Applied Mathematics and Computation 217 (2010) 3207–3215

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Improved particle swarm algorithm for hydrological parameter optimization

Yan Jiang [a,*], Changmin Liu [b], Chongchao Huang [c], Xianing Wu [d]

[a] State Key Laboratory of Urban and Regional Ecology, Research Center for Eco-Environmental Sciences, Chinese Academy of Sciences, Beijing 100085, China
[b] College of Water Sciences, Beijing Normal University, Beijing 100875, China
[c] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
[d] Sinohydro Corporation Limited, Beijing 100044, China

ARTICLE INFO

Keywords:
Global optimization
Particle swarm optimization
Multi-swarms shuffling evolutionary
Hydrological model
Parameter optimization

ABSTRACT

In this paper, a new method named MSSE-PSO (master–slave swarms shuffling evolution algorithm based on particle swarm optimization) is proposed. Firstly, a population of points is sampled randomly from the feasible space, and then partitioned into several sub-swarms (one master swarm and other slave swarms). Each slave swarm independently executes PSO or its variants, including the update of particles' position and velocity. For the master swarm, the particles enhance themselves based on the social knowledge of master swarm and that of slave swarms. At periodic stage in the evolution, the master swarm and the whole slave swarms are forced to mix, and points are then reassigned to several sub-swarms to ensure the share of information. The process is repeated until a user-defined stopping criterion is reached. The tests of numerical simulation and the case study on hydrological model show that MSSE-PSO remarkably improves the accuracy of calibration, reduces the time of computation and enhances the performance of stability. Therefore, it is an effective and efficient global optimization method.

1. Introduction

Hydrological model has been widely applied in hydrology such as planning, design, operation, management, decision and so on. Normally, a hydrological model consists of some modules with a large number of parameters. The successful application of a hydrological model depends on how well the parameters are calibrated. Theoretically they can be assigned values from actual data, but actually it is almost impossible because of temporal and spacial variation of complicated hydrological process. Instead, an inverse problem is solved in which the parameters are optimized by fitting as closely as possible the simulation to the observation. However, affected by weather, climate, terrain, soil, vegetation and human activities, the parameter optimization of a hydrological model is in possession of some characters such as high-dimension, multi-peak values, nonlinear, discontinuous and non-convex and noisy etc., which makes it difficult to be calibrated exactly.

The traditional methods to calibrate hydrological model include Rosenbrock method [1], simplex method [2], genetic algorithm (GA) [3–6] and SCE-UA [7–12]. Both Rosenbrock method and simplex method demand a lot on model structure, and they always converge to local optimal solution. GA is a highly parallel and adaptive optimization algorithm, but it also suffers from premature convergence and tends to get stuck into local optima, especially in complex multi-peak-search problems. SCE-UA has strong global search ability by introducing biologic evolvement to its numerical compute, but different

* Corresponding author.
E-mail address: yanjiang@rcees.ac.cn (Y. Jiang).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2010.08.053

sub-complex approaches to global optimal position from different directions and simplex method is used as the search method in each sub-complex, which reduces the computational efficiency.

Particle swarm optimization (PSO) is a novel global optimization and evolutionary algorithm introduced in 1995 by Eberhart and Kennedy [13,14], who were inspired by the social behavior of animals such as fish schooling and bird flocking. PSO has gained popularity lately and has been applied to wide applications in different fields [15–19]. Since the frame of this algorithm is relatively simple and the operation speed is very fast, PSO has also been applied in hydrological parameter optimization [20,21]. However, it points out that although PSO shows significant performance in the initial iterations, there still exist some problems. When a particle in the swarm finds its current optimal position, the other particles will gather close to it rapidly. If the position is a local optimum, the particle swarm will not be able to search over again in the solution space. Hence the algorithm plunges into local optima.

