Parameter estimation of soil water retention curve with Rao-1 algorithm

Zhongju Wang,[1] Chao Huang,[1] Long Wang[1,2]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing; [2]Shunde Graduate School, University of Science and Technology Beijing, Foshan, China

Abstract

The soil water retention curve (SWRC) has a significant role in determining the unsaturated properties of soil. A stochastic optimisation algorithm named Rao-1 algorithm is employed to estimate the parameters of the SWRC model in this paper. The Rao-1 algorithm is a simple heuristic search algorithm containing only addition and multiplication operations. This paper introduces the method and its application in determining soil water retention this model parameters in detail. In this study, the van Genuchten model is used to depict the SWRC for its good fitting capacity, and the van Genuchten model parameters are determined using Rao-1 algorithm. The feasibility and efficiency of the proposed method are validated via the experimental results of 24 soil samples of 12 soil textural classes. Besides, the performance of Rao-1 algorithm is compared with that of salp swarm algorithm, the particle swarm optimization algorithm, differential evolution algorithm, and RETC program. Through comparative analysis, Rao-1 algorithm outperforms other methods in determining SWRC parameters.

Correspondence: Long Wang, School of Computer and Communication Engineering, University of Science and Technology Beijing, China.
E-mail: lwang@ustb.edu.cn

Key words: Parameter estimation; soil water retention curve; optimization; Rao-1 algorithm.

Funding: this work was supported in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515110431, and in part by the National Nature Science and Foundation of China under Grant 62002016.

Conflict of interest: the authors declare no potential conflict of interest.

Availability of data and materials: available upon request.

Code availability: available upon request.

Received for publication: 25 September 2021.
Accepted for publication: 3 February 2022.

Journal of Agricultural Engineering 2022; LIII:1283
doi:10.4081/jae.2022.1283

Introduction

Soil water retention is an important indicator of the impact of agricultural management on the retention and movement of water (Kassaye *et al*., 2020). The amount of water absorbed and retained by soil depends on the capillary attraction of soil pores and the molecular attraction of soil particles, typically described as soil suction. As a basic property curve reflecting retention and movement of soil water, the soil water retention curve (SWRC), describing the relationship between the soil water potential and the soil water content, tends to be utilised to investigate the hydraulic properties of soil (Kawai *et al*., 2000). Specifically, the conversion between soil water suction and moisture content can be performed by using the SWRC. Furthermore, since the SWRC is highly related to soil physical properties, such as soil texture, structure, and porosity, it can also be used to analyse the distribution of soil pores with different sizes and derive water-retaining capacity of different types of soil. Besides, when applying mathematical and physical methods to measure water movement in soil quantitatively, the SWRC is an indispensable factor (Silva *et al*., 2018). Therefore, accurate estimation of SWRC has a significant role in determining the unsaturated properties of soil (Tan *et al*., 2016). In this context, in order to comprehensively reflect the dynamic process of soil water content changes with soil suction, several methods were discussed in the field of soil physics, generally classified into direct and indirect approaches (Zhai *et al*., 2020). The direct measurement methods balance particular air pressure and water content by using pressure plates, pressure membranes, and tensiometers.

However, such methods are not only time-consuming but also unprecise. Other soil physicists also proposed indirect methods that combined soil hydraulic conductivity, SWRC, and other soil physical properties. Then parameter estimation methods are used to derive soil water movement parameters. Empirical parametric models were proposed to depict the SWRC, such as the Brooks-Corey model (Brooks and Corey, 1964), Gardener model (Gardner *et al*., 1970), Campbell model (Campbell, 1974), Van Genuchten model (Genuchten, 1980), and Rossi-Nimmo model (Rossi and Nimmo, 1994). Among these proposed models, the van Genuchten model is widely applied to depict the SWRC for its good fitting capacity. However, the van Genuchten model is parameterized by some parameters that are to be determined based on the collected data. Therefore, it is meaningful and practical to accurately estimate the parameters of the van Genuchten model for describing the SWRC.

In the literature, various parameter estimation methods were proposed, and they can be generally classified into two types, least-square methods (Duong *et al*., 2015) and heuristic search methods (Wang *et al*., 2018). The least-square method is a mathematical optimisation method used to estimate the actual value of some quantity based on a consideration of errors in measurements (Li, 2018). Therefore, it can easily obtain unknown parameters and minimise the sum of the squares of the errors between the estimated and actual values. However, it requires a lot of time and memory. On the other hand, the RETC program (Talat *et al*., 2020), a typical nonlinear least-square (NLS) method, is widely applied to the hydraulic properties of soil. However, the estimation results of RETC do not guarantee that the parameters of the model are global optimum solutions, which requires the use of prior knowledge to initialise different parameters for different soil types.

On the contrary, the heuristic search method is another technique to search for the global optimum solution faster than the least square methods. It evaluates the available information and decides based on the importance of the information at each branching step. Besides, the heuristic search method is not affected by the initial parameter value compared with the least square method. In this context, the heuristic search methods have a better performance in terms of parameter estimation than the least square methods.

