Method to Estimate Optimal Parameters

Tiantian Yang, Kuolin Hsu, Qingyun Duan, Soroosh Sorooshian, and Chen Wang

Contents

Abstract

Model, data, and parameter estimation are three fundamental elements in hydrologic process modeling and forecasting. Recent progresses in hydrologic modeling have been made toward more efficient and effective estimation of model

T. Yang · S. Sorooshian
University of California, Irvine, CA, USA
e-mail: tiantiay@uci.edu

K. Hsu (✉)
Civil & Environmental Engineering, The Henry Samueli School of Engineering, University of California, Irvine, CA, USA
e-mail: kuolinh@uci.edu

Q. Duan
Faculty of Geographical Science, Beijing Normal University, Beijing, China
e-mail: qyduan@bnu.edu.cn

C. Wang
South China Botanical Garden, Chinese Academy of Sciences, Richland, WA, USA
e-mail: chen.wang@scbg.ac.cn

Q. Duan et al. (eds.), *Handbook of Hydrometeorological Ensemble Forecasting*,
https://doi.org/10.1007/978-3-642-40457-3_26-1

parameters. In this chapter, classical and recently developed parameter optimization methods and their applications in hydrological model calibration are reviewed. Those methods include gradient-based optimization methods, direct search methods, and recently developed stochastic global optimization methods. A recently developed surrogate model approach, with the purpose to reduce computational burden of model which runs through replacing the hydrologic process model with a cheaper-to-run surrogate model, is also discussed. Extending from a single objective function parameter optimization, multi-objective optimization methods and their core concept in deriving trade-offs are also summarized. Examples are provided to demonstrate the strengths and limitations of optimization algorithms summarized in this chapter.

Keywords
Optimization · Hydrologic Model · Evolutionary Algorithm · Automatic Parameter Estimation · Surrogate Model

1 Introduction

Hydrologic models are extensively used in academia, industry, and operating agencies for flood forecasting, streamflow simulation, and water resources management. The successful use of hydrologic models to simulate natural processes depends on many factors, including (1) the mathematical formulation of hydrologic model, i.e., the mathematical representation of natural rainfall-runoff processes in a certain level of sophistication and its corresponding assumptions; (2) sufficiency and accuracy of observation data at proper temporal and spatial resolutions, such as in situ streamflow observations and precipitation measurements from rain gauge, radar network, or remotely sensed information; (3) the properly calibrated model parameters (i.e., the global optimal parameters in the feasible domain), which significantly affect the accuracy and uncertainty of hydrological prediction.

This chapter presents model parameter calibration methods in three parts. The first part reviews recent development of the methods to estimate optimal parameters of hydrologic models, especially those heuristic methods used in automatic parameter estimation. The second part focuses on the search mechanism and procedures employed in different methods. And the third part provides examples to illustrate the strengths and limitations of different methods. An overall review of this chapter is summarized below.

This chapter starts with an introduction of hydrologic models and a general mathematical formulation of parameter estimation from the maximum likelihood perspective. Two classical parameter estimation methods are introduced, namely the *Steepest decent method* and *Newton method*, known as the gradient-based local search methods. Those classical *gradient-based methods* are efficient, but require the objective function to be differentiable, i.e., gradient of the objective function exists for the entire parameter space. Differing from the gradient-based methods, another type of parameter estimation methods is called *direct search methods*, which does not require information or the existence of gradient of the objective function. A *Simplex Downhill method* is presented in detail as one of the efficient and robust *direct search methods*. Both the methods introduced as *gradient-based methods* and the *Simplex Downhill method* belong to the same category of *local search method*, because the search always starts at certain location in the parameter space, evolves toward a gradient decreasing direction in the objective function space, and finally stops when method identifies the gradient of objective function equals or approximately equals zero. As the complexity of hydrologic model and number of parameters to calibrate increase, the response surface of an objective function sometime become multimodal, i.e., there are multiple local optimums instead of a single local optimum for concave or convex problems. The roughness and multimodality place great challenges for those local search methods, and global optimization methods turn out to be powerful in addressing those issues. In the global optimization subsection, a number of nature phenomenon inspired optimization algorithms are summarized in detail, including the *genetic algorithms* (GAs), the *simulated annealing* method (SA), the *particle swarm optimization* (PSO), the *ant colony optimization* (ACO), and the *shuffled complex evolution UA* global optimization scheme (SCE-UA). Though the global optimization methods are effective in finding the global optimal parameter set, they generally require up to tens of thousands of model runs to find the global optimal solution. Given the severe computational constraint on solving such an optimization problem and the further increases of model complexity in operation (e.g., the models require a large amount of CPU time to run), many efforts are made by researchers to reduce the computational burden and replace the expensive simulation model with a cheaper-to-run surrogate model. Some fields also refer to the surrogate modeling as function approximation, metamodeling, response surface method, or model emulation. Once the surrogate model is constructed, a global optimization algorithm can be used to identify the optimal parameter set. These kinds of algorithms are called surrogate modeling-based optimization methods. Last, we summarize several fundamental differences between single and multiple objective optimization due to the fact that many real-world problems are intrinsically multiobjective optimization problems. Examples of using some introduced methods in real-world study are provided at the end of this chapter, through which authors want to deliver the message that the practical use and selection parameter estimation method should be originated from the ultimate goal of application. There will be no single parameter estimation method that is superior than another considering all aspects of performances, i.e., evaluation metrics. One algorithm is inevitably better than another in a certain way, and vice versa. Detailed introduction of each type of method, discussion of the pros and cons, and examples are included in the rest of chapter.

2 Hydrologic Model Parameter Calibration

The rainfall-runoff conversion is a highly nonlinear process, which can be simulated by a hydrologic model (Chiu and Huang 1970; Kulandaiswamy and Subramanian 1967; Pilgrim 1976; Singh 1964). There exists a plethora of rainfall-runoff models of various complexities, from simple black-box models which are derived from statistical relationships between rainfall and runoff observations, to conceptual models based on the physical principles or empirical relationships among hydrological variables, and to the physically-based distributed models which are based on physical laws of mass, energy, and momentum conservation. Many of those models are shown to be effective in forecasting certain important features of the hydrograph, such as the rising limb, the peaking time and the peak flow rate, and/or the flow volume (Kitanidis and Bras 1980; Sorooshian 1983). All of those models contain model parameters which appear in model equations as constants or exponents and are generally nonobservable at the scale of applications. The ability of the rainfall-runoff models to capture the real-world hydrological processes is dependent on how those parameters are specified (Duan et al. 2006). In practice, model parameters are often tuned to improve the fitting between model simulations and observations. This process is also known as model calibration. Because of the highly nonlinear nature of the hydrological processes, calibration of rainfall-runoff models is faced with enormous challenges that require sophisticated mathematical tools, significant amounts of calibration data, and some degree of model knowledge (Duan et al. 1992).

A rainfall-runoff model can be represented as a mathematical function of numerous variables, including the forcing inputs (e.g., precipitation and temperature), the outputs (e.g., streamflow discharge, evapotranspiration), the transfer functions (i.e., the nonlinear equations governing the relationships among variables), the model states (e.g., river stage, soil moisture storage, snow cover, and snow water equivalent), and the model parameters. The transfer functions (g) consist of either a set of physically based or conceptual hydrologic functions or a list of experimental functions:

$$y_t = g(x_0, x_t, I, \theta) + \epsilon_t \tag{1}$$

where x_t represents the model state variables; x_0 is the initial model states; I is the input variables; θ is the model parameter vector; $y_t$is the model outputs and the last term; and ϵ_t is the model estimation error. Model calibration can be formulated as an inverse problem as illustrated in Fig. 1.

As in any inverse problems, a proper objective function must be specified in order to evaluate the goodness-of-fit between the model simulations and the actual observations. The error term $\epsilon_t = y_t - g(x_0,\ x_t,\ I,\ \theta)$ is a function of the parameter vector θ. θ can be treated as a random variable. The objective function can therefore be represented by a likelihood function expressed as below:

$$L(\theta|\text{data}) = f(y_1, y_2, \ldots, y_n|\theta) \tag{2}$$

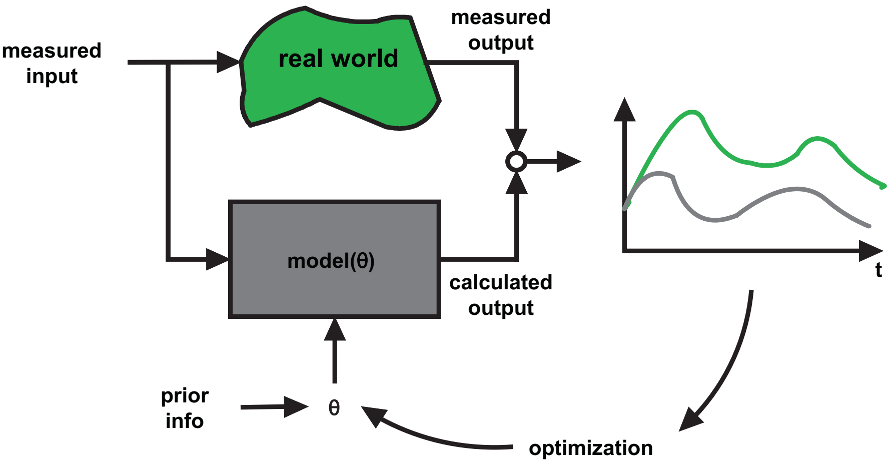

Fig. 1 Strategy for model calibration

where $L(\theta|\ data)$ is the likelihood of θ given the *data*, which is equal to the joint probability density of all observations y_t, $t = 1, \ldots, n$, given θ, $f(y_1, y_2, \ldots, y_n|\ \theta)$. If one assumes statistical independence among y_t, then Eq. (2) can be written as:

$$L(\theta|\text{data}) = \prod_{t=1}^{N} f(y_t|\theta) \tag{3}$$

In practice, it is more convenient to compute the log likelihood function ln. $L(\theta|$ data). For a likelihood function with i.i.d. Gaussian errors, it leads to the equation below:

$$\begin{aligned} \ln.L\big((\theta|\text{data}) &= \ln.\left\{\left(2\pi\sigma_\epsilon^2\right)^{-N/2}\cdot\exp\left[-\frac{1}{2\sigma_\epsilon^2}\left(\sum\nolimits_{t=1}^{N}\epsilon_t^2\right)\right]\right\} \\ &= -N/2.\ln\left(2\pi\sigma_\epsilon^2\right) + \left[-\frac{1}{2\sigma_\epsilon^2}\left(\sum\nolimits_{t=1}^{N}\epsilon_t^2\right)\right] \end{aligned} \tag{4}$$

The maximum likelihood estimate of θ, θ^* can be obtained by solving the following optimization problem:

$$\theta^* = \text{Maximize}_{w.r.t\ \theta}\ln.L\big(\theta|\ \text{data}\big) \cong\ \text{Minimize}_{w.r.t\ \theta}\left(\sum_{t=1}^{N}\epsilon_t^2\right) \tag{5}$$

There are two types of approaches to solving Eq. (5). One approach is known as the deterministic approach which assumes that there exists a unique set of extrema of the objective function (i.e., the maximum or the minimum value of the objective function). Another approach is the stochastic approach or the Bayesian approach which assumes that the optimal solution to Eq. (5) is not a unique set of parameters, but a posterior distribution of θ inferred based on all observations. In this chapter, we focus on estimation of optimal parameters using deterministic approaches. The next chapter discusses the approaches for estimating parameter distributions using stochastic approaches.

3 Overview of Optimal Parameter Estimation Approaches

There are different deterministic approaches to identify the optimal parameter estimates. There is a broad class of local search methods which are presented in classical nonlinear programming textbooks. Those approaches can only guarantee to find a local optimum in the presence of multiple local optima. Local search methods can be further divided into gradient methods which require the calculation of the first and/or the second derivatives of the objective function, and direct methods which do not need the derivative information of the objective function. Obviously, gradient methods require the objective function to be continuous and smooth, which can be a problem for hydrological models as many of them contain threshold parameters (Duan et al. 1992).

Another category of approaches is designed to overcome the limitations of the local search methods. Among the methods in this category, some stochastic and global search mechanisms are often used. Some popular search algorithms include genetic algorithm (GA), the shuffled complex evolution methods developed at the University of Arizona (SCE-UA), particle swarming (PS), ant colony optimization (ACO), and simulated annealing (SA), among others. Many of those algorithms are also called evolutionary algorithms, or EAs because the search strategies follow the evolutionary principles. Another group of global search methods follows the global strategies such as branch and bound methods, cutting plane methods, interval methods, filling function methods, among others. The global search methods usually require smooth and continuous objective functions and are rarely effective in solving high-dimensional hydrological model calibration. Those methods are not reviewed in this chapter. For those interested in those methods, readers may refer to Pintér (1996) and Duan et al. (2006).

More recently, there is a new category of approaches that aim to deal with large complex system models, known as the surrogate modeling-based optimization methods. Those methods are designed to use only a limited number of objective function evaluations to identify the approximate optimal solutions. The idea behind this category of methods is to construct a response surface using a small number of parameter sample sets to approximate the objective function surface. Once this response surface is found to be a reasonable approximation of the objective function, then the search would be conducted on this response surface, which is known as the surrogate model. The solution of this surrogate model would approximate the solution of the dynamic model.

Below, we provide the review of some of the most popular methods that have been used in practice, starting with several local search methods, then global search methods, and finally the surrogate modeling-based optimization methods. We also include a section discussing deterministic multiobjective optimization search algorithms. At the end of this chapter, we provide some examples of the different search algorithms.

3.1 Local Search Methods

3.1.1 Gradient-Based Methods

Parameter estimation methods are used to find the minimum of unimodal functions for which the search is in the direction to improve function value continuously until reaching the local minimum (Singh 1995). If the derivatives of the objective function are used, the local search methods are classified as gradient search methods.

