Balancing exploitation and exploration: A novel hybrid global-local optimization strategy for hydrological model calibration

Giuseppe Brunetti a,*, Christine Stumpp a, Jiří Šimůnek b

a University of Natural Resources and Life Sciences, Vienna (BOKU), Institute for Soil Physics and Rural Water Management, Muthgasse 18, 1180, Vienna, Austria
b University of California, Riverside, Department of Environmental Sciences, CA, 92521, USA

ARTICLE INFO

Keywords:
Optimization
Model calibration
HYDRUS
Environmental modeling

ABSTRACT

Optimization problems in hydrological modeling are frequently solved using local or global search strategies, which either maximize exploitation or exploration. Thus, the elevated performance of one strategy for one class of problems is often offset by poor performance for another class. To overcome this issue, we propose a hybrid strategy, G-CLPSO, that combines the global search characteristics of the Comprehensive Learning Particle Swarm Optimization (CLPSO) with the exploitation capability of the Marquardt-Levenberg (ML) method and implement it into the hydrological model, HYDRUS. Benchmarks involving optimizing non-separable unimodal and multimodal functions demonstrate that G-CLPSO outperforms CLPSO in terms of accuracy and convergence. Synthetic modeling scenarios involving the inverse estimation of soil hydraulic properties are used to compare the G-CLPSO against the original HYDRUS ML solver, the gradient-based algorithm PEST, and the stochastic SCE-UA strategy. Results demonstrate the superior performance of the G-CLPSO, suggesting a potential use in other environmental problems.

1. Introduction

Due to the poor transferability of laboratory-estimated parameters (e.g., Ritter et al., 2003), hydrological models are generally calibrated against transient field measurements for predictive purposes. The main goal is to inversely estimate model parameters by minimizing the deviations between model predictions and observations (i.e., the objective function). The same strategy can also be used for designing purposes (e.g., irrigation system optimization (Roy et al., 2019)). These optimization problems are frequently dealt with using local or global search strategies, which either maximize exploitation or exploration. Thus, the elevated performance of one strategy for one class of problems is often offset by poor performance for another class.

A classic example is the hydrological model HYDRUS (Šimůnek et al., 2016), one of the most widespread mechanistic models for analyzing water flow and transport processes in the vadose zone. HYDRUS includes the Marquardt-Levenberg (ML) algorithm (Levenberg, 1944; Marquardt, 1963) to solve a nonlinear least-squares minimization problem arising from the model calibration (Šimůnek and Hopmans, 2002). The ML is a gradient-based local search method, which has been used extensively in the past in soil physics and vadose zone hydrology (e.g., Kool and Parker, 1988; Šimůnek et al., 1998; Šimůnek and Van Genuchten, 1996; van Dam et al., 1994) due to its efficiency for well-posed low-dimensional unimodal inverse problems. However, calibration problems are often ill-posed and non-separable due to measurement errors and model inadequacy, thus leading to response surfaces characterized by long banana-shaped narrow valleys or multiple local optima (e.g., Abbaspour et al., 2004; Arora et al., 2012; Brunetti et al., 2016; Šimůnek and Van Genuchten, 1996). In these conditions, the ML algorithm becomes extremely sensitive to the initial starting point and can easily converge to a local minimum (Transtrum and Sethna, 2012). To deal with irregularities in the response surface in high dimensions, the PEST software (White et al., 2020) couples an ML solver with a subspace parameter dimensionality reduction based on the singular value decomposition. The resulting optimization strategy generated significant computational gains and was successfully applied in combination with HYDRUS (Watson et al., 2013). Nevertheless, its local search behavior can still lead to a modest performance in complex optimization tasks (e.g., Sedaghatdoost et al., 2019).

To overcome this limitation, multiple studies have externally coupled HYDRUS with global optimization strategies such as the annealing-simplex method (Pan and Wu, 1998), genetic algorithms (Ines...
ant colony optimization (Abbaspour et al., 2001), shuffled complex methods (Duan et al., 1993; Vrugt et al., 2003), and particle swarm optimization (Brunetti et al., 2016, 2018), among many others. Global optimization algorithms outperform local search methods for high-dimensional complex problems due to their high exploration capability. However, according to the so-called “no free lunch” theorem (Wolpert and Macready, 1997): “For any algorithm, any elevated performance over one class of problems is offset by poor performance over another class.” Thus, the high performance of global search strategies on complex multidimensional problems is counterbalanced by their slow convergence on unimodal problems. Nevertheless, since the shape of the response surface (i.e., objective function) is not known a priori when solving real-world problems, it is preferable to apply global optimization methods.

Particle Swarm Optimizers (PSOs) are a promising tool due to their ease of implementation and good performance on moderately high-dimensional inverse problems. To improve their efficiency, multiple studies have coupled them with local search methods (e.g., Cao et al., 2019; Han and Liu, 2015; Noel, 2012; Zhao et al., 2009). For instance, in Noel (2012), the steepest descent scheme is used at every iteration or periodically to survey the vicinity around the swarm’s best solution (exploitation) to complement the PSO’s global exploration (Shi and Eberhart, 1998). However, if the local search is performed too early, the swarm can lose diversity and converge to a local optimum. Recently, Cao et al. (2019) coupled the Comprehensive Learning Particle Swarm Optimization (CLPSO) (Liang et al., 2006) with both the Nelder-Mead and Broyden–Fletcher–Goldfarb–Shannon algorithms and proposed to start the local search only when all particles in the CLPSO enter the global optimum basin. By doing so, both the convergence time and accuracy of the CLPSO improve. However, the limited interaction between the CLPSO and the local search during the optimization process suggests that there is still room for improvement, especially for unimodal problems.

Starting from this background, the main goal of this study is to develop an optimization strategy that combines the exploitation capability of the ML algorithm with the exploration features of the CLPSO, which was shown to outperform other algorithms for multidimensional problems (Liang et al., 2006). We aim to obtain an algorithm that is flexible enough to be applied to multiple optimization tasks without deteriorating its performance. The developed approach is implemented in the HYDRUS source code as an inverse solver to extend its applicability to moderately high-dimensional multidimensional optimization problems. The internal coupling overcomes the usability limitations of external coupling routines by making the approach readily available to all HYDRUS users. The problem is addressed in the following way. First, we describe the ML and CLPSO algorithms. Next, we explain their combination in the Gradient-Based Comprehensive Learning Particle Swarm Optimization (G-CLPSO) and its implementation into the HYD- RUS source code. Non-separable and ill-conditioned unimodal and multimodal test functions are then used to compare the original CLPSO and newly developed strategies and assess achieved improvements. Finally, synthetic modeling scenarios involving the inverse estimation of a layered lysimeter’s soil hydraulic properties are used to test the G-CLPSO performance and assess its benefit against the original HYD- RUS ML solver, the gradient-based PEST strategy (White et al., 2020), and the stochastic search algorithm SCE-UA (Duan et al., 1993).