Since heuristic search methods outperform the least square methods in parameter estimation, numerous researches have been conducted to determine the parameters of the SWRC models using heuristic search algorithms. Maggi (Maggi, 2017) applied the different evolution (DE) algorithm to determine the parameters of the Brooks-Corey model, Rossi-Nimmo model, and the van Genuchten model, respectively. The experimental results showed that this method could find the best solution for model parameters with any random choice of the initial population. Li *et al*. (2018) calculated the SWRC model parameters with particle swarm optimization (PSO) algorithm based on the pre-defined objective functions. The method is not limited to the weight determination compared with the traditional multitarget weighted sum optimisation method. Zhang *et al*. (2018) utilised the Salp Swarm algorithm (SSA) to derive the parameters of the van Genuchten model via nine collected soil samples. The calculated results showed that the SSA is suitable for soil samples with small experimental data. The above-mentioned conventional heuristic search algorithms typically include some algorithm-specific parameters such as mutation factor in DE algorithm (Maggi, 2017) and inertia weight in the PSO algorithm (Li *et al*., 2018) which may affect the estimation performance. Furthermore, it takes much time to adjust these parameters. To tackle this issue, Wang *et al*. (2018) estimated the parameters of the SWRC model with the Jaya algorithm. Jaya algorithm (Rao, 2016; Venkata Rao, 2019) is a swarm intelligence algorithm that iteratively updates solutions by moving solutions towards the global best solution without algorithm-specific parameters. However, the updating procedure of the Jaya algorithm contains complex arithmetic operations, especially the absolute value operation, which means that it is not efficient for embedded implementation. Thus, it is meaningful and valuable to develop new heuristic search algorithms for more simply estimating the SWRC model parameters.

In order to reduce the computational overhead and the complexity of estimating the parameters of the SWRC model, a newly proposed heuristic search algorithm, the Rao-1 algorithm (Rao, 2020), is utilised to identify the parameters of the van Genuchten model for describing the SWRC in this paper. Rao first proposed Rao-1 algorithm in 2019 for solving optimisation problems. Unlike conventional heuristic search algorithms, the proposed algorithm does not require any algorithm-specific control parameters. Instead, the Rao-1 algorithm searches for the global optimum solution depending only on the standard control parameters like population size and the number of iterations. Meanwhile, the Rao-1 algorithm only contains addition and multiplication operations. Therefore, it is very suitable for implementation on embedded devices. To illustrate the superiority of the proposed algorithm, the experimental results are compared with those of SSA, PSO, and DE algorithms and the RETC program.

The van Genuchten model and problem formulation

Van Genuchten first developed the van Genuchten model in 1980 (Genuchten, 1980), which is widely used in the soil science domain for its good fitting capacity. Therefore, the van Genuchten model is employed to depict the SWRC in this study. In the van Genuchten model, the relationship between the water content Θ (cm^3/cm^3) and soil water potential $h(cm^3H_20)$ is described as Eq. 1:

$$\Theta = \theta_r + \frac{(\theta_s - \theta_r)}{\left[1+(\alpha h)^n\right]^m} \tag{1}$$

There are four independent parameters in the Eq. 1 (θ_r, θ_s, α, and n), which have to be estimated from the observed soil water retention data. θ_r is the residual water content, θ_s is the saturated water content, α is an experiential parameter, n is the shape parameter, and $n = 1 - 1/n$.

To identify these model parameters, the problem of estimating parameters is transformed into an optimisation problem that is to minimise the difference between the estimated and actual water content described as Eq. 2:

$$\min_{\theta_r,\theta_s,\alpha,n} \sum_{i=1}^{N} \left[\Theta_i - \hat{\Theta}_i(h_i \mid \theta_r,\theta_s,\alpha,n)\right]^2 \tag{2}$$

where Θ is the actual water content, $\hat{\Theta}$ is the estimated water content. Solution to this optimisation problem is four parameters (θ_r, θ_s, α and n), which are to be estimated for describing SWRC, and N is the number of experimental data of each soil sample.

Rao-1 algorithm

Ravipudi Venkata Rao first developed the Rao-1 algorithm in 2019 for solving optimisation problems (Rao, 2020). This algorithm approaches the global optimum solution based on the best and worst solutions obtained during the update process and the random interactions among the candidate solutions. Suppose that the fitness function $f(\boldsymbol{x})$, d-dimensional vector $\boldsymbol{x}$ and the population size n. In the first place, n candidate solutions are initialised. At any iteration i, the candidate solution is updated as Eq. 3.

$$x'_{j,k,i} = x_{j,k,i} + r_{j,i}(x_{j,best,i} - x_{j,worst,i}) \tag{3}$$

where $x_{j,k,i}$ is the value of the j^{th} element of the k^{th} candidate at iteration i, $x_{j,best,i}$, denotes the value of the j^{th} element of the solution with the best fitness value of $f(\boldsymbol{x})$ while $x_{j,worst,i}$, is the value of the j^{th} element of the solution with the worst fitness value of $f(\boldsymbol{x})$ at iteration i. $x'_{j,k,i}$, is the updated result of $x_{j,k,i}$. $r_{j,i}$ is an independent random number generated from a uniform distribution in the range (0:1) at iteration i, which provides this algorithm a stochastic exploration of the search space. Furthermore, the update process means a tendency of the candidate solution movement toward the best solution while a tendency of the candidate solution leaving from the worst solution. Therefore, iteratively updating the global optimum solution can be easily searched. The process of applying the Rao-1 algorithm to estimate parameters of the van Genuchten model is shown in Figure 1.

Besides, all parameters should be constrained by their boundaries, and the following formula describes the constraints:

$$x_{j,k}=\begin{cases}UB_j, & x_{j,k}\geq UB_j\\ LB_j, & x_{j,k}\leq LB_j\\ x_{j,k}, & otherwise\end{cases} \tag{4}$$

where LB_j and UB_j are the lower and upper bounds of the j^{th} element. The lower and upper bounds of four parameters of the van Genuchten model are described in Table 1, and the detailed algorithm is presented in algorithm 1.

