

Water Science and Engineering, 2010, 3(2): 156-165
doi:10.3882/j.issn.1674-2370.2010.02.004



<http://www.waterjournal.cn>
e-mail: wse2008@vip.163.com

Improved PSO algorithm based on chaos theory and its application to design flood hydrograph

Si-fang DONG*¹, Zeng-chuan DONG¹, Jun-jian MA¹, Kang-ning CHEN²

1. State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering,
Hohai University, Nanjing 210098, P. R. China

2. Department of Remote Sensing Center, China Institute of Water Resources and Hydropower Research,
Beijing 100044, P. R. China

Abstract: The deficiencies of basic particle swarm optimization (bPSO) are its ubiquitous prematurity and its inability to seek the global optimal solution when optimizing complex high-dimensional functions. To overcome such deficiencies, the chaos-PSO (COSPSO) algorithm was established by introducing the chaos optimization mechanism and a global particle stagnation-disturbance strategy into bPSO. In the improved algorithm, chaotic movement was adopted for the particles' initial movement trajectories to replace the former stochastic movement, and the chaos factor was used to guide the particles' path. When the global particles were stagnant, the disturbance strategy was used to keep the particles in motion. Five benchmark optimizations were introduced to test COSPSO, and they proved that COSPSO can remarkably improve efficiency in optimizing complex functions. Finally, a case study of COSPSO in calculating design flood hydrographs demonstrated the applicability of the improved algorithm.

Key words: particle swarm optimization; chaos theory; initialization strategy of chaos factor; global particle stagnation-disturbance strategy; design flood hydrograph

1 Introduction

The particle swarm optimization (PSO) algorithm was first described by Eberhart and Kennedy (1995). It is a swarm intelligence-based algorithm used to find a solution to an optimization problem in a search space, or to model and predict social behavior in the pursuit of objectives. It has been successfully applied to many optimization problems, including learning artificial neural networks and model predictive control (Eberhart and Shi 2001). A brief and complete overview of the principle, technique, and application of the PSO algorithm is provided by Kennedy and Eberhart (2001) and Clerc (2006). Schutte et al. (2005) used the PSO algorithm for biomechanical optimization and concluded that the performance of the PSO algorithm is superior to that of the genetic algorithm.

Just like other evolutionary algorithms of global optimization, PSO has the disadvantages

This work was supported by the National Basic Research Program of China (973 Program) (Grant No. 2006CB403402).

*Corresponding author (e-mail: dongsifang@163.com)

Received Apr. 7, 2010; accepted Apr. 26, 2010

of premature convergence, delay of convergence in later periods, and an excessive decrease of particle variety with more iterations, which may render it incapable of converging at the global optimal solution. Therefore, domestic and foreign scholars have tried to improve the optimization performance of the basic PSO (bPSO) algorithm and developed some representative improved versions with superior performance. These improved models can be classified into four types: (1) those that balance the global and local searching capabilities of the algorithm by fluctuating or adjusting the parameters of PSO (Angeline 1998; Higashi and Iba 2003); (2) those that improve the performance of the algorithm by designing different types of topological structures (Mendes et al. 2004; Shi and Eberhart 2001); (3) those that decrease the gathering of particles by increasing the particle variety in PSO (Baskar and Suganthan 2004; Kennedy and Eberhart 1997); and (4) those that form a blending algorithm with superior performance by combining PSO with other searching technology (Wang et al. 2001; Koay and Srinivasan 2003; Fukuyama and Yoshida 2001).

