Acta Geophysica (2023) 71:2325–2338
https://doi.org/10.1007/s11600-022-00988-0

RESEARCH ARTICLE - HYDROLOGY

Calibration of conceptual rainfall-runoff models by selected differential evolution and particle swarm optimization variants

Jaroslaw J. Napiorkowski[1] · Adam P. Piotrowski[1] · Emilia Karamuz[1] · Tesfaye B. Senbeta[1]

Received: 17 July 2022 / Accepted: 23 November 2022 / Published online: 9 December 2022

Abstract

The performance of conceptual catchment runoff models may highly depend on the specific choice of calibration methods made by the user. Particle Swarm Optimization (PSO) and Differential Evolution (DE) are two well-known families of Evolutionary Algorithms that are widely used for calibration of hydrological and environmental models. In the present paper, five DE and five PSO optimization algorithms are compared regarding calibration of two conceptual models, namely the Swedish HBV model (Hydrologiska Byrans Vattenavdelning model) and the French GR4J model (modèle du Génie Rural à 4 paramètres Journalier) of the Kamienna catchment runoff. This catchment is located in the middle part of Poland. The main goal of the study was to find out whether DE or PSO algorithms would be better suited for calibration of conceptual rainfall-runoff models. In general, four out of five DE algorithms perform better than four out of five PSO methods, at least for the calibration data. However, one DE algorithm constantly performs very poorly, while one PSO algorithm is among the best optimizers. Large differences are observed between results obtained for calibration and validation data sets. Differences between optimization algorithms are lower for the GR4J than for the HBV model, probably because GR4J has fewer parameters to optimize than HBV.

Keywords Conceptual rainfall-runoff models · HBV · GR4J · Particle swarm optimization · Differential evolution

Introduction

Metaheuristics are widely used for the optimization in hydrology (Jahandideh-Tehrani et al. 2020; Maier et al. 2014), especially for conceptual catchment runoff models. Among various kinds of metaheuristics, Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995) and Differential Evolution (DE) (Storn and Price 1997) are two landmark examples of Swarm Intelligence and Evolutionary Algorithms (Boussaid et al. 2013). Both were proposed in the mid-1990's and gained widespread popularity in hydrological applications (Jahandideh-Tehrani et al. 2020; Okkan and Kirdemir. 2020b; Maier et al. 2014; Kisi et al. 2010; Xu et al. 2022). DE turned out also a stepping point for development of Markov Chain Monte Carlo-based differential evolution adaptive Metropolis (DREAM) approach (Vrugt et al. 2009). Both DE and PSO algorithms are popular and widely considered to be effective, so that they have been frequently hybridized together into a single algorithm (Xin et al. 2012; Parouha and Verma 2022). Such PSO-DE hybrids were used in water-related applications like optimal localization of hydrocarbon reservoir wells (Nwankwor et al. 2013), design of water distribution system in big cities (Sedki and Ouazar 2012), design of hydraulic structures (Singh and Duggal 2015) or suspended sediment load estimation (Mohammadi et al. 2021).

Because the performance of various optimization methods may be highly uneven for particular application, one may find numerous large-scale comparisons among optimization algorithms in literature (Kazikova et al. 2021; Tharwat and Schenck 2021; Ezugwu et al. 2020; Price et al. 2019; Bujok et al. 2019; Piotrowski et al. 2017a), including some guidelines how to organize such comparisons (Swan et al. 2022; LaTorre et al. 2021). One may also find many comparison studies in which various kinds of metaheuristics are applied for different types of catchment runoff models. Among papers published during last few years,

Edited by Dr. Michael Nones (CO-EDITOR-IN-CHIEF).

✉ Jaroslaw J. Napiorkowski
j.napiorkowski@igf.edu.pl

[1] Institute of Geophysics, Polish Academy of Sciences, Ks. Janusza 64, 01-452 Warsaw, Poland

Jahandideh-Tehrani et al. (2021) compared PSO against Genetic Algorithm, Adnan et al. (2021) tested PSO against Grey Wolf optimizer, Tikhamarine et al. (2020) compared PSO against Harris-Hawks optimizer, Okkan and Kirdemir (2020a) proposed a hybrid PSO algorithm and compared it against five metaheuristics: the basic PSO, the basic DE, Genetic Algorithm, Invasive Weed Algorithm, and Artificial Bee Colony method, Hong et al. (2018) compared DE against Genetic Algorithm, Piotrowski et al. (2017b) compared large field of 26 diversified metaheuristics, and Tigkas et al. (2016) compared Shuffled Complex Evolution, Genetic Algorithm and Evolutionary Annealing. Good reviews of older studies may be found in Meier et al. (2019) and Reddy and Kumar (2020).

There are, however, two main problems with the application of DE and PSO algorithms in hydrology. First, despite the popularity of both PSO and DE in water-related studies, there is no paper that directly compares various variants from PSO and DE families of methods for catchment runoff modelling. Second, plenty of DE and PSO algorithms have appeared in recent decade (Das et al. 2016; Bonyadi and Michalewicz 2017; Bilal et al. 2020; Shami et al. 2022), and many of them perform much better than the basic DE and PSO versions (e.g. Tanabe and Fukunaga 2014; Piotrowski et al. 2017a; Bujok et al. 2019). However, in many hydrological applications only the simplest, over 20-year-old versions of either DE or PSO are used. As a result, one cannot find out which kind of algorithms are de facto more efficient in solving hydrological problems, especially in calibration of rainfall-runoff models.

In the present paper, we aim at detailed and thorough comparison of DE versus PSO algorithms applied for calibration of rainfall-runoff models. One may as well find plenty of other Evolutionary Algorithms applied to this task (Cantoni et al. 2022; Okkan and Kirdemir 2020a, b; Kumar et al. 2019; Dakhlaoui et al. 2012; Gan and Biftu 1996), but the present study is restricted to the comparison solely between DE and PSO variants. Instead of using historical versions of DE and PSO, we test relatively recently proposed variants that may currently be considered the state of the art. For comparison purposes, we have selected five DE and five PSO variants that were proposed between 2012 and 2022. These ten algorithms are applied for calibration of two conceptual rainfall-runoff models, namely HBV (Hydrologiska Byrans Vattenavdelning model; Bergström 1976; Lindström 1997) and GR4J (modèle du Génie Rural à 4 paramètres Journalier; Perrin et al. 2003). The research is performed on the Kamienna catchment that is located in the central part of Poland. We mainly focus on the relative performance of DE and PSO algorithms in calibration of hydrological models, as we wish to find out which family of methods perform better for this task. Direct comparisons between two hydrological models is considered to be of secondary importance in this paper.

Methodology and materials

Rainfall-runoff models

We consider two lumped conceptual catchment runoff models that are built of interconnected reservoirs with mathematical transfer functions used to describe the transfer of water between reservoirs and into the river.

HBV

The HBV model with a snow routine, proposed by Bergström and Forsman (1973), has been used in dozens of countries around the world. In the majority of these applications, the modified versions of the original HBV model have been used (Bergström 1976; Bergström and Lindström 2015). A block diagram of a particular version of the HBV model applied in this paper is shown in Fig. 1. A detailed description of the HBV model components for the version adopted in this paper is given in Piotrowski et al. (2017b).

The input variables of the model are: daily precipitation, average daily air temperature and daily potential evapotranspiration (PET). Precipitation can take the form of rain,

