CATFISH-EFFECT MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION FOR COORDINATED DISPATCHMENT OF WATER AND SEDIMENT IN A RESERVOIR

PENG, Y.* – JI, C. M. – SHI, Y. L.

School of Renewable Energy, North China Electric Power University, Beijing 102206, China

**Corresponding author*
e-mail: pengyang@ncepu.edu.cn; phone: +86-136-9102-9633; fax: +86-10-6177-2234

(Received 22nd Apr 2019; accepted 12th Jul 2019)

Abstract. In this paper, a catfish-effect multi-objective particle swarm optimization algorithm (CE-MOPSO) is proposed for optimizing the coordinative flow-sediment scheduling in a reservoir. In the proposed CE-MOPSO, the driven effect of catfish particles is introduced to improve the convergence and diversity of solutions. The performance of the proposed CE-MOPSO was verified using a classical bi-objective test function (ZDT3), and it was found that compared with MOPSO and Sigma-MOPSO algorithms, the CE-MOPSO showed better convergence to the true Pareto optimal fronts, and provided better diversity and uniformity for the Pareto frons with smaller values of convergence index and diversity index. After the successful validations in simulation studies, the proposed approach was then applied to a real case study in the Three Gorges Reservoir in China. Our results showed that the obtained Pareto solution set effectively approximated the true Pareto optimal frontier during the process of evolution. The scheduling results of CE-MOPSO revealed the relationship between power generation and sediment deposition in ten years and can be used to develop reservoir operation policies and plan sediment trapping and flow operations in real time. These results suggest that the proposed CE-MOPSO approach is efficient and effective in managing multi-objective water resources and hydrologic problems.

Keywords: *flow-sediment optimal scheduling, particle swarm optimization, catfish's driven, Pareto optimal solution*

Introduction

Reservoir comprehensive utilization and sediment sluicing are contradictory in reservoir operation and management. Sediment deposition can reduce reservoir storage capacity, and subsequently cause decreases in the efficiency of flood control, hydropower generation, and navigation (Yoon, 1992; Yang, 2003). Therefore, the coordinated flow-sediment regulation is very important for effective and successful reservoir operation (Zhu, 1997; Han, 2003; Peng et al., 2014).

The aim of coordinated flow-sediment regulation in a reservoir is to seek maximum comprehensive benefits under the minimum sediment deposition. This can be considered as a multi-objective optimization problem (MOOP) with characteristics such as being multidimensional, dynamic, strong coupling, and nonlinear. In most previous studies, this multi-objective problem was usually solved by being transformed into a single-objective problem using vector optimization technologies, such as weighted method (Lian et al., 2004; Xiao et al., 2013), constraint method (Zhang and Feng, 1988; Xiang et al., 2010), Bayesian model averaging method (Duan et al., 2007; Yan and Moradkhani, 2014; Yan and Moradkhani, 2016), decomposition-polymerization method (Peng et al., 2004, 2014) and so on. However, the values of weight coefficients and constraint thresholds should be set subjectively in advance during the process of solving the model. A single set of fixed weights or constraint thresholds may result in only local optimum on the Pareto front. To obtain the global Pareto optimum, we need to run a

DOI: http://dx.doi.org/10.15666/aeer/1705_1166911686

good number of simulations to examine all the weight combinations or constraint threshold (Reddy and Kumar, 2009).

As an alternative, multi-objective particle swarm optimization algorithm (MOPSO) has been shown as a useful tool for solving large and complex MOOPs, due to its stochastic and implicitly parallel properties to search for multiple local optimal solutions and consequently, to obtain the Pareto optimal solutions set. However, the algorithm of MOPSO has some disadvantages including premature convergence, reduced individual diversity, and enmeshed local optimum (Reyes-Sierra and Coello, 2006). To overcome these issues, several improved MOPSO have been proposed. For instance, Mostaghim and Teich (2003) used a Sigma method in MOPSO for searching the best local guides to speed up the convergence towards the true Pareto front with better distribution. Coello et al. (2004) used a constraint-handling mechanism and a special mutation operator to enhance exploration capability of MOPSO. Reyes-Sierra and Coello (2005) improved the MOPSO with the use of the Pareto dominance and a crowding factor to select the leaders. Leong and Yen (2006) suggested to use an adaptive local archive to promote swarm’s diversity and incorporated an adjusted population size to promote swarm’s competition. Branke and Mostaghim (2006) proposed a few strategies for the selection of global guides. Abido (2008) employed a clustering technique to adjust the optimal Pareto set size. Recently, the MOPSO has been increasingly applied in the field of water resources. Reddy and Kumar (2007, 2009) presented an elitist-mutation MOPSO approach for solving optimal multipurpose reservoir operation problems. Azadnia and Zahraie (Azadnia and Zahraie, 2010) used non-domination sorting and crowding distance techniques in MOPSO and applied it to optimize reservoir operation with two objectives including water supply and sediment removal. Li and Lian (2008) improved the MOPSO with self-adjusting inertia weights and Pareto-optimal archive for optimizing the coordinate reservoir deposition and power generation in reservoir operation. Despite the advances in these studies, there is still room to improve the efficiency of MOPSO, and the application of MOPSO to the coordinative flow-sediment scheduling in a reservoir has remained a challenge.

