

Technical Note

Parameter Estimation for Soil Water Retention Curve Using the Salp Swarm Algorithm

Jing Zhang ^{1,2,3}, Zhenhua Wang ^{4,*} and Xiong Luo ^{1,2,3,*} 

¹ School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China; g20178747@xs.ustb.edu.cn

² Guangdong Provincial Key Laboratory of New and Renewable Energy Research and Development, Guangzhou 510640, China

³ Key Laboratory of Geological Information Technology, Ministry of Land and Resources, Beijing 100037, China

⁴ College of Water Resources and Architectural Engineering, Shihezi University, Shihezi 832003, China

* Correspondence: wzh2002027@163.com (Z.W.); xluo@ustb.edu.cn (X.L.)

Received: 17 May 2018; Accepted: 18 June 2018; Published: 20 June 2018



Abstract: This paper employs an optimization algorithm called the salp swarm algorithm (SSA) for the parameter estimation of the soil water retention curve model. The SSA simulates the behavior of searching for food of the salp swarm and manages to find the optimal solutions for optimization problems. In this paper, parameter estimation of the van Genuchten model based on nine soil samples, covering eight soil textures, is conducted. The optimization problem that minimizes the difference between the measured and the estimated water content is formulated, and the SSA is applied to solve this problem. To validate the competitive advantage of the SSA, the experimental results are compared with Particle Swarm Optimization algorithm, the Differential Evolution algorithm and the RETC program, which indicates that SSA performs better than the three methods.

Keywords: irrigation and drainage; soil characteristics; salp swarm algorithm; evolutionary algorithm

1. Introduction

Soil water retention is a key soil property, playing a significant role in many applications in the fields of irrigation, hydrology, geotechnical engineering, and soil science in general [1]. The soil water retention curve (SWRC), which gives the relation between the amount of water contained in soil and the soil water potential, is widely used to estimate unsaturated soil properties [2]. The method of measuring the SWRC directly can be time-consuming, and improper measurement may lead to greater errors. Thus, mathematical equations to approximate the general form of the SWRC have been suggested, and several different analytical models, including Farrell-Larson [3], Brooks-Corey [4], Fredlund-Xing [3], Gardner [5], van Genuchten [6], Campbell [7], and Russo [8,9] models, have been proposed. These models usually contain several parameters that need to be identified based on collected experimental data. An optimization problem that minimizes the difference between the measured and the estimated water content are formulated to determine the parameters. In order to address the optimization problem, nonlinear least-square (NLS) [6] methods and heuristic search methods [10] are usually considered.

NLS methods, a form of least squares analysis, are used to fit a set of m observations with a model that is non-linear parameterized by n unknown parameters ($m > n$) [11,12]. The basis of the NLS method is to approximate the model by a linear function and to refine the parameters by successive iterations. The RETC algorithm is a representative NLS method [13] and is widely used in parameter estimation of SWRC. However, the estimation result of RETC cannot be guaranteed

to be a globally optimal solution, which requires parameter initialization with prior knowledge of specified soil types. On the contrary, the heuristic search methods do not need prior knowledge to initialize the model parameters, which contributes to a performance that is better than that of the NLS methods in the parameter estimation of SWRC. However, these heuristic search methods require some specified parameters that could have effects on the final estimation solution, and the adjustment of these parameters may be also time-consuming [14], such as the inertia weight, and learning factors in particle swarm optimization (PSO) [15].

There are many related studies on the parameter estimation of SWRC. Hosseini et al. [2] used the van Genuchten and Fredlund-Xing equations for the estimation of the SWRC, which takes the plasticity index and fine content into consideration. Xing et al. [16] proposed a vertical infiltration method to estimate parameters α and n and further develop the SWRC and establish the relationships describing the cumulative infiltration and infiltration rate with the depth of the wetting front. Oh et al. [17] modified van Genuchten's function for SWRC within the range of low matric suction to improve the Mualem hydraulic conductivity (HC), and van Genuchten-Mualem hydraulic conductivity was then modified to integrate the proposed SWRC for each interval decomposed by a tangential curve. The modified model was tested on Korean weathered soils and accurately predicted the unsaturated behavior of the HC functions. Wang et al. [18] proposed a method based on the Jaya algorithm to estimate the parameters of the van Genuchten model via four different soil samples, and the results showed the superiority of the proposed method. Matula et al. [19] applied Woesten's continuous pedotransfer functions to 140 SWRCs measured in a selected location, and the results showed that the general equation of Woesten's pedotransfer functions were not very suitable to estimate the SWRC for the location studied. Maggi [20] used a method based on the differential evolution algorithm for the determination of the characteristic parameters of several water retention models from the experimental data, and the method could find the optimal model parameters without any prior information. Chen et al. [21] used the Monte-Carlo method to analyze the sensitivity of the parameters and uncertainty of the van Genuchten model to obtain the key parameters and a posteriori parameter distribution to guide the parameter identification, and a new type of intelligent algorithm-difference search algorithm was then employed to identify these parameters. Nascimento et al. [22] estimated the parameters of the van Genuchten model using the inverse modeling function of Hydrus-1D, and the results revealed that the Hydrus-1D can simulate well the behavior of matric potential and moisture over time if the field data is fed.