2. Materials and methods

2.1. Optimization strategy

In this section, we first provide a general description of the CLPSO and ML algorithms and then explain their combination in the G-CLPSO, as well as its implementation into the HYD- RUS source code. Considered is the following optimization problem in a bounded parameter space of dimension $D$:

\[
\begin{align*}
\text{minimize} & \quad f(x_1, x_2, \ldots, x_D) \\
\text{subject to} & \quad x_l^i < x_i < x_u^i \quad \text{with} \quad i = 1, 2, \ldots, D
\end{align*}
\]

where $x_l^i$ and $x_u^i$ are the lower and upper bounds of the $i$th parameter, respectively. Since the optimization strategy includes the ML algorithm, the fitness $f$ is the Sum of Squared Residuals (SSQ).

2.1.1. Comprehensive Learning Particle Swarm Optimization

The PSO is a metaheuristic and gradient-free global search strategy based on a social-psychological metaphor involving individuals of a swarm that interact with each other in a social world to achieve an optimum state (Kennedy and Eberhart, 1995; Shi and Eberhart, 1998). By making no assumptions on the problem being optimized, it is especially suited for black-box optimization problems often encountered in environmental sciences (e.g., Brunetti et al., 2018, 2016; Chu and Chang, 2009; Xi et al., 2017).

The swarm consists of $S$ individuals or particles (i.e., potential solutions). Each particle $j$ is characterized by three $D$-dimensional vectors and a scalar value. These are its current position $\mathbf{X}_j$, its best position $\mathbf{X}_j^{\text{best}}$, its velocity $\mathbf{V}_j$, and its best objective function $f_j^{\text{best}}$. The current position is a set of coordinates in the particles’ space, which is updated at each iteration using a specific learning strategy that involves an information exchange with some neighboring individuals or the whole swarm. This communication allows the swarm to progressively locate better solutions and eventually converge to the global optimum.

Intuitively, the core of the PSO is social interaction. In the original PSO, the particle’s position is updated by random averaging of the best position encountered by the particle so far and the best position found by the neighboring particles. However, this trade-off between individual and social interests can reduce the swarm’s diversity, thus leading to premature convergence to a local optimum in complex multimodal problems. The CLPSO (Liang et al., 2006) overcomes this limitation by using a different learning strategy that preserves diversity and reduces swarm behavior. The $j$th particle’s movement in the $i$th dimension is described as follows:

\[
\begin{align*}
X_{ij} & \leftarrow \omega V_{ij} + U_j(0, 1) \cdot (X_{ij}^{\text{best}} - X_{ij}) \\
V_{ij} & \leftarrow \alpha V_{ij} + \beta \cdot X_{ij} - X_{ij}^{\text{best}}
\end{align*}
\]

where $V_{ij}$ and $X_{ij}$ are the velocity and position of the $i$th parameter of the $j$th particle, respectively; $\omega$ is a inertia weight that is reduced as the number of iterations grows to favor exploitation, $U_j$ is a random number in the range $[0, 1]$, $c$ is a learning parameter typically set to 1.4995, and $X_{ij}^{\text{best}}$ defines which particles’ $X_j^{\text{best}}$, the particle $j$ should follow. $X_{ij}^{\text{best}}$ can be the corresponding dimension of any particle’s $X_j^{\text{best}}$ including its own, and the decision depends on probability $P_c$, referred to as the learning probability, which takes different values for different particles:

\[
P_c(j) = 0.05 + 0.45 \frac{\exp \left( \frac{x_j - x_j^{\text{best}}}{m} \right)}{\exp(10)} - 1
\]

For each dimension of the particle $j$, a random number is generated. If this number is lower than $P_c(j)$, the corresponding dimension learns from other particles based on a tournament selection procedure; otherwise, it will learn from its own $X_j^{\text{best}}$. The learning process is continuously monitored. If a particle ceases to improve for a certain number of iterations (i.e., refreshing gap), $m$, the exemplar from which the particle is learning, is reassigned. A thorough description of the selection procedure is reported in Liang et al. (2006).

As problem dimensionality increases, the particles tend to leave the search space and exhibit unwanted roaming behavior. To address this issue, both the particle’s velocity and position should be corrected. In
the CLPSO, the particle’s velocity in each dimension is bounded by a maximum magnitude, $V_{\text{max}}$. If $|V_j|$ exceeds a positive constant value, $V_{\text{max},j}$, then the velocity of that dimension is set to $\text{sign}(V_j)V_{\text{max},j}$. Simultaneously, the fitness value of a particle is calculated and its $\vec{X}_j$ is updated only if its position is within the limits of the search range. This guarantees that the particle will return in the search range in the following iterations.

The main advantage of the CLPSO learning strategy is that all particles can be potentially used to guide the search direction of other individuals, and this learning process can be different for each dimension of the particle. Such reduced emphasis on the global best position preserves the swarm’s diversity and enhances the algorithm’s performance on complex multimodal problems (Liang et al., 2006). However, this high exploration capability delays the convergence for unimodal problems.

### 2.1.2. Marquardt-Levenberg algorithm

The Marquard-Levenberg algorithm is a gradient-based technique that combines the steepest descent and Gauss-Newton methods (Levenberg, 1944; Marquardt, 1963) to solve nonlinear least-squares minimization problems. In the classic steepest descent scheme, the algorithm is started at a selected position in the search space and small steps are made against the direction of the gradient according to:

$$\vec{X}(k+1) = \vec{X}(k) - \beta \nabla C(\vec{X}(k))$$

where $\vec{X}(k+1)$ is the approximation of the local minimum at the iteration $k+1$, $\beta$ is the step size, and $\nabla C(\vec{X}(k))$ is the gradient of the function evaluated at $\vec{X}(k)$. The gradient consists of first-order derivatives, which are calculated using the forward difference approximation. This step can be relatively computationally intensive when the inverse problem’s dimensionality is high and can lead to inaccuracies when the response surface is topologically complex. Large $\beta$ will result in a faster convergence towards the local minimum, yet oscillations will appear once the algorithm reaches the sub-optimal regions as $\vec{X}(k)$ overshoots the minimum. On the other hand, a small step size will improve stability but reduce the algorithm’s convergence rate.

The ML algorithm solves these issues by using the steepest descent method when the objective function is far from its minimum, and the Gauss-Newton method as the minimum is approached. This switch is operated by the parameter $\lambda$, which is set to a modest value at the beginning (e.g., 0.02) and then reduced as the solution approaches the minimum. The algorithm stops when a certain number of iterations, $N_{\text{max}}$, is reached, or no further reductions of the fitness value are obtained. Results are then used to calculate the correlation matrix between estimated parameters, and their confidence intervals, thus providing a statistical basis for uncertainty assessment.

The ML algorithm outperforms stochastic search algorithms for unimodal response surfaces characterized by a single well-defined optimum, or when the initial point lies near the global optimum. In such circumstances, the high exploration ability of stochastic search methods is redundant. However, when the problem’s dimensionality is high, and the response surface is noisy and topologically complex, the ML is likely to converge early to a local optimum (Transtrum and Sethna, 2012).