Therefore, many researchers are devoted into this field to handle its premature convergence. Inspired by GA, Shi et al. [22] present a hybrid evolutionary algorithm based on PSO and GA methods through crossing over PSO and GA, which possess better ability to find the global optimum than the standard PSO algorithm. Wang and Li [23] integrate PSO and simulated annealing to improve the performance of PSO. Inspired by the phenomenon of symbiosis in natural ecosystem, Niu et al. [24] incorporate master–slave mode into PSO and presents a multi-population cooperative particle swarm optimization (MCPSO). Shelokar et al. [25] propose an improved particle swarm optimization hybridized with an ant colony approach, called PSACO (particle swarm ant colony optimization), for optimization of multi-modal continuous functions. The proposed method applies PSO for global optimization and the idea of ant colony approach to update positions of particles to attain rapidly the feasible solution space. Ali and Kaelo [26] study the efficiency and robustness of a number of particle swarm optimization algorithms and identify the cause for their slow convergence. And then propose some modifications in the position update rule of particle swarm optimization algorithm in order to make the convergence faster. Jiang et al. [27] propose an improved PSO algorithm named PSSE-PSO (parallel swarms shuffling evolution algorithm based on particle swarm optimization) for hydrological parameters calibration, by introducing the ideas of population division and biological evolution into the standard PSO to avoid premature convergence. In this paper, we present MSSE-PSO (master–slave swarms shuffling evolution algorithm based on particle swarm optimization) to improve the performance of PSSE-PSO in accordance with the idea of hierarchical evolution. The initialized populations are distributed with different regions of the search space of the entire feasible solution to maintain the diversity of the individuals. They are then partitioned into several sub-swarms (one master swarm and other slave swarms). Each slave swarm executes PSO or its variants, including the update of particles' position and velocity, and the creation of a new local population. When all slave swarms are ready with the new generations, each of them then sends the best local individual to the master one. The master swarm selects the best one of all received individuals and evolves based on the social knowledge of master swarm and the whole slave swarms. At periodic stage in the evolution, they are forced to mix, and points are then reassigned to sub-swarms to ensure the share of information. The process is repeated until a user-defined stopping criterion is reached. Finally, the standard PSO and PSSE-PSO are used for comparison to evaluate the optimization performance of the proposed algorithm MSSE-PSO with tests of numerical simulation and case study on parameter optimization on hydrological model.

This paper is organized as follows: Section 2 presents a review of the standard PSO. Description of the proposed algorithm MSSE-PSO is given in Section 3. Experiment used to illustrate the optimization performance of MSSE-PSO is given in Section 4. Case study on hydrological parameter optimization is given in Section 5. Finally, Section 6 concludes this paper.

2. Overview of the standard PSO

Particle swarm optimization (PSO) is a stochastic, population-based algorithm for solving optimization problems, which is initialized with a group of random particles (solutions) and then searches for optima by updating generations. Suppose that the search space is D-dimensional, and then the particle i of the swarm can be represented by a D-dimensional vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})^T$. The velocity of this particle can be represented by another D-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})^T$. The best previously visited position of the particle i is noted as its individual best position $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})^T$. The position of the best individual of the whole swarm is noted as the global best position $G = (g_1, g_2, \ldots, g_D)^T$. At each step, the velocity of particle and its new position will be assigned according to Eqs. (1) and (2). This process is repeated until a user-defined stopping criterion is reached.

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_d^t - x_{id}^t), \tag{1}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \tag{2}$$

where, t means the current iterative step. $i = 1, 2, \ldots, N$; $d = 1, 2, \ldots, D$, N is the numbers of the total populations. r_1 and r_2 are independently uniformly distributed random variables with range $[0, 1]$. c_1 and c_2 are the acceleration constants. ω is the constriction factor.

3. Master–slave swarms shuffling evolution based on PSO (MSSE-PSO)

For MSSE-PSO, a population of points is sampled randomly from the feasible space. Then the population is partitioned into several sub-swarms (one master swarm and other slave swarms). Each slave swarm independently executes PSO or its variants, including the update of particles' position and velocity. The master swarm enhances the particles based on not only the social knowledge of the master swarm but also that of the slave swarms. This idea is realized by further incorporating a new dimension into the velocity of particles in standard PSO. The resulting equations for the manipulation of the master swarm are:

$$v_{id}^{t+1} = \omega v_{id}^{t} + c_1 r_1 (p_{id}^{t} - x_{id}^{t}) + c_2 r_2 (g_{d}^{t} - x_{id}^{t}) + c_3 r_3 (s_{d}^{t} - x_{id}^{t}), \tag{3}$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}, \tag{4}$$

where c_3 is a positive parameter called migration coefficient, and r_3 is a uniform random variable in the range [0,1]. s_d is the best position of the whole slave swarms so far. As shown in Eq. (3), the first term of summation represents the inertia (the particle keeps moving in the direction it had previously moved), the second term represents memory (the particle is attracted to the best point in its trajectory), the third term represents cooperation (the particle is attracted to the best point found by all particles of master swarm) and the last represents information exchange (the particle is attracted to the best point found by the slave swarms). After a certain number of generations, the master swarm and the whole slave swarms are forced to mix, and then points are reassigned to ensure information sharing. In order to ensure the optimization accuracy, generation is terminated if one of the following three criteria is reached. Criterion 1: the parameters distance corresponding to the optimal objective function value for two iterations is less than a given accuracy. Criterion 2: the difference of the optimal objective function value for two iterations is less than a given accuracy. Criterion 3: iteration has already reached the maximum number of generations. The strategy of MSSE-PSO is presented below and illustrated in Fig. 1.