In order to avoid the additional, unnecessary work caused by adjusting these specified parameter in the SWRC parameter estimation, the SSA was employed to determine these parameters. Taking advantage of the lack of algorithm-specified parameters in SSA [23], the parameter estimation of the SWRC based on nine different soil samples, covering eight soil textures, is performed and the van Genuchten Model is selected as the SWRC model. To verify the efficient performance of this algorithm, the estimation solution is compared with the PSO algorithm, the differential evolution (DE) algorithm, and the RETC program.

2. The Van Genuchten Model

The van Genuchten Model was proposed by van Genuchten in 1980 [6]. Due to the superiorities of simple form and high fitting ability, the van Genuchten Model has gained popularity and is widely used in research on soil water retention. The van Genuchten Model depicts the relation of the water contained $\varphi(h)(cm^3cm^{-3})$ and soil water potential $h(cm^{-1})$. The formula of the van Genuchten Model is shown in (1):

$$\varphi(h) = \varphi_r + (\varphi_s - \varphi_r)(1 + \alpha|h|^n)^{-m} \quad (1)$$

where φ_r is the residual water content, φ_s represents the saturated water content, α is a parameter given by experience, n is the shape parameter of the curve, and $m = 1 - 1/n$.

Referring to the optimization problem formulated, the sum of squared errors (SSE) is utilized as the metric in order to evaluate the difference between the measured and the estimated water content. The SSE is defined as follows:

$$SSE = \sum_{k=1}^N (\varphi(h_k)_r - \varphi(h_k)_e)^2 \quad (2)$$

where the $\varphi(h_k)_r$ is the measured data, while the $\varphi(h_k)_e$ is the estimated value corresponding to the h_k , and N is the total number of measurements in each soil sample.

3. Salp Swarm Algorithm

The salp swarm algorithm was proposed by Seyedali Mirjalili in 2017 [23] and is used in many fields. For example, Sayed et al. [24] used an SSA-based algorithm for feature selection, and the results indicate that the algorithm has the capability of finding an optimal feature subset, which maximizes the classification accuracy and minimizes the number of selected features [24,25]. It can also be used to adjust the hyper-parameters of neural networks and other machine learning models [26–28].

The main idea of SSA is inspired by the predatory behavior of salp, a kind of creature in the ocean. The salps often form a swarm called a salp chain and move forward to the food resource, which has not been able to be explained clearly until now. In SSA, the population is divided into two groups, leaders and followers, according to the position in the chain. The leader is in the front of the chain and the followers follow it. For a specified model, assume that n variables need to be estimated, x denotes the position of a salp, and y defines the food source, which indicates the target of the swarm in the search space. The leader salp updates its position by (3) in the search process:

$$x_i^1 = \begin{cases} y_i + r_1((ub_i - lb_i)r_2 + lb_i) & r_3 \geq 0 \\ y_i - r_1((ub_i - lb_i)r_2 + lb_i) & r_3 < 0 \end{cases} \quad (3)$$

where the x_i^1 is the position of the first salp in the i th dimension, and y_i is the food position in the i th dimension. lb_i and ub_i represent the lower bound and the upper bound of the i th dimension, and r_1, r_2, r_3 are three random numbers.

Among the three random numbers, r_1 occupies the dominant position because it balances exploration and exploitation during the entire search process, and it is defined as

$$r_1 = 2e^{-\left(\frac{l}{L}\right)^2} \quad (4)$$

where L is the maximum iterations, l is the current iteration, and r_2, r_3 are random numbers between [0,1]. For the followers, the following equation is used to update their positions according to Newton's law of motion:

$$x_i^j = \frac{1}{2}\lambda t^2 + \delta_0 t \quad (5)$$

where $j \geq 2$, x_i^j indicates the position of the j th salp in the i th dimension, t is the time, δ_0 is an initial speed, and $\lambda = \frac{\delta_{final}}{\delta_0}$, where $\delta = \frac{x-x_0}{t}$.