In order to improve the PSO algorithm, this paper introduces the chaos optimization mechanism and a global particle stagnation-disturbance strategy, which keeps the particles from the stagnation state. An optimal model called the chaos-PSO (COSPOS) model was set up to calculate the design flood hydrograph based on similar disparity theory. In general, the following problems are encountered when using the conventional homogeneous frequency enlargement method to draw the design flood hydrograph: (1) when the flood peak and flood volume relation is not satisfactory, the homogeneous frequency enlargement method might cause different amplification rates in different time intervals of the hydrograph, leading to mutation or discontinuity of hydrographs between successive time intervals; (2) the conventional method simply duplicates the flood discharge in the same time interval, and cannot maintain the original typical flood process after linking each time interval; (3) the hydrograph is smoothed manually, rather than with a computer science technique, which takes more time and effort. In this study, the COSPOS algorithm was used to solve these problems in calculating the design flood hydrograph.

2 Establishment of COSPSO model

2.1 Initialization strategy

The bPSO algorithm initializes the stochastic location of each particle in the solution space. The uniform distribution of particles cannot be guaranteed in this way and more time is needed for the particles to seek the global optimal solution. The particles' random distribution also increases the randomness of the global optimal solution. If the locations of most particles are significantly inferior to the others, the variety of particles will rapidly decrease as the algorithm runs, ultimately causing prematurity. Therefore, the chaos theory and chaotic optimization mechanism were introduced into the bPSO algorithm to guide the optimal distribution of the particles in the solution space: the locations of the particles were mapped in the optimal chaos space so that particles would stay in the chaos state and move in a chaotic trajectory after they were initialized randomly (Bloch 2005). Most of the particles arrive at a

better location within a given period. Therefore, the rational selection of origin remarkably increases the probability that the PSO algorithm will reach the global optimal solution and reduces the number of iterations. At the same time, the diversity of particles does not decrease rapidly during the running of the algorithm, as the particles have a good initial location. We picked two chaos theory mapping methods, Logistic mapping and Ken mapping, for the following analysis:

(1) Logistic mapping is formulated as follows:

$$x_{k+1} = f(\mu, x_k) = \mu x_k (1 - x_k) \quad (1)$$

where x is the variable ($0 < x < 1$); f is the self-mapping function; k is the number of iterations ($k = 1, 2, \dots, n$); and μ is the control parameter ($0 < \mu \leq 4$).

(2) Kent mapping and Logistic mapping have a mutual transformation relationship and topological conjugacy property. The formula for Kent mapping is as follows:

$$f(x) = \begin{cases} \frac{x}{\beta} & 0 < x \leq \beta \\ \frac{1-x}{1-\beta} & \beta < x \leq 1 \end{cases} \quad (2)$$

where β is the control parameter. Since $0 < \beta \leq 1$, the Lyapunov exponent of Kent mapping is greater than 0, and the mapping is in chaos status. The chaos invariant set of Kent mapping is (0, 1). However, the chaotic sequence of Kent mapping can easily be influenced by such restraints of the computer as finite word length and accuracy.

2.2 Global particle stagnation-disturbance strategy

An important cause of prematurity of the bPSO algorithm is that the $gBest$ particle that reaches the global optimal solution makes no contribution in the later period, and just follows the velocity and direction of the former iteration in the search. It is easily trapped in a local extremum in a complicated environment with lots of local extrema. No strategy is provided to help it escape from local optimization, but only to force calculation. The chaos theory was introduced to carry out a random disturbance strategy for stagnant $gBest$ particles. The following is an example using Logistic mapping:

$$g_{k+1}^B = g_k^B + \sigma r_1 x_{k+1} - \sigma r_2 x_k \quad (3)$$

$$x_{k+1} = r x_k (1 - x_k) \quad (4)$$

$$\sigma = \begin{cases} 1 & r_3 < p_m \\ 0 & r_3 \geq p_m \end{cases} \quad (5)$$

where g_k^B is the global optimal solution function for the k th iteration; r , r_1 , r_2 , and r_3 are random numbers lying between 0 and 1; and p_m is the probability constant, lying between 0.05 and 0.5.

The chaos stagnation-disturbance strategy not only guarantees that the algorithm can enhance the ability of the $gBest$ particles to avoid local optimization when the local extremum is close to the global optimal point, but also provides a continuous transform mechanism to

help *gBest* particles escape from local extrema gradually when the local extrema are far from the global optimal point.