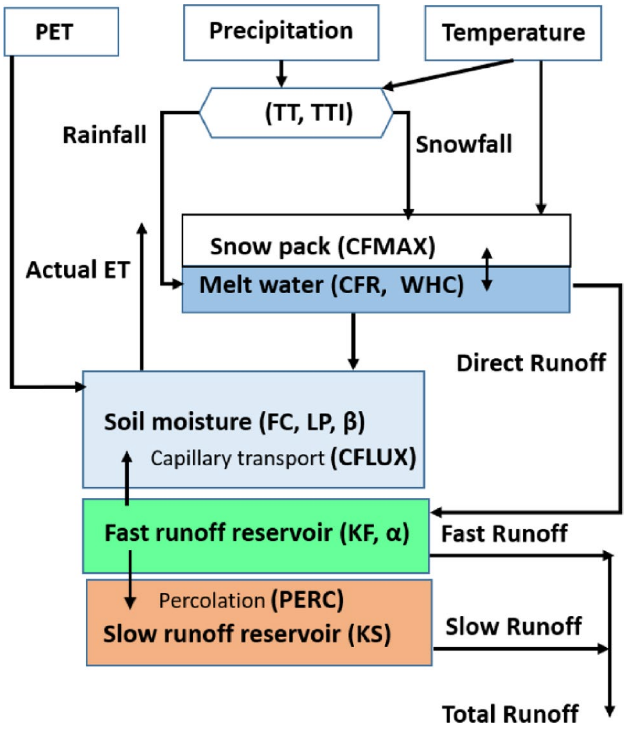

Fig. 1 Structure, conceptual storages and parameters of the HBV model

Table 1 Parameters of the HBV model

Parameter	Description
TT	Threshold temperature
TTI	Temperature interval
CFMAX	The degree-day factor
CFR	The refreezing factor
WHC	The water holding capacity of snow
FC	Field capacity
LP	Factor limiting potential evapotranspiration
β	Nonlinearity parameter is related to the conceptual storage representing soil moisture
CFLUX	The capillary transport between the fast runoff reservoir and the soil moisture reservoir
KF	Parameter of the fast runoff reservoir
α	Another parameter of the fast runoff reservoir
PERC	Percolation—the transport between the fast and slow run-off reservoirs
KS	Parameter of the slow runoff reservoir

snow, or a mixture of snow and rain, which is described using the threshold temperature (TT) and the temperature interval (TTI). At temperatures lower than the lower limit (TT-0.5 TTI) only snow occurs, and at temperatures higher than the upper limit (TT + 0.5 TTI) only rain falls. In the interval between these limits, precipitation is a mixture of rain and snow, decreasing linearly from 100% snow in the lower limit to 0% in the upper limit.

The HBV model used has five state variables representing storage of snow, melt water, soil moisture, fast runoff, and baseflow. The HBV model has 13 parameters defined in Table 1, the values of which are determined using selected optimization procedures. The parameters are grouped into four categories: (1) snow process parameters (TT, TTI, CFMAX, CFR and WHC), (2) soil moisture parameters (FC, LP, β), (3) rapid runoff process parameters (KF, α), and (4) slow runoff (baseflow) parameters (PERC, KS) (Fig. 1).

GR4J

The original GR4J conceptual model is a daily lumped four-parameter catchment runoff model that takes into account changes in soil moisture and can be used for temperatures greater than zero (Perrin et al. 2003). Since our study is concerned with the catchment located in Polish climatic conditions, where snow plays an important role, the original model is extended by adding a snow module (Fig. 2). However, the original name GR4J is retained in this paper. This extended version has seven parameters, three of which (TT, TTI, CFMAX) relate to the snow routine. All GR4J parameters are listed in Table 2 with a brief description. A detailed description of the GR4J model can be found in Perrin et al. (2003).

The input variables to the GR4J model are the same as the HBV model. Similarly to the HBV model, precipitation may take the form of rainfall, snowfall or a mixture of snowfall

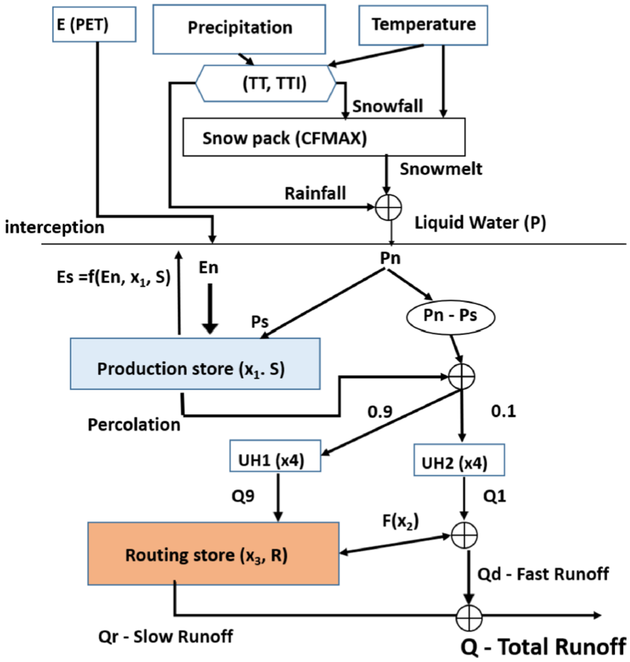

Fig. 2 Structure, conceptual storages and parameters of the GR4J model

Table 2 Parameters of the GR4J model

Parameter	Description
×1	Maximum capacity of the production store
×2	Groundwater exchange coefficient
×3	One-day-ahead maximum capacity of the routing store
×4	Time base of unit hydrograph UH1
TT	Threshold temperature
TTI	Temperature interval
CFMAX	The degree-day factor

and rainfall. Snowmelt is assumed to be directly proportional to the temperature and is computed by means of the degree-day method.

Optimization algorithms

This paper focuses on direct comparison between two families of optimization algorithms: Particle Swarm Optimization (PSO) and Differential Evolution (DE), for conceptual rainfall-runoff model calibration. After a quarter century of research, hundreds of DE and PSO variants could be found in the literature (Das et al. 2016; Bonyadi and Michalewicz 2017). Various variants of DE and PSO may highly differ from each other and are often much more complicated than the basic versions of these algorithms. In this study, we assess modern DE and PSO variants, not their historical, simple versions. Five DE and five PSO relatively recently proposed variants were selected for calibration of the HBV and GR4J models. In the brief introduction below, we define only the classical simple variants, outline the main differences between DE and PSO, and give a brief guide to more advanced DE and PSO algorithms. For the detailed description of the DE and PSO variants being compared, readers are referred to the source papers.

Differential evolution and its variants

The classical Differential Evolution algorithm (Storn and Price 1997) defines a movement of population of NP individuals (solutions vectors) in D-dimensional decision space, where D is the number of parameters to be optimized, in a search for the global optimum. In generation $g=0$, NP individuals: $\mathbf{x}_{i,g}=\left\{x_{i,g}^{1},\dots,x_{i,g}^{\mathrm{D}}\right\}$, $i=1,\dots,\mathrm{NP}$ are initialized at random according to the uniform distribution:

$$x_{i,0}^{j}=L^{j}+\mathrm{rand}_{i}^{j}(0,1)\cdot\left(U^{j}-L^{j}\right);\ j=1,\dots,D;\ i=1,\dots,\mathrm{NP}. \tag{1}$$

Here, rand_{i}^{j} (0,1) is a random value within (0,1) interval that is generated separately for each j-th element of i-th individual. L^j and U^j are lower and upper bounds that define the subset $\prod_{j=1}^{D}\left[L^{j},U^{j}\right]$ of the search space $\mathbf{R}^{\mathrm{D}}$. After initialization of the population of solutions, in each g every individual makes a move across the search space following the three operations: mutation, crossover and selection. In the basic DE, the mutation is defined as:

$$\mathbf{v}_{i,g}=\mathbf{x}_{r1,g}+F\cdot\left(\mathbf{x}_{r2,g}-\mathbf{x}_{r3,g}\right), \tag{2}$$

and is followed by the crossover.

$$u_{i,g}^{j}=\begin{cases}v_{i,g}^{j} & \text{if}\quad \mathrm{rand}_{i}^{j}(0,1)\le \mathrm{CR}\quad \text{or}\quad j=j_{\mathrm{rand},i}\\ x_{i,g}^{j} & \text{otherwise}\end{cases} \tag{3}$$

In Eq. (2) $r1$, $r2$ and $r3$ are three different ($r1\ne r2\ne r3\ne i$) integers that are randomly chosen from the range [1, NP]. In Eq. (3) $j_{\mathrm{rand,i}}$ is another integer, randomly selected within [1, D]. Note that there are three control parameters of the basic DE variant, NP, CR and F, which, in the basic DE, need to be defined by the user.

As much as the search space is often bounded (i.e. the values of model parameters to be calibrated are restricted within some range), some verification is needed after crossover to check whether the new solution $\boldsymbol{u}_{i,g}$ is within the bounds (Kononova et al. 2021). If $\boldsymbol{u}_{i,g}$ turns out to be outside the bounds, it has to be forced to stay within the search domain (e.g. by using some of the methods discussed in Helwig et al. (2013) and Kadavy et al. (2022)). After that the objective function is called for the solution $\boldsymbol{u}_{i,g}$ that is within the bounds and one obtains its goodness of fit $f(\boldsymbol{u}_{i,g})$ that represents the quality of the solution $\boldsymbol{u}_{i,g}$. Finally, selection operation is performed to choose only the better among $\boldsymbol{x}_{i,g}$ and $\boldsymbol{u}_{i,g}$ to enter the next generation.

$$\mathbf{x}_{i,g+1}=\begin{cases}\mathbf{u}_{i,g} & \text{if}\quad f\left(\mathbf{u}_{i,g}\right)\le f\left(\mathbf{x}_{i,g}\right)\\ \mathbf{x}_{i,g} & \text{otherwise}\end{cases} \tag{4}$$

After repeating the above procedures for each individual in the population, the NP individuals proceed to the $g+1$ generation. The algorithm repeats the same steps in the subsequent generations until some stopping conditions are reached. In the present study, the maximum number of function calls set to 20,000 is defined as the stopping condition.