Considering the assumption that $\delta_0 = 0$, and t equals the iteration in an optimization problem, the aforementioned equation can be transformed into the following form:

$$x_i^j = \frac{1}{2}(x_i^j + x_i^{j-1}) \quad (6)$$

where $j \geq 2$. This equation shows that the following salps update their position based on the position of their own and the prior salp.

If some salps move outside the limited search space, they will be brought back to the boundaries though the limitation formula:

$$x_i^j = \begin{cases} lb & \text{if } x_i^j \leq lb \\ ub & \text{if } x_i^j \geq ub \\ x_i^j & \text{otherwise} \end{cases} \quad (7)$$

All of the above updates are iteratively executed until satisfaction of the end condition. It should be pointed that the food source can be updated sometimes because the salp chain may find a better solution by exploring and exploiting the space around the current solution. In other words, the salp chain has the ability to move toward the global optimum solution during the optimization. The procedure of the SSA is illustrated in Algorithm 1, which shows that the SSA starts approximating the global optimum by initiating multiple salps with random positions, updates these positions iteratively, and finally finds a global optimization solution. A flowchart of the SSA is shown in Figure 1.

Algorithm 1 The Procedure of the Salp Swarm Algorithm (SSA) Algorithm.

Require: Initialize the salp population $x_i (i = 1, 2, \dots, n)$ consider ub and lb .

```

while (End condition is not satisfied)
  Calculate the fitness of each search salp
   $F$ =the best search solution
  Update  $r_1$  by Equation (4)
  for each salp ( $x_i$ )
    if ( $i == 1$ )
      Update the position of the leading salp by Equation (3)
    esle
      Update the position of the followers salp by Equation (6)
    end
  end
  Verify the position of salps based on the upper and lower bounds
end
return  $F$ 

```

Benchmarking Algorithms

To assess the performance of the SSA in parameter estimation, its estimation results are compared with the PSO and DE algorithms, as well as the RETC program.

PSO is an evolutionary algorithm–algorithm, which starts from a random solution and finds the global optimum by following the optimal value found in the current research. The advantage of PSO is that it is easy to implement and there are not many parameters to be adjusted. The updating formula of PSO is

$$\delta'_i = \omega\delta_i + r_1c_1(pbest_i - x_i) + r_2c_2(gbest_i - x_i) \quad (8)$$

where ω , c_1 , and c_2 are three specified parameters, δ'_i and δ_i represent the new updated and the current velocities, and $pbest_i$ is the best solution found by the i th particle, while the $gbest_i$ indicates the global best solution.

The RETC program, a computer program which includes an NLS algorithm for optimization, developed by the U.S Salinity Laboratory, can be used to predict the hydraulic conductivity from observed soil water retention data. The executable program can be obtained from the Laboratory's website freely, and the program can work normally on a Windows 10 system.

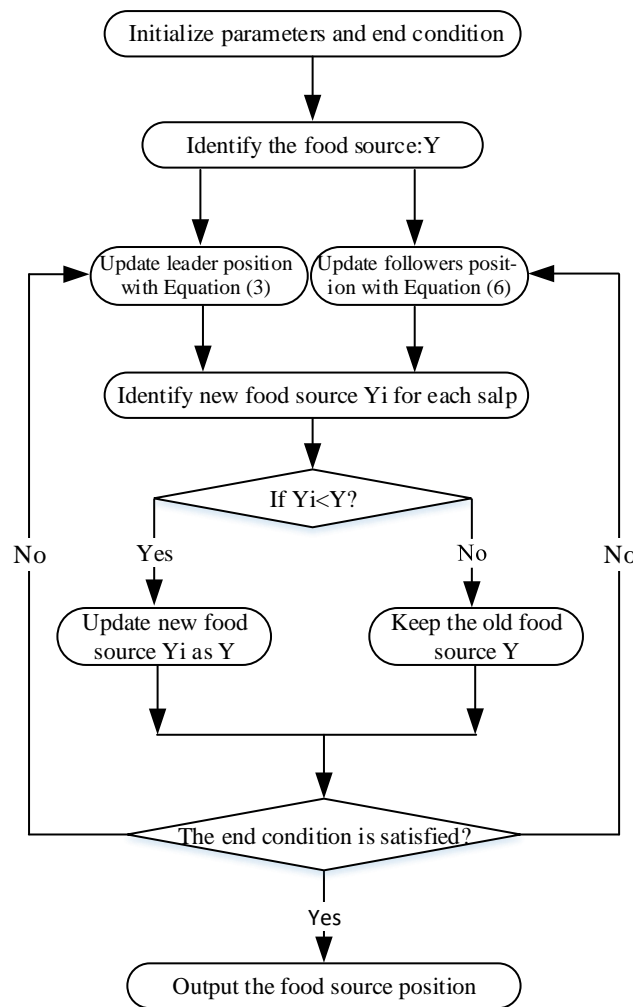


Figure 1. The flowchart of the salp swarm algorithm (SSA).