2.3 Calculation steps

(1) Two initial numbers of iterations are set as $k=1$ and $k'=1$. m chaos variables $x_{i,k+1}$ ($i=1, 2, \dots, m$) with different traces are obtained by assigning x_k in Eq. (1) with m initial values, which have slight differences from one another.

(2) According to the equation $x'_{i,k+1} = c_i + d_i x_{i,k+1}$, where c_i and d_i are constants, the scale transformation between $x_{i,k+1}$ and $x'_{i,k+1}$ is performed. Then, $x'_{i,k+1}$ is used in optimization by utilizing the chaotic carrier.

(3) An iteration search is performed using chaos variables. To calculate the function value $f(x_{i,k})$, it is assumed that $x_{i,k} = x'_{i,k+1}$. Then, we set the initial values $x_i^* = x'_{i,1}$ and $f^* = f(x'_{i,1})$.

(4) It is necessary to determine whether $f(x_{i,k}) \leq f^*$. If not, $x_{i,k'}$ is abandoned. If so, we set $f^* = f(x_{i,k})$ and $x_i^* = x_{i,k}$. Then, $k = k + 1$.

(5) If f^* is unchanged, we go on to the next step. If not, we return to step (2).

(6) Reloading by the chaotic carrier is conducted according to the relationship $x''_{i,k+1} = x_i^* + a_i x_{i,k+1}$, where a_i is the adjustable constant that makes $a_i x_{i,k+1}$ a low-range chaos variable.

(7) We set $x_{i,k'} = x''_{i,k+1}$, and $f(x_{i,k'})$ is calculated. It is necessary to determine whether $f(x_{i,k'}) \leq f^*$. If not, then $k' = k' + 1$, $x_{i,k'}$ is abandoned, and we return to step (5); if so, we set $f^* = f(x_{i,k'})$ and $x_i^* = x_{i,k'}$.

(8) If the global optimal solution remains unchanged for many iterations, then we carry out the disturbance strategy and return to step (2). If not, we go on to the next step.

(9) It is necessary to determine whether $f^* = f(x_{i,k'})$ and $x_i^* = x_{i,k'}$ comply with the termination rule. If not, then step (6) is repeated. If so, then the optimal solution is found: the optimal solution is x^* and the corresponding function is f^* .

3 Verification of COSPSO

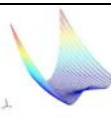
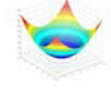
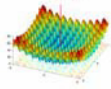
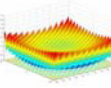
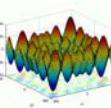
3.1 Introduction of benchmark functions

In order to evaluate the performance of COSPSO based on chaos theory, including the convergence speed of global optimization, five benchmark optimizations (Table 1) were introduced and analyzed (Runarsson and Yao 2000), facilitating comparison of the performance of COSPSO to that of bPSO. These benchmark optimizations, which aim at a global minimum value, are often used to test the performance of a reformed algorithm. They are composed of two single-peak (unimodal) functions and three multi-peak (multimodal) functions, which possess different characteristics and can thus test every aspect of the optimized performance in various problems.

In Table 1, $\min F(x)$ is the global minimum, and f_1 and f_2 are consecutive unimodal functions, usually used to inspect the convergence rate of the algorithm. The Rosenbrock

function is a typical complex function with a smooth trend, and its global optimal point is in an even, narrow, parabola-shaped valley. It is usually used to test the convergence rate of the algorithm because it requires little information and has a small likelihood of convergence to the global optimal point. f_3 , f_4 , and f_5 are complex nonlinear multimodal functions with plenty of local extrema. Therefore, they work effectively for the inspection of the algorithm's global search performance, particle diversity, and ability to allow particles to escape from local extrema, guarantee convergence, and prevent prematurity.