Benchmarking algorithms

In this study, the Rao-1 algorithm is employed to estimate the parameters of the van Genuchten model. Besides, to assess the estimation performance of the Rao-1 algorithm, the Rao-1 algorithm is compared with SSA, PSO, DE algorithms, and RETC program.

SSA algorithm (Mirjalili *et al*., 2017) originated from the foraging behaviour of a kind of creature in the ocean. The SSA algorithm approximates the global optimum by initiating multiple salps with random positions, calculating the fitness of each salp, finding the salp with the best fitness, and updating the positions of the salp iteratively. The best position is considered the source of food to be chased by the salp chain. Thus, this algorithm can effectively improve the initial random solutions and converge towards the optimum. The search process of SSA is shown in Eq. 5:

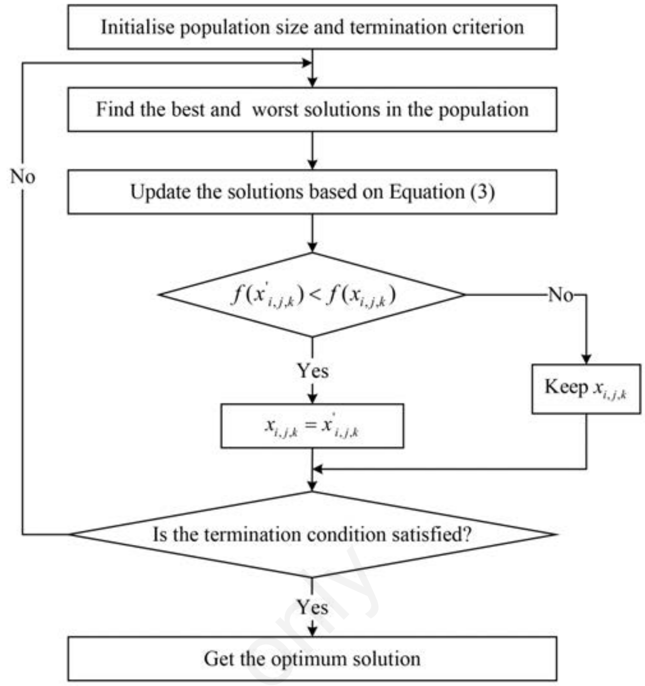

Figure 1. The flowchart of the Rao-1 algorithm.

Table 1. The upper and lower bounds of the model parameters.

Parameter	θ_r (cm^3/cm^3)	θ_s (cm^3/cm^3)	α (cm^{-1})	n
LB	0	0	0	1
UB	1	1	+∞	+∞

Algorithm 1: Rao-1 algorithm

Input: Population size *n*, the upper bound *LB*, the lower bound *LB*, and the max generation *Maxgen*;
Output: Optimum model parameters: x_{best};

```
for i = 1:n
  for j = 1:4
    Initialise x_{i,j};
  end
end
Compute f(x);
Get x_{best,1} and x_{worst,1} based on f(x);
i =1;
while (i ≤ Maxgen)
  for k = 1:n
      for j = 1:4
   x'_{j,k,i} = x_{j,k,i} + r_{j,i}(x_{j,best,i} − x_{j,worst,i});
   bordering x'_{j,k,i} by Equation (4);
      end
      if f(x'_{j,k,i})<f(x_{j,k,i})
        x_{j,k,i+1}=x'_{j,k,i};
        f(x_{j,k,i})=f(x'_{j,k,i});
      else
   x_{j,k,i+1}=x_{j,k,i};
      end
  end
   i = i+1;
   Update x_{best,i} and x_{worst,i}
end
return x_{best};
```

$$x_i^1 = \begin{cases} y_i + r_1((ub_i - lb_i)r_2 + lb_i) & r_3 \geq 0 \\ y_i - r_1((ub_i - lb_i)r_2 + lb_i) & r_3 < 0 \end{cases} \quad (5)$$

where x_i^1 is the position of the first salp in the i^{th} dimension, y_i is the target position in the j^{th} dimension, lb_i and ub_i are the critical values of the i^{th} dimension, and r_1,r_2,r_3 are three random numbers separately.

PSO (Wang *et al.*, 2018), a swarm iteration algorithm, was first developed by Kennedy and Eberhart in 1995. This algorithm starts from a random solution and searches the global optimum solutions by following the optimal particles in the solution space. This algorithm is widely used to solve optimisation problems because it is easy to execute and has fast convergence speed. The iterative process of the PSO algorithm (Yang *et al.*, 2012) is described in Eq. 6:

$$\begin{aligned} v_i' &= wv_i + c_1r_1(Pbest_i - x_i) + c_2r_2(Gbest - x_i) \\ x_i' &= x_i + v_i \end{aligned} \quad (6)$$

where v'_i and v_i represent the new and old velocities, x' and x denote the updated and previous positions, respectively, ω, c_1, and c_2 are three hyper-parameters, r_1 and r_2 are two random numbers, and $Pbest_i$ is the best position that particle i has ever experienced, while $Gbest$ is the best position of all the individual particles.