### 2.1.3. Gradient-based Comprehensive Learning Particle Swarm Optimization

The critical point in hybrid global-local search strategies is to find a trade-off between their conflicting explorative and exploitative interests. One of the main advantages of the CLPSO over other PSOs is the use of selective pressure in the learning strategy. Sutton et al. (2007) noted how the application of selective pressure to the optimization of a non-separable function leads to significant improvements in differential evolution strategies by focusing the search. This was further confirmed by Stanovov et al. (2019), who concluded that rank and tournament selection could significantly enhance the performance of differential evolution strategies by increasing their exploitation capabilities while not affecting exploration significantly. However, one inherent danger with high selective pressure is the rapid loss of diversity within the population. Thus, it is reasonable to assume that the coupling with a local search algorithm will further increase selective pressure, with detrimental effects for multimodal fitness landscapes.

In the Gradient-based Comprehensive Learning Particle Swarm Optimization (G-CLPSO) developed here, we propose a loose coupling strategy between the CLPSO and ML. The CLPSO is used first for $N_1$ iterations. Then, one random individual $\text{rand}$ is selected from the swarm and $\vec{X}_{\text{rand}}$ is used as the starting point for the ML local search. If the calculated fitness value($\vec{X}_{\text{rand}}$), is lower than the corresponding personal best $\vec{X}_{\text{best}}$, then $\vec{X}_{\text{rand}}$ is replaced by the optimum found by the local search, $\vec{X}_{\text{best}}$. By doing so, the new $\vec{X}_{\text{best}}$ can enter the CLPSO tournament selection procedure and improve the swarm without significantly reducing the diversity of the swarm. The main advantage is the possibility to start the ML algorithm after every $N_1$ iterations at different points, with practical implications for different types of problems:

- **Multimodal problems**: This allows to progressively identify different local minima, which will then enter and improve the tournament selection procedure without hampering the swarm’s diversity.
- **Unimodal problems**: The ML will calculate similar fitness values for different starting points, thus driving the entire swarm towards the global optimum in a few iterations.

The algorithm stops when the total number of function evaluations exceeds a user-defined threshold value, $N_{\text{FEs},\text{max}}$, or the difference between the best and worst objective function values is below a user-defined tolerance value, $\tau$. The latter criterion has proven to be reliable for multiple problems (Zielinski and Laur, 2007). The pseudocode of the G-CLPSO is shown in Fig. 1.

### 2.2. Algorithm testing

This section describes the test functions and synthetic case studies used to assess the performance of the G-CLPSO for both widely used benchmark functions and classic vadose zone inverse estimation problems. In particular, test functions are used only to highlight numerical improvements of the G-CLPSO against the original CLPSO and investigate the influence of G-CLPSO input parameters, $S$ and $N_1$, on the algorithm’s performance. Conversely, the synthetic scenarios serve to compare the G-CLPSO, the original HYDRUS inverse solver, the gradient-based PEST optimization strategy (White et al., 2020), and the stochastic search algorithm SCE-UA (Duan et al., 1993) for specific hydrological problems. Algorithms are evaluated in terms of accuracy, which is the capability to approximate better the global optimum, and convergence speed, which is the number of function calls required to identify the optimum or meet the termination criteria.

The use of test functions and synthetic scenarios is an often-used practice in hydrological modeling for testing newly developed algorithms. For instance, Duan et al. (1993) tested the shuffled complex evolution approach against six benchmark functions and one synthetic dataset obtained using the SIXPAR hydrological model. In another study, Duan et al. (1994) used a synthetic sequence of streamflow to test the SCE-UA optimization strategy. Multiple numerical experiments were used by Vrugt and Robinson (2007) and Ter Braak and Vrugt (2008) to assess the performance of the AMALGAM and Differential Evolution Monte Carlo algorithms, respectively. The a priori knowledge of the global optimum and topological features of the response surface guarantees a fair assessment of the proposed optimization strategy.
2.2.1. Test functions

As the G-CLPSO is supposed to outperform the original CLPSO in unimodal problems while preserving its performance for multimodal functions, we selected one unimodal and two multimodal functions from the list of benchmark problems used at the IEEE Congress on Evolutionary Computation (Liang et al., 2014) to test the proposed strategy. The choice of the test functions is not casual but intentionally directed towards the inclusion of mathematical functions whose landscapes closely resemble the response surfaces of real hydrological optimization problems. All functions are non-separable since model parameters often exhibit high interactions in hydrological inverse problems. To this aim, different rotation matrices, \( M \), are assigned to each function, which is also shifted by using a vector \( o \). The search range is \([-100,100]^D\) for all problems, which are tested in ten and thirty dimensions. The maximum number of function evaluations \( N_{\text{FES}} \) and the convergence tolerance, \( \text{tol} \), are set to 10,000D and 0.001, respectively. The properties and formulas of selected functions are listed in Table 1, while their fitness landscapes are shown in Fig. 2.

Twenty-five combinations of \( S \) and \( N_L \) are generated. For each of them, the average final fitness and the total number of function evaluations for each test function are calculated by averaging the results of ten independent G-CLPSO optimizations. Finally, a color map is used to explain the sensitivity of the G-CLPSO performance to changes in the population size and local search frequency. The better combination is identified, and then it is used in the following comparison between the G-CLPSO and the CLPSO.

2.2.2. Implementation of G-CLPSO into the HYDRUS code

The main reason to implement the G-CLPSO directly into the HYDRUS code is to extend the capabilities of the existing HYDRUS inverse algorithm. The pseudocode of the G-CLPSO is shown in Fig. 1.

**Algorithm 1 G-CLPSO**

1. Initialize particles’ position \( \mathbf{X} \), velocity \( \mathbf{V} \), and learning probability \( P_e \).
2. Calculate initial particles’ fitness \( f^\text{best} \) and set \( \mathbf{X}^\text{best} = \mathbf{X} \).
3. Identify the global best \( G = X^\text{best}(f^\text{best}) \).
4. Assign initial learning indices \( f(i) \).
5. Initialize local search counter \( N = 0 \).
6. for iteration = 1, 2, ... do
7. Calculate \( \omega \).
8. for \( j = 1, 2, ..., S \) do
9. if \( f^\text{flag}_j \geq m \) then
10. Reassign learning indices \( f^\text{flag}(i) \).
11. end if
12. for \( i = 1, 2, ..., D \) do
13. \( V_{ij} \leftarrow \omega \cdot V_{ij} + U(0,1) \cdot c \cdot (X^\text{best}_j - X_{ij}) \).
14. if \( |V_{ij}| > V_{\text{max}} \) then
15. \( V_{ij} = \text{sign}(V_{ij}) \cdot V_{\text{max}} \).
16. end if
17. \( X_{ij} \leftarrow X_{ij} + V_{ij} \).
18. end for
19. if \( x^l < X_j < x^u \) then
20. Calculate fitness value \( f(X_j) \).
21. if \( f(X_j) < f^\text{best} \) then
22. \( f^\text{best} = f(X_j) \).
23. \( X^\text{best} = X_j \).
24. \( f^\text{flag}_j = 0 \).
25. else
26. \( f^\text{flag}_j = f^\text{flag}_j + 1 \).
27. end if
28. end if
29. end for
30. if \( N = N_L \) then
31. Select a random particle \( X_{\text{rand}} \).
32. Perform local search with starting point \( X_{\text{rand}} \).
33. Calculate fitness \( f(X_{\text{rand}}) \) and the new position \( X_{\text{new}} \).
34. if \( f(X_{\text{new}}) < f^\text{best} \) then
35. \( f^\text{best} = f(X_{\text{new}}) \).
36. \( X^\text{best} = X_{\text{new}} \).
37. end if
38. \( N = 0 \).
39. else
40. \( N = N + 1 \).
41. end if
42. Update the global best \( G = X^\text{best}(f^\text{best}) \).
43. if Termination criteria are satisfied then
44. break
45. end if
46. end for