Table 1 Five benchmark functions

Code of functions	Optimization function	Three-dimensional graph
f_1	$F_{\text{Rosenbrock}}(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (1 - x_i)^2], \quad -2.048 \leq x_i \leq 2.048$ $\min F_{\text{Rosenbrock}}(x) = F(1, 1, \dots, 1) = 0$	
f_2	$F_{\text{Sphere}}(x) = \sum_{i=1}^n x_i^2, \quad -5.12 \leq x_i \leq 5.12$ $\min F_{\text{Sphere}}(x) = F(0, 0, \dots, 0) = 0$	
f_3	$F_{\text{Rastrigin}}(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)], \quad -5.12 \leq x_i \leq 5.12$ $\min F_{\text{Rastrigin}}(x) = F(0, 0, \dots, 0) = 0$	
f_4	$F_{\text{Griewank}}(x) = 1 + \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \left[\cos\left(\frac{x_i}{\sqrt{i}}\right) \right], \quad -512 \leq x_i \leq 512$ $\min F_{\text{Griewank}}(x) = F(0, 0, \dots, 0) = 0$	
f_5	$F_{\text{Schwefel}}(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i }), \quad -512 \leq x_i \leq 512$ $\min F_{\text{Schwefel}}(x) = F(420.968\ 750, \dots, 420.968\ 750) = -418.982\ 887\ 27n$	

The functions with more dimensions, wider independent variable scopes, and higher target accuracy are more difficult to be optimized. In order to facilitate the comparison between the original algorithm and the improved algorithm, and to give priority to the performance of the latter, this study selected the most rigorous parameter sets (Eberhart and Kennedy 1995), shown in Table 2.

Table 2 Function parameter settings

Code of functions	Variable range	Iterations	Particle quantity	Function dimension
f_1	[-100,100]	7 000	30	30
f_2	[-100,100]	7 000	30	30
f_3	[-100,100]	7 000	30	30
f_4	[-600,600]	7 000	30	30
f_5	[-500,500]	7 000	30	30

3.2 Parameter setting of COSPSO

For better comparison of the performances of COSPSO and bPSO, the same parameter values were used, in addition to some new parameters. In bPSO the tactic of linear dynamic descending within the range of 0.4 to 0.9 was adopted for the inertia weight w (Parsoploulos et al. 2001), while a constant value of $w = 0.9$ was used for the COSPSO.

The same value of 2 was used for the acceleration coefficients c_1 and c_2 of COSPSO and bPSO. The particle quantity was 30. There were 500 low-accuracy searches and 200 high-accuracy searches in COSPSO. When the control parameter was 4, it could be ensured that Logistic mapping would be in the chaotic state. In this way, the efficient optimization performance of COSPSO was fully examined.

3.3 Unimodal function comparison

Fig. 1 and Fig. 2 show that the performance of COSPSO with the Logistic and Kent mapping functions presents a great improvement over bPSO. In the figures, L-MAP and K-MAP are the processes of COSPSO with Logistic mapping and Kent mapping, respectively, and $\ln y$ indicates the accuracy of iterative calculation. When solving the sphere function, COSPSO always maintains a speedy convergence and could be said to have a tendency of accelerating convergence compared with PSO. The global optimal solution of COSPSO in the initial stage was by no means better than that of bPSO. However, the initial individual value of the particles was much superior to that of randomly initialized bPSO. Therefore, the descending speed of the iterative curve of COSPSO was higher than that of bPSO, thus guaranteeing speed, efficiency, and accuracy in convergence. As seen in Fig. 1, it took at least 1 000 iterations before bPSO reached a relatively high convergence accuracy, while in COSPSO only 150 iterations were needed. Furthermore, the accuracy of e^{-160} at the 7000th iteration of COSPSO was far higher than the accuracy of e^{-50} of bPSO, as shown in Fig. 2.