The DE algorithm (Storn and Price, 1997) is a heuristic evolutionary global optimisation approach especially suited for continuous spaces optimisation problems. In the DE algorithm, the random population of P parent parameter vectors is initialised in Eq. 7:

$$x_{i,j} = l_i + r_{i,j}(u_i - l_i) \quad (7)$$

where $r_{i,j}$ is a random number in range (0:1). There are three randomly chosen parent vectors $x_{1,k}$, $x_{2,k}$, and $x_{3,k}$. A mutant vector is computed according to Eq. 8:

$$y_k = x_{1,k} + M_f(x_{2,k} - x_{3,k}) \quad (8)$$

where M_f is a mutation factor in range (0:2) that controls the differential variation. Then the trial vector is obtained as follows:

$$z_{i,j} = \begin{cases} y_{i,j} & \text{if } r_{i,j} \leq C_f \text{ or } j = L \\ x_{i,j} & \text{otherwise} \end{cases} \quad (9)$$

where C_f is a crossover constant in range (0:2), and L is a random integer in the range (0:4). The vector with the best solution will be the parent vector by comparing $x_{i,j}$ and $z_{i,j}$ in the next iteration. The process is executed continuously until the end condition is satisfied.

Table 2. The physical properties of soil samples employed in this study.

Soil sample code	Texture	Number of data points	Bulk density	Location
1011	Loamy Sand	9	1.52	Union Springs, AL, USA
1012	Loamy Sand	10	1.4	Union Springs, AL, USA
1022	Sand	10	1.6	Blackville, SC, USA
1023	Sand	9	1.67	Blackville, SC, USA
1091	Sandy Loam	15	1.63	Lillington, NC, USA
1101	Sandy Loam	9	1.83	Blackville, SC, USA
1102	Sandy Clay Loam	9	1.71	Blackville, SC, USA
1103	Sandy Clay Loam	9	1.56	Blackville, SC, USA
1135	Sandy Clay	15	1.63	Blackstone, VA, USA
1174	Sandy Clay	11	1.4	Clemson, SC, USA
1173	Clay Loam	11	1.38	Clemson, SC, USA
1191	Clay Loam	6	1.53	Payne County, OK, USA
2681	Clay	8	1.51	Irchel, Switzerland
2691	Clay	5	1.67	Baden, Switzerland
1212	Loam	10	1.61	Tillman County, OK, USA
2321	Loam	7	1.74	Hasenholz, Germany
1330	Silt	21	1.37	Ohlendorf, Hannover, Germany
3214	Silt	13	1.37	Dickey Co., ND, USA
1280	Silt Loam	10	1.34	Ohlendorf, Hannover, Germany
1281	Silt Loam	10	1.48	Ohlendorf, Hannover, Germany
1361	Silty Clay	10	1.49	Reinhausen, Germany
2031	Silty Clay	8	1.25	Oahu, HI, USA
2050	Silty Clay Loam	8	1.26	Oahu, HI, USA
2051	Silty Clay Loam	8	1.38	Oahu, HI, USA

RETC program (Borek and Bogdał, 2018) was proposed by the U.S. Salinity Laboratory. It can be used to analyse the soil water retention and hydraulic conductivity functions of unsaturated soils and predict the hydraulic conductivity. The program is publicly accessible on the Laboratory's website http://www.pc-progress.com. The precompiled version is used in this study.

To assess the estimation performance of all considered methods, the sum of squared error (SSE) is employed; it is defined in

$$SSE = \sum_{i=1}^{N} (\Theta - \hat{\Theta})^2 \tag{10}$$

Case study

To assess the availability and superiority of the proposed method, the parameters of the van Genuchten model are estimated