4. Estimation Algorithms and Dataset

In this paper, the parameter estimation of the van Genuchten Model using the SSA is performed. In order to show the universality of SSA, nine different soil samples that consist of eight soil textures are employed to conduct the experiment. Furthermore, the PSO and DE algorithms, as well as the RETC program, are utilized as benchmarks for comparison.

The DE algorithm is a type of evolutionary algorithms for solving the global optimal solution in multidimensional space [29]. In the DE algorithm, M parent vectors, each of which consists of j -dimensional vectors, are initialized as

$$x_{i,j} = L_i + r_j(U_i - L_i) \tag{9}$$

where r_j is randomly generated in $(0, 1)$. In the g th generation, three different individual vectors, x_{i1} , x_{i2} , and x_{i3} , are selected randomly to compute the mutant vector p_i :

$$p_i(g) = x_{i1}(g) + F [x_{i2}(g) - x_{i3}(g)] \tag{10}$$

where F is a constant factor in $[0, 2]$. Then the trial vector is formed:

$$u_{i,j}(g) = \begin{cases} p_{i,j}(g) & \text{if } (r_j \leq R_j) \text{ or } (j = R_m) \\ x_{i,j}(g) & \text{otherwise} \end{cases} \tag{11}$$

where R_j is a parameter specified by user which is a constant in $[0, 1)$, R_m is generated randomly in set $\{0, 1, \dots, M - 1, M\}$. In the process of vector updating, the vector in $x_{i,j}(g)$ and $u_{i,j}(g)$, which has a smaller fitness value for the objective function would be selected as the parent vector for the next generation. The iteration process ends until the set condition is satisfied.

Data Description

The data for this experiment are collected from the UNSODA database and can be obtained freely. The nine soil samples data recorded contain the soil water potential and the water content. More details of physical properties are represented in Table 1.

Table 1. The physical properties of nine soil samples.

Soil Sample ID	Location	Bulk Density	Data Number	Soil Type
3020	Moscow, Russia	1.21	5	Sand
1120	Rome, AL, USA	1.63	10	Sandy Loam
3154	Dickey Co., ND, USA	1.53	10	Sand
1330	Hannover, Germany	1.37	21	Silt
1173	Clemson, SC, USA	1.38	11	Clay Loam
1102	Blackville, SC, USA	1.71	9	Sandy Clay
1162	Watkinsville, GA, USA	1.54	15	Clay
1361	Reinhausen (Goettingen), Germany	1.49	11	Silty Clay
2400	Cass County, ND, USA	1.08	17	Loam

From Table 1, we can see that the data are different in type, bulk density, and location, and the number of data points covers from 5 to 21, which indicates that the effectiveness of the proposed method is validated with different datasets to examine the universality of this algorithm. The lower bound and upper bounds of parameters, φ_r , φ_s , α , and n of the van Genuchten Model are set and presented in Table 2.

Table 2. The lower and upper bounds of the four model parameters.

Parameter	φ_r	φ_s	α	n
Lower Bound	0	0	0	1
Upper Bound	1	1	100	100

5. Estimation Results and Discussion

To estimate the parameters of the van Genuchten Model efficiently using the SSA, the population size was set to 50 and the maximum number of iteration was 30,000. In this experiment, the end condition was that the current iteration reaches the maximum iteration. Furthermore, in order to search the set of parameters corresponding to each soil sample, data with different sizes were used, and the estimation procedure was repeated.

For the three benchmarking methods, the algorithm-specific parameters in DE and PSO were set according to [20,30], and a precompiled version RETC was applied. The source code of the SSA was written in Matlab, and a Windows 10 system with an Intel Core i5@3.20GHz CPU and 4GB memory was used to conduct the experiment. The computational results of the experiment are shown in Table 3.

Table 3. Estimated parameters and the sum of squared errors (SSE) of the van Genuchten model using SSA, differential evolution (DE), particle swarm optimization (PSO), and RETC.