To further enhance the robustness of MOPSO, we propose a catfish-effect MOPSO algorithm (CE-MOPSO) in this paper. The catfish effect introduces a driven effect of catfish particles on individuals (i.e. the sardine particles), which was studied in the PSO algorithm and proven that the introduced catfish particle can improve the performance of the PSO (Chuang et al., 2008, 2011; Ji et al., 2011; De Souza et al., 2014). By taking advantage of such catfish effect, this paper incorporates catfish effect mechanism into the MOPSO algorithm. Different from the work in references (Chuang et al., 2008, 2011; De Souza et al., 2014), in which catfish particles were added at the extreme points of the search region and replaced 10% of original sardine particles having the smallest fitness value, this paper suggests another way to generate catfish particles and a different way that catfish particles work. The proposed algorithm was applied to optimize the coordinative flow-sediment scheduling in a real reservoir. In summary, the main contributions of the proposed CE-MOPSO algorithm are as follows.

(1) In the CE-MOPSO, the catfish particles are generated from the external archive, which is used to store the intermediate non-inferior solutions (i.e., elite particles with greater fitness value during the optimization process). Thus, the generated catfish particles have greater vigor and competitiveness than sardine particles. When the diversity of particle swarm is less than a defined threshold, catfish particles are added into the search region. The added catfish particles have driven effect on sardine

particles, which make the sardine particles escape from them. Thus, the sardine particles may jump out of the “locked” status and be guided to moving towards new area of the search space. This helps the MOPSO algorithm improve the diversity of the population, and converge fast towards true Pareto fronts in further generations.

(2) The global optimal value of the sardine particle driven-influenced by catfish particles, is decided by the Sigma value of catfish particle, which is the second close to the Sigma value of sardine particle instead of the closest one. This contributes the CE-MOPSO avoid trapping in the local optimal solutions caused by the Sigma method and finding the global optimal solution with deeper search and increased convergence speed.

(3) The proposed CE-MOPSO performance is first compared with MOPSO (Coello et al., 2004) and Sigma-MOPSO (Mostaghim and Teich, 2003) algorithms through an optimization test with a bi-objective test function (ZDT3). The results show that the CE-MOPSO have better performance in convergence and diversity. Then, the proposed algorithm is applied to solving the optimization of reservoir flow-sediment regulation in a case study. Results show that the proposed CE-MOPSO is able to obtain a well-distributed Pareto optimal front, which can reflect the relationship between power generation and reservoir deposition.

The remaining sections are organized as follows. In Section 2, we establish a reservoir flow-sediment optimal dispatching model. The proposed CE-MOPSO algorithm is elaborated in Section 3 and its performance is verified in Section 4 by comparing it with the MOPSO and Sigma-MOPSO algorithms. A case study of using the CE-MOPSO is presented in Section 5. Finally, we conclude the paper in Section 6.

Materials and methods

Materials

The Three Gorges Reservoir (TGR) in China is taken up as a case study for analyzing non-inferior relationship between hydropower generation and sediment deposition. The Three Gorges dam is located in the upper reach of the Yangtze River, as shown in *Figure 1*. The TGR is the largest multi-purpose hydropower project in China, which has great benefits of flood controlling, power generation, navigation improving, water supply, etc. The TGR began to impound water on 1 June 2003, and was in full operation in 2009. The main features of the reservoir are given in *Table 1*.