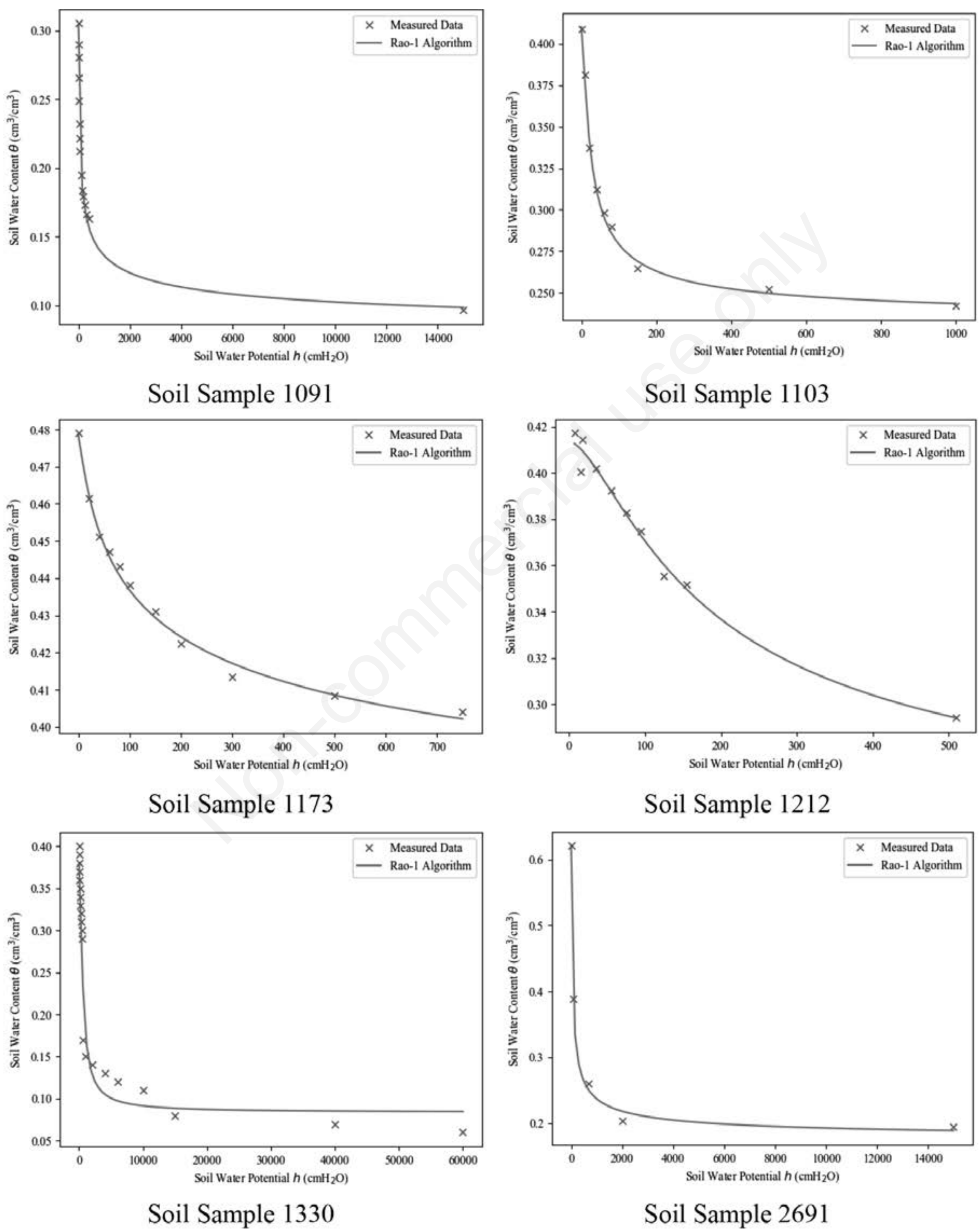

Figure 2. Fitting result of six selected soil samples using Rao-1 algorithm.

Table 3. Estimation results and computing time of all methods.

Soil sample code	Method	θ_r (cm³/cm³)	θ_s (cm³/cm³)	α (cm⁻¹)	n	SSE (10^{-3})	Computing time (s)
1011	Rao-1	0.07375	0.39916	0.02839	3.39946	*1.297186*	*38.73*
	DE	0.07284	0.40023	0.02866	3.30151	1.303650	41.41
	PSO	0.07374	0.39915	0.02839	3.39778	1.297191	73.45
	SSA	0.07375	0.39917	0.02839	3.39857	1.297187	71.33
	RETC	0.07378	0.39913	0.02838	3.40220	1.297192	-
1012	Rao-1	0.06179	0.37015	0.02964	3.08251	*1.512023*	*33.25*
	DE	0.06129	0.37117	0.03000	3.01733	1.516182	37.26
	PSO	0.06202	0.36984	0.02959	3.09136	1.512316	70.55
	SSA	0.06179	0.37015	0.02964	3.08253	*1.512023*	76.38
	RETC	0.06181	0.37012	0.02963	3.08531	1.512029	-
1022	Rao-1	0.05138	0.36662	0.11422	1.87686	*0.304936*	*43.23*
	DE	0.05138	0.36662	0.11422	1.87686	*0.304936*	47.06
	PSO	0.05137	0.36662	0.11423	1.87678	*0.304936*	79.95
	SSA	0.05138	0.36662	0.11422	1.87686	*0.304936*	76.11
	RETC	0.05138	0.36662	0.11422	1.87691	*0.304936*	-
1023	Rao-1	0.04238	0.34034	0.13397	1.94023	*0.570740*	*38.02*
	DE	0.04238	0.34034	0.13397	1.94023	*0.570740*	41.20
	PSO	0.04245	0.34024	0.13370	1.94209	0.570765	71.47
	SSA	0.04235	0.34033	0.13398	1.93974	0.570742	70.71
	RETC	0.04243	0.34033	0.13389	1.94107	0.570743	-
1091	Rao-1	0.06608	0.30440	0.07259	1.28474	*0.235061*	*60.71*
	DE	0.06608	0.3044	0.07259	1.28474	*0.235061*	64.43
	PSO	0.06879	0.30426	0.06949	1.29478	0.237333	116.70
	SSA	0.06608	0.3044	0.07259	1.28474	*0.235061*	100.4
	RETC	0.06612	0.30439	0.07255	1.28487	*0.235061*	-
1101	Rao-1	0.09259	0.24552	0.08503	1.58018	*0.324006*	*37.43*
	DE	0.09259	0.24552	0.08503	1.58018	*0.324006*	41.64
	PSO	0.09258	0.24558	0.08520	1.57946	0.324014	70.88
	SSA	0.09259	0.24552	0.08503	1.58017	*0.324006*	74.22
	RETC	0.09260	0.24552	0.08502	1.58026	*0.324006*	-
1102	Rao-1	0.12175	0.34674	0.15905	1.29729	*0.240458*	*37.83*
	DE	0.1226	0.34668	0.15752	1.30022	0.240490	40.36
	PSO	0.12122	0.34663	0.15930	1.29558	0.240495	74.47
	SSA	0.12175	0.34674	0.15905	1.29729	*0.240458*	72.10
	RETC	0.12171	0.34675	0.15911	1.29716	*0.240458*	-
1103	Rao-1	0.23193	0.41032	0.08380	1.61958	*0.135185*	*38.58*
	DE	0.23193	0.41032	0.08380	1.61958	*0.135185*	43.54
	PSO	0.23237	0.41085	0.08430	1.62166	0.135702	76.62
	SSA	0.23192	0.41033	0.08385	1.61929	0.135186	70.94
	RETC	0.23194	0.41032	0.08379	1.61972	0.135186	-
1135	Rao-1	0.26477	0.41033	0.00098	2.47450	*0.254249*	*58.32*
	DE	0.26427	0.41075	0.00099	2.44342	0.255082	61.35
	PSO	0.26478	0.41033	0.00098	2.47456	*0.254249*	116.30
	SSA	0.26777	0.40955	0.00097	2.65691	0.266914	85.79
	RETC	0.26477	0.41033	0.00098	2.47433	0.254264	-
1174	Rao-1	0.34238	0.47187	0.02134	1.39909	*0.049498*	*50.04*
	DE	0.34239	0.47187	0.02134	1.39916	*0.049498*	53.01
	PSO	0.34438	0.47211	0.02152	1.40676	*0.049908*	80.82
	SSA	0.34235	0.47187	0.02135	1.39895	*0.049498*	77.01
	RETC	0.34232	0.47187	0.02135	1.39871	*0.049498*	-
1173	Rao-1	0.30150	0.47850	0.04958	1.15555	*0.037083*	*49.01*
	DE	0.33389	0.47810	0.04370	1.21165	0.038211	52.01
	PSO	0.30423	0.47855	0.04831	1.16113	0.037566	80.20
	SSA	0.33105	0.47822	0.04474	1.20452	0.037954	76.06
	RETC	0.30130	0.47851	0.04964	1.15526	*0.037083*	-
1191	Rao-1	0.31245	0.37668	0.04202	2.59342	*0.110539*	*27.71*
	DE	0.31231	0.37712	0.04302	2.55034	0.110650	30.75
	PSO	0.31245	0.37668	0.04202	2.59342	*0.110539*	49.04
	SSA	0.31245	0.37668	0.04202	2.59336	*0.110539*	64.38
	RETC	0.31244	0.37671	0.04208	2.59105	*0.110539*	-