Step 1: Initializing. Assign M and N ($M \geqslant 1, N \geqslant 1$), where M is the number of sub-swarms (one master swarm and $M-1$ slave swarms) and N is the number of points in each sub-swarm. Compute the size of sample $S = M \times N$. Sample S points $X_1, X_2, \ldots, X_S$ in the feasible space. Compute the function value f_i for each point X_i, $i = 1, 2, \ldots, S$.

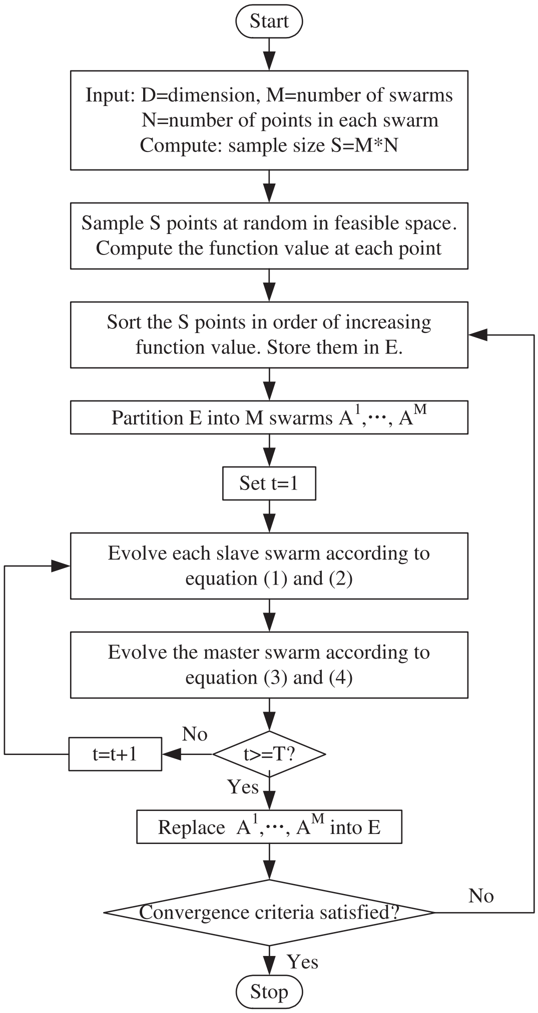

Fig. 1. The flow chart of MSSE-PSO.

Step 2: Ranking. Arrange these points by the ascending order of function value and store them in an array $E = \{X_i, f_i, i = 1, 2, \ldots, S\}$.
Step 3: Partitioning. Partition E into M sub-swarms $A^1, A^2, \ldots, A^M$, each of which contains N particles, like this: $A^k = \{X_j^k, f_j^k | X_j^k = X_{k+p(j-1)}, f_j^k = f_{k+M(j-1)}, j = 1, 2, \ldots, N\}$.
Step 4: Evolving. Each slave swarm adapts itself according to Eqs. (1) and (2) independently. The master swarm adapts itself according to Eqs. (3) and (4).
Step 5: Iterating. Iterate by repeating **Step 4** T times, where T is a user-specified parameter which determines how fast each sub-swarm should evolve.
Step 6: Shuffling. Replace $A^1, A^2, \ldots, A^M$ into E. Arrange E by the ascending order of function value.
Step 7: Convergence checking. If the convergence criteria are satisfied, stop. Otherwise, return to **Step 3**.

Similar to PSSE-PSO [27], MSSE-PSO combines the strengths of the particle swarm optimization, competitive evolution and sub-swarm shuffling, which greatly enhances survivability by sharing the information gained independently by each swarm. Besides, MSSE-PSO adopts the hierarchical idea, by which the master swarm guides the whole group to the optimal direction to control the balance between exploration and exploitation.

4. The test of numerical simulation

In this section, four nonlinear benchmark functions that are commonly used in literature [28] are performed for test. Each function is designed to have minimum in the region. Name, formula, dimension, variable range, optimum and goal value of each function are listed in Table 1.