Fig. 1. The pseudocode of the G-CLPSO.
Table 1: Formulas and properties of the test functions.

| ID  | Function               | Formula                                                                 | Properties                                                                 |
|-----|------------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------|
| F1  | Rotated Elliptic       | $f_1(x) = \sum_{i=1}^{n} \left(10^i \left(x_i^2 - 1\right)^2\right)$ | - Unimodal, - Non-separable                                               |
|     | Conditioned            |                                                                         | Quadratic ill-conditioned                                                 |
| F2  | Shifted and Rotated    | $f_2(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1}^2 - x_i^2)^2 + (x_i - 1)^2\right]$ | - Multimodal, - Non-separable                                             |
|     | Rosenbrock’s Function  |                                                                         | Having a narrow valley from a local optimum to a global optimum          |
| F3  | Shifted and Rotated    | $f_3(x) = f_1 \left(\frac{\sum_{i=1}^{n} \frac{100}{100} (x_i^2 - M_i^2)}{100}\right)$ | - Multimodal, - Non-separable                                             |
|     | Rastrigin’s Function   |                                                                         | The number of local optima is huge                                        |

The new G-CLPSO algorithm has been implemented in a separate FORTRAN file (GCLPSO.for) to preserve the original HYDRUS code’s structure as much as possible. The file contains only four subroutines: GCLPSO, GRADIENT, TOURNAMENT, and FITNESS. The first two names are self-explanatory, and the subroutines contain the instructions to run the GCLPSO and ML algorithms as described above in Sections 2.1.1 and 2.1.2. TOURNAMENT includes the CLPSO learning strategy, while FITNESS is a subroutine used to call the HYDRUS model and calculate SSQ. The main file is modified to load the G-CLPSO parameters from an external input file (i.e., FIT.in), which are then passed to the GCLPSO subroutine. At this stage of development, an integer, iOptimization, loaded from an external input file, is used to choose between the original HYDRUS inverse solver (i.e., iOptimization = 0) or the new G-CLPSO algorithm (i.e., iOptimization = 1). Modifications of the Graphical User Interface are currently ongoing to facilitate the input. At each G-CLPSO iteration k, the number of HYDRUS calls, the lowest SSQ calculated thus far, and its corresponding best position in the parameter space are saved in an external output file (i.e., GCLPSO.out).

2.2.3. Synthetic modeling scenarios and model settings

We use three synthetic modeling scenarios focused on the inverse estimation of a layered lysimeter’s soil hydraulic properties to test the performance of the G-CLPSO in vadose zone-related problems and assess its benefit against the original HYDRUS ML solver, the gradient-based PEST optimization strategy (White et al., 2020), and the stochastic search algorithm SCE-UA (Duan et al., 1993). The synthetic lysimeter is a widely used numerical experiment for the inverse estimation of soil hydraulic properties (e.g., Durner et al., 2008; Schelle et al., 2012).

A 150 cm deep monolithic lysimeter installed in Gumpenstein (Austria) has been used as a reference (Herndl et al., 2011). The lysimeter consists of three soil layers, classified from top to bottom as loam, silt loam, and silt. The vegetation consists of grass with homogeneous roots distributed to a depth of 20 cm. Climatic data collected by a weather station are used to calculate daily potential evapotranspiration (Hargreaves, 1994). The evapotranspiration demand is then partitioned into potential evaporation and transpiration fluxes using the Leaf Area Index (Sutanto et al., 2012), which is constant during the simulation period and equal to 2.0. Precipitation is measured using a tipping bucket rain collector placed nearby. Data from January to September 2012 are used in the numerical simulations.

HYDRUS-1D simulates variably-saturated water flow and root water uptake in the lysimeter by numerically solving the Richards equation (Richards, 1931) and the Feddes macroscopic model (Feddes et al., 1978), respectively. The Van Genuchten-Mualem functions (VGM) (van Genuchten, 1980) describe the unsaturated soil hydraulic properties of the three layers, and their parameters are initially set according to Carsel and Parrish (Carlsel and Parrish, 1988) to generate synthetic data (Table 2). Field volumetric water content measurements at a pressure head of −15,000 cm are used to adjust the residual water content, $\theta_r$. The domain is discretized into 100 finite elements. An atmospheric boundary condition is applied at the soil surface, while a seepage face boundary condition is set at the bottom.

Warm-up simulations covering a period of 3 months (1st Jan – 31st Mar 2012) are performed to reduce the influence of the initial condition, which is set constant and equal to a pressure head of −100 cm. Water flow from 1st Apr to 30th Sept 2012 is calculated to generate the synthetic data used in the following inverse modeling scenarios. In particular, simulated daily seepage outflow and volumetric water contents at three different depths (i.e., $z = [-10, -45, -130]$ cm) are retrieved and combined in different modeling scenarios.

Forward simulation results are perturbed with a normally distributed measurement error with mean zero and standard deviation $\sigma$. We used $\sigma_\theta = 1.1 \times 10^{-4}$ cm$^3$ cm$^{-2}$ and $\sigma_q = 0.01$ cm for the water content and cumulative outflow, respectively (Schelle et al., 2012). Synthetic perturbed data are then combined in three different modeling scenarios.

![Fig. 2. The fitness landscapes of the three functions described in Table 1.](image-url)
focused on the inverse estimation of the saturated water content and conductivity, $\theta_1$ and $K_s$, respectively, and the VGM shape parameters, $\alpha$ and $n$, for the three soil layers (i.e., a total of 12 soil hydraulic parameters). Scenarios M1 and M2 use separately synthetic seepage outflow and volumetric water contents at three different depths, respectively, while scenario M3 combines the two time series.