To be continued on next page

Table 3. Continued from previous page.

Soil sample code	Method	θ_r (cm³/cm³)	θ_s (cm³/cm³)	α (cm⁻¹)	n	*SSE* (10⁻³)	Computing time (s)
2681	Rao-1	0.17116	0.52681	0.55626	1.18844	*0.570045*	*35.91*
	DE	0.17190	0.52646	0.54744	1.18959	0.570076	39.27
	PSO	0.17109	0.52681	0.55944	1.18831	0.570164	64.89
	SSA	0.11002	0.56641	2.21531	1.12696	0.659117	70.01
	RETC	0.17129	0.52678	0.55512	1.18863	*0.570045*	-
2691	Rao-1	0.17148	0.62266	0.05039	1.49196	*0.325667*	*24.45*
	DE	0.17148	0.62266	0.05039	1.49196	*0.325667*	27.24
	PSO	0.16544	0.61991	0.05433	1.45643	0.345431	42.85
	SSA	0.17148	0.62266	0.05039	1.49196	*0.325667*	60.71
	RETC	0.17146	0.62266	0.05041	1.49180	0.325668	-
1212	Rao-1	0.22710	0.41367	0.00998	1.61331	*0.180792*	*42.99*
	DE	0.21466	0.41451	0.01019	1.54398	0.181684	45.74
	PSO	0.21108	0.4125	0.00918	1.55443	0.196422	77.61
	SSA	0.22710	0.41367	0.00998	1.61331	*0.180792*	74.74
	RETC	0.22714	0.41367	0.00998	1.61355	0.180793	-
2321	Rao-1	0.21378	0.37169	0.10356	1.16942	*0.020588*	*31.56*
	DE	0.23571	0.37133	0.09163	1.21645	0.021253	34.24
	PSO	0.21771	0.37150	0.10364	1.17481	0.020783	57.47
	SSA	0.22054	0.37164	0.10077	1.18134	0.020633	66.39
	RETC	0.21371	0.37169	0.10358	1.16930	*0.020588*	-
1330	Rao-1	0.08362	0.38004	0.00259	2.11588	*11.25378*	*84.68*
	DE	0.07149	0.38676	0.00310	1.81632	11.56176	88.84
	PSO	0.08392	0.38013	0.00258	2.12511	11.25432	168.30
	SSA	0.08362	0.38004	0.00259	2.11589	*11.25378*	106.04
	RETC	0.08374	0.37998	0.00259	2.12003	11.25387	-
3214	Rao-1	0.02210	0.63262	0.03612	1.12399	*0.211204*	*57.63*
	DE	0.10587	0.63028	0.03271	1.15239	0.212123	60.98
	PSO	0.02245	0.62949	0.03452	1.12325	0.223096	99.19
	SSA	0.17953	0.62759	0.02934	1.19018	0.216027	83.40
	RETC	0.05081	0.63183	0.03496	1.13245	0.211283	-
1280	Rao-1	0.05376	0.41102	0.00629	1.53053	*0.640241*	*42.80*
	DE	0.04230	0.41592	0.00730	1.46636	0.682748	45.72
	PSO	0.05367	0.41115	0.00629	1.53078	0.640320	79.89
	SSA	0.05375	0.41102	0.00629	1.53050	*0.640241*	74.88
	RETC	0.05373	0.41103	0.00629	1.53039	0.640243	-
1281	Rao-1	0.07583	0.38946	0.00474	1.84252	*1.417541*	*43.36*
	DE	0.07188	0.39168	0.00503	1.78249	1.432588	46.49
	PSO	0.07533	0.38931	0.00474	1.84096	1.418170	78.05
	SSA	0.07583	0.38946	0.00474	1.84253	*1.417541*	75.12
	RETC	0.07582	0.38946	0.00474	1.84239	1.417562	-
1361	Rao-1	0.16460	0.43307	0.00430	1.25777	*0.266939*	*43.97*
	DE	0.15406	0.43372	0.00455	1.23896	0.267767	45.67
	PSO	0.16416	0.43307	0.00430	1.25734	0.266960	78.21
	SSA	0.16460	0.43307	0.00430	1.25777	*0.266939*	76.70
	RETC	0.16449	0.43308	0.00431	1.25757	0.266944	-
2031	Rao-1	0.34946	0.52096	0.04675	1.71563	*0.028071*	*34.49*
	DE	0.34947	0.52100	0.04680	1.71539	0.028073	36.39
	PSO	0.34953	0.52126	0.04697	1.71575	0.028107	62.61
	SSA	0.34946	0.52096	0.04675	1.71564	*0.028071*	70.01
	RETC	0.32697	0.54776	0.09670	1.44009	0.092460	-
2050	Rao-1	0.31766	0.51179	0.01443	1.82296	*0.063841*	*38.72*
	DE	0.31607	0.51256	0.01477	1.79689	0.064247	41.32
	PSO	0.31764	0.51182	0.01445	1.82210	0.063844	64.50
	SSA	0.31766	0.51179	0.01443	1.82297	*0.063841*	70.75
	RETC	0.31769	0.51178	0.01443	1.82339	*0.063841*	-
2051	Rao-1	0.30012	0.50552	0.02319	1.43696	*0.032243*	*35.27*
	DE	0.29671	0.50681	0.02431	1.41657	0.033245	38.14
	PSO	0.29726	0.5061	0.02364	1.42117	0.033587	64.42
	SSA	0.08398	0.51868	0.04950	1.12801	0.102485	69.66
	RETC	0.30017	0.50551	0.02318	1.43722	*0.032243*	-