To evaluate the optimization performance of the proposed algorithm MSSE-PSO, the standard PSO and PSSE-PSO are used for comparison. For those PSO-type algorithms above, the parameter used is recommended from [29] with a linearly decreasing, which changes from 0.9 to 0.4. The acceleration constants c_1, c_2 are both 2.0 for PSO and PSSE-PSO. The acceleration constants $c_1 = c_2 = 2.05$ and migration coefficient $c_3 = 0.8$ are used in MSSE-PSO. For each optimization experiment, x and v are randomly initialized in the range $[x_{min}, x_{max}]$ shown in Table 1. Population size of particles is assigned by $S = 40$, 80, 120 in turn for test. The maximum number of generations Max_T is set as 3000. For PSSE-PSO and MSSE-PSO, $M = 4$, $N = S/M$, $T = D$ are used as default, where M is the number of swarms, N is the number of points in each sub-swarm, T is the number of evolution steps taken by each sub-swarm and D is the dimension size.

In order to avoid the influence of randomicity, 20 runs are conducted for each algorithm. In the test, the goal value of each function is recorded and the average, maximum and minimum function value are calculated. If the goal is not reached within the maximum number of generations, the run is considered unsuccessful. In different initial conditions, we use the standard deviation of function values to characterize the randomicity of the three algorithms. The smaller the standard deviation is, the stabler the algorithm is. At the same time Wilcoxon Rank Sum Test is used to compare the results of PSO, PSSE-PSO and MSSE-PSO to evaluate the significance. For these three algorithms, the results of the 20 independent runs form independent samples. For any two algorithms (A and B), the distribution of their results (F_A and F_B) are compared by means of the null hypothesis H_0: $F_A = F_B$ and the alternative H_0: $F_A < F_B$. The tests are made at a significance level of $\alpha = 0.01$ and the significance comparison between two algorithms is displayed by a 3×3 matrix $A = (a_{ij})_{i,j \in [1,3]}$, in which an entry "+" at position a_{ij} denotes that algorithm i is significantly better than algorithm j, an entry "−" at position a_{ij} denotes that algorithm i is not significantly better than algorithm j, and the entries a_{ij} on the main diagonal are left empty. We call the corresponding matrix a significance matrix (S-matrix).

The performances of PSO, PSSE-PSO and MSSE-PSO are shown in Tables 2–5. In most cases, increasing the number of particles can decrease the maximum, minimum and average function values. Whatever the population size is, the success rate (the probability of algorithm's finding the global optimal solution) of MSSE-PSO is superior to those of PSO. And the function values obtained by MSSE-PSO are smaller than those of PSO and PSSE-PSO. Table 6 lists standard deviation of function values and mean CPU time by 20 independent runs of these three algorithms for each function. It shows that MSSE-PSO has stronger stability and can rapidly converge to the global optimal solution in a short time. The main reason is that for PSSE-PSO and MSSE-PSO, the strategy of swarm separation is used to maintain the diversity of population and the introduction of competition evolution can enhance the adaptability of population. Moreover for MSSE-PSO, the master swarm which enhances itself based on the social knowledge of the master swarm and that of the slave swarms can guide the whole group to optimal

Table 1
Benchmark functions for test.

Name	Formula	Dim	Range	Opt.	Goal
Sphere	$f_0(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0	0.01
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$	30	$[-30,30]^n$	0	100
Rastrigrin	$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12,5.12]^n$	0	100
Griewank	$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600,600]^n$	0	0.1

Table 2
The computational results for sphere function.

Size	Algorithm	Max.	Min.	Avg.	Success rate (%)	*S*-Matrix		
						1	2	3
40	PSO	9.94	2.03E−04	1.82	50		–	–
	PSSE-PSO	8.87E−06	1.95E−08	2.88E−06	100	+		–
	MSSE-PSO	4.34E−34	0	4.17E−35	100	+	+	
80	PSO	0.642	8.90E−05	0.093	50		–	–
	PSSE-PSO	3.67E−12	2.50E−13	1.03E−12	100	+		–
	MSSE-PSO	3.63E−44	0	1.40E−45	100	+	+	
120	PSO	5.57E−05	1.67E−08	8.83E−06	100		–	–
	PSSE-PSO	1.29E−13	4.25E−14	7.02E−14	100	+		–
	MSSE-PSO	0	0	0	100	+	+	

Table 3
The computational results for Rosenbrock function.