The majority of modern DE variants are much more complicated than the basic version from 1997, defined above (e.g. see Mohamed et al. 2021). A detailed review of DE variants may be found in Das et al. (2016), Al-Dabbagh et al. (2018) or Opara and Arabas (2018). The modern variants often adaptively modify the control parameters F and CR (Brest et al. 2006; Tanabe and Fukunaga 2014; Zuo et al. 2021; Ghosh et al. 2022), use variable population size NP (Tanabe and Fukunaga 2014; Piotrowski 2017; Polakova et al. 2019), implement multiple search (Qin et al. 2009; Wu et al. 2018; Yi et al. 2022) or crossover strategies (Zaharie 2009; Islam et al. 2012; Wang et al. 2022), introduce new procedures or use operators proposed for other metaheuristics (Piotrowski 2018; Caraffini and Neri 2019; Cai et al. 2020). DE is also sometimes hybridized with other metaheuristics (Gong et al. 2010; Xin et al. 2012; Awad et al. 2017). In the present study, we compare five advanced DE variants which are defined in Table 3. The detailed description of these algorithms may be found in the source papers. The control parameters of algorithms are the

Table 3 Compared variants of differential evolution and particle swarm optimization algorithms

Algorithm's number	Short name	Descriptive name	References	Comments
1	HARD-DE	Hierarchical archive-based DE	Meng and Pan (2019)	Population size is parabolically (quicker at the end of the search) decreased during run from $25ln(D)\sqrt{D}$ to 4. Algorithm performed best on other hydrological problem—air2water model calibration (Zhu et al. 2021)
2	MDE_pBX	Memetic adaptive DE with new mutation and crossover	Islam et al. (2012)	Population size = 100. One among older classical adaptive DE variants
3	L-SHADE	SHADE with linear population size reduction	Tanabe and Fukunaga (2014)	A version of Successful History Adaptive DE with population size linearly reduced during run from 18D at the beginning to 4 at the end. State-of-the-art algorithm that was a kick-off point for many other Differential Evolution variants
4	OLSHADE-CS	DE orthogonal array-based initialization and new selection strategy	Kumar et al. (2022)	Algorithm proposes a new DE selection technique and a new initialization method. The population size is decreasing linearly from $6D^2$ to 4 during the search
5	EnsDE	Ensemble of DE algorithms	Wu et al. (2018)	Population size = 100. State-of-the-art ensemble of DE variants
6	DEPSO	Dual environmental PSO	Zhang et al. (2019)	Population size = 50. Velocities are initialized randomly within [MIN_i,MAX_i] interval
7	EPSO	Ensemble of PSO variants	Lynn and Suganthan (2017)	Population size = 40, divided into two uneven swarms. Velocities are initialized randomly within [MIN_i,MAX_i] interval. Classical algorithm that manages an ensemble of different PSO variants
8	PPSO	Pyramid PSO	Li et al. (2022)	Population size = 64. A recent algorithm that builds a 4-layer pyramid with number of particles from the top to the bottom layer set to 4, 8, 20, 32. Some particles in each layer may communicate solely with particles from the same layer, others may communicate with particles from the same layer and from the one upper layer. Velocities are initialized with ± 0.1·[MIN_i-MAX_i]
9	PSO-sono	PSO for single-objective problems	Meng et al. (2022)	Population size = 100. A very recent PSO variant that uses a ring topology. Velocities are initialized randomly within [− 30,30] interval
10	TAPSO	Triple-archive PSO	Xia et al. (2020)	Population size = 60. Velocities are initialized randomly within [MIN_i,MAX_i] interval, but during run are effectively restricted to ± 0.2·[MIN_i-MAX_i]

In the Comments column, D means problem dimensionality, [MIN_i, MAX_i] are bounds set for i-th dimension. Algorithms 1–5 are Differential Evolution based, and 6–10 are Particle Swarm Optimization based same as suggested in the source papers, but we provide an information on the population size used in Table 3.

Particle swarm optimization variants

Particle Swarm Optimization (Kennedy and Eberhart 1995) is a very popular stochastic population-based algorithm, inspired by the behavior of the swarm of animals. In PSO, the solutions (called particles) move across the *D*-dimensional search space all the time, but remember the best location they have visited so far. As in DE, initial positions $\mathbf{x}_{i,0}$ of NP PSO particles ($i = 1,\ldots,$ NP) are usually generated randomly within the bounds of the search space (Eq. (1)). However, in PSO each particle has an associated velocity vector. Depending on the specific PSO variant, the initial velocities $\mathbf{v}_{i,0}$ of each particle are either set to 0 or are generated from some pre-specified interval, which frequently depend on the differences between upper and lower bounds of the search space. The fitness value $f(\mathbf{x}_{i,0})$ is evaluated for each newly generated particle. Then in each generation g the particles are moving through the search space according to the following equation:

$$\begin{aligned}v^{j}_{i,g+1} &= w \cdot v^{j}_{i,g} + c_1 \cdot \text{rand}\,1^{j}_{i,g}(0,1) \cdot \left(\text{pbest}^{j}_{i,g} - x^{j}_{i,g}\right) \\ &\quad + c_2 \cdot \text{rand}\,2^{j}_{i,g}(0,1) \cdot \left(\text{gbest}^{j}_{g} - x^{j}_{i,g}\right) \\ x^{j}_{i,g+1} &= x^{j}_{i,g} + v^{j}_{i,g+1},\end{aligned} \tag{5}$$

where $j = 1,\ldots,D$, $\mathbf{pbest}_{i,g}$ and $\mathbf{gbest}_{g}$ are the best positions visited during the search by i-th particle and the best position visited by any particle in the swarm, respectively. $\text{rand1}_{i,g}{}^{j}(0,1)$ and $\text{rand2}_{i,g}{}^{j}(0,1)$ are two random numbers generated at each generation from [0,1] interval for each i and j index separately, and c_1 and c_2 are acceleration coefficients (algorithm parameters to be set by the user). As may be seen, for each i-th particle three vectors are remembered—its current position $\mathbf{x}_{i,g}$, the best position $\mathbf{pbest}_{i,g}$ visited by the i-th particle since the initialization of the search and i-th particle current velocity $\mathbf{v}_{i,g}$. The parameter w is the so-called inertia weight that was first introduced by Shi and Eberhart (1998). Like in the case of DE, modern PSO variants are often much more complicated than the initial version—for a survey readers are referred to Bonyadi and Michalewicz (2017), Cheng et al. (2018), and Shami et al. (2022). Modern PSO variants use different topologies—under this term we mean the communication possibilities between individuals (Lynn et al. 2018; Xia et al. 2020; Li et al. 2022), theoretically or empirically modify the values of control parameters (Clerc and Kennedy 2002; Harrison et al. 2018; Piotrowski et al. 2020; Cleghorn and Stapleberg 2022; Meng et al. 2022), introduce novel equations for movement of particles (Santos et al. 2020; Li et al. 2021; Houssein et al. 2021), bring together an ensemble of different PSO variants (Lynn and Suganthan 2017; Wu et al. 2019; Liu and Nishi 2022), or hybridize PSO with other algorithms (Aydilek 2018; Sengupta et al. 2019; Xu et al. 2019; Dziwinski and Bartczuk 2020). The details of the five specific PSO variants used in the current study are given in the second half of Table 3.

Major differences between DE and PSO

Technically, PSO is a major algorithm within a family of Swarm Intelligence that is based on the communal behavior of animals, and DE is a version of Evolutionary Computation that is based on the evolutionary principles of life. However, such inspiration-focused differences are irrelevant from optimization point of view (Tzanetos and Dounias 2021; Molina et al. 2020; Sorensen 2015). What's important is that, although both types of algorithms are population-based metaheuristics (Del Ser et al. 2019; Boussaid et al. 2013), they perform a search in much different way.