SSA, salp swarm algorithm; PSO, particle swarm optimization algorithm; DE, differential evolution algorithm. The italics for some values here is to highlight the best performance among all methods for comparison of computational results conveniently.

for 24 soil samples covering 12 soil textural classes. Meanwhile, the estimation results of the Rao-1 algorithm are compared with those of SSA, PSO, DE, and RETC.

Data description

The soil samples tested in this study were obtained from the UNSODA database (Nemes *et al*., 2001), containing the soil water potential and water content data of various soil types. The physical properties of soil samples used for this study are shown in Table 2.

According to Table 2, it is evident that the studied soil samples are of different types, bulk densities, locations, and sizes. Thus, the effectiveness and universal applicability of the proposed method are validated accordingly.

Experimental results and discussion

During the estimation process, the population size and maximum iteration of all the considering evolutionary computation algorithms, including Rao-1, PSO, DE, and SSA, are set to 200 and 3000, respectively. The evolutionary computation algorithms are implemented using Python 3.7. All methods are executed on a PC with an Intel Core i3@3.6GHz CPU and 8GB memory. Both the estimated results and computing time are recorded, presented in Table 3. Especially since the results of RETC are derived by using professional software, it is hard to measure the computing time accurately. Therefore, the computing time of RETC is not recorded in Table 3.

According to the results presented in Table 3, it is observed that the Rao-1 algorithm yields the smallest estimation error and the shortest computing time over all soil samples. To be specific, Rao-1 outperforms other benchmarks for Sample 1011 and 3214. The Rao-1 algorithm can still generate the best estimation results for other samples and is more stable than other evolutionary computation algorithms. Regarding the computing time, SSA and PSO need more computing time than Rao-1 and DE. However, Rao-1 generally uses 3s less computing time than DE. Through the above comparative analysis, three advantages of the Rao-1 algorithm are observed: good optimisation performance, fast convergence speed, and fewer control parameters. Therefore, it is very competitive to use the Rao-1 algorithm for parameter estimation of the van Genuchten model.

In order to illustrate the performance of the Rao-1 algorithm intuitively, Figure 2 shows SWRC curves of six selected soil samples using the Rao-1 algorithm. From Figure 2, it can be seen that the estimated curves using the Rao-1 algorithm fit measured data well for different types of soil. In addition, the Rao-1 algorithm requires only the standard control parameters like population size and number of iterations and does not require any algorithm-specific control parameters. Thus, it is greatly suitable to apply the Rao-1 algorithm to the SWRC model parameters estimation.

Due to the simple updating operation of the Rao-1 algorithm, it can be easily implemented on embedded devices. In order to verify the computing performance of Rao-1 on embedded devices, Raspberry Pi 3 with a 64-bit quad-core ARMv8 CPU and 1 GB memory is utilised. SSE values of Rao-1 executed on a PC and Raspberry Pi 3 are listed in Table 4. From Table 4, it is observable that the same estimation accuracy of the Rao-1 algorithm can be obtained regardless of the computing architecture.

Table 4. SSE values of Rao-1 on PC and Raspberry Pi 3.

Soil sample code	PC (10^{-3})	Raspberry Pi 3 (10^{-3})
1011	1.297186	1.297186
1012	1.512023	1.512023
1022	0.304936	0.304936
1023	0.570740	0.570740
1091	0.235061	0.235061
1101	0.324006	0.324006
1102	0.240458	0.240458
1103	0.135185	0.135185
1135	0.254249	0.254249
1174	0.049498	0.049498
1173	0.037083	0.037083
1191	0.110539	0.110539
2681	0.570045	0.570045
2691	0.325667	0.325667
1212	0.180792	0.180792
2321	0.020588	0.020588
1330	11.25378	11.25378
3214	0.211204	0.211204
1280	0.640241	0.640241
1281	1.417541	1.417541
1361	0.266939	0.266939
2031	0.028071	0.028071
2050	0.063841	0.063841
2051	0.032243	0.032243

Conclusions

A new parameter estimation approach based on the Rao-1 algorithm was applied to estimate the soil water retention model parameters in this paper. Compared with the conventional heuristic search methods, the Rao-1 algorithm is a relatively simple heuristic search algorithm that only contains addition and multiplication operations, which can save a lot of computing overhead. Therefore, it is very efficient for embedded implementation. In addition, the Rao-1 algorithm does not contain any algorithm-specific parameters, which guarantees the universality of the proposed method for various soil types.