Size	Algorithm	Max.	Min.	Avg.	Success rate (%)	*S*-Matrix		
						1	2	3
40	PSO	488.22	8.19	161.54	55		–	–
	PSSE-PSO	2.89	4.33E−06	0.308	100	+		–
	MSSE-PSO	1.48	2.28	0.249	100	+	+	
80	PSO	53.6	0.105	10.1	100		–	–
	PSSE-PSO	0.0542	8.57E−07	0.01	100	+		–
	MSSE-PSO	1.55E−06	0	7.08E−09	100	+	+	
120	PSO	0.276	1.88E−09	0.028	100		–	–
	PSSE-PSO	1.54E−05	1.58E−10	1.57E−06	100	+		–
	MSSE-PSO	0	0	0	100	+	+	

Table 4
The computational results for Rastrigrin function.

Size	Algorithm	Max.	Min.	Avg.	Success rate (%)	*S*-Matrix		
						1	2	3
40	PSO	150.07	54.94	105.63	40		–	–
	PSSE-PSO	27.86	1.24	20.94	100	+		–
	MSSE-PSO	20	0	5.76	100	+	+	
80	PSO	120.49	4.09	60.50	85		–	–
	PSSE-PSO	16.91	1.17E−08	5.04	100	+		–
	MSSE-PSO	3.98	0	5.76	100	+	+	
120	PSO	93.74	1.53E−05	28.74	100		–	–
	PSSE-PSO	8.95	1.28E−13	2.64	100	+		–
	MSSE-PSO	0	0	0	100	+	+	

Table 5
The computational results for Griewank function.

Size	Algorithm	Max.	Min.	Avg.	Success rate (%)	*S*-Matrix		
						1	2	3
40	PSO	1.97	0.009	1.22	5		–	–
	PSSE-PSO	0.039	7.34E−04	0.013	100	+		+
	MSSE-PSO	0.084	0	0.022	100	+	–	
80	PSO	0.143	1.16E−04	0.038	95		–	–
	PSSE-PSO	7.82E−05	2.23E−12	1.91E−05	100	+		+
	MSSE-PSO	0.061	0	0.007	100	+	–	
120	PSO	0.064	2.41E−06	0.008	100		–	–
	PSSE-PSO	5.17E−12	2.55E−15	3.89E−13	100	+		–
	MSSE-PSO	0	0	0	100	+	+	

Table 6
The standard deviation of function values and CPU time for PSO, PSSE-PSO and MSSE-PSO.

Function	Size	Standard deviation of function values			CPU (s)		
		PSO	PSSE-PSO	MSSE-PSO	PSO	PSSE-PSO	MSSE-PSO
Sphere	40	2.583	2.67E−06	1.21E−34	20	16	24
	80	0.178	9.1E−13	0	66	40	38
	120	1.57E−05	2.08E−14	0	137	68	80
Rosenbrock	40	164.709	0.71	0.434	28	40	26
	80	12.688	0.019	3.46E−07	89	84	61
	120	0.063	0.000	0	194	136	104
Rastrigrin	40	28.485	7.919	5.274	24	24	16
	80	28.205	5.045	1.308	83	52	43
	120	27.561	2.647	0	158	92	78
Griewank	40	0.427	0.01	0.02	27	32	21
	80	0.039	2.78E−05	0.016	86	64	47
	120	0.016	1.15E−12	0	160	108	89

direction, which reduces the effect of randomicity and enhances the stability to some extent. Figs. 2–5 show the average fitness by 80 particles for each function respectively. From the Figures, MSSE-PSO outperforms PSO and PSSE-PSO significantly for almost all the cases. Therefore, the proposed algorithm can be considered efficient and effective.

5. Application of MSSE-PSO to calibrate the Xinanjiang model

The Xinanjiang model [30], developed in 1973 by Prof. Zhao, is a conceptual hydrological model with 15 parameters needed to be calibrated. The range of these parameter values and their physical meaning are given in Table 7. In the Xinanjiang model, the basin is divided into a set of sub-basins, the outflow hydrograph from each of which is first simulated and then routed down the channels to the main basin outlet.

Objective function is used to evaluate the fitting degree between measured and simulated discharge. Different objective function describes the different characteristics of hydrological processes. The function expressed as formula (5) is selected as the objective function, which is designed to have the minimum.

$$f(x) = \frac{1}{N}\sum_{i=1}^{N}(Q_{obs,i} - Q_{sim,i})^2\left(1 + \frac{|\overline{Q}_{obs} - \overline{Q}_{sim}|}{\overline{Q}_{obs}}\right), \tag{5}$$

where, the variable x is parameters needed to be calibrated. $Q_{obs,i}$ and $Q_{sim,i}$ are the measured and simulated discharge in the period of calibration respectively. $\overline{Q}_{obs}$ and $\overline{Q}_{sim}$ are the average value of measured and simulated discharge in the period of calibration respectively. N is the number of time series.