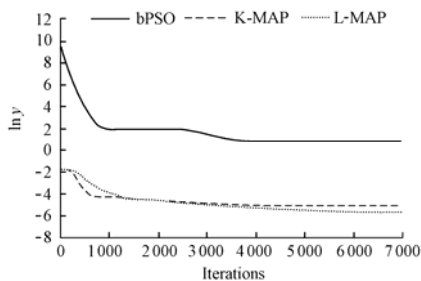


Fig. 1 Optimization process of Rosenbrock function

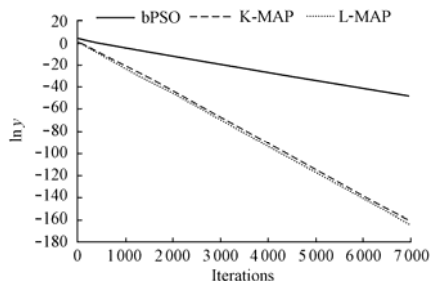


Fig. 2 Optimization process of Sphere function

3.4 Multimodal function comparison

In the optimization of multimodal functions (f_3 , f_4 , and f_5), the abundance of local extrema easily ran the algorithm into local optimization and then caused prematurity, and thus limited the ability of the algorithm to escape from the local extremum and converge at global optimization. Fig. 3, Fig. 4, and Fig. 5 show that the convergence accuracy of COSPSO was the same as that of the bPSO in the Griewank function. However, in other respects, such as the ability to escape from local extrema, convergence speed, and global optimization, COSPSO was evidently better than bPSO. This was well proven by its speedy convergence in the initial 500 to 1000 iterations. Besides, the initial accuracy of the new algorithm was significantly higher than that of bPSO. All of these factors lead to a global optimal solution. COSPSO can improve the efficiency of multimodal function optimization mainly because of the chaotic

optimization process, in which the initial locations of particles are calculated with high accuracy and searching directions are highly precise, thus increasing the variety of particles. The variety does not diminish as iteration continues, and it can enhance particles' ability to escape from the local extremum, thus avoiding the prematurity in bPSO. This can be seen clearly from the results of the experiment.

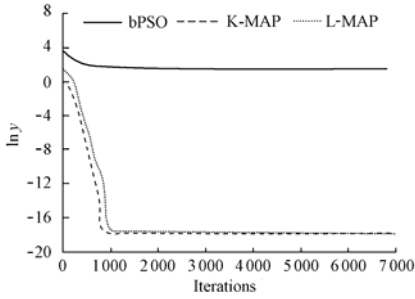


Fig. 3 Optimization process of Rastrigin function

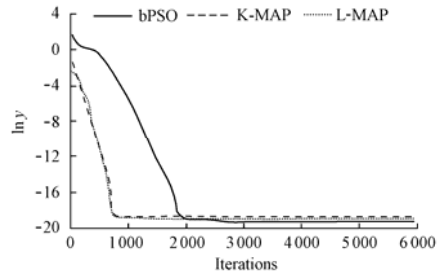


Fig. 4 Optimization process of Griewank function

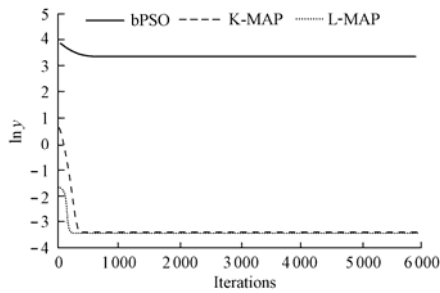


Fig. 5 Optimization process of Schwefel function

4 Design flood hydrograph model

The design flood hydrograph model was established based on complete water level and discharge data from Changba Hydrological Station, 4 500 m upstream of the Wantou hydro-junction, for the period of 1953 to 2004. This model was meant for dealing with the following problems in solving the design flood hydrograph by enlarging a typical flood hydrograph at the same frequency: (1) When the flood peak and flood volume relation is not satisfactory, a mutation on the hydrograph may occur at the connecting point of two successive periods. (2) The original typical flood mode can easily be destroyed (CWRC 2001). Similar disparity theory was applied to establish the object function of the design flood hydrograph model:

$$\min f = \frac{1}{n} \sum_{i=1}^{n-1} |k_i - k'_i| - \frac{1}{n} \sum_{i=1}^{n-1} (k_i - k'_i) + \frac{1}{n} \sum_{i=1}^{n-1} |k_i - k'_i| \quad (6)$$