### 2.2.4. Algorithms’ settings

The PEST software (White et al., 2020) is based on a Gauss-Levenberg-Marquardt algorithm similar to that implemented in HYDRUS, which calculates the gradient through numerical differentiation. The main differences are the treatment of the parameter $\lambda$ and the PEST use of regularization to restrict the parameter search to identifiable parameters either by adding additional constraints to the parameters (Tikhonov Regularization) or separating identifiable parameters from non-identifiable parameters (Subspace Regularization). In the present study, PEST with subspace regularization based on singular value decomposition is used. Optimization control parameters are set for this simulation, which can slow the calculations but avoids their frequent initialization the ML solver at different points in the curved narrow valley of the Rosenbrock function improves the swarm. The main differences are the treatment of the parameter $\lambda$ and the PEST use of regularization to restrict the parameter search to identifiable parameters either by adding additional constraints to the parameters (Tikhonov Regularization) or separating identifiable parameters from non-identifiable parameters (Subspace Regularization). In the present study, PEST with subspace regularization based on singular value decomposition is used. Optimization control parameters are set for this simulation, which can slow the calculations but avoids their frequent initialization the ML solver at different points in the curved narrow valley of the Rosenbrock function improves the swarm. The main differences are the treatment of the parameter $\lambda$ and the PEST use of regularization to restrict the parameter search to identifiable parameters either by adding additional constraints to the parameters (Tikhonov Regularization) or separating identifiable parameters from non-identifiable parameters (Subspace Regularization). In the present study, PEST with subspace regularization based on singular value decomposition is used. Optimization control parameters are set for this simulation, which can slow the calculations but avoids their frequent initialization the ML solver at different points in the curved narrow valley of the Rosenbrock function improves the swarm.

For G-CLPSO termination, only iterations encompassing a gradient-based search are considered.

### 3. Results and discussion

#### 3.1. G-CLPSO: sensitivity analysis

Fig. 3 shows the mutual influence of the swarm size, $S$, and local search frequency, $N_L$, on the normalized final SSQ and FES (Function Evaluations) for each 10D test function. Results indicate that $N_L$ has a small effect on the final fitness value (SSQ) for both the Elliptic and Rosenbrock functions, while it influences more the outcomes for the Rastrigin problem. The combined effect of $S$ and $N_L$ on the latter is nonlinear, suggesting that specific parameter combinations can yield better results for multimodal problems. However, in general, a more frequent local search ($1 < N_L < 10$) leads to better results for both unimodal and multimodal problems independently from the population size.

This also holds for the total number of function evaluations (FES), though the behavior is more varied among different functions. A frequent local (ML) search significantly accelerates the convergence of the algorithm for the Rosenbrock problem, while its effect is more nuanced for the Elliptic and Rastrigin functions. This is intuitive since frequently initializing the ML solver at different points in the curved narrow valley of the Rosenbrock function improves the swarm’s convergence towards the global optimum. This behavior is less critical but still appreciable for the unimodal elliptic function characterized by a more regular fitness landscape.

These findings are exacerbated when the dimensionality (30D) of the problem increases (Fig. 4). On the one hand, a more frequent local search is again enormously beneficial for both the Elliptic and Rosenbrock functions, which are only minimally affected by the swarm size.

### Table 2

Soil hydraulic parameters and their search bounds in the inverse estimation.

| Parameter | Parameter description | Original parameters | Parameter bounds |
|-----------|-----------------------|---------------------|-----------------|
| Soil Layer 1 (0 < z < -25 cm; Loam) | $\theta_1$ Residual water content | 0.07 | 0.07 |
| | $\theta_1$ [cm$^3$cm$^{-3}$] | 0.44 | (0.4, 0.6) |
| | $r_1$ [cm$^{-1}$] VGM shape parameter | 0.036 | (0.001, 0.1) |
| | $n_1$ [-] VGM shape parameter | 1.56 | (1.1, 2.5) |
| | $K_s$ [cm] Saturated hydraulic permeability | 24.96 | (1, 100) |
| | day$^{-1}$ conductivity | |
| Soil Layer 2 (-25 < z < -55 cm; Silt loam) | $\theta_2$ Residual water content | 0.15 | 0.15 |
| | $\theta_2$ [cm$^3$cm$^{-3}$] | 0.45 | (0.4, 0.6) |
| | $r_2$ [cm$^{-1}$] VGM shape parameter | 0.02 | (0.001, 0.1) |
| | $n_2$ [-] VGM shape parameter | 1.41 | (1.1, 2.5) |
| | $K_s$ [cm] Saturated hydraulic permeability | 10.8 | (1, 100) |
| | day$^{-1}$ conductivity | |
| Soil Layer 3 (-55 < z < -150 cm; Silt) | $\theta_3$ Residual water content | 0.07 | 0.07 |
| | $\theta_3$ [cm$^3$cm$^{-3}$] | 0.46 | (0.4, 0.6) |
| | $r_3$ [cm$^{-1}$] VGM shape parameter | 0.016 | (0.001, 0.1) |
| | $n_3$ [-] VGM shape parameter | 1.37 | (1.1, 2.5) |
| | $K_s$ [cm] Saturated hydraulic permeability | 6.00 | (1, 100) |
| | day$^{-1}$ conductivity | |

---

### Table 3

Termination criteria for G-CLPSO, ML, PEST, and SCE-UA in scenarios M1, M2, and M3.

| Algorithm | ML | PEST | SCE-UA |
|-----------|----|------|--------|
| Maximum number of HYDRUS executions | 10,000 | 10,000 | 10,000 |
| Minimum relative SSQ reduction | minimum | minimum | minimum |
| Maximum number of consecutive iterations for which the minimum SSQ reduction is achieved | 5 | 5 | 5 |
| Maximum number of consecutive iterations for which there is no SSQ reduction | 5 | 5 | 5 |

For G-CLPSO termination, only iterations encompassing a gradient-based search are considered.
On the other hand, the interaction between $S$ and $N_L$ is nonlinear for the multimodal Rastrigin problem.

The main findings of the sensitivity analysis are:

- It is not possible to identify a single combination $S$-$N_L$ that guarantees the best performance for all problems. Conversely, the algorithm consistently exhibits good performance for multiple parameters' combinations for all benchmark functions, thus suggesting the overall robustness of the optimization approach. The use of only two parameters confirms the flexibility of the proposed search strategy.
- The algorithm shows limited sensitivity to the swarm size for low- and moderately high-dimensional unimodal and multimodal problems. Piotrowski et al. (2020) compared the CLPSO with other PSOs and concluded that a swarm of 20 individuals is optimal for the CLPSO. With a larger population, particles’ movement in the CLPSO becomes too chaotic, reducing the convergence to any optima. We do not observe this problem in the G-CLPSO. This may be mainly related to the effects of the ML solver on the search behavior of the particles and the corresponding reduction in their roaming behavior. However, no general conclusions can be drawn since the maximum investigated swarm size was 40.
- The analysis suggests that performing the local search every 1 to 10 G-CLPSO iterations is beneficial in terms of convergence and computational cost for low- and moderately high-dimensional unimodal and multimodal problems. We argue that this is again related to the local search’s magnifying effect on selective pressure in the tournament selection procedure of the CLPSO.