First of all, in DE the particular individual verifies the new solution in each generation, but moves to the new location only if it is not inferior to the solution at which it was located at the beginning of the generation. It means that DE population may test new locations, but stay in the former ones until some promising region of the search space is found. Because the probability of visiting particular part of the search space is a function of the current location of individuals in DE population, such lack of movements may lead to stagnation (Weber et al. 2009) and hamper proofs of convergence (Hu et al. 2016; Opara and Arabas 2018). However, this feature assures that each individual is always located in the best place it has visited so far, and the location of the whole population is a kind of space-based memory of the high-quality solutions. On the contrary, the PSO particle in each generation moves and stays in the new location, irrespective how poor it is. As a result, the particle requires an additional memory in which a best solution it has visited so far is remembered. PSO particles may fly all around, and may have a problem with returning to the promising solutions (Van der Bergh and Engelbrecht 2006). This inspired researchers to determine the relations between the trajectories of PSO and the values of control parameters or topologies in an analytical way (Clerc and Kennedy 2002; Harrison et al. 2018; Cleghorn and Stapleberg 2022).

Another main difference between DE and PSO is the crossover Eq. (3). In almost all DE variants, the sampled solution is a mix of the former solution and a solution that comes up as a result of initial move (which in most DE variants will be an extended version of Eq. (2)). The crossover is useful as it allows keeping some information from the previous solution within the newly tested one. It limits the diversity, but enhances the chance of finding a better solution; without that the number of successful steps would often be very low in DE, and population could stagnate for a long time in the former location. As PSO performs moves all the time, crossover is not necessary (although it has been tested in some PSO variants, e.g. Engelbrecht 2016; Gong et al. 2016; Molaei et al. 2021).

Finally, the majority of new DE algorithms use adaptive control parameters and population size modification schemes (Al-Dabbagh et al. 2018). Although, as noted above, adaptive and variable population-size PSO variants are also numerous, they are not as clearly superior to the variants with fixed but carefully chosen values (Bonyadi 2020). PSO variants that adaptively modify acceleration coefficients (c_1, c_2) are relatively rare (for examples see Harrison et al. 2018) and do not much improve the performance.

Description of the study area

The study is performed on the Kamienna River catchment. The Kamienna Catchment is located in the Central Vistula basin in the Polish Upland area and covers 2020 km^2 (Fig. 3).

The main river of the catchment is the Kamienna (left tributary of the Vistula River), whose sources are located at the border of the Masovian and Świętokrzyskie provinces above the town of Skarżysko-Kamienna in the mountainous area. The river is 156 km long and runs from west to east, predominantly through the Świętokrzyskie Province. The catchment elevation varies from about 130–600 m a.m.s.l. There are large variations of the longitudinal slope of the channel in the upper part of the Kamienna River (around 10%). This part has a mountainous character up to Skarżysko-Kamienna, from where the slope gradually decreases, reaching near Kunów about 0.7% (Lenar-Matyas et al. 2006). The catchment area is prone to natural and human hazards (FramWat 2019). Human activities focused on increase in water retention in the catchment by constructing many small artificial reservoirs and two large ones: Wióry and Brody Iłżeckie.

According to the climate classification of Köppen–Geiger (adapted by Peel et al. 2007), the Kamienna Catchment climate is "cold" with no dry season and a warm summer. Annual areal precipitation for the period 1968–2018 varies from 410 to 920 mm, with a long-term annual mean of 600 mm, while the long-term monthly mean varies from about 30–90 mm (Senbeta and Romanowicz 2021). The minimum and maximum precipitations occur in winter and summer, respectively. The mean monthly temperature in the watershed in the same observational period varies from − 3.1 to 18.3 °C, with the minimum and maximum in January and July, respectively (Senbeta and Romanowicz 2021). The land use structure of the study catchment is dominated by agriculture (46.3%), a significant part of the area is also occupied by forest and semi-natural land (43.3%); other parts are artificial land and water bodies, 10% and 0.4%, respectively.

Dataset

Data used include daily hydrological and climatological variables, namely streamflow, air temperature, precipitation and potential evapotranspiration (PET) in and around the watershed. These data were collected for the historical period 1968–1982, during which the catchment could be considered free from anthropogenic influences. After 1982, the artificial reservoirs were constructed in the catchment, which changed the flow regime. The periods 1968–1970, 1971–1976 and 1977–1982 were used for warm-up, calibration and validation, respectively. Hydroclimatic data were obtained from the Institute of Meteorology and Water Management (https://dane.imgw.pl/).

The temperature-based method was used to estimate PET at each meteorological station. As both the HBV and GR4J models are lumped, temperature, precipitation and PET in the catchment were averaged using Thiessen polygon method.

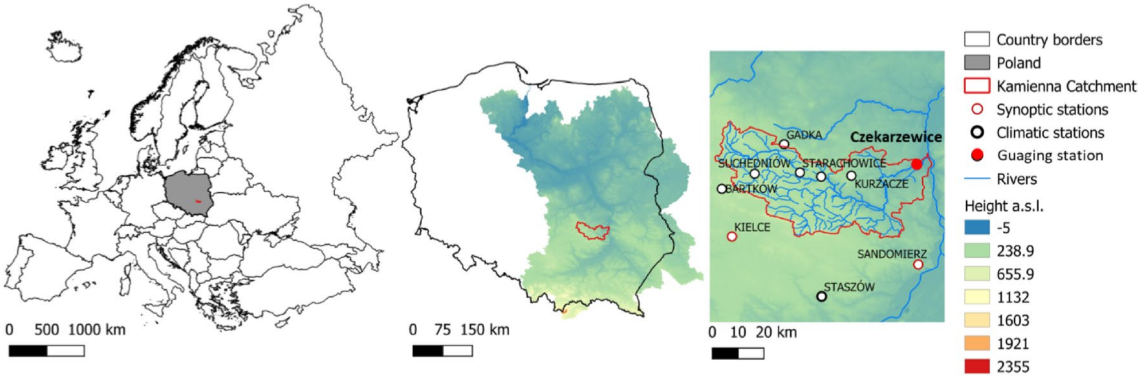

Fig. 3 The Kamienna River catchment, flow gauging station and meteorological stations

Comparison criteria

Both HBV and GR4J models are calibrated using mean square error (MSE). As a result, we compare algorithms using exactly the same criterion that was used as objective function during search. Each algorithm is run 30 times on every model (HBV and GR4J). The mean, median, best, and worst performances from these 30 runs obtained for calibration and testing datasets are used for comparison. This is a frequently adopted compromise between the confidence in the result's quality (the more runs, the more reliable results), and the applicability (more runs means more time for computation). As shown in Vecek et al. (2017), the number of runs has only moderate impact on the final conclusions from the research. In addition, we also report the standard deviation of the results obtained based on 30 runs. This allows us to discuss both the averaged performance, extremes, and the consistency of solutions found by particular optimization algorithm.

Results and discussion

Calibration of the HBV model

Each time the model is calibrated, some data need to be set aside and not be available to calibration algorithm—we call this data set validation (or testing) data. This validation dataset is important, because it detects a potential model's overfitting. For obvious reason, we want our model to work correctly not just for the data that were used during calibration (calibration set), but also for some future, unknown data. Therefore, validation data set is needed to verify the practical effects of calibration. Thus the discussion of the results may be divided into two parts—the first covers the comparison based on the calibration data, and the second includes the comparison based on the validation data (see Table 4).

Based on the calibration dataset, two algorithms, namely PPSO and HARD-DE, appear to be the best ones for the HBV model. When comparing the performance based on the mean or median from 30 runs, the results obtained by PPSO are the best. In particular, the low median obtained by PPSO (14.039) shows that this algorithm often leads to the results with high performance. Three algorithms (HARD-DE, MDE_pBX and L-SHADE) achieve equal median (14.505) indicating all these algorithms frequently find a similar, although sub-optimal solution. However, according to the average values HARD-DE performs better than MDE_pBX and L-SHADE.