Two statistical indices are selected to evaluate the performance of PSO, PSSE-PSO and MSE-PSO, which are model coefficient $R^2 : R^2 = 1 - \frac{\sum_{i=1}^{N}(Q_{obs,i}-Q_{sim,i})^2}{\sum_{i=1}^{N}(Q_{obs,i}-\overline{Q}_{obs})^2}$ and water balance error $RE : RE = \frac{|\overline{Q}_{obs}-\overline{Q}_{sim}|}{\overline{Q}_{obs}} \times 100\%$. When water balance error is smaller and model coefficient is bigger, it means that simulated discharges are closer to observed discharges.

Compared with PSO and PSSE-PSO, the performance of MSSE-PSO for the Xinanjiang model calibration is tested. The range of parameter values in the model is given in Table 7 and the objective function is expressed as formula (5). The studied area is the catchment of Tianfumiao reservoir in Hubei province of China. Data used for calibration include five

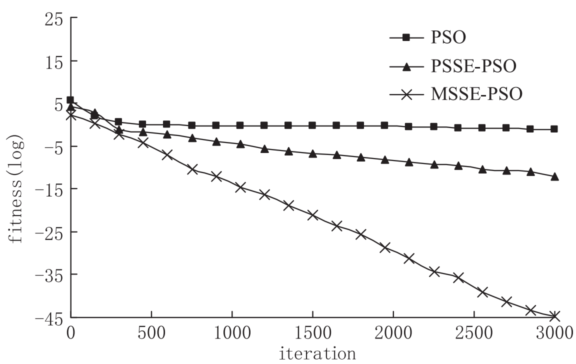

Fig. 2. Evolution of logarithmic average fitness of Sphere function for PSO, PSSE-PSO and MSSE-PSO.

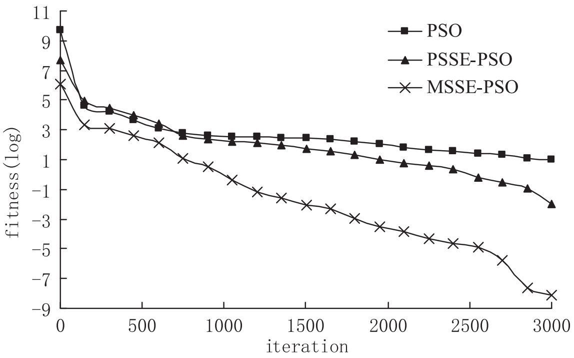

Fig. 3. Evolution of logarithmic average fitness of Rosenbrock function for PSO, PSSE-PSO and MSSE-PSO.

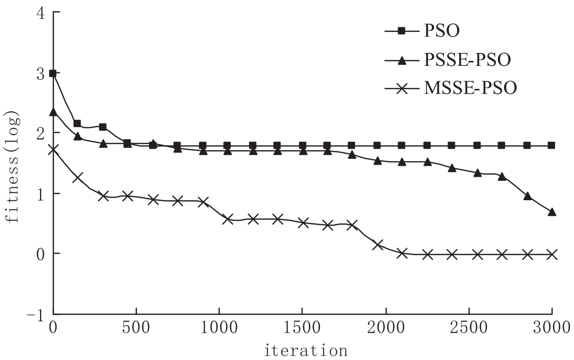

Fig. 4. Evolution of logarithmic average fitness of Rastrigrin function for PSO, PSSE-PSO and MSSE-PSO.

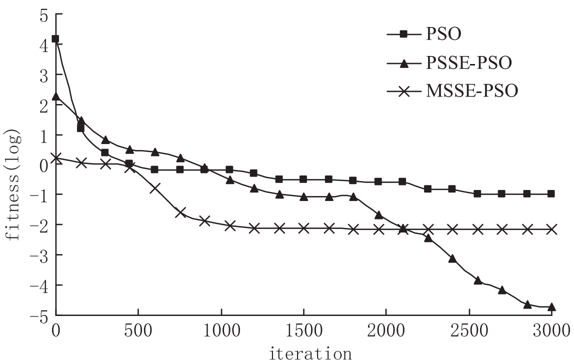

Fig. 5. Evolution of logarithmic average fitness of Griewank function for PSO, PSSE-PSO and MSSE-PSO.

years of daily rainfall, evapotranspiration and runoff, starting from January 1st 1988. Moreover, two years of data are used for verification, starting from January 1st 1995. Population size is assigned by 80. In order that the randomizer generates different initial starting populations of points, 5 runs are made with different initial seeds. Each run consists of 3000 objective function evaluations. The average value of each parameter for 5 runs are shown in Table 7. Table 8 Compares the performance for PSO, PSSE-PSO and MSSE-PSO and Fig. 6 shows the average function values of the best particles for PSO, PSSE-PSO and MSSE-PSO.