Steepest Descent

Steepest descent method is a gradient-based search method for unconstraint optimization. The search iteration can be written as follows:

$$\theta_{k+1} = \theta_k - \eta_k g_k \tag{6}$$

where η_k is the step size or learning rate; g_k is gradient direction of the likelihood function.

Newton Method

Newton method is also a second-order gradient-based method by taking the curvature of the space into consideration. The interactive form of the search algorithm is as follows:

$$\theta_{k+1} = \theta_k - \eta_k H_k^{-1} g_k \tag{7}$$

This method requires the estimation of H (Hessian matrix), which is a positive defined second-order derivative.

3.1.2 Direct Search Methods

Direct search methods are very popular because of their simplicity and do not require information about the gradient of the objective function.

Downhill Simplex

A direct search method is widely used for the objective function which is not directly differentiable. Downhill Simplex Method (Nelder and Mead 1965) is one popular direct search algorithm. The method uses the concept of a simplex of n dimensional parameters to set $n + 1$ test points to form a simplex. The objective function is evaluated at each test point and all of the simplex points are ordered (sorted) according to the function values from low to high values; meanwhile, the centroid of the simplex is estimated. A new test point is evaluated based on the reflection, expansion, and contraction of the centroid toward the worst point and the worst point is replaced if the new test point is better than the worst point. At the

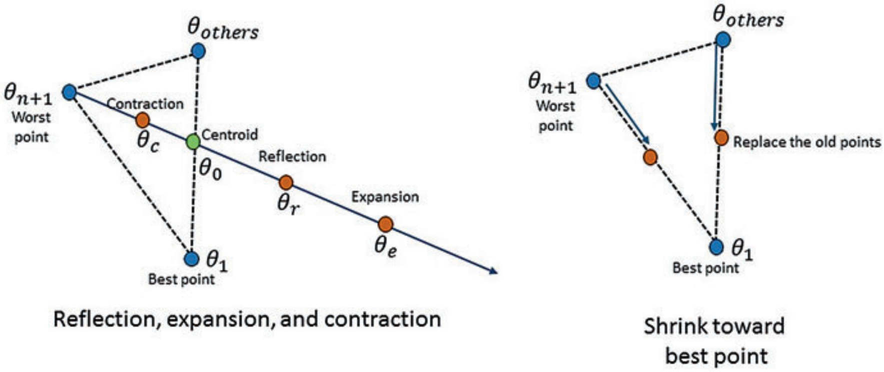

Fig. 2 Reflection, expansion, contraction, and shrinking stage of Simplex search

same time, a new simplex is formed and the same process is repeated to move the simplex forward. On the other hand, if this new test point is not able to have better fitness value than the previous value, the simplex is then shrunk toward its best point. This strategy allows the simplex to continue to evolve till the convergence criteria are reached.

The key simplex evolving steps are shown in Fig. 2 and listed below:

- For an n-dimension parameter space, the test points $\theta_1, \ldots, \theta_{n+1}$are sorted based on their function values, i.e., $g(\theta_1) \leq g(\theta_2) \ldots g(\theta_{n+1})$, and the location of the centroid of the simplex is calculated and represented with θ_0.
- Test of reflection, expansion, contraction points:

$$\text{Reflection} : \theta_{r=}\theta_0 + \alpha(\theta_0 - \theta_{n+1})$$

$$\text{Expansion} : \theta_{e=}\theta_0 + \beta(\theta_r - \theta_0)$$

$$\text{Contraction} : \theta_{c=}\theta_0 + \gamma(\theta_{n+1} - \theta_0)$$

$\alpha, \beta, \gamma > 0$; in general, α, β, γ are set to 1.0, 2.0, and 0.5, respectively.

- Shrink the simplex if the function values of above test points $\{\theta_r, \theta_e, \theta_c\}$ are not better than the function value of the worst point $g(\theta_{n+1})$:
Shrink: $\theta_{i=}\theta_1 + \delta(\theta_i - \theta_1)$ *for all* $i = 2..n + 1\&\delta = 0.5$.

Gradient-based and downhill simplex methods are local search methods, which are capable of finding the local minimum, but have no guarantee to find the global optimal parameter solution. Duan et al. (1992) conducted an analysis of the properties of the response surface associated with a rainfall-runoff model and found that the surface (1) contains more than one main region of attraction, (2) has many local optima within each region of attraction, (3) is rough with discontinuous derivatives,

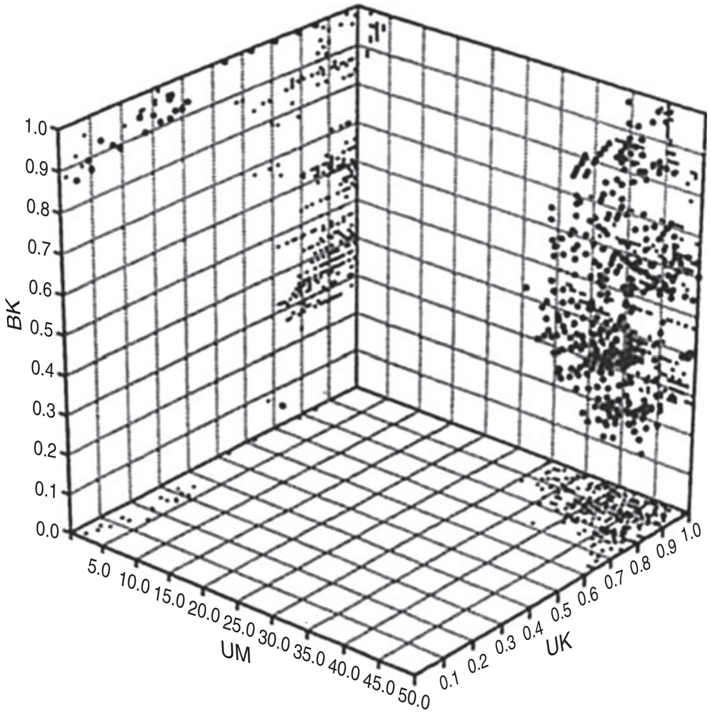

Fig. 3 Three parameters subspace of a simple conceptual catchment model (SIXPAR, Duan et al. 1992), showing locations of multiple local optima

(4) is flat near the optimum with significantly different parameter sensitivities, and (5) includes long and curved ridges. Figure 3 shows many local optima within each region of attraction for a conceptual catchment model of three parameters (Duan et al. 1992). The response surface with above characteristics makes it very difficult to find the global minimum solution.

Finding global optimal solution for a complex optimization problem is of great importance to many real-world applications. Because of a large number of local minimum, it is a challenge for the local search algorithms to find the global optimal solution. Duan et al. (1992) developed a new optimization scheme, which combines the strength of the simplex method with the concept of information sharing from the evolution-based algorithms, and termed it as the shuffled complex evolution global optimization scheme – University of Arizona (SCE-UA). The concept of the SCE-UA algorithm is similar to those used in evolutional algorithms. For the past two to three decades, SCE-UA has been widely used in the parameter estimation of hydrologic models. In the next chapter, we will specifically introduce the mechanisms and concept of the SCE-UA algorithm along with other commonly used global search methods.

3.2 Global Search Methods

3.2.1 Genetic Algorithms

The genetic algorithms (GAs) belong to one of the most popular evolutionary algorithms that mimic the processes of natural selection (Golberg 1989; Holland 1975). The natural selection is defined as the processes that organisms correspondingly survive and produce offspring with the tendency to adapt their environment. There are different types of natural selection processes, including chromosome heredity, mutation, crossover, and selection. The following Fig. 4 illustrates a conceptual GA algorithm, in which a population consists of a group of individuals (dashed-box 1). Each individual is denoted as a chromosome, which has a set of properties as genes. The chromosomes are able to mutate, carryover, and crossover with other chromosomes to produce a new generation of offspring. The production of next offspring generation is repeated following the natural selection criteria. After producing a number of new generations, the entire population will gradually carry the "good" genes from the chromosomes that have better fitness with respects to their environment, and eliminate the "bad" genes which are not suitable for surviving. In other words, the natural selection processes tend to generate offspring with better suitability to survive under the pressures from their living environment.

According to Simpson et al. (1994), the optimization of a particular problem using GA is achieved through the following concepts. First, the initial population is created by randomly selecting a number of individuals in the searching space, and each individual is called a chromosome. Second, the objective function value for each feasible solution, or individual, is defined as the fitness of chromosome to its environment. The individual (chromosome) with better objective function value (or fitness) is assumed to possess better genes (parameters), and therefore, has a higher chance to be selected to produce next generation. Last, when producing offspring from the selected parent individuals (chromosomes), the percentages of genes in each parent chromosome to crossover, carryover, and mutate are defined as algorithm parameters, i.e., crossover rate, elite rate, and mutation rate, respectively.

A generalized procedure of implementing GA is summarized as follows:

1. *Define objective function*:
   Assign objective function (see Eq. (5)): $f(\theta_1, \theta_2, \ldots, \theta_n)$, where n is the number of dimension.
2. *Initialization:*
   Randomly sample k individuals in the parameter space to form the population $P = \{p_1, p_2, \ldots, p_k\}$. Each p_i, $i \epsilon 1, 2, \ldots n$ is defined as an individual chromosome as shown in the dashed-box 1 of Fig. 4.
3. *Selection:*
   Evaluate the fitness, i.e., objective function values for all individuals in population, and recursively select two individuals as parents for producing offspring. In the example shown in Fig. 4, chromosome 1 and 2 are selected as parents. Elite members, i.e., the individuals with high fitness values, are also selected and directly copied to next generation without any changes.

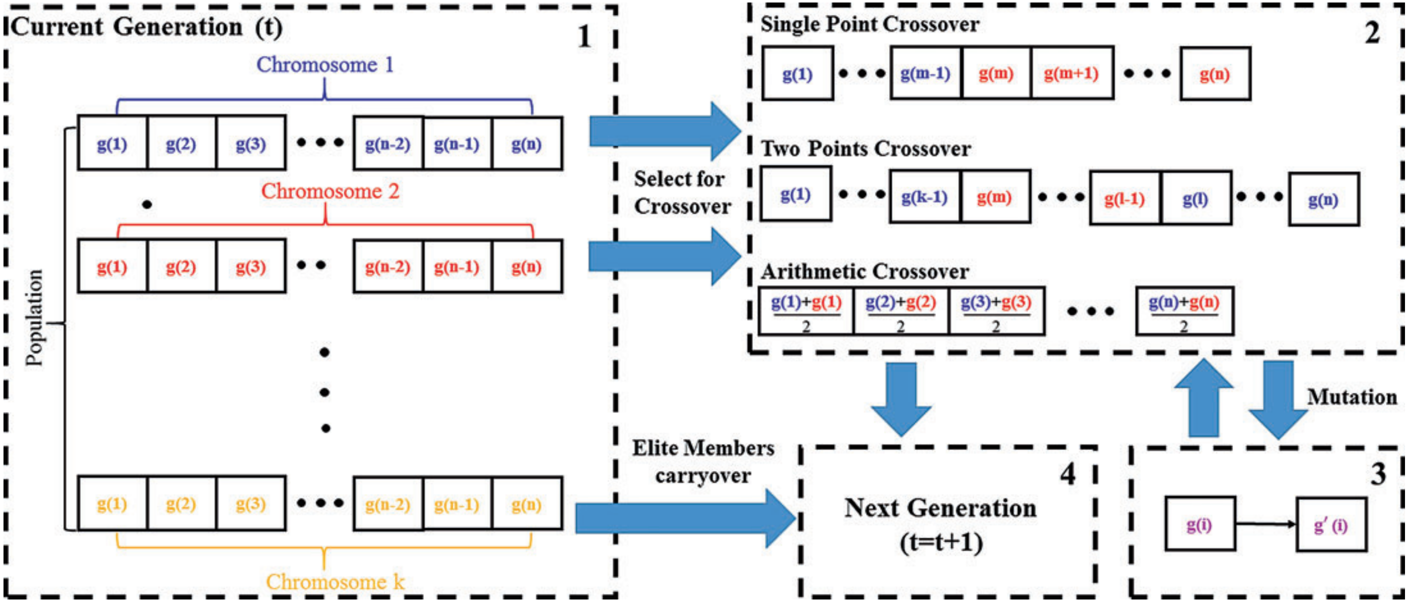
Current Generation (t)
1
Chromosome 1
g(1) g(2) g(3) • • • g(n-2) g(n-1) g(n)
Chromosome 2
g(1) g(2) g(3) • • g(n-2) g(n-1) g(n)
Population
g(1) g(2) g(3) • • • g(n-2) g(n-1) g(n)
Chromosome k
Select for Crossover
Elite Members carryover
2
Single Point Crossover
g(1) • • • g(m-1) g(m) g(m+1) • • • g(n)
Two Points Crossover
g(1) • • • g(k-1) g(m) • • • g(l-1) g(l) • • • g(n)
Arithmetic Crossover
$\frac{g(1)+g(1)}{2}$ $\frac{g(2)+g(2)}{2}$ $\frac{g(3)+g(3)}{2}$ • • • $\frac{g(n)+g(n)}{2}$
Mutation
4
Next Generation
(t=t+1)
3
g(i) g′ (i)

Fig. 4 A conceptual diagram showing the procedures of GA, i.e., crossover, carryover, and mutation operations from generation t to t + 1

4. *Crossover:*
   The production of offspring is firstly conducted by crossover operation on the selected parental individuals. There are many types of crossover operations as shown in the dashed-box 2 of Fig. 4: (a) Single-point crossover, in which a gene location m ($m \epsilon$ 2, 3…n − 1) is defined and the genes of offspring before this location are from the first parent and the rest genes come from the second parent. (b) Two-point crossover, in which another gene location l (l ϵ 2, 3…n − 1, and $l > m$) in the offspring is further defined and the genes after location l are copied from the first parent again instead of using the genes from the second parent. (3) Arithmetic crossover, in which the offspring genes are obtained from both parents with a average of the values of genes.
5. *Mutation:*
   As shown in the dashed-box 3 of Fig. 4, after crossover operation, a portion of offspring genes is able to mutate from $g(i)$ to $g'(i)$, where i ϵ 1, 2…, n, i.e., for real-value encoded GA, the mutation is conducted by adding a small random number to the values of offspring genes who are selected to mutate during an iteration. This mutation strategy prevents the population from trapping in local minima, and premature converging. The new genes $g'(i)$ are copied back to each offspring and replace the original genes $g(i)$.
6. *Next Generation:*
   The new population for generation (t + 1) consists of the offspring from the crossover and mutation operations, as well as the elite members directly copied from previous generation (t).
7. *Termination:*
   Repeat steps (3)–(6) until stopping criteria are met, i.e., total number of function evaluation reaches user-defined maximum, total number of generation reaches user-defined maximum, or the average relative changes in the objective function values over a number of generation is less than the function tolerance, etc.

