osuch, M. & Kundzewicz, Z. W. 2017 Are modern metaheuristics successful in calibrating simple conceptual rainfall–runoff models? *Hydrological Sciences Journal* **62** (4), 606–625. doi:10.1080/02626667.2016.1234712.
- Piotrowski, A. P., Napiorkowski, J. J. & Osuch, M. 2019 Relationship between calibration time and final performance of conceptual rainfall-runoff models. *Water Resources Management* **33** (1), 19–37. doi:10.1007/s11269-018-2085-3.
- Qi, W., Zhang, C., Fu, G. & Zhou, H. 2016 Quantifying dynamic sensitivity of optimization algorithm parameters to improve hydrological model calibration. *Journal of Hydrology* **533**, 213–223. doi:10.1016/j.jhydrol.2015.11.052.
- Qin, Y., Kuczera, G. & Kavetski, D. 2017 Comparison of Newton-type and SCE optimisation algorithms for the calibration of conceptual hydrological models. *Australasian Journal of Water Resources* **20** (2), 169–176. doi:10.1080/13241583.2017.1298180.
- Qin, Y., Kavetski, D. & Kuczera, G. 2018 A robust Gauss-Newton algorithm for the optimization of hydrological models: benchmarking against industry-standard algorithms. *Water Resources Research* **54** (11), 9637–9654. doi:10.1029/2017wr022489.
- Rathore, A. & Sharma, H. 2017 Review on inertia weight strategies for particle swarm optimization. In: *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, Vol. 546, pp. 76–86.
- Refsgaard, J. C. & Knudsen, J. 1996 Operational validation and intercomparison of different types of hydrological models. *Water Resources Research* **32** (7), 2189–2202. doi: 10.1029/96wr00896.
- Sha, D. Y. & Hsu, C.-Y. 2006 A hybrid particle swarm optimization for job shop scheduling problem. *Computers & Industrial Engineering* **51**, 791–808. doi:10.1016/j.cie.2006.09.002.
- Shahid, M., Cong, Z. & Zhang, D. 2018 Understanding the impacts of climate change and human activities on streamflow: a case study of the Soan River basin, Pakistan. *Theoretical and Applied Climatology* **134** (1–2), 205–219. doi:10.1007/s00704-017-2269-4.
- Storn, R. & Price, K. 1997 Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* **11**, 341–359.
- Tayfur, G. 2017 Modern optimization methods in water resources planning, engineering and management. *Water Resources Management* **31** (10), 3205–3233. doi:10.1007/s11269-017-1694-6.
- Tekleab, S., Uhlenbrook, S., Mohamed, Y., Savenije, H. H., Temesgen, M. & Weninger, J. 2011 Water balance modeling of Upper Blue Nile catchments using a top-down approach. *Hydrology and Earth System Sciences* **15** (7), 2179–2193. doi:10.5194/hess-15-2179-2011.
- Thomas, H. A. 1981 *Improved Methods for National Water Assessment*. Report, Contract WR 15249270. U.S. Water Resources Council, Washington, DC.
- Thyer, M., Kuczera, G. & Bates, B. C. 1999 Probabilistic optimization for conceptual rainfall-runoff models: a comparison of the shuffled complex evolution and simulated annealing algorithms. *Water Resources Research* **35** (3), 767–773. doi:10.1029/1998wr900058.
- Tigkas, D., Christelis, V. & Tsakiris, G. 2016 Comparative study of evolutionary algorithms for the automatic calibration of the Medbasin-D conceptual hydrological model. *Environmental Processes* **3** (3), 629–644. doi:10.1007/s40710-016-0147-1.
- Tolson, B. A. & Shoemaker, C. A. 2007 Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research* **43** (1). doi:10.1029/2005wr004723.

- Wang, Q. 1997 Using genetic algorithms to optimise model parameters. *Environmental Modelling & Software* **12** (1), 27–34. doi:10.1016/s1364-8152(96)00030-8.
- Wang, Y., Yu, P. & Yang, T. 2010 Comparison of genetic algorithms and shuffled complex evolution approach for calibrating distributed rainfall-runoff model. *Hydrological Processes* **24** (8), 1015–1026. doi:10.1002/hyp.7543.
- Xu, D., Qiu, L. & Chen, S. 2012 Estimation of nonlinear Muskingum model parameter using differential evolution. *Journal of Hydrologic Engineering* **17** (2), 348–353. doi:10.1061/(asce)he.1943-5584.0000432.
- Yip, S., Ferro, C. A. T., Stephenson, D. B. & Hawkins, E. 2011 A simple, coherent framework for partitioning uncertainty in climate predictions. *Journal of Climate* **24** (17), 4634–4643.
- Zhang, J.-R., Zhang, J., Lok, T.-M. & Lyu, M. R. 2007 A hybrid particle swarm optimization – back-propagation algorithm for feed forward neural network training. *Applied Mathematics and Computation* **185** (2), 1026–1037. doi:10.1016/j.amc.2006.07.025.
- Zhang, L., Potter, N., Hickel, K., Zhang, Y. & Shao, Q. 2008 Water balance modeling over variable time scales based on the Budyko framework – model development and testing. *Journal of Hydrology* **360** (1–4), 117–131. doi:10.1016/j.jhydrol.2008.07.021.
- Zhang, X., Srinivasan, R., Zhao, K. & Liew, M. V. 2009 Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes* **23** (3), 430–441. doi:10.1002/hyp.7152.

First received 17 January 2020; accepted in revised form 14 April 2020. Available online 8 May 2020