The constraint conditions are as follows:

$$\begin{cases} Q_{\max} = Q'_{\max} \\ Q_1 = \int_{t_{s1}}^{t_{e1}} Q(t) dt = Q'_1, \quad Q_3 = \int_{t_{s3}}^{t_{e3}} Q(t) dt = Q'_3, \quad Q_7 = \int_{t_{s7}}^{t_{e7}} Q(t) dt = Q'_7 \end{cases} \quad (7)$$

where $k_i = \frac{Q(t_i + \Delta t) - Q(t_i)}{\Delta t}$; $k'_i = \frac{Q'(t_i + \Delta t) - Q'(t_i)}{\Delta t}$; Q_{\max} , $Q(t)$, Q_1 , Q_3 , and Q_7 are, respectively, the peak discharge, the discharge at time t , and the maximum flood volumes after one day, three days, and seven days of the design flood; Q'_{\max} , $Q'(t)$, Q'_1 , Q'_3 , and Q'_7 are, respectively, the peak discharge, the discharge at time t , and the one-day, three-day, and seven-day maximum flood volumes of a typical flood at a definite frequency; t_{sj} and t_{ej} ($j = 1, 3, 7$) are the beginning and ending time of the durations of one day, three days, and seven days; i is the number of the time interval: $i = 1, 2, \dots, n$; and Δt is the time step.

The constrained nonlinear optimization problem of Eq. (6) can be converted into an unconstrained optimization problem by Eq. (8):

$$\min f = \frac{1}{n} \sum_{i=1}^{n-1} \left| k_i - k'_i - \frac{1}{n} \sum_{i=1}^{n-1} (k_i - k'_i) \right| + \frac{1}{n} \sum_{i=1}^{n-1} |k_i - k'_i| + \sigma_1 (Q_{\max} - Q'_{\max})^M + \sigma_2 \left[\sum_{i=1}^{n_1} Q(t_i) \Delta t - Q_1 \right]^M + \sigma_3 \left[\sum_{i=1}^{n_3} Q(t_i) \Delta t - Q_3 \right]^M + \sigma_4 \left[\sum_{i=1}^{n_7} Q(t_i) \Delta t - Q_7 \right]^M \quad (8)$$

where M is a positive even number, which is set as 2 in the following calculation; and σ_i ($i = 1, 2, 3, 4$) are the penalty function factors, which are related to the current number of iterations n' . σ_i ($i = 1, 2, 3, 4$) is small at the beginning and increases gradually, which is helpful to the search for the optimal solution at a large scale, leading to the final solution to the original question:

$$\sigma_i = \left[\frac{2}{1 + \exp(-\alpha n'/T)} - 1 \right] \sigma_i^* \quad (i = 1, 2, 3, 4) \quad (9)$$

where α is a positive coefficient; σ_i^* ($i = 1, 2, 3, 4$) are random numbers larger than 500; T is the upper limit of iterations; $\sum_{i=1}^{n_1} Q(t_i) \Delta t$, $\sum_{i=1}^{n_3} Q(t_i) \Delta t$, and $\sum_{i=1}^{n_7} Q(t_i) \Delta t$ are the discrete forms of $\int_{t_{s1}}^{t_{e1}} Q(t) dt$, $\int_{t_{s3}}^{t_{e3}} Q(t) dt$, and $\int_{t_{s7}}^{t_{e7}} Q(t) dt$, referring to the maximum flood volumes for one day, three days, and seven days; and n_1 , n_3 , and n_7 are the number of time intervals during the periods of one day, three days, and seven days, respectively.