In the following comparison, $S$ and $N_L$ in the G-CLPSO are set to 10 and 1, respectively. This choice is a compromise based on the sensitivity analysis results and is intended to highlight the effect of a very frequent local search on the algorithm performance. The population size in the
CLPSO is set to 20, according to Piotrowski et al. (2020), who carried out an extensive analysis on the effect of the population size in multiple particle swarm optimization strategies.

3.2. Test functions: comparison between G-CLPSO and CLPSO

The convergence characteristics of the G-CLPSO (black) and CLPSO (grey) on the selected test functions in both 10 and 30 dimensions are shown in Fig. 5, which reports the fitness value vs. the number of function evaluations for ten independent optimizations. The mean and the standard deviation of the final fitness value for both competing algorithms are reported in Table 4. Results indicate that the G-CLPSO significantly outperforms the CLPSO in terms of both accuracy and convergence speed for low- and moderately high-dimensional unimodal problems (i.e., $F_1$). The local search quickly focuses the search of the swarm on the global optimum basin and reduces the roaming behavior of the CLPSO. This also happens, although to a lower extent, for the multimodal Rosenbrock problems, where more function evaluations are needed to identify the global minimum. Nevertheless, the accuracy gain of the G-CLPSO against the CLPSO remains significant also for this function (Table 4).

Similar behavior is encountered for the multimodal Rastrigin problem, for which the G-CLPSO identifies a better minimum. Interestingly, the use of the local search does not hinder the exploration of the individuals in the early stage of the optimization, while it focuses their search towards the end when the global optimum basin is likely to be identified. This is a desirable property to avoid premature convergence in local minima. Finally, Table 4 confirms the better performance of the G-CLPSO for all problems, thus indicating good flexibility of the developed strategy over a wide range of potential applications.

3.3. Synthetic modeling scenarios: Comparison between G-CLPSO, ML, PEST, and SCE-UA

The convergence characteristics of the ML (grey), PEST (red), SCE-UA (blue), and G-CLPSO (black) for the three inverse modeling scenarios (M1, M2, and M3) are shown in Fig. 6, which reports the fitness value vs. the number of HYDRUS evaluations for ten independent optimizations. The mean and standard deviation of the final fitness value for all competing algorithms are reported in Table 4. On the first inspection, it is evident that the G-CLPSO outperforms other solvers in terms of accuracy and convergence speed in all scenarios. The local search strategies, ML and PEST, have similar behaviors and exhibit early convergence in local minima due to their sensitivity to the initialization point. Their accuracy is comparable (Table 4), with PEST outperforming ML in scenario M2. Conversely, ML is generally less computationally expensive than PEST, and this can be related to the scaling of the parameter $\lambda$ during the optimization process.

The global stochastic optimization strategy SCE-UA ranks second in terms of accuracy (Table 4) but exhibits slow convergence (Fig. 6).

![Fig. 5. Convergence analysis of the G-CLPSO (black) and CLPSO (grey) for three analyzed functions (Elliptic (left), Rosenbrock (middle), and Rastrigin (right)) for 10D (top) and 30D (bottom) test functions.](image-url)

Table 4

| Function/Scenario | G-CLPSO | CLPSO | ML | PEST | SCE-UA |
|-------------------|---------|-------|----|------|--------|
| Mean SSQ (standard deviation) |
| $F_1$ (10D) | 0 (0) $^{\text{a}}$ | 0.05 | – | – | – |
| $F_2$ (10D) | 0 (0) | 3.28 | – | – | – |
| $F_3$ (10D) | 3.14 (1.27) | 21.02 | – | – | – |
| $F_1$ (30D) | 0 (0) | 2.5e-4 | – | – | – |
| $F_2$ (30D) | 0 (0) | 101 | – | – | – |
| $F_3$ (30D) | 48 (10.15) | 112 (7.58) | – | – | – |
| M1 | 4.5e-3 (0.9e-3) | 0.45 | 0.54 | 0.017 |
| M2 | 3.2e-5 | 0.31 | 0.02 | 5.9e-3 |
| M3 | 3.1e-3 | 0.65 | 0.75 | 0.016 |

$^{\text{a}}$ Values lower than 1e-10 are reported as zero.
Multiple optimization runs were terminated because the algorithm exceeded the maximum number of allowed model executions (i.e., 10,000). The algorithm can consistently identify the global optimum basin, but then a significant number of model executions is required to achieve small accuracy gains. This behavior was already observed by Wöhling et al. (2008), and it is common for global optimization methods, which generally lack exploitation capabilities. Compared to ML and PEST, SCE-UA has a much higher computational cost, hindering its applicability when the model execution time increases (e.g., in reactive solute transport models).

On the other hand, the G-CLPSO always reaches the global optimum region in a reasonable amount of function evaluations. In particular, the number of HYDRUS model executions is, on average, only slightly higher than for ML and PEST and significantly lower than for SCE-UA. This is extremely important since the computational cost is generally a bottleneck in the calibration of environmental models, especially when using global optimization strategies. Table 4 confirms the superior performance of the G-CLPSO, the convergence characteristics of which exhibit a recurrent pattern characterized by early exploration followed by an exploitation phase induced by the local search. After this, minor improvements are achieved, suggesting that the optimal region has been reached and that this region is likely to be flat and similar to the Rosenbrock fitness landscape (Fig. 2). This shape of the objective function is frequent in inverse hydrological problems (Abbaspour et al., 2004; Šimůnek and Van Genuchten, 1996). As Fig. 5 shows, more iterations are needed to identify precisely the global minimum in these cases. More stringent convergence criteria would have further reduced the spread of the final optimized value among different runs. However, this is not particularly important for practical applications since the modeler is more interested in knowing the fitness landscape’s shape near the optimum than a single value characterizing the parameter’s uncertainty. This is automatically done in the G-CLPSO, which calculates the parameters’ confidence interval around the final global optimum.

3.4. Strengths and limitations

We consider this study to be innovative in three main aspects:

- Performance: The analysis demonstrates that the newly developed strategy outperforms both its “parent” algorithms, the CLPSO and ML, and the gradient-based and stochastic search strategies PEST and SCE-UA, respectively, in terms of accuracy and convergence speed.
In particular, it preserves the ML solver’s exploitation and robustness without sacrificing the global search capabilities of the CLPSO. This allows the algorithm to be successfully used in moderately high-dimensional unimodal and multimodal problems frequently encountered in vadose zone hydrology and environmental sciences.

- **Computational cost:** The G-CLPSO does not replace the well-tested and robust ML solver but extends its capabilities without dramatically increasing the run time. This is crucial for several applications, but especially for multi-dimensional problems and problems involving reactive solute transport, for which high computational costs make global search strategies impractical.

- **Ease of implementation and coupling:** The G-CLPSO needs only a few additional parameters (i.e., $S$ and $N_{f}$) compared to the original HYDRUS local search solver. This makes it easy to use and simplifies its implementation into the HYDRUS graphical user interface, which makes the ML algorithm to be favored by HYDRUS users due to its internal coupling into the HYDRUS source code.