In contrast, OLSHADE-CS leads to by far the poorest results, with MSE that is over 30% (median MSE = 20.365) higher than that of PPSO and HARD-DE. In terms of the mean and median, PSO-based PPSO looks as the winner,

Table 4 MSE results obtained for the HBV model

Measure	Variants of differential evolution and particle swarm optimization algorithms									
	HARD-DE	MDE-pBX	L-SHADE	OLSHADE-CS	EnsDE	DEPSO	EPSO	PPSO	PSO-sono	TAPSO
Calibration dataset										
Mean	14.154	14.504	14.666	20.242	14.624	15.200	14.647	14.088	15.727	15.041
Rank mean	2	3	6	10	4	8	5	1	9	7
Median	14.505	14.505	14.505	20.365	14.603	14.926	14.629	14.039	16.066	15.065
Rank median	2	2	2	10	5	7	6	1	9	8
Best	12.158	14.487	14.487	18.389	14.505	14.506	14.303	13.042	14.572	13.270
Rank best	1	5	5	10	7	8	4	2	9	3
Worst	14.505	14.585	16.294	22.327	14.892	16.295	14.983	16.161	16.349	16.008
Rank worst	1	2	7	10	3	8	4	6	9	5
Standard deviation	0.811	0.017	0.481	0.939	0.086	0.608	0.163	0.988	0.710	0.607
Validation dataset										
Mean	39.314	40.365	40.450	46.087	42.072	38.832	40.600	37.196	40.339	39.410
Rank mean	3	6	7	10	9	2	8	1	5	4
Median	40.026	40.027	40.025	44.484	41.886	39.367	39.937	37.118	39.974	39.349
Rank median	7	8	6	10	9	3	4	1	5	2
Best	32.350	39.964	39.354	36.918	37.665	37.076	38.811	35.099	38.859	37.291
Rank best	1	10	9	3	6	4	7	2	8	5
Worst	41.718	41.738	43.285	60.342	45.759	41.264	45.108	40.014	43.525	43.710
Rank worst	3	4	5	10	9	2	8	1	6	7
Standard deviation	2.644	0.677	0.886	5.908	1.545	1.197	1.578	1.329	1.256	1.315

and DE-based OLSHADE-CS as the worst method. However, it does not mean the superiority of PSO over DE according to the mean and median measures, as the four remaining PSO variants (DEPSO, EPSO, PSO-sono and TAPSO) perform poorer than the majority of DE variants (HARD-DE, MDE-pBX, EnsDE, and often L-SHADE) on calibration data set.

The mean and median are not the only metrics to compare algorithms. Many users would simply be rather interested in the best results found. When one compares the best and the worst solutions obtained during 30 runs, HARD-DE becomes a winner. Indeed, the best solution found by HARD-DE (12.158) is about 7% better than the best solution found by PPSO (13.047). Moreover, HARD-DE never found a solution worse than the median (14.505), while the worst solution found by PPSO is over 10% poorer (16.161). We may also observe that the ranking of algorithms based on the best solutions found is generally different than rankings based on the mean or median. One should especially note that TAPSO and EPSO were able to find best solutions with lower MSE than the best solutions found by DE algorithms except HARD-DE. However, DEPSO and PSO-sono were not able to outperform DE methods anyway. Hence, according to the ranking based on the best solutions found, DE still outperforms PSO in general, but the relative positions of specific algorithms are different, and the whole picture is more complicated.

The superiority of DE over PSO algorithms on the calibration data set is probably an effect of the behavior of both families of algorithms. In the recent, efficient DE variants, the control parameters are often made adaptive. Hence they are flexibly being modified during search (Das et al. 2016; Al-Dabbagh et al. 2018), whereas the control parameters of PSO are more frequently set fixed throughout the whole search (e.g. Clerc and Kennedy 2002; Harrison et al. 2018). This flexibility of control DE parameters may give DE algorithm additional chances to cope with complicated fitness landscape of each specific problem, in cases where the PSO variants with fixed control parameter values are more conservative. Another difference that may partly justify better performance of DE is the selection operator. DE algorithms reject poorer solutions that are found during search, and move only to the better locations. Hence, the current DE population is composed of solutions that are better than all their predecessors. On the contrary, PSO algorithms keep moving all the time, and will produce final generations in both better or poorer locations. As a result, PSO variants may be considered more chaotic, and less effective in finding the precise location of the optima.

The results obtained for the validation dataset are over twice poorer than the results that were obtained for the calibration data set. It is, however, rather up to the data, not the calibration process. Considering mean and median measures, the PPSO algorithm is the winner for testing dataset, as it was for calibration dataset. However, the quality of solutions found by HARD-DE, MDE_pBX and L_SHADE is frequently not confirmed on testing dataset. All three algorithms achieved the second-best median on the calibration dataset, but the median MSE for the validation dataset is only 6th–8th, which is 10% poorer than the median MSE obtained by PPSO. In contrast to the calibration dataset, the PSO-based algorithms do not perform poorer than the DE-based ones on the validation dataset. It may suggest that finding the exact optimum based on the calibration data set is of moderate importance when the incoming data would significantly change (e.g. Beven 2006; Beven et al. 2022). Nonetheless, the overall best solution found by any method for the validation dataset again belongs to HARD-DE (32.350) and is again about 8% better than the best solution found by PPSO (35.099). This means that, in some sense, DE is still better than PSO on validation data, as one of DE variants is able to find much better result than all other competing algorithms. Whether one prefers to look at the mean or the best results is up to the user's taste.

Calibration of the GR4J model

Contrary to the HBV model, the calibration of the GR4J model seems to be much simpler, and almost all algorithms, apart from OLSHADE-CS, lead to almost the same median and best results. Only mean MSE slightly vary, and the DE-based methods (excluding OLSHADE-CS), especially HARD-DE, achieve clearly better mean results than the PSO algorithms (Table 5). This indicates that algorithms rather compete in terms of failures, not in finding the best results. The poor performance of OLSHADE-CS may be due to its very slow convergence, and may be a side-effect of the fact that OLSHADE-CS was initially tested on, and probably fitted to, problems with very large number of allowed function calls (see Kumar et al. 2022 for initial tests).

The mean MSE obtained by HARD-DE (15.819) is by about 7% better than the mean MSE obtained by PPSO (16.567). This difference is also kept for the validation dataset, where HARD-DE is also about 8% better than PPSO. However, for the validation dataset surprisingly the best solution found by PPSO is better than the best solution found by HARD-DE. Moreover, the overall best solution of the GR4J model for the validation dataset is even found by the other PSO-based method, PSO-sono. This may look as the opposite finding compared with the one noted for HBV model. Nonetheless, the results obtained for the GR4J model are much less diversified than those for the HBV model. This may be due to much smaller number of parameters to be optimized—small number of parameters may lead to lower differences in performance between algorithms.

Table 5 Results obtained for the GR4J model

Measure	Variants of differential evolution and particle swarm optimization algorithms									
	HARD-DE	MDE-pBX	L-SHADE	OLSHADE-CS	EnsDE	DEPSO	EPSO	PPSO	PSO-sono	TAPSO
Calibration set										
Mean	15.819	16.127	16.091	19.909	15.824	16.946	15.979	16.567	16.661	16.567
Rank mean	1	5	4	10	2	9	3	6	8	6
Median	15.719	15.719	15.719	19.871	15.719	15.828	15.737	15.828	15.815	15.828
Rank median	1	1	1	10	1	7	5	7	6	7
Best	15.719	15.719	15.719	17.030	15.719	15.719	15.719	15.719	15.719	15.719
Rank best	1	1	1	10	1	1	1	1	1	1
Worst	18.819	18.821	18.821	21.529	18.819	18.868	18.823	18.874	18.863	18.874
Rank worst	1	3	3	10	1	7	5	8	6	8
Standard deviation	0.557	1.054	0.932	1.344	0.556	1.521	0.764	1.356	1.423	1.356
Validation set										
Mean	30.279	30.279	31.296	38.513	30.240	35.873	30.837	33.913	34.104	33.913
Rank mean	2	2	5	10	1	9	4	6	8	6
Median	29.759	29.759	29.759	35.996	29.757	30.275	29.759	30.221	29.759	30.221
Rank median	2	2	2	10	1	9	2	7	2	7
Best	29.759	29.759	29.749	28.874	29.049	28.840	29.079	28.840	28.816	28.840
Rank best	8	8	8	5	6	2	7	2	1	2
Worst	45.870	45.870	45.816	46.794	45.816	45.518	45.535	45.518	45.871	45.518
Rank worst	7	7	5	10	5	1	4	2	9	2
Standard deviation	2.894	2.894	4.765	5.992	2.894	7.753	3.939	6.937	7.372	6.937

Conclusion

The present paper discusses numerous state-of-the-art variants from the PSO and DE optimization methods which were applied to calibration of the catchment runoff models. In the literature dealing with computational optimization methods, no broader comparison of performance between PSO and DE families has been presented so far. We have chosen five DE and five PSO variants proposed between 2012 and 2022. These ten algorithms were applied for calibration of two conceptual rainfall-runoff models: HBV and GR4J, on the Kamienna catchment located in the middle part of Poland. We aimed at finding whether the DE or PSO algorithms would be better suited for calibration of rainfall-runoff models. Furthermore, we focused on the relative performance obtained by the algorithms from the two different modern families in calibration of hydrological models, rather than comparing the results obtained by both conceptual rainfall-runoff models.