The applications of GA in hydrological model calibration are extensive. Some early attempts to using GA in the field of automatic calibration of hydrological model parameters include the works by Wang (1991), Franchini (1996), Franchini and Galeati (1997), Wang (1997), Balascio et al. (1998), Savic et al. (1999), and Whigham and Crapper (1999). The usefulness of GA has also been demonstrated in many different hydrological models, such as the HYMOD model, SWAT model, the Xinanjiang Model, etc., as demonstrated by numerous studies (Babovic and Keijzer 2002; Liong et al. 2002; Srivastava et al. 2002; Cheng et al. 2006; Francés et al. 2007; Lin and Wang 2007; Zhang et al. 2009b; Wu et al. 2012). Some recent comparisons of GA against other stochastic optimization schemes are available from Wang et al. (2010) and Arsenault et al. (2013) for interested readers.

3.2.2 Simulated Annealing

The simulated annealing (SA) algorithm was originally introduced by Kirkpatrick (1984) as a robust global optimizer for addressing the issue of trapping in local minimums of classical gradient descent method. The concept of SA was inspired by the process of annealing in metal work, in which a metal material was repeatedly heated and cooled down to improve the stiffness of metals. The heating process allows the metal molecules to vibrate in their neighborhood, and partially breaks the molecular bonds. The cooling process re-forms the molecular structure, and re-combines a stronger molecular bond, so that the whole physical system reaches an entropy maximum state. This metal work annealing concept can be creatively used for finding global optimums on multimodality response surfaces (Eglese 1990), and many real-world problems, i.e., the travel salesman (Černý 1985).

In SA implementation, it follows two conditions: (1) when the temperature is high, the status of the system is free to move to other energy states through random work and (2) when the temperature is lower, the system states are becoming restricted and therefore, the solutions can only move toward regions where energy states are lower. In each iteration, a nearby region near the current solution is tested. If the objective function of the new test point is better than the old one, the new point is used to replace the old point. Otherwise, a probability being a function of the annealing temperature is assigned to the new test point to decide whether this test point is acceptable. When the annealing temperature continues to move lower, the acceptable probability for a worse solution becomes lower. To accept worse solutions can be referred as a "hill-climbing" procedure, whereas this search strategy allows the algorithm to have the capability of escaping from local minimums. As the annealing temperature decreases, the chance of accepting worse solution will decrease. It is expected that only better solution is acceptable when annealing temperature reaches minimum. The annealing temperature can be controlled by a cooling scheme specifying how it should be progressively reduced over iteration. Theoretical study has proved that the algorithm can converge toward the optimal solution in a asymptotic manner (Granville et al. 1994).

The SA algorithm in general follows six steps as shown below:

1. *Define objective/energy function:*
   Assign objective function or an energy function (see Eq. (5)): $f(\theta)$.
2. *Initialization:*
   Assign initial parameters ($\theta_{t=0}$), terminal temperature (T_E), cooling rate ($\propto$), for t = 0.
   Find the solution ($f(\theta_{t=0}$) of the initial parameters $\theta_{t=0}$.
3. *Selection of a new point (iteration):*
   Set t = t + 1; generate a new parameters θ_t near θ_{t-1}.
4. *Selection/rejection of the selected point:*
   Use Metropolis acceptance rule to accept or reject θ_t:
   i.e., estimate $\Delta E = f(\theta_t) - f(\theta_{t-1})$;
   For a minimization problem, θ_t is accepted to replace θ_{t-1} if $\Delta E<0$.
   Otherwise, θ_t can be accepted to replace θ_{t-1} with a probability of $p = e^{-\Delta E/T}$.
5. *Adjustment of anneal temperature:*
   One way to adjust the anneal temperature is to reduce the temperature over time, such as T=$\propto \times T$ and $\propto \in [0, 1]$.

6. *Terminate:*

   If T>T_E, repeat steps (3)–(5).

   Otherwise, terminate the process. The final solution is assigned to θ_t. In addition, once other user-defined stopping criteria are met, i.e., the maximum of number of function evaluation is reached, or as the stagnation time of annealing temperature becomes zero, etc., the process is terminated.

With the extensive uses of SA in various fields, particularly, in the field of automatic calibration of hydrological model parameters, many studies have shown the usefulness of SA with different case studies. Bates (1994) pioneered in the use of SA for calibrating an SFB conceptual rainfall-runoff model. Sumner et al. (1997) applied a modified SFB model and SA optimization scheme for a large-scale case study in Australia. Thyer et al. (1999) and Madsen et al. (2002) compared SA strategy with many other population-based optimization schemes with regard to its performances in calibrating conceptual rainfall-runoff models. Bárdossy and Das (2006) applied the SA to a semidistributed HBV model and shown a good capability of SA in tuning model parameters. A number of other stochastic optimization algorithms, such as the Shuffled Complex Evolution Metropolis algorithm – University of Arizona (SCEM-UA) (Vrugt et al. 2003a), were also developed based on the combination uses of annealing concept of SA, and the Metropolis-Hasting algorithm (Hastings 1970).

3.2.3 Particle Swarm Optimization

Similar to the GAs, the particle swarm optimization (PSO) is another extensively used, population-based global optimizer, which simulates the social-individual behaviors of bird flocking and fish schooling (Kennedy 2011; Kennedy et al. 2001). The particle swarm optimization belongs to one type of swarm intelligence, in which the *particles* that mimic the behaviors of social animals are swarming in a manner of strategic movements, instead of randomly moving in the searching domain. Different from the adopted natural selection criteria in GA, i.e., mutation or crossover, in PSO the offspring production is based on the fitness of *particles* and their movement velocities toward the locations of current best, as well as the historical best location so far. This is a simplified social behavior of bird foraging. According to Eberhart and Kennedy (1995) and Shi (2001), there are three assumptions when interpreting the birds foraging behavior into PSO algorithms. First, all birds (*particles*) are assumed to be blind with regards to (i.e., do not know) the location of best food source (global optimum). Therefore, one of the effective foraging strategies for all birds (*particles*) is to fly toward the bird which is nearest to the food (the *particle* that has the best fitness). Secondly, each bird (*particle*) is assumed to be intelligent enough to remember the distance of the historical locations to food source (i.e., the fitness values during all movements during the entire search). Last, each bird (*particle*) is able to collectively adjust its next movement direction and position (i.e., the next moving direction and distance). Therefore, in PSO the population is updated by recursively approaching two best positions: (1) the best location that gives the best fitness value within current population, and (2) the historical best location that gives the best fitness value through the entire evolution that the algorithm has achieved so far. In summary, the information sharing mechanism sorely relies on the best individual instead of chromosomes exchanges and mutations. In PSO, the movement of population is always toward the best two members, while in GA, the individuals move as a group approaching the global optimum (Panduro et al. 2009). One similarity between PSO and GA is that the population is randomly sampled from the feasible solution space (Arsenault et al. 2014).

A generalized procedure of implementing PSO is summarized as follows:

1. *Define objective function*:
   Assign objective function (see Eq. (5)): $f(\theta_1, \theta_2, \ldots, \theta_n)$, where n is the number of dimension.
2. *Initialization:*
   Randomly sample k individuals (*particles*) in the parameter space to form the population $P = \{p_1, p_2, \ldots, p_k\}$, and define each particle's neighborhood as $\mathcal{N}_i \in \mathrm{P}$.
3. *Evaluation:*
   Evaluate the fitness, i.e., objective function values for all *particles.*
4. *Swarming:*
   Swarm each *particle* (p_i) in its neighborhood ($\mathcal{N}_i$), and store the neighborhood best location l_i, called *local best* location, for each *particle.* The *local best* location also includes the particle's current location before swarming. Then, evaluate the historical best location each particle reached so far, and store as g_i, called *global best location*. Note that *global best* for each particle is not worse than the *local best*, i.e., $f(l_i) \leq f(g_i), \forall p_i \in \mathcal{N}_i$.
5. *Update velocity*
   The movement velocity for *particle* (p_i) at step t is defined as:

$$v_i^{t+1} = wv_i^t + c_1 U_1^t \left(g_i^t - x_i^t\right) + c_2 U_2^t \left(l_i^t - x_i^t\right) \tag{8}$$

   where w is an user-defined algorithm parameter called *inertia weight*; c_1 and c_2 are also user-defined algorithm parameters called *acceleration coefficients*; U_1^t andU_2^t are n by n diagonal matrixes with diagonal components randomly drawn from a uniform distribution in the interval of [0, 1).
6. *Update particle location*
   The next movement location for each particle (p_i) is denoted as x_i^{t+1}, and it is updated based on previous location (x_i^t) and the movement velocity v_i^{t+1} that is obtained from step (5). The equation for updating *particle* location is expressed as:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{9}$$

7. *Termination:*
   Repeat steps (3)–(6) until stopping criteria are met, i.e., total number of function evaluation reaches user-defined maximum, total number of generation reaches

user-defined maximum, or the average relative changes in the objective function value over a number of generation is less than the function tolerance, etc.

The use of PSO in the field of automatic calibration of hydrologic model parameter is extensive. Gill et al. (2006) tested both single-objective PSO and multiobjective PSO in calibrating the sacramental soil moisture accounting model, which has 13 parameters. Chau (2008) applied PSO to train a data-driven rainfall-runoff model. Zhang et al. (2009c) compared both GA and PSO with regard to their performances of calibrating Soil and Water Assessment Tool (SWAT) model. Zhang and Chiew (2009) applied PSO scheme in both the Xinanjiang and SIMHYD hydrological models. Further investigation of parameter sensitivity of the Xinanjiang model using PSO is also conducted by Kuok and Chan (2012) and Lü et al. (2013). Kamali et al. (2013) applied both single- and multiple objective PSO algorithms to the HEC-HMS model. A recent comparison of many stochastic optimization schemes in calibrating hydrological models is made available by Arsenault et al. (2014). In addition, PSO is a very useful tool in various fields, such as water stage forecasting (Chau 2007), power system design (Abido 2002), ground water management (Zambrano-Bigiarini and Rojas 2013), etc.

3.2.4 Ant Colony Optimization

The ant colony optimization (ACO) belongs to another type of swarm intelligence, which was first introduced by Dorigo (1992) in his Ph.D. dissertation, and further developed by Marco Dorigo and his colleagues (Dorigo et al. 2006; Dorigo and Blum 2005; Dorigo and Stützle 2009). The concept of ACO followed the social foraging behavior of social insects, in particular, the strategy that ants find food sources and the development of optimal paths from food sources to their nest. According to Marco Dorigo's description and the biology study by Deneubourg et al. (1983), ants initially are able to explore the area near their nest in a random manner. When ants are randomly moving on the ground, a chemical pheromone trail is left by each individual ant, which is detectable by other ants. An individual ant tends to follow the path, in probability, has the strongest pheromone concentrations that marked by other ants. Once a food source is located by an individual ant, this ant will evaluate the quantity and quality of the food source, and carry a small portion of food back to its nest. On the way back to the nest, the pheromone left by this ant will correspondingly change based on the quantity and quality of food source, so that other ants can be guided to this discovered food source. By using the pheromone trails, ants are able to indirectly exchange information of the location, quantity, and quality of food source. This communication strategy via pheromone trails is proven to be effective allowing ants to find the shortest paths between the food sources and nest (Deneubourg et al. 1990).

In ACO, an instantiated decision variable $X_i = v_i^j$ (i.e., a variable X_i with a value v_i^j assigned from its parameter domain θ_i) is termed a *solution component* and denoted by c_{ij}, where i and j are the locations connecting a searching domain. τ_{ij} is the pheromone value or intensity associated with each solution component c_{ij}, and it is continuously updated based on time and the behavior of all ants. All possible *solution components* and feasible solutions consist a complete set of $\mathcal{N}\left(s^k\right)$, where s^k is a partial feasible solution that constructed from an empty set s by adding the 1st, 2nd … and kth *solution component* from the complete feasible solution set $\mathcal{N}$. A generalized procedure for ACO is as follows:

1. *Define the objective function:*
   Assign objective function for the optimization problem (see Eq. (5)): $f(\theta)$.
2. *Initialization:*
   Assign the number of ants (M) in ASO, locations of the ants in searching space, and randomly assign pheromone values τ_{ij} for each *solution components* c_{ij} that connect location i and j.
3. *Define the pheromone model:*
   A commonly used pheromone model for ant system (Dorigo et al. 1996) is

$$p(c_i|s) = \frac{\left[\tau_{ij}\right]^{\alpha}\cdot\left[\eta(c)_{ij}\right]^{\beta}}{\sum_{c_{il}\epsilon\mathcal{N}(s^k)}\left[\tau_{il}\right]^{\alpha}\cdot\left[\eta(c)_{il}\right]^{\beta}} \tag{10}$$

   where α and β are algorithm parameters that control the significance of pheromone value τ_{ij}, and the visibility of pheromone trials $\eta(c)_{ij}$; the visibility $\eta(c)_{ij}$ is defined as the inverse Euclidean distance between location i and j.
4. *Movements of ants*
   When individual ant (m) is in the position i and so far constructed the partial solution s^k, the probability of moving from position i to position j is given by the pheromone model in step (3). Each ant will move to its next position until all the M ants finish their movements.
5. *Update pheromone values:*
   The pheromone values τ_{ij} are updated for all the M ants according to the following equation:

$$\tau_{ij,t} = (1-\rho)\cdot\tau_{ij,t-1} + \sum_{m=1}^{M}\Delta\tau_{ij,t-1}^{m} \tag{11}$$

   where ρ is an evaporation rate of pheromone, which uniformly decreases all the pheromone values in order to prevent the algorithm from a rapid convergence toward suboptimal; $\Delta\tau_{ij,t-1}^{m}$ is the quantity of pheromone left on a path connecting position i and j by mth ant during the previous movement.
6. *Termination:*
   Repeat steps (3)–(5) until stopping criterion are met, i.e., the number of function evaluation reaches user-defined maximum, the number of total number of ants movement cycle reaches user-defined maximum, or the fitness (objective function values) for all M ants are limited within function tolerance (i.e., convergence is reached).