Despite these major benefits, there are still some aspects that must be further investigated:

- The synthetic lysimeter experiment is limited and does not represent other inverse modeling scenarios frequently encountered in vadose zone hydrology. Therefore, the algorithm should be further tested on synthetic and real case studies to identify potential issues and then resolve them.

- The effect of the local search on selective pressure in the tournament procedure is empirically hypothesized but not mathematically explained. More numerical tests are needed to fully understand the interaction between the local and global search, particularly how the refreshing gap $m$ of the CLPSO might interact with $N_{f}$.

### 4. Conclusions

This study’s main aim was to extend the capabilities of the HYDRUS inverse solver to moderately high-dimensional multimodal calibration problems by combining its exploitation capability with the global search characteristics of the PSO. To this aim, we developed a new loosely coupled strategy, the G-CLPSO, which couples the CLPSO and ML algorithms. The G-CLPSO has been internally coupled into the HYDRUS source code, facilitating its use and implementation in the HYDRUS graphical user interface. Multiple numerical and synthetic tests confirmed that the hybrid strategy outperforms both the CLPSO and ML on unimodal and multimodal problems by appropriately balancing exploration and exploitation. Furthermore, synthetic modeling scenarios on the inverse estimation of soil hydraulic properties showed G-CLPSO outperforming the original HYDRUS ML solver, the gradient-based strategy PEST, and the stochastic global optimization method SCE-UA in terms of accuracy and convergence speed.

Despite the promising results presented in this study, more research is needed to test the developed optimization strategy further, better understand its structure, and improve its performance. While this is generally accomplished by using a probabilistic framework, the inclusion of an approximate appraisal of the parameters’ uncertainty in the numerical framework is desirable to better characterize the model predictive performance. Nevertheless, this represents an important step toward efficiently embodying global search strategies in vadose zone model calibration. Finally, we would like to emphasize that while we have developed the G-CLPSO strategy for the HYDRUS model, the algorithm can be easily adapted for other hydrological or environmental models.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

Partial financial support from the Austrian Academy of Sciences (ÖAW) is acknowledged for funding the project RechAUT (Variability of Groundwater Recharge and its Implication for Sustainable Land Use in Austria) within the Earth System Sciences research programme. A Python implementation of the developed algorithm is available upon request from the corresponding author.

### References

Abbaspour, K.C., Johnson, C.A., van Genuchten, M.T., 2004. Estimating uncertain flow and transport parameters using a sequential uncertainty fitting procedure. Vadose Zone J. 3, 1340–1352. https://doi.org/10.2113/3.4.1340.

Abbaspour, K.C., Schulin, R., van Genuchten, M.T., 2001. Estimating unsaturated soil hydraulic parameters using an colony optimization. Adv. Water Resour. 24, 827–841. https://doi.org/10.1016/S0309-1708(01)00018-5.

Arora, B., Mohanty, B.P., McGuire, J.T., 2012. Uncertainty in dual permeability model parameters for structured soils. Water Resour. Res. 48 https://doi.org/10.1029/2011WR010500.

Brunetti, G., Porti, M., Piro, P., 2018. Multi-level numerical and statistical analysis of the hygrothermal behavior of a non-vegetated green roof in a mediterranean climate. Appl. Energy 221, 204–219. https://doi.org/10.1016/j.apenergy.2018.03.190.

Brunetti, G., Siminek, J., Piro, P., 2016. A comprehensive numerical analysis of the hydraulic behavior of a permeable pavement. J. Hydrol. 540, 1146–1161. https://doi.org/10.1016/j.jhydrol.2016.07.030.

Cao, Y., Zhang, H., Li, W., Zhou, M., Zhang, Y., Chaolatifalvongse, W.A., 2019. Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. IEEE Trans. Evol. Comput. https://doi.org/10.1109/TEVC.2018.2885075.

Carserl, R.F., Parrish, R.S., 1988. Developing joint probability distributions of soil water retention characteristics. Water Resour. Res. 24, 755–769. https://doi.org/10.1029/WR024i005p00755.

Chu, H.-J., Chang, L.-C., 2009. Applying particle swarm optimization to parameter estimation of the nonlinear Muskingum model. J. Hydrol. Eng. 14, 1024–1027. https://doi.org/10.1061/(ASCE)HE.1943-5584(2009)0.

Duan, Q., Sorooshian, S., Gupta, V.K., 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. J. Hydrol. 158, 265–284. https://doi.org/10.1016/0022-1694(94)90057-4.

Duan, Q.Y., Gupta, V.K., Sorooshian, S., 1993. Shuffled complex evolution approach for effective and efficient global minimization. J. Optim. Theor. Appl. 76, 501–521. https://doi.org/10.1007/BF00935360.

Durner, W., Jansen, U., Iden, S.C., 2008. Effective hydraulic properties of layered soils at the lysimeter scale determined by inverse modelling. Eur. J. Soil Sci. https://doi.org/10.1111/j.1365-2389.2007.00972.x.

Fedden, R.A., Kowalk, P.J., Zaradny, H., 1978. Simulation of Field Water Use and Crop Yield, PUDOC, Wageningen.

Han, F., Liu, Q., 2015. An improved hybrid PSO based on ARPSO and the quasi-Newton method. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-319-20466-4.

Hargreaves, G.H., 1994. Defining and using reference evapotranspiration. J. Irrig. Drain. Eng. 120, 6. https://doi.org/10.1061/(ASCE)0733-9437.1942.

Herming, M., Fösch, E.M., Böhmer, A., Kandolf, M., 2011. Lysimeter als Bestandteil eines technischen Vernuchkonzepzes zur Simulation der Erderwärmung im Grünland. In: 14. Gumpensteinner Lysimetertagung “Lysimeter in Der Klimafolgenforschung Und Bildung”. 126, pp. 119–126.

Ines, A.V.M., Droogers, P., 2002. Inverse modeling in estimating soil hydraulic retention characteristics. Water Resour. Res. 38, 441–456. https://doi.org/10.1029/2001WR000903.

Kool, J.B., Parker, J.C., 1988. Analysis of the inverse problem for transient unsaturated flow. Water Resour. Res. https://doi.org/10.1029/WR024i006p000817.

Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. 10, 281–295. https://doi.org/10.1109/TEVC.2005.857610.

Liang, J.J., Qu, B.Y., Suganthan, P.N., 2014. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. 2014 IEEE Congr. Evol. Comput.

Marquardt, D.W., 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11, 431–441. https://doi.org/10.1137/0111036.

Noel, M.M., 2012. A new gradient based particle swarm optimization algorithm for accurate computation of global minimum. Appl. Soft Comput. J. https://doi.org/10.1016/j.asoc.2011.08.037.
Pan, L., Wu, L., 1998. A hybrid global optimization method for inverse estimation of hydraulic parameters: annealing-simplex method. Water Resour. Res. 34, 2261–2269. https://doi.org/10.1029/98WR01672.