We show that the results obtained by different optimizers applied are roughly similar for the GR4J model, which has very few parameters. For the GR4J model, one may rather point at an inferior algorithm—OLSHADE-CS, rather than a winner, as many optimizers performed very similarly. No clear difference between PSO and DE methods could be found. This is probably because the GR4J model has low number of parameters that are relatively simple to calibrate. However, among the best results found during many runs, those found by two PSO variants (PPSO and PSO-sono) are better than those found by their DE competitors.

In the case of the HBV model, the results were much different. OLSHADE-CS showed the poorest performance as well, but the results obtained by other algorithms were diversified. Which exact method could be a winner depends on whether one focuses on calibration, or validation dataset, and whether one is interested in the mean/median performance, or in finding the best possible solution in one among 30 runs. Overall, two algorithms, PSO-based PPSO and DE-based HARD-DE performed best on the HBV model calibration. Comparison between both families of methods reveals that, in general, the DE algorithms slightly outperformed the PSO ones. The difference was, however, clearer for the calibration dataset than the validation dataset. We may recommend using adaptive variants of algorithms for model calibration, especially those that have flexible control parameters (e.g. HARD-DE) or advanced topology (e.g. PPSO) that may automatically tune the speed of information exchange between individuals within the population managed by the algorithm. DE algorithms seems to be more appropriate choice for the calibration of rainfall-runoff models than PSO variants, but the difference between their final performances is limited and depends on the measure that is used to create the ranking of algorithms.

Acknowledgements This work was supported within statutory activities No 3841/E-41/S/2022 of the Ministry of Science and Higher Education of Poland, and by the project HUMDROUGHT, carried out in the Institute of Geophysics Polish Academy of Sciences and funded by National Science Centre (2018/30/Q/ST10/00654).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Adnan RM, Mostafa RR, Kisi O, Yaseen ZM, Shahid S, Zounemat-Kermani M (2021) Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization. Knowl Based Syst 230:107379. https://doi.org/10.1016/j.knosys.2021.107379

Al-Dabbagh RD, Neri F, Idris D, Baba MS (2018) Algorithmic design issues in adaptive differential evolution schemes: re-view and taxonomy. Swarm Evol Comput 43:284–311. https://doi.org/10.1016/j.swevo.2018.03.008

Awad NH, Ali MZ, Suganthan PN, Reynolds RG (2017) CADE: a hybridization of cultural algorithm and differential evolution for numerical optimization. Inf Sci 378:215–241. https://doi.org/10.1016/j.ins.2016.10.039

Aydilek IB (2018) A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems. Appl Soft Comput 66:232–249. https://doi.org/10.1016/j.asoc.2018.02.025

Bergström S (1976) Development and application of a conceptual runoff model for Scandinavian catchments. Norrköping: Svergies Meteorologiska och Hydrologiska Institut, SMHI Report RHO 7

Bergström S, Forsman A (1973) Development of a conceptual deterministic rainfall-runoff model. Hydrol Res 4(3):147–170. https://doi.org/10.2166/nh.1973.0012

Bergström S, Lindström G (2015) Interpretation of runoff processes in hydrological modelling—experience from the HBV approach. Hydrol Process 29(15):3535–3545

Beven K (2006) A manifesto for the equifinality thesis. J Hydrol 320(1–2):18–36. https://doi.org/10.1016/j.jhydrol.2005.07.007

Beven K, Lane S, Page T, Kretzschmar A, Hankin B, Smith P, Chappell N (2022) On (in)validating environmental models. 2. Implementation of a turing-like test to modelling hydrological processes. Hydrol Process 36:e14703. https://doi.org/10.1002/hyp.14703

Bilal PM, Zaheer H, Garcia-Hernandez L, Abraham A (2020) Differential evolution: a review of more than two dec-ades of research. Eng Appl Artif Intell 90:103479. https://doi.org/10.1016/j.engappai.2020.103479

Bonyadi MR (2020) A theoretical guideline for designing an effective adaptive particle swarm. IEEE Trans Evol Comput 24(1):57–68. https://doi.org/10.48550/arXiv.1802.04855

Bonyadi MR, Michalewicz Z (2017) Particle swarm optimization for single objective continuous space problems: a review. Evol Comput 25(1):1–54. https://doi.org/10.1162/EVCO_r_00180

Boussaid I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117. https://doi.org/10.1016/j.ins.2013.02.041

Brest J, Greiner S, Boškovic B, Mernik M, Žumer V (2006) Self-adapting control parameters in differential evolution: a com-parative study on numerical benchmark problems. IEEE Trans Evol Comput 10:646–657. https://doi.org/10.1109/TEVC.2006.872133

Bujok P, Tvrdik J, Polakova R (2019) Comparison of nature-inspired population-based algorithms on continuous optimization problems. Swarm Evol Comput 50:100490. https://doi.org/10.1016/j.swevo.2019.01.006

Cai Y, Wu D, Zhou Y, Fu S, Tian H, Du Y (2020) Self-organizing neighborhood-based differential evolution for global optimization. Swarm Evol Comput 56:100699. https://doi.org/10.1016/j.swevo.2020.100699

Cantoni E, Tramblay Y, Grimaldi S, Salamon P, Dakhlaoui H, Dezetter A, Thiemig V (2022) Hydrological performance of the ERA5 reanalysis for flood modeling in Tunisia with the LISFLOOD and GR4J models. J Hydrol Reg Stud 42:101169. https://doi.org/10.1016/j.ejrh.2022.101169

Caraffini F, Neri F (2019) A study on rotation invariance in differential evolution. Swarm Evol Comput 50:100436. https://doi.org/10.1016/j.swevo.2018.08.013

Cheng S, Lu H, Lei X, Shi Y (2018) A quarter century of particle swarm optimization. Complex Intell Systems 4:227–239. https://doi.org/10.1007/s40747-018-0071-2

Cleghorn CW, Stapleberg B (2022) Particle swarm optimization: stability analysis using -informers under arbitrary coefficient distributions. Swarm Evol Comput 71:101060. https://doi.org/10.1016/j.swevo.2022.101060

Clerc M, Kennedy J (2002) The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6(1):58–73. https://doi.org/10.1109/4235.985692

Dakhlaoui H, Bargaoui Z, Bárdossy A (2012) Toward a more efficient calibration schema for HBV rainfall–runoff model. J Hydrol 444–445:161–179. https://doi.org/10.1016/j.jhydrol.2012.04.015

Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution—an updated survey. Swarm Evol Comput 27:1–30. https://doi.org/10.1016/j.swevo.2016.01.004

Del Ser J, Osaba E, Molina D, Yang XS, Salcedo-Sanz S, Camacho D, Das S, Suganthan PN, Coello Coello CA, Herrera F (2019) Bio-inspired computation: where we stand and what's next. Swarm Evol Comput 48:220–250. https://doi.org/10.1016/j.swevo.2019.04.008

Dziwinski P, Bartczuk L (2020) A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic. IEEE Trans Fuzzy Syst 28(6):1140–1154. https://doi.org/10.1109/TFUZZ.2019.2957263

Engelbrecht AP (2016) Particle swarm optimization with crossover: a review and empirical analysis. Artif Intell Rev 45:131–165. https://doi.org/10.1007/s10462-015-9445-7

Ezugwu AE, Adeleke OJ, Akinyelu AA, Viriri S (2020) A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. Neural Comput Appl 32:6207–6251. https://doi.org/10.1007/s00521-019-04132-w

FramWat (2019) The result of tests Kamienna Pilot Catchment. DT1.3.1 reports from pilot action—testing the prototype of the FroGIS tool in the river basins. Version 2. https://www.interreg-central.eu. Accessed 4 Aug 2022

Gan TY, Biftu GF (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

Ghosh A, Das S, Das AK, Senkerik R, Viktorin A, Zelinka I, Masegosa AD (2022) Using spatial neighborhoods for parameter adaptation: an improved success history based differential evolution. Swarm Evol Comput 71:101057. https://doi.org/10.1016/j.swevo.2022.101057

Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. Soft Comput 15:645–665. https://doi.org/10.1007/s00500-010-0591-1

Gong YJ, Li JJ, Zhou Y, Li Y, Chung HSH, Shi YH, Zhang J (2016) Genetic learning particle swarm optimization. IEEE Trans Cybern 46(10):2277–2290. https://doi.org/10.1109/TCYB.2015.2475174

Harrison HR, Engelbrecht AP, Ombuki-Berman BM (2018) Self-adaptive particle swarm optimization: a review and analysis of convergence. Swarm Intell 12:187–226. https://doi.org/10.1007/s11721-017-0150-9