A flood process in 1976 was considered as a typical example. Frequency analysis was carried out based on the long-term hydrological data series and the eigenvalue of a typical flood at Changba Hydrological Station. The typical flood process was mapped with 168 actual measurement points and the time step was one hour. The COSPSO algorithm was applied to draw the design flood hydrograph of Changba Hydrological Station, and, further, to calculate the design flood standard hydrographs of various frequencies.

Some of the COSPSO parameters were set in Section 3.2. In addition, the initial values of penalty function factors were set as $\sigma_i = 5$ and $\sigma_i^* = 10^3$ ($i = 1, 2, 3, 4$). There were 500 iterations during the whole optimization process and 700 iterations during chaos optimization.

Table 3 shows the eigenvalues of a typical flood and design floods for different frequencies at Changba Hydrological Station. Fig. 6 shows different frequency design flood

hydrographs, which basically maintain the distribution pattern of a typical flood process and demonstrate the good effect of curve-fitting.

Table 3 Eigenvalues of typical flood and design floods for different frequencies

Flood frequency	Peak discharge Q_{\max} (m^3/s)	Flood volume (10^6m^3)		
		1 d	3 d	7 d
1976 typical flood	4 640	359.4	700.1	925.4
5%	4 950	379.6	804.5	1 375.8
3.33%	5 360	411.9	878.7	1 511.8
2%	5 860	451.6	968.8	1 685.1
1%	6 510	504.1	1 091.3	1 911.7
0.50%	7 160	556.5	1 211.7	2 140.4
0.20%	8 010	624.2	1 372.5	2 446.0
0.10%	8 640	675.5	1 486.8	2 663.7
0.05%	9 260	725.5	1 604.7	2 886.5

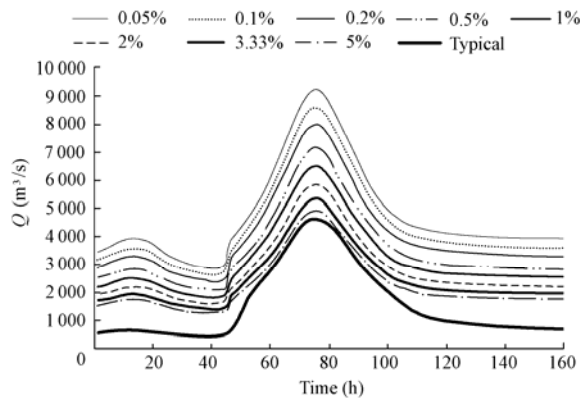


Fig. 6 Different frequency design flood hydrographs of Changba Station

5 Conclusions

In COSPSO, a particle's initial movement trajectory is assumed to be chaotic instead of the desultory stochastic trajectory of other models. The chaos factor is used to guide the particle's path and a disturbance strategy is used to keep the global particles from stagnation. In the optimization solution tests of two unimodal functions and three multimodal functions, the COSPSO displayed its main advantages, including: (1) speedy convergence to a global optimum solution, (2) high efficiency in the search of particles' initial direction, (3) refrainment from prematurity, (4) guarantee of particles' initial variety, and (5) high ability to keep particles from the stagnation state and allow them to maintain chaotic movement.

Based on the similar disparity theory, the scaling model for calculating a design flood was established and a new method of processing penalty function constraints was put forward. In addition, COSPSO, a cluster intelligent algorithm, was applied to optimize the solution, thus radically solving the problems that arise from drawing the design flood hydrograph using the conventional homogeneous frequency enlargement method. With Changba Hydrological Station as an example, the design flood hydrographs for frequencies ranging from 0.05% to

5% were drawn by the flood hydrograph enlargement model with COSPSO. The process is speedy and practical, and the results are in agreement with the typical flood process in 1976. The case study shows that the new method is effective for calculating the peak discharge and flood volumes of the design flood, and that the design flood is consistent with the typical flood patterns. This method allows the user to avoid the randomness and complexity of manual modification.