PEST, 2021. PEST - Model-Independent Parameter Estimation User Manual.
Piotrowski, A.P., Napiorkowski, J.J., Piotrowska, A.E., 2020. Population size in particle swarm optimization. Swarm Evol. Comput. 58, 100718. https://doi.org/10.1016/j.swevo.2020.100718.

Python Software Foundation, 2013. Python Language Reference, Version 2.7. Python Software Foundation.

Raj Shrestha, R., Rode, M., 2008. Multi-objective calibration and fuzzy preference selection of a distributed hydrological model. Environ. Model. Software 23, 1384–1395. https://doi.org/10.1016/j.envsoft.2008.04.001.

Richards, L.A., 1931. Capillary conductions of liquids through porous mediums. Physics 1, 318. https://doi.org/10.1103/physrev.1.318.

Ritter, A., Hupet, F., Muı́oz-Carpena, R., Lambot, S., Vanclooster, M., 2003. Using inverse methods for estimating soil hydraulic properties from field data as an alternative to direct methods. Agric. Water Manag. https://doi.org/10.1016/S0378-3774(02)00160-9.

Roy, P.C., Guber, A., Abouali, M., Nejadhashemi, A.P., Deb, K., Smucker, A.J.M., 2019. Estimating unsaturated soil hydraulic functions from multistep outflow experiments. Soil Sci. Soc. Am. J. https://doi.org/10.2136/sssaj2019.03615995005800030002x.

Schielle, H., Iden, S.C., Fank, J., Durner, W., 2012. Inverse Estimation of Soil Hydraulic and Root Distribution Parameters from lysimeter Data. Vadose Zone J. https://doi.org/10.2136/vzj2011.0169.

Sedaghatdoost, A., Ebrahimian, H., Liaghat, A., 2019. An inverse modeling approach to calibrate parameters for a drainage model with two optimization algorithms on homogeneous/Heterogeneous soil. Water Resour. Manag. 33, 1383–1395. http://doi.org/10.1007/s11269-019-2191-x.

Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer. In: 1998 IEEE Int. Conf. Evolutionary Comput. Proceedings. IEEE World Congr. Comput. Intell. (Cat. No.98TH8360) 69–73. https://doi.org/10.1109/ICEC.1998.699146.

Šiminek, J., Hopmans, J.W., 2002. 1.7 parameter optimization and nonlinear fitting. In: Methods of Soil Analysis, Part 1, Physical Methods. pp. 139–157. https://doi.org/10.2136/xasbooksre8.4.7.

Šiminek, J., Van Genuchten, M.T., 1996. Estimating unsaturated soil hydraulic properties from tension disc infiltrometer data by numerical inversion. Water Resour. Res. https://doi.org/10.1029/96WR01525.

Šiminek, J., Van Genuchten, M.T., Sejna, M., 2016. Recent developments and applications of the HYDRUS computer software packages. Vadose Zone J. 15, 25. https://doi.org/10.2136/vzj2016.04.0033.

Šiminek, J., Van Genuchten, M.T., Wendroth, O., 1998. Parameter estimation analysis of the evaporation method for determining soil hydraulic properties. Soil Sci. Soc. Am. J. 62, 894. https://doi.org/10.2136/sssaj1998.0361599500620040894x.

Sorooshian, S., Duan, Q., Gupta, V.K., 1993. Calibration of rainfall-runoff models: application of global optimization to the sacramento soil moisture accounting model. Water Resour. Res. 29, 1185–1194. https://doi.org/10.1029/92WR02617.

Stanovov, V., Akhmedova, S., Sementin, E., 2019. Selective Pressure Strategy in differential evolution: exploitation improvement in solving global optimization problems. Swarm Evol. Comput. https://doi.org/10.1016/j.swevo.2018.10.014.

Sutanto, S.J., Wenninger, J., Coenders-Gerrits, A.M.J., Ulenbrook, S., 2012. Partitioning of evaporation into transpiration, soil evaporation and interception: a comparison between isotope measurements and a HYDRUS-1D model. Hydrol. Earth Syst. Sci. 16, 2605–2616. https://doi.org/10.5194/hess-16-2605-2012.

Sutton, A.M., Lunacek, M., Whitley, L.D., 2007. Differential evolution and non-separability: using selective pressure to focus search. In: Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference. https://doi.org/10.1145/1276958.1277221.

Ter Braak, C.J.F., Vrugt, J.A., 2008. Differential Evolution Markov chain with snooker updater and fewer chains. Stat. Comput. 18, 435–446. https://doi.org/10.1007/s11222-008-9104-9.

Transtrum, M.K., Sethna, J.P., 2012. Improvements to the Levenberg-Marquardt Algorithm for Nonlinear Least-Squares Minimization.

van Dam, J.C., Stricker, J.N.M., Droogers, P., 1994. Inverse method to determine soil hydraulic functions from multistep outflow experiments. Soil Sci. Soc. Am. J. https://doi.org/10.2136/sssaj1994.03615995005800030002x.

van Genuchten, M.T., 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated Soils1. Soil Sci. Soc. Am. J. 44, 892–898. https://doi.org/10.2136/sssaj1980.03615995004400050002x.

Vrugt, J.A., Gupta, H.V., Bouten, W., Sorooshian, S., 2003. A Stiffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. Water Resour. Res. 39 https://doi.org/10.1029/2002WR001642.

Vrugt, J.A., Robinson, B.A., 2007. Improved evolutionary optimization from genetically adaptive multimethod search. Proc. Natl. Acad. Sci. Unit. States Am. 104, 708–711. https://doi.org/10.1073/pnas.0610471104.

Watson, T.A., Doherty, J.E., Christensen, S., 2013. Parameter and predictive outcomes of model simplification. Water Resour. Res. 49, 3952–3977. https://doi.org/10.1002/wrcr.20145.

White, J.T., Hunt, R.J., Fienen, M.N., Doherty, J.E., 2020. Approaches to Highly Parameterized Inversion: PEST ++ Version 5, a Software Suite for Parameter Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis. Techniques and Methods. https://doi.org/10.3133/tm7c26.

Wohling, T., Vrugt, J.A., Barke, G.F., 2008. Comparison of three multiojective optimization algorithms for inverse modeling of vadose zone hydraulic properties. Soil Sci. Soc. Am. J. 72, 305–319. https://doi.org/10.2136/sssaj2007.0176.

Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput. https://doi.org/10.1109/4235.658593.

Xu, M., Lu, D., Gai, D., Qi, Z., Zhang, G., 2017. Calibration of an agricultural-hydrological model (RZWQM2) using surrogate global optimization. J. Hydrol. 544, 456–466. https://doi.org/10.1016/J.JHYDROL.2016.11.051.

Zhao, S.Z., Liang, J.J., Suganthan, P.N., Taretiren, M.F., 2008. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. In: 2008 IEEE Congress on Evolutionary Computation, CEC 2008. https://doi.org/10.1109/CEC.2008.4631280.