Helwig S, Branke J, Mostaghim S (2013) Experimental analysis of bound handling techniques in particle swarm optimization. IEEE Trans Evol Comput 17(2):259–271

Hong H, Panahi M, Shirzadi A, Ma T, Liu J, Zhu AX, Chen W, Kougias I, Kazakis N (2018) Flood susceptibility assessment in Hengfeng area coupling adaptive neuro-fuzzy inference system with genetic algorithm and differential evolution. Sci Total Environ 621:1124–1141. https://doi.org/10.1016/j.scitotenv.2017.10.114

Houssein EH, Gad AG, Hussain K, Suganthan PN (2021) Major advances in particle swarm optimization: theory, analysis, and application. Swarm Evol Comput 63:100868. https://doi.org/10.1016/j.swevo.2021.100868

Hu Z, Su Q, Yang X, Xiong Z (2016) Not guaranteeing convergence of differential evolution on a class of multimodal functions. Appl Soft Comput 41:479–487. https://doi.org/10.1016/j.asoc.2016.01.001

Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans Syst Man Cybern B Cybern 42(2):482–500. https://doi.org/10.1109/TSMCB.2011.2167966

Jahandideh-Tehrani M, Bozog-Haddad O, Loaiciga HA (2020) Application of particle swarm optimization to water management: an introduction and overview. Environ Monit Assess 192:1–18. https://doi.org/10.1007/s10661-020-8228-z

Jahandideh-Tehrani M, Jenkis G, Helfer F (2021) A comparison of particle swarm optimization and genetic algorithm for daily rainfall-runoff modelling: a case study for Southeast Queensland, Australia. Optim Eng 22:29–50. https://doi.org/10.1007/s11081-020-09538-3

Kadavy T, Viktorin A, Kazikova A, Pluhacek M, Senkerik R (2022) Impact of boundary control methods on bound-constrained optimization benchmarking. IEEE Trans Evol Comput. https://doi.org/10.1109/TEVC.2022.3204412

Kazikova A, Pluhacek M, Senkerik R (2021) How does the number of objective function evaluations impact our understanding of metaheuristics behavior? IEEE Access 9:44032–44048. https://doi.org/10.1109/ACCESS.2021.3066135

Kennedy J, Eberhart RC. (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Perth, Australia. IEEE, Piscataway, NJ, USA IV:1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kisi O (2010) River suspended sediment concentration modeling using a neural differential evolution approach. J Hydrol 389(1–2):227–235. https://doi.org/10.1016/j.jhydrol.2010.06.003

Kononova AV, Caraffini F, Back T (2021) Differential evolution outside the box. Inf Sci 581:587–604. https://doi.org/10.1016/j.ins.2021.09.058

Kumar S, Kaushal DR, Gosain AK (2019) Evaluation of evolutionary algorithms for the optimization of storm water drainage network for an urbanized area. Acta Geophys 67:149–165. https://doi.org/10.1007/s11600-018-00240-8

Kumar A, Biswas PP, Suganthan PN (2022) Differential evolution with orthogonal array-based initialization and a novel selection strategy. Swarm Evol Comput 68:101010. https://doi.org/10.1016/j.swevo.2021.101010

LaTorre A, Molina D, Osaba E, Poyatos J, DelSer J, Herrera F (2021) A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. Swarm Evol Comput 67:100973. https://doi.org/10.1016/j.swevo.2021.100973

Lenar-Matyas A, Witkowska H, Żak A (2006) Kamienna river—changes in time and a proposition of restoration. Infrastruct Ecol Rural Areas 4(2):79–88

Li D, Guo W, Lerch A, Li Y, Wang L, Wu Q (2021) An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization. Swarm Evol Comput 60:100789. https://doi.org/10.1016/j.swevo.2020.100789

Li T, Shi J, Deng W, Hu Z (2022) Pyramid particle swarm optimization with novel strategies of competition and cooperation. Appl Soft Comput 121:108731. https://doi.org/10.1016/j.asoc.2022.108731

Lindström G (1997) A simple automatic calibration routine for the HBV model. Nord Hydrol 28(3):153–168. https://doi.org/10.2166/nh.1997.0009

Liu Z, Nishi T (2022) Strategy dynamics particle swarm optimizer. Inf Sci 582:665–703. https://doi.org/10.1016/j.ins.2021.10.028

Lynn N, Suganthan PN (2017) Ensemble particle swarm optimizer. Appl Soft Comput 55:533–548. https://doi.org/10.1016/j.asoc.2017.02.007

Lynn N, Ali MZ, Suganthan PN (2018) Population topologies for particle swarm optimization and differential evolution. Swarm Evol Comput 39:24–35. https://doi.org/10.1016/j.swevo.2017.11.002

Maier HR, Kapelan Z, Kasprzyk J, Kollat J, Matott LS, Cunha MC, Dandy GC, Gibbs MS, Keedwell E, Marchi A, Ostfeld A, Savic D, Solomatine DP, Vrugt JA, Zecchin AC, Minsker BS, Barbour EJ, Kuczera G, Pasha F, Castelletti A, Giuliani M, Reed PM (2014) Evolutionary algorithms and other metaheuristics in water resources: current status, research challenges and future directions. Environ Mode Softw 62:271–299. https://doi.org/10.1016/j.envsoft.2014.09.013

Maier HR, Razavi S, Kapelan Z, Matott LS, Kasprzyk J, Tolson BA (2019) Introductory overview: optimization using evolutionary algorithms and other metaheuristics. Environ Mode Softw 114:195–213. https://doi.org/10.1016/j.envsoft.2018.11.018

Meng Z, Pan JS (2019) HARD-DE: hierarchical archive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization. IEEE Access 7:12832–12854. https://doi.org/10.1109/ACCESS.2019.2893292

Meng Z, Zhong Y, Mao G, Liang Y (2022) PSO-sono: a novel PSO variant for single-objective numerical optimization. Inf Sci 586:176–191. https://doi.org/10.1016/j.ins.2021.11.076

Mohamed AW, Hadi AA, Mohamed AK (2021) Differential evolution mutations: taxonomy, comparison and convergence analysis. IEEE Access 9:68629–68662. https://doi.org/10.1109/ACCESS.2021.3077242

Mohammadi B, Guan Y, Moazenzadeh R, Safari MJS (2021) Implementation of hybrid particle swarm optimization-differential evolution algorithms coupled with multi-layer perceptron for suspended sediment load estimation. CATENA 198:105024. https://doi.org/10.1016/j.catena.2020.105024

Molaei S, Moazen H, Najjar-Ghabel S, Farzinvash L (2021) Particle swarm optimization with an enhanced learning strategy and crossover operator. Knowl Based Syst 215:106768. https://doi.org/10.1016/j.knosys.2021.106768

Molina D, Poyatos D, Del Ser J, Garcia S, Hussain A, Herrera F (2020) Comprehensive taxonomies of nature- and bio-inspired optimization: inspiration versus algorithmic behavior, critical analysis recommendations. Cogn Comput 12:897–939. https://doi.org/10.1007/s12559-020-09730-8

Nwankwor E, Nagar AK, Reid DC (2013) Hybrid differential evolution and particle swarm optimization for optimal well placement. Comput Geosci 17:249–268. https://doi.org/10.1007/s10596-012-9328-9

Okkan U, Kirdemir U (2020a) Towards a hybrid algorithm for the robust calibration of rainfall-runoff models. J Hydroinf 22(4):876–899. https://doi.org/10.2166/hydro.2020a.016

Okkan U, Kirdemir U (2020b) Locally tuned hybridized particle swarm optimization for the calibration of the nonlinear Muskingum flood routing model. J Water Clim Chang 11(S1):343–358. https://doi.org/10.2166/wcc.2020.015

Opara K, Arabas J (2018) Comparison of mutation strategies in differential evolution—a probabilistic perspective. Swarm Evol Comput 39:53–69. https://doi.org/10.1016/j.swevo.2017.12.007

Parouha RP, Verma P (2022) A systematic overview of developments in differential evolution and particle swarm optimization with their advanced suggestion. Appl Intell 52:10448–10492. https://doi.org/10.1007/s10489-021-02803-7

Peel MC, Finlayson BL, Mcmahon TA (2007) Updated world map of the Köppen-Geiger climate classification. Hydrol Earth Syst Sci 11:1633–1644. https://doi.org/10.5194/hess-11-1633-2007

Perrin C, Michel C, Andreassian V (2003) Improvement of a parsimonious model for streamflow simulation. J Hydrol 279:275–289. https://doi.org/10.1016/S0022-1694(03)00225-7