The original ACO was initially invented for solving optimization problems with discrete decision variable domain, and for finding optimal combinations of components, such as the travel-sale-man problem. Socha and Dorigo (2008) further developed the original ACO to solve optimization problems with a continuous domain. According to the literature, there are many successful applications of the original ACO and its developed versions to solve different problems in various fields, such as soil hydraulic parameters calibration (Abbaspour et al. 2001), water quality (Bowden et al. 2002), optimal open channel design (Nourani et al. 2009), hydrological model calibration (Olarte and Obregon 2004), water distribution system design and planning (Maier et al. 2003; Wang and Guo 2010; Zecchin et al. 2003), and optimal reservoir operation (Kumar and Reddy 2006; Madadgar and Afshar 2009; Zecchin et al. 2012). Two most recent reviews of ACO and its applications can be found in Afshar et al. (2015) and Ostfeld (2011) for interested readers.

3.2.5 Shuffled Complex Evolution-UA (SCE-UA)

The SCE-UA algorithm is a global search algorithm, which combines a number of different strategies, including the Downhill Simplex, the Controlled Random Search, the Competitive Evolution, and the Complex Shuffling scheme (Duan et al. 1992, 1994). Extensive testing of the SCE-UA algorithm by numerous researchers has proven its effectiveness and efficiency in reliably finding the global solution, when a unique solution exists. The SCE-UA algorithm includes the following steps:

1. *Initialization:*
   Generate parameter samples θ_i from the feasible parameter space. Calculate the objective function value of each sample $f(\theta_i)$. Set initial sample size $s = pm$, where p is the number of complexes and m is the number of points in each complex.
2. *Rank samples:*
   Sort the s samples based on $f(\theta_i)$ from small to large values, i.e., $f(\theta_i) \leq f(\theta_{i+1})$.
3. *Partitioning into complexes:*
   The s samples are partitioned into p complexes, such that the complex k includes samples of $\{\theta_k, \theta_{p+k}, .., \theta_{(m-1)*p+k}\}$ and function value of samples: $\{f(\theta_k), f(\theta_{p+k}), .., f(\theta_{(m-1)*p+k})\}$; $k = 1..m$ complexes.
4. *Evolution of complexes:*
   Based on a trapezoidal probability distribution, in which higher probability is assigned to lower function values, select a subcomplex of q samples from each complex. Use downhill simplex algorithm to evolve the samples in each subcomplex.
5. *Complex shuffling:*
   Include all samples from the subcomplex in the sample pool.
6. *Termination:*
   Repeat steps (2)–(5) until the stopping criteria are reached.

SCE-UA has been extensively used in hydrologic modeling and has shown to be robust and efficient for hydrologic model calibration. Gan and Biftu (1996) applied

SCE-UA to multiple operational Conceptual Rainfall-Runoff (CRR) models and proved the effectiveness of SCE-UA scheme in calibrating different CRR models. Eckhardt and Arnold (2001) used SCE-UA algorithm to calibrate the parameters of the SWAT model over the Dietzhölze catchment in central Germany and reached high accuracy. Skahill and Doherty (2006) demonstrated the strengths of SCE-UA and other automatic parameter calibration scheme on an operational hydrological model for the Wildcat Creek watershed located in Kitsap County, Washington. Yang et al. (2008) demonstrated the effectiveness and efficiency of SCE-UA over other automatic parameter calibration schemes using the SWAT model over the Chaohe Basin in China. Ludwig et al. (2009) compared a fully distributed, a semidistributed, and a lumped hydrological model using SCE-UA as automatic calibration scheme over the Ammer basin in the Southern Bavaria, Germany. Khakbaz et al. (2012) employed SCE-UA for calibrating the SAC-SMA models over the Illinois River Basin at Siloam Springs, Arkansas, and produced satisfactory streamflow simulation. Liu et al. (2017) applied SCE-UA to calibrate an operational hydrological model used by Tibet Government for simulating the streamflows for the Upper Yellow and Upper Yangtze River basins of China. There are numerous other successful applications of SCE-UA algorithm in the field of hydrological model parameter calibration. Authors are only able to provide limited references. The original publication of SCE-UA ranked as the top three most cited articles in Water Resources Research (data collected in Sep 2016). However, when the dimension of a given problem increases extensively, the global convergence of SCE-UA algorithm might not be guaranteed. The population could collapse to a subspace of the full span of the parameter space, which impedes SCE-UA algorithm to exploit the parameter space (Chu et al. 2010). Chu et al. (2010, 2011) further improved the SCE-UA algorithm with a principal component analysis for a remedy of this issue. The enhanced version of SCE-UA is termed as the Shuffle Complex Evolution global optimization with Principal Component Analysis – University of California, Irvine (SP-UCI). In SP-UCI, principal component analysis (PCA) is used to detect the occurrence of population degeneration. The PC coordinate system is determined by the data samples. By adding new particles along the PC with zero (or relatively small) variance, the search is ensured to maintain the diversity of the entire population, especially along the collapsed dimension. The SP-UCI algorithm is also proven to be effective and efficient for many high dimensional real-world applications (Chu et al. 2014; Yang et al. 2015, 2017a).

3.3 Surrogate Modeling-Based Methods

The global optimization methods generally require up to tens of thousands of model runs to find the global optimal solution. This may place significant computational burden on solving such an optimization problem, if the underlying model requires a large amount of CPU time to run. One approach to reduce the computational burden is to approximate and replace the expensive simulation model with a cheaper-to-run surrogate model. Some fields also refer to the surrogate modeling as function approximation, meta-modeling, response surface method, or model emulation (Blanning 1975; O'Hagan 2006). Once the surrogate model is constructed, a global optimization algorithm can be used to identify the optimal parameter set. These kinds of algorithms are called surrogate modeling based optimization methods (Simpson et al. 2001; Jin et al 2001; Queipo et al. 2005; Razavi et al. 2012).

A surrogate model can be understood as a "model of model." It is a statistical model of the response surface of a simulation model. A surrogate model describes the relationship between inputs (i.e., model's adjustable parameters) and outputs (i.e., the performance measure of the simulation model). Training an accurate surrogate model needs adequate input–output data, which are obtained by running the simulation model with different sets of parameters selected in the feasible parameter space. Previous studies use the "one-shot" approach (i.e., using a set of samples at once) to obtain input-output data to construct the surrogate model. This method directly establishes a surrogate model on the utilized data. Then it runs global optimization algorithm on the surrogate model. A high number of model runs may be required to ensure that the surrogate model represents the response surface of the original simulation model well. One way to economically construct a surrogate model for optimization is to use adaptive sampling. Adaptive sampling means that a certain number of points are sampled in the initial stage, and then a number of additional points, which can most effectively increase the accuracy of the surrogate model, are adaptively sampled and added to the initial sampling. For the purpose of finding an optimum, it is not necessary to map out the whole surface in a surrogate model exploration. An adaptive sampling strategy can quickly move the experiment to a region containing the optimum of the input variables. Only within this region is a thorough exploration of the surrogate model warranted to find the optimum (Wu and Hamada 2009). Below is the procedure of adaptive surrogate modeling based optimization algorithm:

1. Generate an initial experimental design spread over the entire input space and do the costly function evaluations at the points.
2. Use function evaluations to fit a statistical surrogate model for the objective function.
3. Use the surrogate model to predict the objective function values at unsampled points in the variable domain to decide at which point to do the next expensive function evaluation.
4. Do the expensive function evaluation at the point selected in Step 3. Then, use the new data point to update the surrogate model.
5. Iterate through steps (3)–(4) until the stopping criterion has been met. Take the final optimal value on the newly updated surrogate model as the global optimization result.

There are three main components of adaptive surrogate modeling-based optimization algorithms, named initial sampling, constructing the surrogate model, and adaptive sampling. Below we described these three steps more specifically.

1. *Initial sampling*

The initial sampling is the sampling plan in the design variable space, also called experimental design. It is a body of techniques that enable an investigator to conduct better experiments, analyze data efficiently, and make the connections between the conclusions from the analysis and the original objectives of the investigation (Wu and Hamada 2009). Generally, at this stage, the location of the points is only required to satisfy some space-filling criteria. Because of the absence of prior knowledge of the problem under consideration, uniformity of the design points throughout the domain is favorable. The optimum size of initial sample points is an open question. Some people think it should be ten times the number of dimensions (Jones et al. 1998). Others think it should keep the initial sample size to a minimum (for example: two times the number of dimensions) (Sóbester et al. 2005; Regis and Shoemaker 2007). However, the initial sample size is highly correlated with the initial sample methods. For Latin Hypercube method, too small sample size leads to a slow convergence for the optimization problem. For a more uniform low-discrepancy quasi-Monte Carlo method, a small sample size makes the better results (Wang et al. 2014).

2. *Constructing the surrogate model*

Generally, surrogate model construction methods are statistical regression methods that estimate response surface of a simulation model. A variety of approximation techniques have been developed and applied as the surrogates of an original simulation model: polynomial regression (Fen et al. 2009), regression tree method (Breiman et al. 1984; Yang et al. 2016), Random Forest method (Breiman 2001; Yang et al. 2017b), Multivariate Adaptive Regression Splines (Friedman 1991), Support Vector Machines (Zhang et al. 2009a), Artificial Neural Networks (Behzadian et al. 2009), and Gaussian Process (Rasmussen and Williams 2006; Snelson 2007). At the highest level, response surfaces can be differentiated based on whether they are noninterpolating (i.e., it minimizes the sum of squared errors from some predetermined functional form) or interpolating (i.e., it passes through all points). It has been suggested that noninterpolating surfaces, such as fitted quadratic surfaces, are unreliable for surrogate-based optimization because the surface may not sufficiently capture the shape of the function (Jones 2001). On the other hand, interpolating methods can get more and more accurate as new points are added, eventually converging to the true function.

3. *Adaptive sampling*

Adaptive sampling methods (also called sequential design methods) are iterative algorithms that use data acquired from previous iterations to guide future sample selections. The points we selected are also called infill samples (Sóbester et al. 2005). All of the aforementioned approaches select a sample point by optimizing an auxiliary function (minimize the bumpiness measure, maximize the expected improvement, minimize the response surface), which is in general itself a global optimization problem. Adaptive sampling methods allow significant reduction in the number of simulations of the original simulation model because they only search the area that may contain the optimum of the input variables.

3.4 Deterministic Multiobjective Search Methods

The multiobjective optimization scheme can be referred as an application of single-objective optimization for handling multiple objectives (Deb 2001). A classical approach of solving a multiobjective optimization problem is to create a new composite function by giving individual weight to each single-objective function and adding them together. Then, the new weighted objective function is optimized using classical gradient-based methods or direct search schemes. This is a very straightforward approach. However, the fundamental differences between multi-objective optimization and single-objective optimization are ignored when using the transformation of multiple single-objective functions into a weighted composite function.

One of the fundamental differences between multiobjective optimization and single-objective is the existence of trade-offs among different competitive objective functions. In other words, for any multiobjective optimization problem, any gain with respect to the fitness of one objective function requires a sacrifice of the fitness in another objective function. This is due to the nature of constraints associated with any given multiobjective optimization problem. For example, conceptually a shopper can only get either item A or item B from a supermarket, because the total costs of both items A and B is beyond his/her fixed total budget. Under another circumstance, there are five different brands of the identical item (A, A', A'', A''', and A'''') available with different qualities and costs (subject to the fact that a higher quality item is associated with a higher cost). The item quality that the shopper gets from item A and his/her remaining budget become two competitive objective functions. These two shopper's situations will be further explained with illustration in the following sections and Fig. 5. Many real-word problems are essentially balancing different objective functions and benefit gains. For instance, in California, USA, the water in the Northern California is diverted by the California State Water Project to multiple water demand sectors in the Central and Southern parts of California, including ecosystem, industry, resident, agriculture, etc. The amounts of water allocation to different sectors are conflicting objectives with the constraint of total available surface water. In another reservoir operation example, assuming water can be released through either spill gates or hydropower turbines, the turbine flows and spills become two competing objectives. This is because that once the water is spilled, it is not able to be retrieved for hydropower generation. The constraints include the risks of dam seepage, flooding of downstream areas, and other facility engineering requirements.