Soil Sample ID	Algorithm	φ_r (cm ³ cm ⁻³)	φ_s (cm ³ cm ⁻³)	α (cm ⁻¹)	n	SSE (10 ⁻³)
3020	SSA	0.19166	0.44901	0.06244	2.40869	0.011171
	DE	0.19166	0.44901	0.01796	2.40869	0.011724
	RETC	0.19166	0.44901	0.01796	2.40885	0.011718
	PSO	0.19220	0.44900	0.01780	2.43342	0.011732
1120	SSA	0.07951	0.28940	0.02583	1.88130	0.261462
	DE	0.07951	0.28940	0.01238	1.88130	0.261464
	RETC	0.07951	0.28940	0.01239	1.88119	0.261465
	PSO	0.07951	0.28940	0.01238	1.88131	0.261463
3154	SSA	0.06941	0.41611	0.06706	2.69126	0.214173
	DE	0.06941	0.41611	0.02813	2.69128	0.214174
	RETC	0.06942	0.41609	0.02813	2.69178	0.214175
	PSO	0.06941	0.41611	0.02813	2.69131	0.214174
1330	SSA	0.08362	0.38004	0.00337	2.11588	11.25378
	DE	0.08362	0.38004	0.00259	2.11588	11.25378
	RETC	0.08373	0.37998	0.00259	2.11992	11.25386
	PSO	0.08344	0.38021	0.00260	2.10977	11.25391
1173	SSA	0.29087	0.47857	0.03335	1.14316	0.037139
	DE	0.30479	0.47850	0.03022	1.16004	0.037951
	RETC	0.30130	0.47850	0.04960	1.15530	0.499869
	PSO	0.03266	0.47909	0.05867	1.04872	0.041419
1102	SSA	0.12175	0.34674	0.09208	1.29730	0.240458
	DE	0.12198	0.34673	0.09168	1.29806	0.240460
	RETC	0.12170	0.34670	0.15910	1.29720	1.402827
	PSO	0.12183	0.34677	0.09208	1.29753	0.240460
1162	SSA	0.29374	0.41333	0.01369	1.31096	2.711846
	DE	0.29372	0.41334	0.01371	1.31075	2.711846
	RETC	0.29400	0.41330	0.03770	1.31270	4.622924
	PSO	0.29311	0.41346	0.01424	1.30620	2.711884
1361	SSA	0.16454	0.43307	0.00105	1.25767	0.266938
	DE	0.16448	0.43308	0.00106	1.25746	0.266939
	RETC	0.16450	0.43319	0.00430	1.25760	10.92369
	PSO	0.16128	0.43329	0.00112	1.25138	0.267039
2400	SSA	0.19672	0.45414	0.00139	1.58303	0.144518
	DE	0.19633	0.45419	0.00139	1.58259	0.144576
	RETC	0.19670	0.45410	0.00157	1.58310	0.418468
	PSO	0.19455	0.45444	0.00147	1.57232	0.144657

As shown in Table 3, we can see that the SSA performs better than the three benchmarking methods for Soil Samples 3020, 1120, 3154, 1102, 1361, 1173, and 2400, which indicates that SSA can find more accurate parameter sets in the van Genuchten model. However, it is observable that the SSA has the same SSE value with DE for Soil Samples 1330 and 1162. Meanwhile, it can be seen that the RETC program shows a good performance for Soil Samples 3020, 1120, 3154, and 1330, but it has poor performance for the remaining soil samples, especially Soil Sample 1361, where the SSE of RETC is 1.092369×10^{-2} . For PSO, its performance is similar to that of DE. The results of the nine soil samples show the superiority of heuristic algorithms compared with the NLS method for the SWRC parameter estimation problem.

In order to illustrate the difference between the measured data and the estimated data, Figure 2 shows the details of SWRCs of all eight soil samples corresponding to the eight soil textures.