To assess the estimation performance of the Rao-1 algorithm, it was compared with SSA, PSO, DE, and RETC. The experimental results proved that the Rao-1 algorithm outperformed other benchmarks over 24 soil samples of 12 soil textural classes. Furthermore, due to the simple updating operation, applying the Rao-1 algorithm to practical agricultural applications is promising.

References

Borek Ł., Bogdał A. 2018. Soil water retention of the Odra river alluvial soils (Poland): estimating parameters by retc model and laboratory measurements. Appl. Ecol. Environ. Res. 16:4681-99.

Brooks R.H., Corey A.T. 1964. Hydraulic properties of porous media and their relation to drainage design. Trans. ASAE 7:0026-8.

Campbell, G. S. 1974. A simple method for determining unsaturated conductivity from moisture retention data. Soil Sci. 117:311-4.

Duong V.-H., Bastawrous H.A., Lim K.C., See K.W., Peng Z., Dou S.X. 2015. Online state of charge and model parameters estimation of the LiFePO4 battery in electric vehicles using multiple adaptive forgetting factors recursive least-squares. J. Power Sour. 296:215-24.

Gardner W.R., Hillel D., Benyamini Y. 1970. Post-irrigation movement of soil water: 1. Redistribution. Water Resour. Res. 6:851-61.

Genuchten M.Th van. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. Am. J. 44:892-98.

Kassaye K.T., Boulange J., Lam Th.Van, Saito H., Watanabe H. 2020. Monitoring soil water content for decision supporting in agricultural water management based on critical threshold values adopted for andosol in the temperate monsoon climate. Agric. Water Manage. 229:105930.

Kawai K., Kato S., Karube D. 2000. The model of water retention curve considering effects of void ratio. In: Unsaturated soils for Asia. CRC Press, Boca Raton, FL, USA.

Li M.Y. 2018. Parameter estimation and nonlinear least-squares methods. In: Michael Y. Li (Ed.), An introduction to mathematical modeling of infectious diseases. Mathematics of Planet Earth. Cham: Springer International Publishing, Berlin, Germany, pp. 103-124.

Li Y.-B., Liu Y., Nie W.-B., Ma X.-Y. 2018. Inverse modeling of soil hydraulic parameters based on a hybrid of vector-evaluated genetic algorithm and particle swarm optimization. Water 10:84.

Maggi S. 2017. Estimating water retention characteristic parameters using differential evolution. Comput. Geotechn. 86:163-72.

Mirjalili S., Gandomi A.H., Mirjalili S.Z., Saremi S., Faris H., Mirjalili S.M. 2017. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Engine. Softw. 114:163-91.

Nemes A., Schaap M.G., Leij F.J., Wösten J.H.M. 2001. Description of the unsaturated soil hydraulic database UNSODA Version 2.0. J. Hydrol. 251:151-62.

Rao R. 2016. Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int. J. Indust. Engine. Comput. 7:19-34.

Rao R.V. 2020. Rao algorithms: three metaphor-less simple algorithms for solving optimization problems. Int. J. Ind. Engine. Comput. 6:107-30.

Rossi C., Nimmo J.R. 1994. Modeling of soil water retention from saturation to oven dryness. Water Resour. Res. 30:701-8.

Silva M.L.d.N., Libardi P.L., Setti Gimenes F.H. 2018. Soil water retention curve as affected by sample height. Rev. Brasil. Ciênc. Do Solo 42:2018.

Storn R., Price K. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimizat. 11:341-59.

Talat A.E., Galal M.E., Yeser A., Saad El-Dein A. A. 2020. Quantifying the hydraulic properties of some Egyptian soils using RETC Code. Arab Univ. J. Agric. Sci. [Epub ahead of print].

Tan F., Zhou W.-H., Yuen K.-V.. 2016. Modeling the soil water retention properties of same-textured soils with different initial void ratios. J. Hydrol. 542:731-43.

Venkata Rao R. 2019. Jaya optimization algorithm and its variants. In: Venkata Rao R. (Ed.), Jaya: an advanced optimization algorithm and its engineering applications. Cham: Springer International Publishing, Berlin, Germany, pp 9–58.

Wang D., Dapei T., Lei L. 2018. Particle swarm optimization algorithm: an overview. Soft Comput. 22:387-408.

Wang L., Chao H., Lingmiao H. 2018. Parameter estimation of the soil water retention curve model with Jaya algorithm. Comput. Electron. Agric. 151:349-53.

Wang L., Zijun Z., Chao H., Kwok Leung T. 2018. A GPU-accelerated parallel jaya algorithm for efficiently estimating Li-Ion battery model parameters. Appl. Soft Comput. 65:12-20.

Yang X., Xue Y.Y., Min J. 2012. Determining the soil water characteristic curve in term of van genuchten parameters by the particle swarm optimization. Appl. Mechan. Mater. 160:130-34.

Zhai Q., Harianto R., Alfrendo S., Guoliang D., Yan Z. 2020. Framework to estimate the soil-water characteristic curve for soils with different void ratios. Bull. Engine. Geol. Environment 79:4399-409.

Zhang J., Zhenhua W., Xiong L. 2018. Parameter estimation for soil water retention curve using the salp swarm algorithm. Water 10:815.