Given an optimization problem with two or more competitive objective functions, the trade-offs among those objective functions essentially mean any gain in the

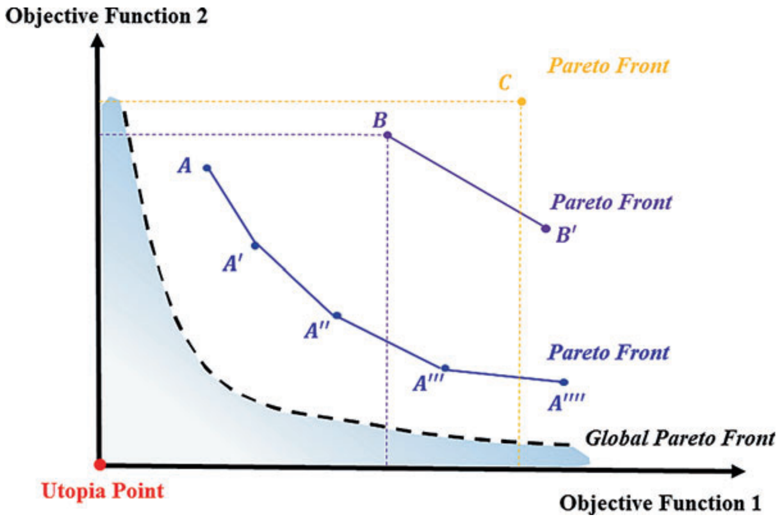

Fig. 5 An example for a two-objective minimization problem

fitness of one of the objective functions will call for the loss of fitness in at least one other objective functions of the optimization problem. Based on Eqs. (2) and (5), the mathematical expression of a multiobjective minimization problem with k objective functions is as follows:

$$\begin{cases} L_1(\theta_1|\text{data}) = f_1(y_1, y_2, \ldots, y_n|\theta_1) \\ L_2(\theta_2|\text{data}) = f_2(y_1, y_2, \ldots, y_n|\theta_2) \\ \vdots \\ L_k(\theta_k|\text{data}) = f_k(y_1, y_2, \ldots, y_n|\theta_k) \end{cases} \tag{12}$$

$$\begin{cases} \theta_1^* = \text{Maximize}_{w.r.t\ \theta_1} \ln.L_1(\theta_1|\text{ data}) \cong \text{ Minimize}_{w.r.t\ \theta_1}\left(\sum_{t=1}^{N}\epsilon_t^2\right) \\ \theta_2^* = \text{Maximize}_{w.r.t\ \theta_2} \ln.L_2(\theta_2|\text{ data}) \cong \text{ Minimize}_{w.r.t\ \theta_2}\left(\sum_{t=1}^{N}\epsilon_t^2\right) \\ \vdots \\ \theta_k^* = \text{Maximize}_{w.r.t\ \theta_k} \ln.L_k(\theta_k|\text{ data}) \cong \text{ Minimize}_{w.r.t\ \theta_k}\left(\sum_{t=1}^{N}\epsilon_t^2\right) \end{cases} \tag{13}$$

where $\theta(\theta_1, \theta_2, \ldots, \theta_k)$ are subject to constraints $\Omega(\Omega_1, \Omega_2, \ldots, \Omega_k)$, respectively.

Another difference between multi- and single-objective optimization is the notion of global optimality. In single-objective optimization, there is only one global optimum, while in the multiobjective context there exists multiple solutions that form a global optimal solution set, called *Global Pareto Optima* or *Global Pareto Front*. For example, in a two-objective minimization problem shown in Fig. 5, the x- and y-axis represent the first and second objective function, respectively. The origin represents the minimization for all objectives without any constraint. However, due to imposed constraints, this point, called Utopia Point, as well as the the light blue area in Fig. 5 are not feasible in reality. In the example of Fig. 5, the dashed line indicates the global optimal solutions of the conceptual two objective minimization problem, in which multiple global optimal solutions form a solution set, and this set of solutions is termed Global Pareto Optima or Global Pareto Front. For problems with higher number of dimensions, the *Global Pareto Optima* or *Global Pareto Front* can be a 3-D surface or a high-dimensional subspace. The actual shape and location of *Global Pareto Front* are dependent on the dimensionality, the conflicting characteristics of selected objective functions, and the parameter bounds.

In multiobjective optimization, the fitness of candidate solutions is defined based on the concepts of *Pareto Optimality* and *Pareto Front*, instead of simply using objective function values as solution fitness. For any multiobjective minimization problem, according to Deb (2001), a solution x is said to dominate the other solution x', if (1) solution x is no worse than x' in all objectives, i.e., $\forall j\epsilon(1, 2, \ldots, k) : f_j(x) \leq f_j(x')$ and (2) solution x is strictly better than x' for at least one objective, i.e., $\exists j\epsilon(1, 2, \ldots, k) : f_j(x) < f_j(x')$. Among the eight solutions shown in Fig. 5, for instance, solution B dominates solution C, while solution B' does not dominate solution C. Similarly, both solutions A' and A''' dominate solution B and solution C. If there is no single solution dominating any others among a set of solutions, then this set of solutions forms a *Pareto Front* and is defined as a *nondominated solution set*. In Fig. 5, three different *Pareto Fronts* are defined with different colors, namely, the solution set $(A,\ \ A',\ \ A'',\ A''', A'''')$, $(B,\ \ B')$, and (C). In each of the *nondominated solution set*, each individual solution is treated as equally important when comparing to others, i.e., solutions that belong to the same *Pareto Front* have equal fitness values even the associated objective function values can be different. Back to our shopper's examples, our shopper prefers item A than B because the fitness of item A is better than that of item B. Note that item A is associated with consistently smaller objective function values as compared to that with item B, i.e., item A is located on a *Pareto Front* that is closer to the *Global Pareto Optima* than item B. Furthermore, all five items $A,\ \ A',\ \ A'',\ \ A'''$, and A'''' are referred to the nondominated solutions, and are located on the same *Pareto Front* as shown in Fig. 5. The quality of our shopper gets from a single selection from items $A,\ \ A',\ \ A'',\ \ A'''$, and A'''', and his/her remaining budget are the two objective functions to be optimized.

Theoretically, any single-objective optimization algorithm can be extended to solving multiobjective optimization if the fitness of population and updating rules are properly defined. For example, the GAs for multiobjective optimization (Deb 2001; Deb et al. 2000, 2002), the multiobjective PSO (Coello et al. 2004; Coello and Lechuga 2002), the multiobjective ACO (Alaya et al. 2007; Angus and Woodward 2009; Doerner et al. 2004; Gao et al. 2013), the multiobjective SA (Bandyopadhyay et al. 2008; Czyzżak and Jaszkiewicz 1998; Serafini 1994; Suman 2004; Suppapitnarm et al. 2000), and different versions of SCE-UA algorithm for multiobjective optimization (Yang et al. 2015; Yapo et al. 1998) are all well developed and successful transformations from single-objective optimization algorithm to multiobjective searching schemes.

All kinds of multiobjective optimization schemes are very useful in real-world applications, by which the trade-offs among completing objectives can be analyzed and investigated. One of the commonly used multiobjective optimization algorithms is called the multiobjective complex evolution – University of Arizona (MOCOM-UA) global optimization method (Yapo et al. 1998). The MOCOM-UA method is one of the successors of the SCE-UA algorithm with a general purpose for solving global multiobjective optimization problems. The uses of MOCOM and other multiobjective optimization algorithms in the field of automatic hydrological model parameter calibration, and trade-off analysis are also extensive. Numbers of different variations of multiobjective algorithms and hydrological models have been tested, such as the studies conducted by Gupta et al. (1998), (2003), Madsen (2000, 2003), Tang et al. (2005), Bekele and Nicklow (2007), Hejazi et al. (2008), Moussa and Chahinian (2009), Shafii and Smedt (2009), Zhu et al. (2017), Zhang et al. (2010), Kollat et al. (2012), Sun et al. (2014), Reed et al. (2013), Asadzadeh et al. (2014), Yapo et al. (1998), and Vrugt et al. (2003b).

4 Examples of Hydrological Applications

Different optimization algorithms have their own strengths and limitations. In practical uses, it is suggested that users choose the most proper algorithm that meets the calibration requirements. However, the selection of algorithm could be tedious for some cases. In this section, we briefly introduce the practical uses of three different optimization algorithms and demonstrate the strengths of (1) the SCE-UA (Duan et al. 1992) algorithm for its global convergence; (2) a surrogate modeling scheme (ASMO) (Wang et al. 2014) and its improvements of computational efficiency; and (3) a multiobjective optimization scheme, termed MOSPD (Yang et al. 2015, 2017c), for its effectiveness of producing *Pareto Optimality.*

The case study is carried on a real-world application, the Sacramento Soil Moisture Accounting Model (SAC-SMA) model. The SAC-SMA model is a conceptual rainfall-runoff model that represents the soil column with upper and lower zones of multiple storages (Burnash 1995). It has been used extensively in both research and operational applications for river forecasting by the National Weather Service River Forecast Centers across the United States. According to literature, the SAC-SMA model is also one of the benchmark models in testing the performances of different automatic parameter calibration methods (Duan et al. 1992, 1994; Yapo et al. 1998; Chu et al. 2014). Figure 6 shows the structure of the SAC-SMA model. There are 16 parameters in the SAC-SMA model. We consider only 13 of them as adjustable parameters, whose feasible ranges and descriptions are listed in Table 1. Three parameters RSERV, RIVA, and SIDE are fixed at prespecified values according to Brazil (1988).

The study area is the South Branch Potomac River basin near Springfield, West Virginia, USA. It is one of the 12 experimental watersheds of the Model Parameter Estimation Experiment (MOPEX) (Duan et al. 2006). The total drainage area of the basin (U.S. Geological Survey Station No. 01608500) is about 3810 km^2. Historical precipitation, potential evapotranspiration, and streamflow observations from

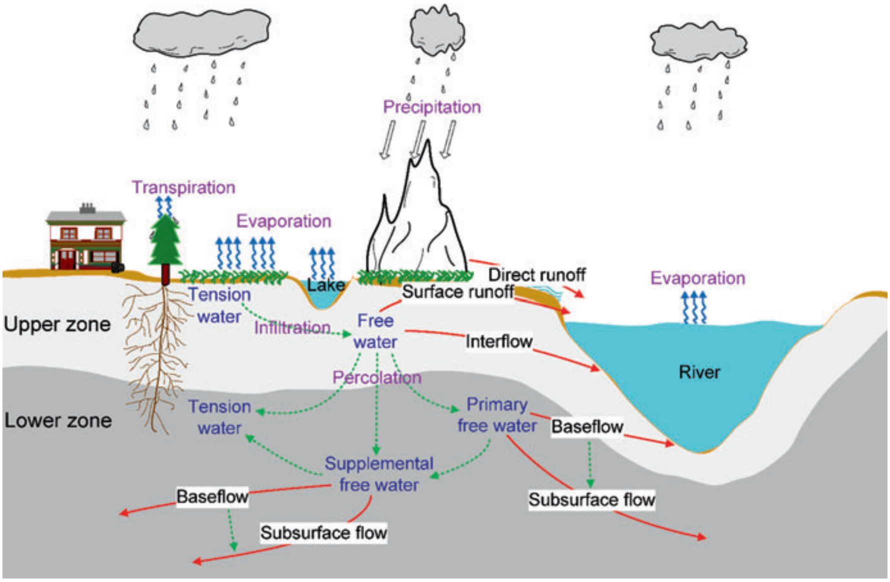

Fig. 6 A schematic of the SAC-SMA model (Source: Gan et al. 2014)

January 1, 1960 to December 31, 1979 were obtained from the MOPEX database for this study. The annual average precipitation over this period is 1021 mm, annual average potential evapotranspiration is 762 mm, and annual average streamflow discharge is 39.5 m^3/s. The hydrological simulations are run with a 6-h time step for each combination of model tunable parameters. Additional physical characteristics of the study area were presented by Duan et al. (2006). The purpose of a model calibration (i.e., parameter optimization) is to find the optimal parameter set for the SAC-SMA model such that the simulated streamflow would have the best overall match with the observed streamflow through minimizing the objective function value. In this study, we used the root mean square error (RMSE) as the objective function, which is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{\mathrm{n}}\left(Q_{s,t} - Q_{o,t}\right)^2}$$

where t is the number of time steps, $Q_{s,t}$ is the simulated flow for time step t, and $Q_{o,t}$ is the observed flow for time step t. In the model simulation process, we used a period of 3 months to remove any impact of uncertain initial conditions, i.e., in computing RMSE, the streamflow values for the first 3 months of 1960 were intentionally removed from consideration in order to warm up the model.

A comparison of calibration performance of SCE-UA algorithm against the multistart downhill simplex (MSDS) algorithms (Duan et al. 1994) is shown in Figs. 7 and 8 for 4 of the 13 parameters in SAC-SMA model.

Table 1 The 13 parameters of the SAC-SMA model and their feasible ranges

No.	Parameter	Description	Range
1	UZTWM	Upper zone tension water maximum storage (mm)	[10, 300]
2	UZFWM	Upper zone free water maximum storage (mm)	[5, 150]
3	UZK	Upper zone free water lateral drainage rate (day^{-1})	[0.1, 0.75]
4	PCTIM	Impervious fraction of the watershed area (decimal fraction)	[0, 0.1]
5	ADIMP	Additional impervious area (decimal fraction)	[0, 0.2]
6	ZPERC	Maximum percolation rate (dimensionless)	[5, 350]
7	REXP	Exponent of the percolation equation (dimensionless)	[1, 5]
8	LZTWM	Lower zone tension water maximum storage (mm)	[10, 500]
9	LZFSM	Lower zone supplemental free water maximum storage (mm)	[5, 400]
10	LZFPM	Lower zone primary free water maximum storage (mm)	[10, 1000]
11	LZSK	Lower zone supplemental free water lateral drainage rate (day^{-1})	[0.01, 0.35]
12	LZPK	Lower zone primary free water lateral drainage rate (day^{-1})	[0.001, 0.05]
13	PFREE	Fraction of water percolating from upper zone directly to lower zone free water (decimal fraction)	[0.0, 0.9]
14	RIVA	Riverside vegetation area (decimal fraction)	0.3
15	SIDE	Ration of deep recharge to channel base flow (dimensionless)	0
16	RSERV	Fraction of lower zone free water not transferrable to lower zone tension water (decimal fraction)	0

As shown in Fig. 7, for 100 independent model runs, the parameters tuned by the MSDS algorithm all fail to converge to a single value at the end of evolution, which indicate the optimal solutions derived by the MSDS algorithm are local optima. On the contrary, all parameters with the SCE-UA algorithm are able to converge to consistent values (Fig. 8), suggesting the detection of global optimum for only ten independent model runs.