From Table 8 and Fig. 6, it can be seen that PSO, PSSE-PSO and MSSE-PSO can control water balance error in the allowed range (less than 5%), but MSSE-PSO has higher calibration accuracy and computational efficiency than PSO and PSSE-PSO. Therefore, MSSE-PSO performs better than PSO and PSSE-PSO for the parameter calibration of Xinanjiang model.

Table 7
Range of parameter values and their physical meanings.

Parameter	Physical meanings	Mm[a]	Max[b]	Avg.[c]
WM (mm)	The areal mean tension water capacity	80	170	100.532
WUM (mm)	The mean tension water capacities of the upper soil layer	5	20	5
WLM (mm)	The mean tension water capacities of the lower soil layer	60	90	60
K	The ratio of potential to pan evapotranspiration	0.2	1.5	0.55
IMP (%)	The ratio of impervious area to the total area of the basin	0.01	0.05	0.01
B	The exponential of distribution water capacity	0.1	0.4	0.109
C	The evapotranspiration coefficient of deeper layer	0.1	0.2	0.181
SM (mm)	The free water storage capacity	10	50	41.144
EX	The exponential of distribution water capacity	0.5	2	0.501
KG	The outflow coefficient of freewater storage to the groundwater flow	0.01	0.7	0.187
KSS	The outflow coefficient of free water storage to the interflow	0.01	0.7	0.513
KKG	The recession constant of ground water storage	0.95	0.998	0.990
KKSS	The recession constant of lower interflow storage	0.5	0.9	0.5
KE	Muskingum coefficient	1	10	1.917
XE	The residence time of water	1	10	8.955

[a] The minimal parameter value.
[b] The maximal parameter value.
[c] The average parameter value for calibrating for 5 runs.

Table 8
Comparison of performance for PSO, PSSE-PSO and MSSE-PSO.

Algorithm	Avg	CPU	The period of calibration		The period of test	
			RE (%)	R^2	RE (%)	R^2
PSO	51.904	54’46”	4.28	0.822	2.90	0.831
PSSE-PSO	50.677	51’38”	4.74	0.825	2.80	0.831
MSSE-PSO	50.659	44’27”	4.48	0.825	2.78	0.831

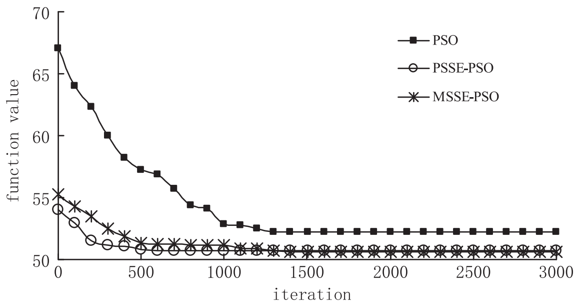

Fig. 6. Evolution of function value for PSO, PSSE-PSO and MSSE-PSO.

6. Conclusions

This paper presents MSSE-PSO (master–slave swarms shuffling evolution algorithm based on particle swarm optimization) to improve the performance of PSO and PSSE-PSO (parallel swarms shuffling evolution algorithm based on particle swarm optimization). For MSSE-PSO, firstly a population of points is sampled randomly from the feasible space, and then the population is partitioned into several sub-swarms (one master swarm and other slave swarms). Each slave swarm independently executes PSO or its variants, including the update of particles' position and velocity. For the master swarm, the particles enhance themselves based on the social knowledge of master swarm and that of slave swarms. At periodic stage in the evolution, the master swarm and the whole slave swarms are forced to mix and points are reassigned to sub-swarms to ensure information sharing. The process is repeated until a user-defined stopping criterion is reached. By simulation of four benchmark test functions, it is shown that MSSE-PSO possesses better ability to find the global optimum than PSO and PSSE-PSO. Compared with PSO and PSSE-PSO, MSSE-PSO is applied to identify the parameter of hydrological model. The results show that MSSE-PSO remarkably improves the accuracy of calibration and it is an effective method to calibrate hydrologic model.