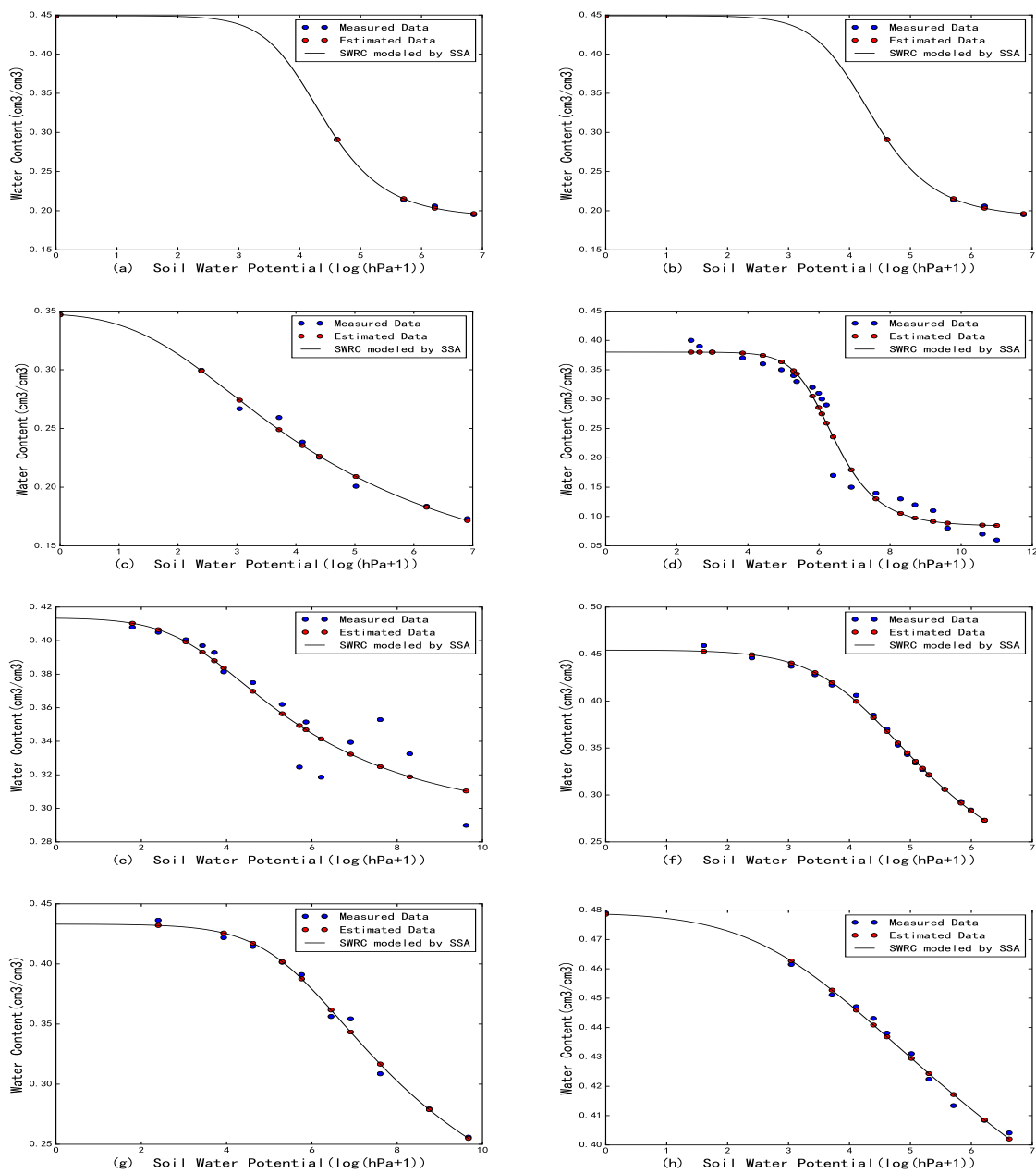


Figure 2. The estimated SWRCs of eight soil samples using SSA. (a) Sample 3020, (b) Sample 1120, (c) Sample 1102, (d) Sample 1330, (e) Sample 1162, (f) Sample 2400, (g) Sample 1361, and (h) Sample 1173.

Figure 2 visualizes the estimation results, and it can be observed that the estimated curves with the modeled data can fit the measured data accurately for Samples 3020, 1120, 2400, and 1173 (see in Figures 2a,b,f,h), which have fewer data points than the rest of the soil samples. Correspondingly, it is observable that, for Soil Samples 1330 and 1162, which have more data points, the SWRC cannot perform as well as it can for Soil Samples 3020 and 1120. From the values of SSEs recorded in Table 2 and the eight subgraphs, it can be observed that the SSA can estimate parameters of the van Genuchten model accurately even with a smaller dataset. Apart from the lower and upper bounds of the set of parameters need to be estimated, the SSA does not have any other model-specified parameters. Due to this advantage, the SSA is applicable to estimate parameters of the van Genuchten model simulating the SWRC.

It is not practical to perform the parameter estimation with a normal computer in the field. Thus, a portable computing device is a better option. To verify the performance of the SSA on

embedded devices, we implemented the parameter estimation algorithm on Raspberry Pi 3 using the Octave language. The hardware specification of the Raspberry Pi 3 platform was a 64-bit quad core processor, with on-board WIFI, Bluetooth, and USB boot capabilities, and 1 GB of memory. The lower and upper bounds, population size, and maximum iteration were set to the same value as those on the Windows 10 system in the verification experiment. Soil Samples 2400, 3020, 1361, 1330, and 1102 were selected, and the estimation SSEs are given in Table 4.