The SCE-UA algorithm is famous for its global convergence when the number of evaluation is not limited. However, it generally requires up to tens of thousands of model runs to find the global optimal solution. This may place severe computational constraint on solving such an optimization problem, if the underlying model requires a large amount of CPU time to run. For this situation, the surrogate modeling-based optimization algorithm (e.g., ASMO) is a good choice. Figure 9 shows the comparison between SCE-UA and ASMO applied to the 13 parameters SAC-SMA model. In this study, we set the number of complexes = 4 for SCE-UA and set maximum model evaluations = 200 for ASMO. From Fig. 9, we note that, for SCE-UA, after 2606 model evaluations, the optimization search converges to its optimal solution with an objective function value of 0.92722. For ASMO, the optimized objective function value is 0.92895424449 after 200 model runs. From this case, we conclude that ASMO has an obvious advantage in convergence speed over SCE-UA, with the former needing about 200 total sample points and the latter needing close to 1900

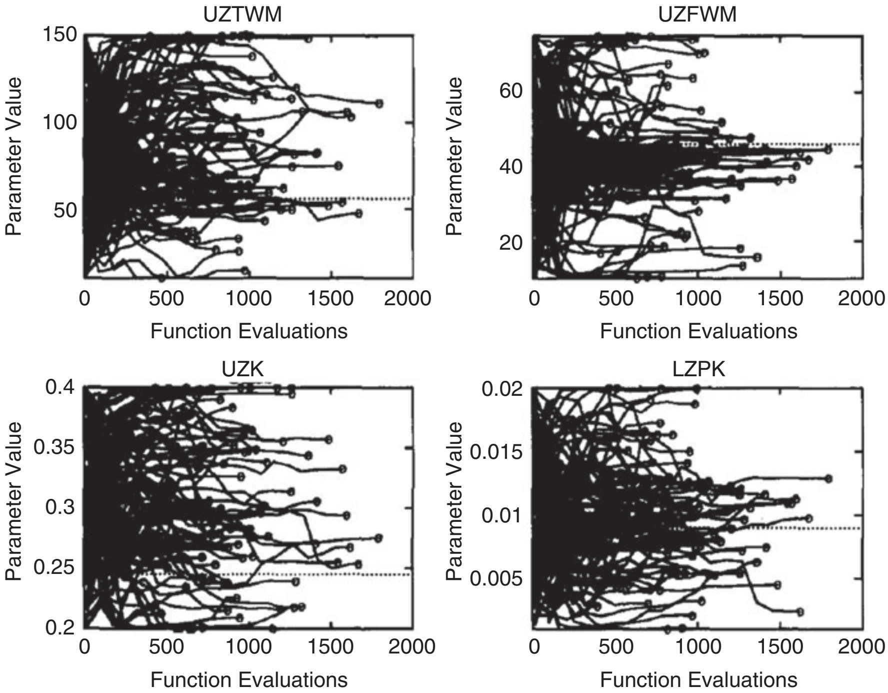

Fig. 7 MSDS search and convergence behavior for four SAC-SMA parameters (100 independent trails). (Source: Duan et al. 1994)

sample points to reach similar objective function value. The SCE-UA method possesses an edge over the surrogate-based optimization in converging to the "true" optimal parameter set if there is no limit on the number of sample points. In other words, the SCE-UA algorithm is capable of finding the exact "true" parameter values, while the ASMO can provide approximate optimization results with relatively small costs. There are also different types of ASMO algorithm available, for instance, the Multiobjective ASMO (Gong et al. 2016), and distribution-based parameter estimation with surrogate model ASMO-PODE (Gong and Duan 2017) for interested readers.

The parameter calibration in a multiobjective context is different from that in the single-objective framework as demonstrated in the previous examples of SCE-UA and AMSO algorithm. In the multiobjective optimization framework, the ultimate goal is to produce *nondominated solutions* that match *Global Pareto Optima* or *Global Pareto Front* in the objective function space. The less differences between the *nondominated solutions* and the solutions located on the *Global Pareto Optima* or *Global Pareto Front,* the better fitness the *nondominated solutions* have. In reality, to define many objective functions, which are completely competing to each other, is a tedious task for a given problem, and the *Global Pareto Front* for most of the cases cannot be explicitly formulated or even does not exist.

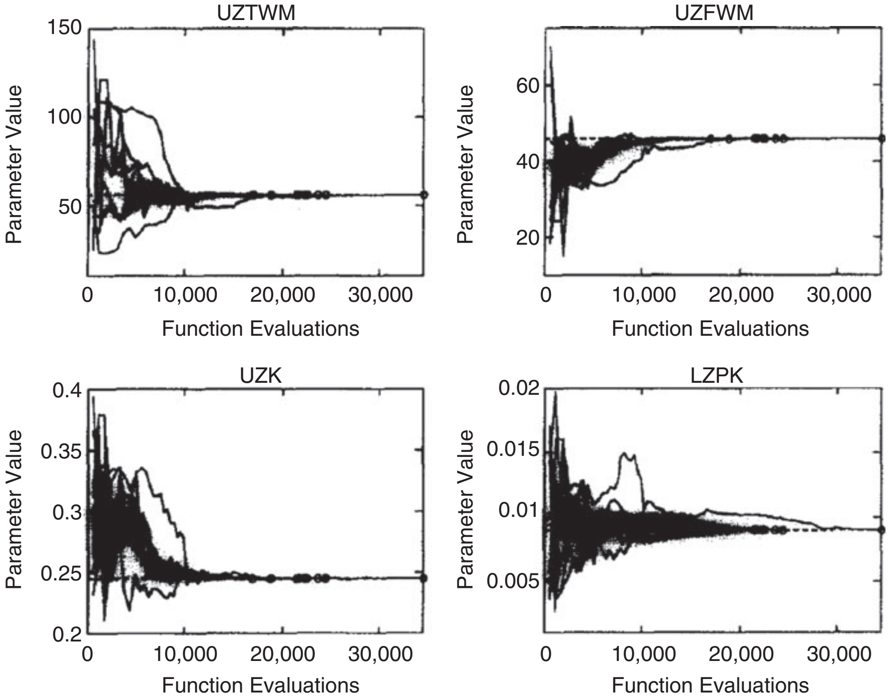

Fig. 8 SCE-UA search and convergence behavior for four SAC-SMA parameters (10 independent trails). (Source: Duan et al. 1994)

Therefore, many human-built, benchmark, conceptual test functions are used in both literature and real-world application to evaluate the performance of multiobjective optimization algorithms. In the following sections, we demonstrate the strength of a Multiobjective Shuffle Complex Evolution with Principal Component Analysis and Crowing Distance (MOSPD) (Yang et al. 2015) on two conceptual, benchmark, composite test functions with known *Global Pareto Front.* The two test functions are KUR function (Kursawe 1991) and ZDT1 function (Zitzler et al. 2000; Zitzler and Thiele 1999), which are both commonly used test functions in the literture. The detailed objective functions, dimentionality, parameter bounds, and characteristics are listed in Table 2. The total number of individuals in population is set to 124 for all simulation. The simulation results (red dots), along with the known *Global Pareto Front* (black line), are shown in Fig. 10a, b. The population in the objective function space during the entire evolution is shown in Fig. 10c, d with different color legends represents the locations of population during the evolution.

In the multiobjective context, a higher number of parameters will not only increase the dimensionality for any single-objective function, but also result in a more complex shape of *Global Pareto Front* in the objective functions' space. As shown in Fig. 10a, b, the final nondominated solutions derived by MOSPD generally match well with the known *Global Pareto Front* for both KUR and ZDT1 functions.

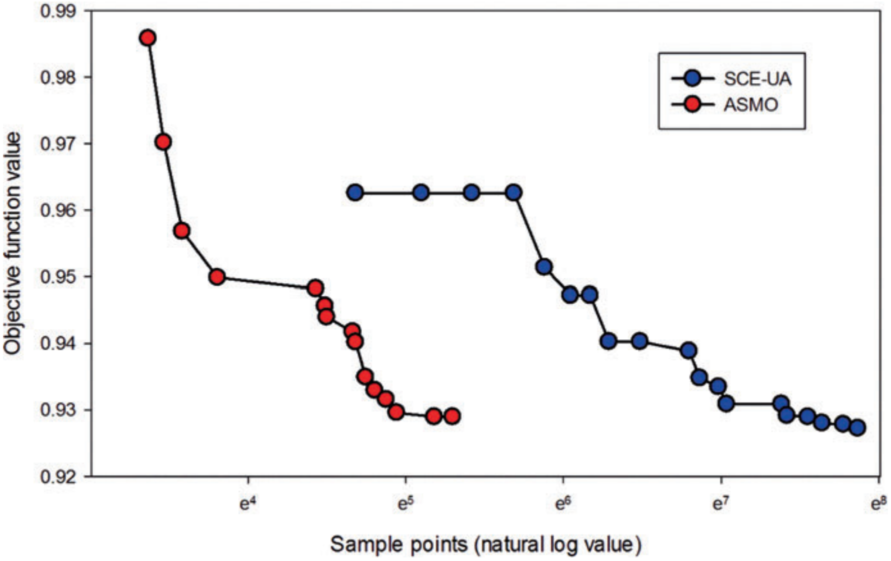

Fig. 9 Comparison of optimization results of SCE-UA and ASMO

Table 2 Details on test functions, including name, number of parameters (N), range, objective functions, optimal solutions, and shapes

Name	N	Variable range	Objective functions	Optimal solutions	Shape
KUR	3	[−5,5]	$f_1(x) = \sum_{i=1}^{n-1}\left(-10\exp\left(\sqrt{x_i^2 + x_{i+1}^2}\right)\right)$	$x_1 \in [0, 1]$ $x_i = 0,$	Convex, disconnected
			$f_2(x) = \sum_{i=1}^{n}\left(\lvert x_i\rvert^{0.8} + 5\sin\left(x_i^3\right)\right)$	i = 2,. . .,n	
ZDT1	30	[0,1]	$f_1(x) = x_1$	$x_1 \in [0, 1]$	Convex
			$f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right]$	$x_i = 0,$	
			$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(\mathrm{n} - 1))$	i = 2,. . .,n	

Different from the previous examples with single-objective function, the population evolution during the entire search (Fig. 10c, d) is gradually toward the *Global Pareto Front*, instead of optimizing any single-objective function value in a consistently decreasing pattern. According to Deb (2001), the evaluation criteria of nondominated solutions for any algorithm have to take two aspects into consideration: (1) the closeness of *nondominated solutions* toward the *Global Pareto Front*, i.e., the convergence of solutions; and (2) the spread of nondominated solutions along the *Global Pareto Front*, i.e., the diversity of solutions that represents the coverage of the *nondominated solutions* on extreme values of objective functions. In the demonstrated examples (Fig. 10a, b), the produced *nondominated solutions* have satisfactory performances with regard to both convergence and diversity on the KUR and ZDT1 functions. More examples of different heuristic multiobjective optimization

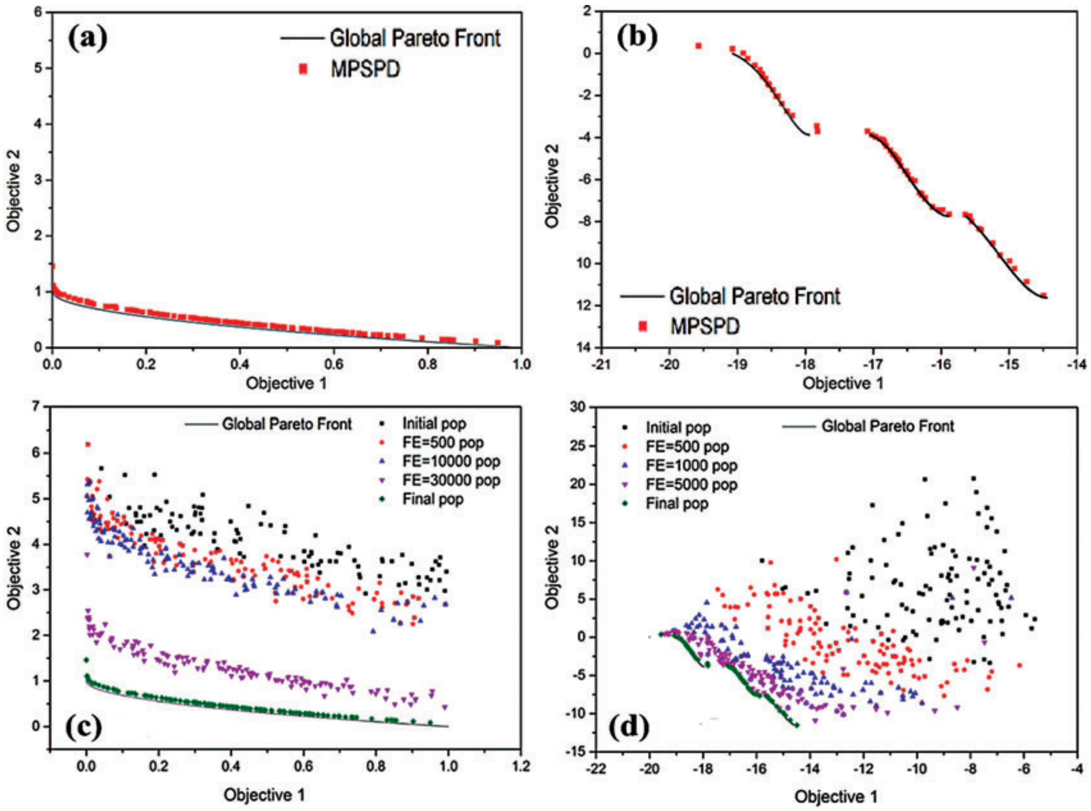

Fig. 10 The final nondominated solutions and Global Pareto Front for (**a**) KUR function and (**b**) ZDT 1 function, and the evolution process of MOSPD on (**c**) KUR function and (**d**) ZDT1 function. Note: the acronym "pop" in the figure legend refers to the number of individuals in a population

algorithms on a large-scale tests are provided in Zitzler et al. (2000), Zitzler and Thiele (1999), Yang et al. (2015), Gong et al. (2016), and Gong and Duan (2017) for intersted readers.