Acknowledgements

This research is supported by National Natural Science Foundation of China (50809004)

References

[1] H.H. Rosenbrock, An automatic method for finding the greatest or least value of function, Comput. J. 3 (1960) 175–183.
[2] J.A. Nelder, R.A. Meade, Simplex method for function minimization, Comput. J. 7 (1965) 308–313.
[3] Q.J. Wang, The genetic algorithm and its application to calibrating conceptual rainfall-runoff models, Water Resour. Res. 27 (1991) 2467–2471.
[4] M. Franchina, Use of a genetic algorithm combined with a lacal search method for the automatic calibration of conceptual rainfall-runoff models, Hydrol. Sci. 41 (1996) 21–39.
[5] Q.J. Wang, Using genetic algorithms to optimize model parameters, Environ. Model. Software 12 (1997) 27–34.
[6] C.T. Cheng, C.P. Ou, K.W. Chau, Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration, J. Hydrol. 268 (2002) 72–86.
[7] Q.Y. Duan, S. Sorooshian, V.J. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resour. Res. 28 (1992) 1015–1031.
[8] Q.Y. Duan, V.K. Gupta, S. Sorooshian, Shuffled complex evolution approach for effective and efficient global minimization, J. Opt. Theor. Appl. 76 (1993) 501–521.
[9] Q.Y. Duan, S. Sorooshian, V.J. Gupta, Optimal use of the SCE-UA global optimization method for calibrating watershed models, J. Hydrol. 158 (1994) 265–284.
[10] P.O. Yapo, H.V. Gupta, S. Sorooshian, Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data, J. Hydrol. 181 (1996) 23–48.
[11] J.A. Vrugt, H.V. Gupta, W. Bouten, S. Sorooshian, A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resour. Res. 39 (2003) 1201.
[12] A. Vrugt, H.V. Gupta, L.A. Bastidas, W. Bouten, Effective and efficient algorithm for multiobjective optimization of hydrologic models, Water Resour. Res. 39 (2003) 1214.
[13] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings Sixth Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995, pp. 39–43.
[14] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
[15] S. Paterlini, T. Krink, Differential evolution and particle swarm optimization in partitional clustering, Comput. Stat. Data Anal. 50 (1) (2006) 1220–1247.
[16] C. Elegbede, Structural reliability assessment based on particle swarm optimization, Struct. Saf. 27 (2) (2005) 171–186.
[17] K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, Nat. Comput. 1 (2002) 235–306.
[18] X.T. Feng, B.R. Chen, C. Yang, H. Zhou, X. Ding, Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm, Int. J. Rock Mech. Min. 43 (2006) 789–801.
[19] P.C. Fourie, A.A. Groenwold, The particle swarm optimization algorithm in size and shape optimization, Struct. Multidiscip. Optim. 23 (4) (2002) 259–267.
[20] Y. Jiang, T.S. Hu, F.L. Gui, X.N. Wu, Z.X. Zeng, Application of particle swarm optimization to parameter calibration of Xin'anjiang model, J. Wuhan Univ. (Engineering Science) 39 (2006) 14–17.
[21] M.K. Gill, Y.H. Kaheil, A. Khalil, M. McKee, L. Bastidas, Multiobjective particle swarm optimization for parameter estimation in hydrology, Water Resour. Res. 42 (2006) 7417.
[22] X. Shi, Y. Lu, C. Zhou, H. Lee, W. Lin, Y. Liang, Hybrid evolutionary algorithms based on PSO and GA, in: Proceedings of IEEE Congress on Evolutionary Computation, 2003, pp. 2393–2399.
[23] X.H. Wang, J.J. Li, Hybrid particle swarm optimization with imulated annealing, Proceedings of the Third International Conference on Machine Learning and Cybernetics, 26–29, 2006, pp. 2402–2405.
[24] B. Niu, Y.L. Zhu, X.X. He, Multi-population cooperative particle swarm optimization, Lecture Notes Artif. Intell. 3630 (2005) 874–883.
[25] P.S. Shelokar, P. Siarry, V.K. Jayaraman, B.D. Kulkarni, Particle swarm and ant colony algorithms hybridized for improved continuous optimization, Appl. Math. Comput. 188 (2007) 129–142.
[26] M.M. Ali, P. Kaelo, Improved particle swarm algorithms for global optimization, Appl. Math. Comput. 196 (2008) 578–593.
[27] Y. Jiang, T.S. Hu, C.C. Huang, X.N. Wu, An improved particle swarm optimization algorithm, Appl. Math. Comput. 193 (2007) 231–239.
[28] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Info. Process. Lett. 85 (2003) 317–325.
[29] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, NJ, 1998, pp. 69–73.
[30] R.J. Zhao, The Xinanjiang model applied in China, J. Hydrol. 135 (1992) 371–381.