Piotrowski AP (2017) Review of differential evolution population size. Swarm Evol Comput 32:1–24. https://doi.org/10.1016/j.swevo.2016.05.003

Piotrowski AP (2018) L-SHADE optimization algorithms with population-wide inertia. Inf Sci 468:117–141. https://doi.org/10.1016/j.ins.2018.08.030

Piotrowski AP, Napiorkowski MJ, Napiorkowski JJ, Rowinski PM (2017a) Swarm intelligence and evolutionary algorithms: performance versus speed. Inf Sci 384:34–85. https://doi.org/10.1016/j.ins.2016.12.028

Piotrowski AP, Napiorkowski MJ, Napiorkowski JJ, Osuch M, Kundzewicz ZW (2017b) Are modern metaheuristics successful in calibrating simple conceptual rainfall–runoff models? Hydrol Sci J 62(4):606–625. https://doi.org/10.1080/02626667.2016.1234712

Piotrowski AP, Napiorkowski JJ, Piotrowska AE (2020) Population size in particle swarm optimization. Swarm Evol Comput 58:100718. https://doi.org/10.1016/j.swevo.2020.100718

Polakova R, Tvrdik J, Bujok P (2019) Differential evolution with adaptive mechanism of population size according to current population diversity. Swarm Evol Comput 50:100519. https://doi.org/10.1016/j.swevo.2019.03.014

Price KV, Awad NH, Ali MZ, Suganthan PN (2019) The 2019 100-digit challenge on real-parameter, single-objective op-timization: analysis of results. Nanyang Technological University, Singapore, Tech Rep, http://www.ntu.edu.sg/home/epnsugan

Qin AK, Huang VL, Suganthan PN (2009) Differential Evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417. https://doi.org/10.1109/TEVC.2008.927706

Reddy MJ, Kumar DN (2020) Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. H2Open J 3(1):135–188. https://doi.org/10.2166/h2oj.2020.128

Santos R, Borges G, Santos A, Silva M, Sales C, Costa JCWA (2020) A rotationally invariant semi-autonomous particle swarm optimizer with directional diversity. Swarm Evol Comput 56:100700. https://doi.org/10.1016/j.swevo.2020.100700

Sedki A, Ouazar D (2012) Hybrid particle swarm optimization and differential evolution for optimal design of water distribution systems. Adv Eng Inform 26(3):582–591. https://doi.org/10.1016/j.aei.2012.03.007

Senbeta TB, Romanowicz RJ (2021) The role of climate change and human interventions in affecting watershed runoff responses. Hydrol Process 35(12):e14448. https://doi.org/10.1002/hyp.14448

Sengupta S, Basak S, Peters RA II (2019) Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives. Mach Learn Knowl Extr 19(1):157–191. https://doi.org/10.48550/arXiv.1804.05319

Shami TM, El-Saleh AA, Alswaitti M, Al-Tashi Q, Summakieh MA, Mirjalili S (2022) Particle swarm optimization: a comprehensive survey. IEEE Access 10:10031–10061. https://doi.org/10.1109/ACCESS.2022.3142859

Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceeding of the IEEE congress on evolutionary computation (CEC). IEEE World Congress on Computational Intelligence, Anchorange, AC, USA, pp 69–73

Singh RM, Duggal SK (2015) Optimal design of hydraulic structures with hybrid differential evolution multiple particle swarm optimization. Can J Civ Eng 42(5):303–310. https://doi.org/10.1139/cjce-2014-0441

Sorensen K (2015) Metaheuristics—the metaphor exposed. Int Trans Oper Res 22:3–18. https://doi.org/10.1111/itor.12001

Storn R, Price KV (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

Swan J, Adriaensen S, Brownlee AEI, Hammond K, Johnson CG, Kheiri A, Krawiec F, Merelo JJ, Minku LL, Ozcan E, Pappa GL, Garcia-Sanchez P, Sorensen K, Voss S, Wagner M, White DR (2022) Metaheuristics "in the large." Eur J Oper Res 29:393–406. https://doi.org/10.1016/j.ejor.2021.05.042

Tanabe R, Fukunaga A (2014) Improving the search performance of SHADE using linear population size reduction. In: Proceedings of IEEE congress evolutional computation, Bejing, pp 1658–1665. https://doi.org/10.1109/CEC.2014.6900380

Tharwat A, Schenck W (2021) A conceptual and practical comparison of PSO-style optimization algorithms. Expert Syst Appl 67:114430. https://doi.org/10.1016/j.eswa.2020.114430

Tigkas D, Christelis V, Tsakiris G (2016) Comparative study of evolutionary algorithms for the automatic calibration of the medbasin-D conceptual hydrological model. Environ Process 3:629–644. https://doi.org/10.1007/s40710-016-0147-1

Tikhamarine Y, Souag-Gamane D, Ahmed AN, Sammen SS, Kisi O, Huang YF, El-Shafief A (2020) Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs. particle swarm optimization. J Hydrol 589:125133. https://doi.org/10.1016/j.jhydrol.2020.125133

Tzanetos A, Dounias G (2021) Nature inspired optimization algorithms or simply variations of metaheuristics? Artif Intell Rev 54:1841–1862. https://doi.org/10.1007/s10462-020-09893-8

Van Der Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176(8):937–971. https://doi.org/10.1016/j.ins.2005.02.003

Vecek N, Crepinsek M, Mernik M (2017) On the influence of the number of algorithms, problems, and independent runs in the comparison of evolutionary algorithms. Appl Soft Comput 54:23–45. https://doi.org/10.1016/j.asoc.2017.01.011

Vrugt JA, ter Braak CJF, Diks CGH, Robinson BA, Hyman JM, Higdon D (2009) Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. Int J Nonlinear Sci Numer Simul 10(3):273–290. https://doi.org/10.1515/IJNSNS.2009.10.3.273

Wang Z, Chen Z, Wang Z, Wei J, Chen X, Li Q, Zheng Y, Sheng W (2022) Adaptive memetic differential evolution with multi-niche sampling and neighborhood crossover strategies for global optimization. Inf Sci 583:121–136. https://doi.org/10.1016/j.ins.2021.11.046

Weber M, Neri F, Tirronen V (2009) Distributed differential evolution with explorative–exploitative population families. Genet Program Evolvable Mach 10(4):343–371. https://doi.org/10.1007/s10710-009-9089-y

Wu G, Shen X, Li H, Chen H, Lin A, Suganthan PN (2018) Ensemble of differential evolution variants. Inf Sci 423:172–186. https://doi.org/10.1016/j.ins.2017.09.053

Wu G, Mallipeddi R, Suganthan PN (2019) Ensemble strategies for population-based optimization algorithms—a survey. Swarm Evol Comput 44:695–711. https://doi.org/10.1016/j.swevo.2018.08.015

Xia X, Gui L, Yu F, Wu H, Wei B, Zhang YL, Zhan ZH (2020) Triple archives particle swarm optimization. IEEE Trans Cybern 50(12):4862–4875. https://doi.org/10.1109/TCYB.2019.2943928

Xin B, Chen J, Zhang J, Fang H, Peng ZH (2012) Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. IEEE Trans Syst Man Cybern C Appl Rev 42(5):744–767. https://doi.org/10.1109/TSMCC.2011.2160941

Xu Y, Hu C, Wu Q, Jian S, Li Z, Chen Y, Zhang G, Zhang Z, Wang S (2022) Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation. J Hydrol 608:127553. https://doi.org/10.1016/j.jhydrol.2022.127553

Xu P, Luo W, Lin X, Qiao Y, Zhu T (2019) Hybrid of PSO and CMA-ES for global optimization. In: 2019 IEEE congress on evolutionary computation (CEC), pp. 27–33. https://doi.org/10.1109/CEC.2019.8789912

Yi W, Chen Y, Pei Z, Lu J (2022) Adaptive differential evolution with ensembling operators for continuous optimization problems. Swarm Evol Comput 69:100994. https://doi.org/10.1016/j.swevo.2021.100994

Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. Appl Soft Comput 9(3):1126–1138. https://doi.org/10.1016/j.asoc.2009.02.012

Zhang JQ, Zhu XX, Wang YH, Zhou MC (2019) Dual-environmental particle swarm optimizer in noisy and noise-free environments. IEEE Trans Cybern 49(6):2011–2021. https://doi.org/10.1109/TCYB.2018.2817020

Zuo M, Dai G, Peng L (2021) A new mutation operator for differential evolution algorithm. Soft Comput 25:13595–13615. https://doi.org/10.1007/s00500-021-06077-6