5 Summary and Conclusion

As a summary of this chapter, many types of deterministic optimization algorithms have been developed over the past decades, which have been useful for different types of optimization problems in general. With a limited amount of content of this chapter, authors only review few of the most popular methods in the field of hydrological model calibration, and there exists many other effective algorithms, which may fit better to various real-world optimization problems. It is also a fact that the development of optimization algorithm in the research community is rapid, and the number of new optimization algorithms has been increasing as time goes. However, it is worth mentioning that even the development of optimization algorithm has been prosperous over the years; all algorithms obey the No-Free-Lunch algorithm (Wolpert and Macready 1997). In other words, different algorithms have their own strengths and limitations regarding the efficiency, effectiveness, suitability, etc. to a particular problem. To reduce the computational burden of calibrating a complex model, one of the approaches is to approximate and replace the expensive simulation model with a cheaper-to-run surrogate model. On the other hand, if the computational resource is not a concern by users, many recently developed paralleled computing techniques and hybrid optimization approaches are also promising. Some challenges and future directions with respects to the development and applications of evolutionary algorithms can be found in a recent review paper by Maier et al. (2014).

References

K. Abbaspour, R. Schulin, M.T. Van Genuchten, Estimating unsaturated soil hydraulic parameters using ant colony optimization. Adv. Water Resour. **24**(8), 827–841 (2001)

M.A. Abido, Optimal design of power-system stabilizers using particle swarm optimization. IEEE Trans. Energy Convers. **17**(3), 406–413 (2002)

A. Afshar, F. Massoumi, A. Afshar, M.A. Mariño, State of the art review of ant colony optimization applications in water resource management. Water Resour. Manag. **29**(11), 3891–3904 (2015)

I. Alaya, C. Solnon, K. Ghedira, *Ant Colony Optimization for Multi-objective Optimization Problems* (Citeseer, Patras, 2007), pp. 450–457. https://doi.org/10.1109/ICTAI.2007.108

D. Angus, C. Woodward, Multiple objective ant colony optimisation. Swarm Intell. **3**(1), 69–85 (2009)

R. Arsenault, A. Poulin, P. Côté, F. Brissette, Comparison of stochastic optimization algorithms in hydrological model calibration. J. Hydrol. Eng. **19**(7), 1374–1384 (2013)

R. Arsenault, A. Poulin, P. Côté, F. Brissette, Comparison of stochastic optimization algorithms in hydrological model calibration. J. Hydrol. Eng. **19**(7), 1374–1384 (2014)

M. Asadzadeh, B.A. Tolson, D.H. Burn, A new selection metric for multiobjective hydrologic model calibration. Water Resour. Res. **50**(9), 7082–7099 (2014)

V. Babovic, M. Keijzer, Rainfall runoff modelling based on genetic programming. Hydrol. Res. **33**(5), 331–346 (2002)

C. Balascio, D. Palmeri, H. Gao, Use of a genetic algorithm and multi-objective programming for calibration of a hydrologic model. Trans. ASAE **41**(3), 615 (1998)

S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A simulated annealing-based multiobjective optimization algorithm: AMOSA. IEEE Trans. Evol. Comput. **12**(3), 269–283 (2008)

A. Bárdossy, T. Das, Influence of rainfall observation network on model calibration and application. Hydrol. Earth Syst. Sci. Discuss. **3**(6), 3691–3726 (2006)

B. Bates, Calibration of the SFB model using a simulated annealing approach. Water Down Under 94: Surface Hydrology and Water Resources Papers; Preprints of Papers, 1 (1994)

K. Behzadian, Z. Kapelan, D. Savic, A. Ardeshir, Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks. Environ. Model. Softw. **24**(4), 530–541 (2009)

E.G. Bekele, J.W. Nicklow, Multi-objective automatic calibration of SWAT using NSGA-II. J. Hydrol. **341**(3), 165–176 (2007)

R.W. Blanning, Construction and implementation of metamodels. Simulation **24**(6), 177–184 (1975)

G. Bowden, G. Dandy, H. Maier, Ant colony optimisation of a general regression neural network for forecasting water quality, in *Hydroinformatics 2002: Proceedings of the FIFTH INTERNATIONAL Conference on Hydroinformatics*, ed. by R.A. Falconer et al., Cardiff (IWA Publishing, 2002), pp. 692–698

L.E. Brazil, *Multilevel Calibration Strategy for Complex Hydrologic Simulation Models* (Colorado State University, Fort Collins, 1988)

L. Breiman, Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A.1010933404324

L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees* (Wadsworth, Belmone, 1984)

R.J.C. Burnash, The NWS river forecast system: Catchment modeling, in *Computer Models of Watershed Hydrology*, ed. by V.P. Singh (Water Resources Publications, Highlands Ranch, 1995), pp. 311–366

V. Černý, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. J. Optim. Theory Appl. **45**(1), 41–51 (1985)

K. Chau, A split-step particle swarm optimization algorithm in river stage forecasting. J. Hydrol. **346**(3), 131–135 (2007)

K. Chau, Application of a particle swarm optimization algorithm to hydrological problems, in *Water Resources Research Progress*, (Nova Science Publishers, New York, 2008), pp. 3–12

C.-T. Cheng, M.-Y. Zhao, K. Chau, X.-Y. Wu, Using genetic algorithm and TOPSIS for Xinanjiang model calibration with a single procedure. J. Hydrol. **316**(1), 129–140 (2006)

C.L. Chiu, J. Huang, Nonlinear time varying model of rainfall-runoff relation. Water Resour. Res. **6**(5), 1277–1286 (1970)

W. Chu, X. Gao, S. Sorooshian, Improving the shuffled complex evolution scheme for optimization of complex nonlinear hydrological systems: Application to the calibration of the Sacramento soil-moisture accounting model. Water Resour. Res. **46**(9), W09530 (2010)

W. Chu, X. Gao, S. Sorooshian, A new evolutionary search strategy for global optimization of high-dimensional problems. Inf. Sci. **181**(22), 4909–4927 (2011)

W. Chu, T. Yang, X. Gao, Comment on "High-dimensional posterior exploration of hydrologic models using multiple-try DREAM (ZS) and high-performance computing" by Eric Laloy and Jasper A. Vrugt. Water Resour. Res. **50**(3), 2775–2780 (2014)

C.C. Coello, M.S. Lechuga, *MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization* (IEEE, Honolulu, 2002), pp. 1051–1056

C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 256–279 (2004)

P. Czyzżak, A. Jaszkiewicz, Pareto simulated annealing – A metaheuristic technique for multiple-objective combinatorial optimization. J. Multi-Criteria Decis. Anal. **7**(1), 34–47 (1998)

K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms* (Wiley, Chichester, 2001)

K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II* (Springer, Berlin, 2000), pp. 849–858

K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)

J.-L. Deneubourg, J.M. Pasteels, J.-C. Verhaeghe, Probabilistic behaviour in ants: A strategy of errors? J. Theor. Biol. **105**(2), 259–271 (1983)

J.-L. Deneubourg, S. Aron, S. Goss, J.M. Pasteels, The self-organizing exploratory pattern of the argentine ant. J. Insect Behav. **3**(2), 159–168 (1990)

K. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Ann. Oper. Res. **131**(1–4), 79–99 (2004)

M. Dorigo, Optimization, learning and natural algorithms. Ph.D. Thesis, Politecnico di Milano (in Italian) 1992

M. Dorigo, C. Blum, Ant colony optimization theory: A survey. Theor. Comput. Sci. **344**(2), 243–278 (2005)

M. Dorigo, T. Stützle, *Ant Colony Optimization: Overview and Recent Advances*. Techreport, IRIDIA, Universite Libre de Bruxelles (2009)

M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B Cybern. **26**(1), 29–41 (1996)

M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2006)

Q. Duan, S. Sorooshian, H.V. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour. Res. **28**, 1015 (1992)

Q. Duan, S. Sorooshian, V.K. Gupta, Optimal use of the SCE-UA global optimization method for calibrating watershed models. J. Hydrol. **158**, 265 (1994)

Q. Duan, J. Schaake, V. Andreassian, S. Franks, G. Goteti, H.V. Gupta, Y.M. Gusev, F. Habets, A. Hall, L. Hay, T. Hogue, M. Huang, G. Leavesley, X. Liang, O.N. Nasonova, J. Noilhan, L. Oudin, S. Sorooshian, T. Wagener, E.F. Wood, Model parameter estimation experiment (MOPEX): An overview of science strategy and major results from the second and third workshops. J. Hydrol. **320**(1–2), 3–17 (2006)

R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in *Micro Machine and Human Science, 1995, MHS '95. Proceedings of the Sixth International Symposium on*, Nagoya, 4–6 October 1995 (IEEE, New York, 1995), pp. 39–43. https://doi.org/10.1109/MHS.1995.494215

R. Eglese, Simulated annealing: A tool for operational research. Eur. J. Oper. Res. **46**(3), 271–281 (1990)

C. Fen, C. Chan, H. Cheng, Assessing a response surface-based optimization approach for soil vapor extraction system design. J. Water Resour. Plann. Manag. **135**(3), 198–207 (2009)

F. Francés, J.I. Vélez, J.J. Vélez, Split-parameter structure for the automatic calibration of distributed hydrological models. J. Hydrol. **332**(1), 226–240 (2007)

M. Franchini, Use of a genetic algorithm combined with a local search method for the automatic calibration of conceptual rainfall-runoff models. Hydrol. Sci. J. **41**(1), 21–39 (1996)

M. Franchini, G. Galeati, Comparing several genetic algorithm schemes for the calibration of conceptual rainfall-runoff models. Hydrol. Sci. J. **42**(3), 357–379 (1997)

J. Friedman, Multivariate adaptive regression splines. Ann. Stat. **19**(1), 1–67 (1991)

T.Y. Gan, G.F. Biftu, Automatic calibration of conceptual rainfall-runoff models: Optimization algorithms, catchment conditions, and model structure. Water Resour. Res. **32**(12), 3513–3524 (1996)

Y. Gan, Q. Duan, W. Gong, C. Tong, Y. Sun, W. Chu, A. Ye, C. Miao, Z. Di, A comprehensive evaluation of various sensitivity analysis methods: A case study with a hydrological model. Environ. Model. Softw. **51**, 269–285 (2014)

Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. J. Comput. Syst. Sci. **79**(8), 1230–1242 (2013)

M.K. Gill, Y.H. Kaheil, A. Khalil, M. McKee, L. Bastidas, Multiobjective particle swarm optimization for parameter estimation in hydrology. Water Resour. Res. **42**(7), 417–431 (2006). https://doi.org/10.1029/2005WR004528

D.E. Golberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addion Wesley, Estados Unidos, 1989), p. 102

W. Gong, Q. Duan, An adaptive surrogate modeling-based sampling strategy for parameter optimization and distribution estimation (ASMO-PODE). Environ. Model Softw. **95**, 61–75 (2017)

W. Gong, Q. Duan, J. Li, C. Wang, Z. Di, A. Ye, C. Miao, Y. Dai, Multiobjective adaptive surrogate modeling-based optimization for parameter estimation of large, complex geophysical models. Water Resour. Res. **52**(3), 1984–2008 (2016)

V. Granville, M. Krivánek, J.-P. Rasson, Simulated annealing: A proof of convergence. IEEE Trans. Pattern Anal. Mach. Intell. **16**(6), 652–656 (1994)

H.V. Gupta, S. Sorooshian, P.O. Yapo, Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. Water Resour. Res. **34**(4), 751–763 (1998)

H.V. Gupta, S. Sorooshian, T.S. Hogue, D.P. Boyle, Advances in automatic calibration of watershed models, in *Calibration of Watershed Models*, (American Geophysical Union, Washington, DC, 2003), pp. 9–28

W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**(1), 97–109 (1970)

M.I. Hejazi, X. Cai, D.K. Borah, Calibrating a watershed simulation model involving human interference: An application of multi-objective genetic algorithms. J. Hydroinf. **10**(1), 97–111 (2008)

J.H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (University of Michigan Press, Ann Arbor, 1975)

R. Jin, W. Chen, T.W. Simpson, Comparative studies of metamodelling techniques under multiple modeling criteria. Struct. Multidisc. Optim. **23**, 1–13 (2001)

D. Jones, A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. **21**, 345–383 (2001)

D. Jones, M. Schonlau, W. Welch, Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**(4), 455–492 (1998)

B. Kamali, S.J. Mousavi, K.C. Abbaspour, Automatic calibration of HEC-HMS using single-objective and multi-objective PSO algorithms. Hydrol. Process. **27**(26), 4028–4042 (2013)

J. Kennedy, *Encyclopedia of Machine Learning* (Springer, Berlin, 2011), pp. 760–766

J. Kennedy, J.F. Kennedy, R.C. Eberhart, Y. Shi, *Swarm Intelligence* (Morgan Kaufmann, San Francisco, 2001)

B. Khakbaz, B. Imam, K. Hsu, S. Sorooshian, From lumped to distributed via semi-distributed: Calibration strategies for semi-distributed hydrologic models. J. Hydrol. **418**, 61–77 (2012)

S. Kirkpatrick, Optimization by simulated annealing: Quantitative studies. J. Stat. Phys. **34**(5–6), 975–986 (1984)

P.K. Kitanidis, R.L. Bras, Real-time forecasting with a conceptual hydrologic model: 2. Applications and results. Water Resour. Res. **16**(6), 1034–1044 (1980)

J. Kollat, P. Reed, T. Wagener, When are multiobjective calibration trade-offs in hydrologic models meaningful? Water Resour. Res. **48**(3), 520–539 (2012). https://doi.org/10.1029/2011WR011534

V. Kulandaiswamy, C. Subramanian, A nonlinear approach to runoff studies, in *Proceedings of the International Hydrology Symposium*, vol. **1**, (Colorado State University, Fort Collins, 1967), pp. 72–79

D.N. Kumar, M.J. Reddy, Ant colony optimization for multi-purpose reservoir operation. Water Resour. Manag. **20**(6), 879–898 (2006)