Table 4. SSEs of SSA on Raspberry Pi 3 and a Windows 10 system.

Soil Sample Id	SSE on Raspberry Pi 3 (10^{-3})	SSE on Windows 10 System (10^{-3})
2400	0.144518	0.144518
3020	0.011171	0.011171
1361	0.266938	0.266938
1102	0.240458	0.240458
1330	11.25378	11.25378

From Table 4, the estimated SSEs are the same as those on the Windows system. The results show that the SSA can perform well on the Raspberry Pi 3, a embedded computing device, which makes the algorithm applicable on portable devices in the field.

6. Conclusions

A new heuristic algorithms SSA was utilized to estimate the parameters of the van Genuchten model, and nine soil samples, covering eight soil textures, were collected to validate the universality of the SSA.

In the nine estimation experiments, the SSA performed better than the three benchmarking methods, including DE and PSO algorithms, and the RETC program. The SWRC using parameters estimated by SSA based on the van Genuchten model could fit the measured data well and provided minimum values of SSE among all the algorithms. Furthermore, by comparing the estimated SSEs on both Raspberry Pi 3 and Windows 10 platforms, the SSA can obtain the same results, so it is suitable for the embedded implementation.

Author Contributions: Conceptualization, Z.W. and X.L.; Methodology, J.Z.; Writing, J.Z.

Funding: This research was funded by [Guangdong Provincial Key Laboratory of New and Renewable Research and Development] grant number [Y807s61001], [National Key Research and Development Program of China] grant number [2016YFC0600510], [University of Science and Technology Beijing - National Taipei University of Technology Joint Research Program] grant number [TW2018008], and [Fundamental Research Funds for the Central Universities] grant number [06500078].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Leong, E.; Rahardjo, H. Review of Soil-Water Characteristic Curve Equations. *J. Geotech. Geoenviron. Eng.* **1997**, *123*. [[CrossRef](#)]
2. Hosseini, S.M.M.M.; Ganjian, N.; Pisheh, Y.P. Estimation of the water retention curve for unsaturated clay. *Can. J. Soil Sci.* **2011**, *91*, 543–549. [[CrossRef](#)]
3. Too, V.K.; Omuto, C.T.; Biamah, E.K.; Obiero, J.P. Review of Soil Water Retention Characteristic (SWRC) Models between Saturation and Oven Dryness. *Open J. Mod. Hydrol.* **2014**, *4*, 173–182. [[CrossRef](#)]
4. Brooks, R.H.; Corey, A.T. Hydraulic properties of porous media and their relation to drainage design. *Trans. ASAE* **1964**, *7*, 26–0028. doi:10.13031/2013.40684. [[CrossRef](#)]
5. Gardner, W.R.; Hillel, D.; Benyamini, Y. Post-Irrigation Movement of Soil Water 2. Simultaneous Redistribution and Evaporation. *Water Resour. Res.* **1970**, *6*, 1148–1153. [[CrossRef](#)]