References

- Angeline, P. J. 1998. Using selection to improve particle swarm optimization. *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*, 84-89. Piscataway: IEEE Press. [doi: 10.1109/ICEC.1998.699327]
- Baskar, S., and Suganthan, P. N. 2004. A novel concurrent particle swarm optimization. *Proceedings of the 2004 Congress on Evolutionary Computation*, 1, 792-796. Piscataway: IEEE Press. [doi:10.1109/CEC.2004.1330940]
- Bloch, D. P. 2005. Complexity, chaos, and nonlinear dynamics: A new perspective on career development theory. *Career Development Quarterly*, 53(3), 194-207.
- Changjiang Water Resources Commission (CWRC). 2001. *The Calculation Handbook of Design Flood for Water Conservancy and Hydropower Project*. Beijing: China WaterPower Press. (in Chinese)
- Clerc, M. 2006. *Particle Swarm Optimization*. London: ISTE.
- Eberhart, R. C., and Kennedy, J. 1995. A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, 39-43. New York: IEEE Press. [doi: 10.1109/MHS.1995.494215]
- Eberhart, R. C., and Shi, Y. H. 2001. Particle swarm optimization: Developments, applications and resources. *Proceedings of the 2001 Congress on Evolutionary Computation*, 1, 81-86. Piscataway: IEEE Press. [doi: 10.1109/CEC.2001.934374]
- Fukuyama, Y, and Yoshida, H. 2001. Particle swarm optimization for reactive power and voltage control in electric power systems. *Proceedings of the 2001 Congress on Evolution Computation*, 1, 87-93. Piscataway: IEEE Press. [doi:10.1109/CEC.2001.934375]
- Higashi, N, and Iba, H. 2003. Particle swarm optimization with Gaussian mutation. *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 72-79. Piscataway: IEEE Press. [doi:10.1109/SIS.2003.1202250]
- Kennedy, J., and Eberhart, R. C. 1997. A discrete binary version of the particle swarm algorithm. *Proceedings of the 1997 IEEE International Conference on Computational Cybernetics and Simulation*, 5, 4104-4108. Piscataway: IEEE Press. [doi:10.1109/ICSMC.1997.637339]
- Kennedy, J., and Eberhart, R. C. 2001. *Swarm Intelligence*. San Francisco: Morgan Kaufmann Publishers.
- Koay, C. A., and Srinivasan, D. 2003. Particle swarm optimization-based approach for generator maintenance scheduling. *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 167-173. Piscataway: IEEE Press. [doi:10.1109/SIS.2003.1202263]
- Mendes, R., Kennedy, J., and Neves, J. 2004. The Fully Informed Particle Swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3), 204-210. [doi:10.1109/TEVC.2004.826074]
- Parsopoulos, K. E., Magoulas, G. D., Plagianakos, V. P., Vrahatis, M. N. 2001. Stretching technique for obtaining global minimizers through Particle Swarm Optimization. *Proceedings of the Workshop on PSO*. Indianapolis: Purdue School of Engineering and Technology, INPUI.
- Runarsson, T. P., and Yao, X. 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3), 284-294. [doi:10.1109/4235.873238]
- Schutte, J. F., Koh, B. I., Reinbolt, J. A., Haftka, R. T., George, A. D., and Fregly, B. J. 2005. Evaluation of a particle swarm algorithm for biomechanical optimization. *Journal of Biomechanical Engineering*, 127(3), 465-474. [doi:10.1115/1.1894388]
- Shi, Y. H., and Eberhart, R. C. 2001. Fuzzy Adaptive Particle Swarm Optimization. *Proceedings of the 2001 Congress on Evolutionary Computation*, 1, 101-106. Piscataway: IEEE Press. [doi:10.1109/CEC.2001.934377]
- Wang, L., Zheng, D. Z., and Li, Q. S. 2001. Survey on chaotic optimization methods. *Computing Technology and Automation*, 20(1), 1-5. (in Chinese)