C. Kuok, C.P. Chan, Particle swarm optimization for calibrating and optimizing Xinanjiang model parameters. Int. J. Adv. Sci. Appl. **3**, 115 (2012)

F. Kursawe, Parallel Problem Solving from Nature: 1st Workshop, PPSN I Dortmund, FRG, October 1–3, 1990 Proceedings, ed. by H.-P. Schwefel, R. Männer (Springer Berlin Heidelberg, Berlin, 1991), pp. 193–197

G.-F. Lin, C.-M. Wang, A nonlinear rainfall–runoff model embedded with an automated calibration method – Part 2: The automated calibration method. J. Hydrol. **341**(3–4), 196–206 (2007)

S.Y. Liong, T.R. Gautam, S.T. Khu, V. Babovic, M. Keijzer, N. Muttil, Genetic programming: a new paradigm in rainfall runoff modeling. J. Am. Water Resour. Assoc. **38**(3), 705–718 (2002)

X. Liu, T. Yang, K. Hsu, C. Liu, S. Sorooshian, Evaluating the streamflow simulation capability of PERSIANN-CDR daily rainfall products in two river basins on the Tibetan plateau. Hydrol. Earth Syst. Sci. **21**(1), 169 (2017)

H. Lü, T. Hou, R. Horton, Y. Zhu, X. Chen, Y. Jia, W. Wang, X. Fu, The streamflow estimation using the Xinanjiang rainfall runoff model and dual state-parameter estimation method. J. Hydrol. **480**, 102–114 (2013)

R. Ludwig, I. May, R. Turcotte, L. Vescovi, M. Braun, J.-F. Cyr, L.-G. Fortin, D. Chaumont, S. Biner, I. Chartier, The role of hydrological model complexity and uncertainty in climate change impact assessment. Adv. Geosci. **21**, 63–71 (2009)

S. Madadgar, A. Afshar, An improved continuous ant algorithm for optimization of water resources problems. Water Resour. Manag. **23**(10), 2119–2139 (2009)

H. Madsen, Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. J. Hydrol. **235**(3), 276–288 (2000)

H. Madsen, Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. Adv. Water Resour. **26**(2), 205–216 (2003)

H. Madsen, G. Wilson, H.C. Ammentorp, Comparison of different automated strategies for calibration of rainfall-runoff models. J. Hydrol. **261**(1), 48–59 (2002)

H.R. Maier, A.R. Simpson, A.C. Zecchin, W.K. Foong, K.Y. Phang, H.Y. Seah, C.L. Tan, Ant colony optimization for design of water distribution systems. J. Water Resour. Plan. Manag. **129**(3), 200–209 (2003)

H.R. Maier, Z. Kapelan, J. Kasprzyk, J. Kollat, L.S. Matott, M. Cunha, G.C. Dandy, M.S. Gibbs, E. Keedwell, A. Marchi, Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. Environ. Model Softw. **62**, 271–299 (2014)

R. Moussa, N. Chahinian, Comparison of different multi-objective calibration criteria using a conceptual rainfall-runoff model of flood events. Hydrol. Earth Syst. Sci. **13**(4), 519–535 (2009)

J.A. Nelder, R. Mead, A simplex method for function minimization. Comput. J. **7**(4), 308–313 (1965)

V. Nourani, S. Talatahari, P. Monadjemi, S. Shahradfar, Application of ant colony optimization to optimal design of open channels. J. Hydraul. Res. **47**(5), 656–665 (2009)

A. O'Hagan, Bayesian analysis of computer code outputs: a tutorial. Reliab. Eng. Syst. Saf. **91**(10–11), 1290–1300 (2006)

R.E. Olarte, N. Obregon, Comparison between a simple GA and an ant system for the calibraton of a rainfall-runoff model, in *6th International Conference on Hydroinformatics* (in **2** volumes, with CD-ROM) (World Scientific Publishing Company, Singapore, 2004), pp. 842–849, ISBN 981-238-787-0

A. Ostfeld, Ant colony optimization for water resources systems analysis–Review and challenges, in *Ant Colony Optimization Methods and Applications* (Technion Israel Institute of Technology, Israel, 2011), p. 147

M.A. Panduro, C.A. Brizuela, L.I. Balderas, D.A. Acosta, A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. Prog, Electromagn. Res. B **13**, 171–186 (2009)

D. Pilgrim, Travel times and nonlinearity of flood runoff from tracer measurements on a small watershed. Water Resour. Res. **12**(3), 487–496 (1976)

J. Pintér, Continuous global optimization software: A brief review. Optima **52**(1–8), 270 (1996)

N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Kevin Tucker, Surrogate-based analysis and optimization. Prog. Aerosp. Sci. **41**(1), 1–28 (2005)

C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA, 2006)

S. Razavi, B.A. Tolson, D.H. Burn, Review of surrogate modeling in water resources. Water Resour. Res. **48**(7), 401–433 (2012). https://doi.org/10.1029/2011WR011527

P.M. Reed, D. Hadka, J.D. Herman, J.R. Kasprzyk, J.B. Kollat, Evolutionary multiobjective optimization in water resources: The past, present, and future. Adv. Water Resour. **51**, 438–456 (2013)

R.G. Regis, C.A. Shoemaker, A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J. Comput. **19**, 497–509 (2007)

D.A. Savic, G.A. Walters, J.W. Davidson, A genetic programming approach to rainfall-runoff modelling. Water Resour. Manag. **13**(3), 219–231 (1999)

P. Serafini, *Multiple Criteria Decision Making* (Springer, Berlin, 1994), pp. 283–292

M. Shafii, F.D. Smedt, Multi-objective calibration of a distributed hydrological model (WetSpa) using a genetic algorithm. Hydrol. Earth Syst. Sci. **13**(11), 2137–2149 (2009)

Y. Shi, *Particle Swarm Optimization: Developments, Applications and Resources* (IEEE, Seoul, 2001), pp. 81–86. https://doi.org/10.1109/CEC.2001.934374

A.R. Simpson, G.C. Dandy, L.J. Murphy, Genetic algorithms compared to other techniques for pipe optimization. J. Water Resour. Plan. Manag. **120**(4), 423–443 (1994)

T.W. Simpson, J.D. Peplinski, P.N. Koch, J.K. Allen, Metamodels for computer-based engineering design: Survey and recommendations. Eng. Comput. **17**, 129–150 (2001)

K.P. Singh, Nonlinear instantaneous unit hydrograph theory. J. Hydraul. Div. Am. Soc. Civ. Eng. **90**, 313–347 (1964)

V.P. Singh, *Computer Models of Watershed Hydrology* (Water Resources Publications, Englewood, 1995)

B.E. Skahill, J. Doherty, Efficient accommodation of local minima in watershed model calibration. J. Hydrol. **329**(1), 122–139 (2006)

E. Snelson, Flexible and efficient Gaussian process models for machine learning. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London 2007

A. Sóbester, S. Leary, A. Keane, On the design of optimization strategies based on global response surface approximation models. J. Glob. Optim. **33**(1), 31–59 (2005)

K. Socha, M. Dorigo, Ant colony optimization for continuous domains. Eur. J. Oper. Res. **185**(3), 1155–1173 (2008)

S. Sorooshian, Surface water hydrology: On-line estimation. Rev. Geophys. **21**(3), 706–721 (1983)

P. Srivastava, J. Hamlett, P. Robillard, R. Day, Watershed optimization of best management practices using AnnAGNPS and a genetic algorithm. Water Res. Res. **38**(3), 3-1 (2002)

B. Suman, Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. Comput. Chem. Eng. **28**(9), 1849–1871 (2004)

N.R. Sumner, P.M. Fleming, B.C. Bates, Calibration of a modified SFB model for twenty-five Australian catchments using simulated annealing. J. Hydrol. **197**(1), 166–188 (1997)

Q. Sun, D. Kong, C. Miao, Q. Duan, T. Yang, A. Ye, Z. Di, W. Gong, Variations in global temperature and precipitation for the period of 1948 to 2010. Environ. Monit. Assess. **186**(9), 5663–5679 (2014)

A. Suppapitnarm, K. Seffen, G. Parks, P. Clarkson, A simulated annealing algorithm for multiobjective optimization. Eng. Optim. **33**(1), 59–85 (2000)

Y. Tang, P. Reed, T. Wagener, How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? Hydrol. Earth Syst. Sci. Discuss. **2**(6), 2465–2520 (2005)

M. Thyer, G. Kuczera, B.C. Bates, Probabilistic optimization for conceptual rainfall-runoff models: A comparison of the shuffled complex evolution and simulated annealing algorithms. Water Resour. Res. **35**(3), 767–773 (1999)

J.A. Vrugt, H.V. Gupta, W. Bouten, S. Sorooshian, A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. Water Resour. Res. **39**(8), 1201–1213 (2003a). https://doi.org/10.1029/2002WR001642

J.A. Vrugt, H.V. Gupta, L.A. Bastidas, W. Bouten, S. Sorooshian, Effective and efficient algorithm for multiobjective optimization of hydrologic models. Water Resour. Res. **39**(8), 1214–1233 (2003b). https://doi.org/10.1029/2002WR001746

Q. Wang, The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. Water Resour. Res. **27**(9), 2467–2471 (1991)

Q. Wang, Using genetic algorithms to optimise model parameters. Environ. Model Softw. **12**(1), 27–34 (1997)

H. Wang, W. Guo, ACO Optimizing Neural Network for Macroscopic Water Distribution System Modeling (IEEE, Kuala Lumpur, 2010), pp. 367–370. https://doi.org/10.1109/ICICCI.2010.109

Y.C. Wang, P.S. Yu, T.C. Yang, Comparison of genetic algorithms and shuffled complex evolution approach for calibrating distributed rainfall–runoff model. Hydrol. Process. **24**(8), 1015–1026 (2010)

C. Wang, Q.Y. Duan, W. Gong, A.Z. Ye, Z.H. Di, C.Y. Miao, An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. Environ. Model. Softw. **60**, 167–179 (2014)

P.A. Whigham, P.F. Crapper, Time series modelling using genetic programming: An application to rainfall-runoff models. Adv. Genet. Program **3**, 89–104 (1999)

D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)

C.F.J. Wu, M. Hamada, *Experiments: Planning, Analysis, and Optimization*, 2nd edn. (Wiley, New York, 2009)

S.-J. Wu, H.-C. Lien, C.-H. Chang, Calibration of a conceptual rainfall–runoff model using a genetic algorithm integrated with runoff estimation sensitivity to parameters. J. Hydroinf. **14**(2), 497–511 (2012)

J. Yang, P. Reichert, K.C. Abbaspour, J. Xia, H. Yang, Comparing uncertainty analysis techniques for a SWAT application to the Chaohe Basin in China. J. Hydrol. **358**(1–2), 1–23 (2008)

T. Yang, X. Gao, S.L. Sellars, S. Sorooshian, Improving the multi-objective evolutionary optimization algorithm for hydropower reservoir operations in the California Oroville–Thermalito complex. Environ. Model Softw. **69**, 262–279 (2015)

T. Yang, X. Gao, S. Sorooshian, X. Li, Simulating California reservoir operation using the classification and regression-tree algorithm combined with a shuffled cross-validation scheme. Water Resour. Res. **52**(3), 1626–1651 (2016)

T. Yang, A.A. Asanjan, M. Faridzad, N. Hayatbini, X. Gao, S. Sorooshian, An enhanced artificial neural network with a shuffled complex evolutionary global optimization with principal component analysis. Inf. Sci. **418**, 302–316 (2017a)

T. Yang, A.A. Asanjan, E. Welles, X. Gao, S. Sorooshian, X. Liu, Developing reservoir monthly inflow forecasts using artificial intelligence and climate phenomenon information. Water Resour. Res. **53**(4), 2786–2812 (2017b)

T. Yang, Y. Tao, J. Li, Q. Zhu, L. Su, X. He, X. Zhang, Multi-criterion model ensemble of CMIP5 surface air temperature over China. Theor. Appl. Climatol. **132**(3), 1057–1072 (2017c). https://doi.org/10.1007/s00704-017-2143-4

P.O. Yapo, H.V. Gupta, S. Sorooshian, Multi-objective global optimization for hydrologic models. J. Hydrol. **204**(1), 83–97 (1998)

M. Zambrano-Bigiarini, R. Rojas, A model-independent particle swarm optimisation software for model calibration. Environ. Model Softw. **43**, 5–25 (2013)

A.C. Zecchin, H.R. Maier, A.R. Simpson, A. Roberts, M.J. Berrisford, M. Leonard, Max-min ant system applied to water distribution system optimization. Proc. Int. Congr. Model. Simul. (MODSIM) **2**, 795–800 (2003)

A.C. Zecchin, A.R. Simpson, H.R. Maier, A. Marchi, J.B. Nixon, Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem. Water Resour. Res. **48**(9), 795–800 (2012)

Y. Zhang, F.H.S. Chiew, Relative merits of different methods for runoff predictions in ungauged catchments. Water Res. Res. **45**(7), 412–425 (2009). https://doi.org/10.1029/2008WR007504

X. Zhang, R. Srinivasan, M. Van Liew, Approximating SWAT model using artificial neural network and support vector machine. J. Am. Water Resour. Assoc. **45**(2), 460–474 (2009a)

X. Zhang, R. Srinivasan, D. Bosch, Calibration and uncertainty analysis of the SWAT model using genetic algorithms and Bayesian model averaging. J. Hydrol. **374**(3), 307–317 (2009b)

X. Zhang, R. Srinivasan, K. Zhao, M.V. Liew, Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. Hydrol. Process. **23**(3), 430–441 (2009c)

X. Zhang, R. Srinivasan, M.V. Liew, On the use of multi-algorithm, genetically adaptive multi-objective method for multi-site calibration of the SWAT model. Hydrol. Process. **24**(8), 955–969 (2010)

Q. Zhu, K.I. Hsu, Y.P. Xu, T. Yang, Evaluation of a new satellite-based precipitation data set for climate studies in the Xiang River basin, southern China. Int. J. Climatol. **37**, 4561 (2017)

E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput. **8**(2), 173–195 (2000)