6. Van Genuchten, M. A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Sci. Soc. Am. J.* **1980**, *44*, 892. [[CrossRef](#)]
7. Campbell, G. A Simple Method for Determining Unsaturated Conductivity From Moisture Retention Data. *Soil Sci.* **1974**, *117*, 311–314. [[CrossRef](#)]
8. Omuto, C. Biexponential model for water retention characteristics. *Geoderma* **2009**, *149*, 235–242. [[CrossRef](#)]
9. Fredlund, D.G.; Xing, A. Equations for the Soil–Water Characteristic Curve. *Can. Geotech. J.* **1994**, *31*, 521–532. [[CrossRef](#)]
10. Guangzhou Chen, L.J.; Li, X. Sensitivity Analysis and Identification of Parameters to the Van Genuchten Equation. *J. Chem.* **2016**, *2016*. [[CrossRef](#)]
11. Luo, X.; Zhang, D.; Yang, L.T.; Liu, J.; Chang, X.; Ning, H. A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems. *Future Gener. Comput. Syst.* **2016**, *61*, 85–96. [[CrossRef](#)]
12. Luo, X.; Deng, J.; Wang, W.; Wang, J.H.; Zhao, W. A Quantized Kernel Learning Algorithm Using a Minimum Kernel Risk-Sensitive Loss Criterion and Bilateral Gradient Technique. *Entropy* **2017**, *19*. doi:10.3390/e19070365. [[CrossRef](#)]
13. Van Genuchten, M.V.; Leij, F.J.; Yates, S.R. *The RETC Code for Quantifying Hydraulic Functions of Unsaturated Soils*; EPA/600/2-91/065, R.S.; U.S. Environmental Protection Agency: Ada, OK, USA, 1991; Volume 83.
14. Luo, X.; Cao, L.; Wang, L.; Zhao, Z.; Huang, C. Parameter Identification of the Photovoltaic Cell Model with a Hybrid Jaya-NM Algorithm. *Optik* **2018**. [[CrossRef](#)]
15. Li, Y.B.; Liu, Y.; Nie, W.B.; Ma, X.Y. Inverse Modeling of Soil Hydraulic Parameters Based on a Hybrid of Vector-Evaluated Genetic Algorithm and Particle Swarm Optimization. *Water* **2018**, *10*, 84. [[CrossRef](#)]
16. Xing, X.; Wang, H.; Ma, X. Brooks–Corey Modeling by One-Dimensional Vertical Infiltration Method. *Water* **2018**, *10*, 593. [[CrossRef](#)]
17. Oh, S.; Kim, Y.K.; Kim, J.W. A Modified van Genuchten-Mualem Model of Hydraulic Conductivity in Korean Residual Soils. *Water* **2015**, *7*, 5487–5502. [[CrossRef](#)]
18. Wang, L.; Huang, C.; Huang, L. Parameter Estimation of the Soil Water Retention Curve Model with Jaya Algorithm. *Comput. Electron. Agric.* **2018**, *151*, 349–353. [[CrossRef](#)]
19. Matula, S.; Miháliková, M.; Batkova, K. Estimation of the Soil Water Retention Curve (SWRC) Using Pedotransfer Functions (PTFs). *Soil Water Res.* **2007**, *2*, 113–122. [[CrossRef](#)]
20. Maggi, S. Estimating water retention characteristic parameters using differential evolution. *Comput. Geotech.* **2017**, *86*, 163–172. [[CrossRef](#)]
21. Bayat, H.; Zadeh, G.E. Estimation of the soil water retention curve using penetration resistance curve models. *Comput. Electron. Agric.* **2018**, *144*, 329–343. [[CrossRef](#)]
22. Do Nascimento, Í.V.; de Assis Júnior, R.N.; de Araújo, J.C.; de Alencar, T.L.; Freire, A.G.; Lobato, M.G.R.; da Silva, C.P.; Mota, J.C.A.; do Nascimento, C.D.V. Estimation of van Genuchten Equation Parameters in Laboratory and through Inverse Modeling with Hydrus-1D. *J. Agric. Sci.* **2018**, *10*, 102. [[CrossRef](#)]
23. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 16–191. [[CrossRef](#)]
24. Sayed, G.I.; Khoriba, G.; Haggag, M.H. A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl. Intell.* **2018**. [[CrossRef](#)]
25. Luo, X.; Deng, J.; Liu, J.; Wang, W.; Ban, X.; Wang, J.H. A quantized kernel least mean square scheme with entropy-guided learning for intelligent data analysis. *China Commun.* **2017**, *14*, 127–136. [[CrossRef](#)]
26. Luo, X.; Jiang, C.; Wang, W.; Xu, Y.; Wang, J.H.; Zhao, W. User behavior prediction in social networks using weighted extreme learning machine with distribution optimization. *Future Gener. Comput. Syst.* **2018**. doi:10.1016/j.future.2018.04.085. [[CrossRef](#)]
27. Luo, X.; Xu, Y.; Wang, W.; Yuan, M.; Ban, X.; Zhu, Y.; Zhao, W. Towards enhancing stacked extreme learning machine with sparse autoencoder by correntropy. *J. Frankl. Inst.* **2018**, *355*, 1945–1966. [[CrossRef](#)]
28. Luo, X.; Yang, X.; Jiang, C.; Ban, X. Timeliness online regularized extreme learning machine. *Int. J. Mach. Learn. Cybern.* **2018**, *9*, 465–476, doi:10.1007/s13042-016-0544-9. [[CrossRef](#)]

29. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359, doi:10.1023/A:1008202821328. [[CrossRef](#)]
30. Wang, L.; Zhang, Z.; Huang, C.; Tsui, K.L. A GPU-accelerated parallel Jaya algorithm for efficiently estimating Li-ion battery model parameters. *Appl. Soft Comput.* **2018**, *65*, 12–20. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).