The storage capacity of TGR is 393 × 10^8 m^3, and every 1 m increase in water head will bring TGR great benefit of power generation. Meanwhile, reservoir sedimentation is a serious issue in TGR operation and management, because it reduces TGR storage capacity and causes decreasing capability in flood control, hydropower generation, and navigation. Increasing power generation requires increasing the water head, which result in increase in sediment deposition. Power generation and reservoir sedimentation are contradictory in the TGR operation and management. Therefore, relationship between power generation and sediment trapping need to be revealed for effective reservoir operations.

The hydropower generation and sedimentation of TGR were calculated based on the 1961~1970 hydrological series. The time interval for the scheduling calculation is ten days in June to September and one month in other months. Therefore, the total number of time intervals is 20 in a year and 200 in ten years. The range of sedimentation calculation of the TGR was from Jiangjin to the dam site of the TGR, with the total length of 689.2 km. 210 sections were set with average section spacing of about 3,280 m. The typical cross-sections are also shown in *Figure 1*.

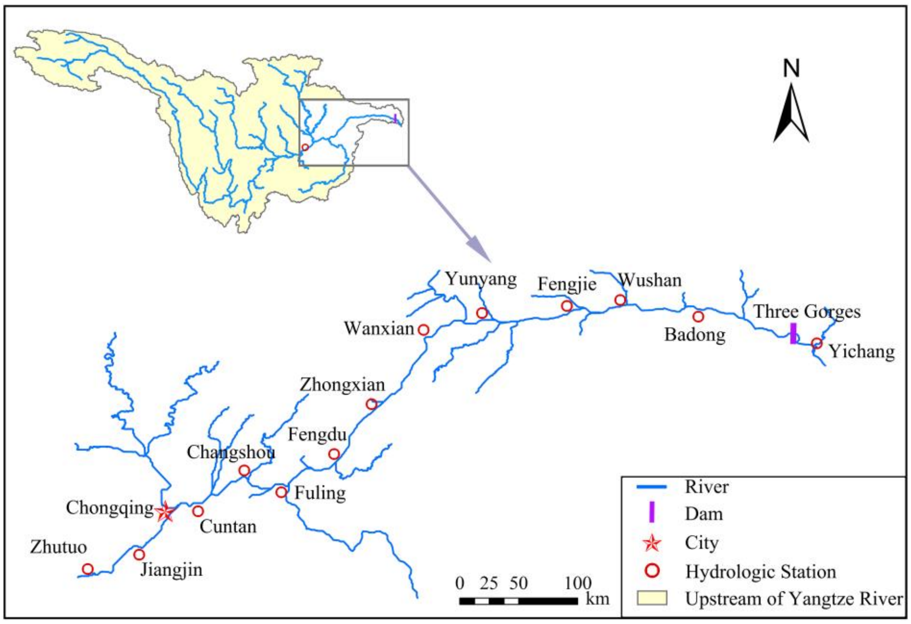

***Figure 1.** Location of the studied area*

***Table 1.** Primary features of the Three Gorges Reservoir*

Reservoir parameters	Quantity
Total reservoir capacity (10^8 m^3)	393
Active capacity (10^8 m^3)	165
Flood control capacity (10^8 m^3)	221.5
Dam crest elevation (m)	185
Normal water level (m)	175
Flood limited water level (m)	145
Lowest drawdown water level in dry season (m)	155
Installed capacity (10^6 kW)	22.40
Mean annual power generation (10^9 kW·h)	847

Model formulation for reservoir flow-sediment optimal scheduling

In this study, the flow-sediment optimal scheduling model takes into account the non-inferior relationships between hydropower generation and sediment siltation, with the consideration of the constraints including reservoir flood control and navigation. The developed optimization model is generalized as follows.

Objective function

The optimal regulation model includes two objectives: maximize hydropower generation and minimize sediment deposition in reservoir. The mathematical functions of the objectives are expressed as follows (Peng et al., 2004; Yoo, 2009):

$$\max E = max\sum_{t=1}^{T}\eta H_t Q_t \Delta t / 3600 = f_1\left(\boldsymbol{X}\right) \tag{Eq.1}$$

$$\min V_s = min\sum_{t=1}^{T} V_{st} = f_2\left(\boldsymbol{X}\right) \tag{Eq.2}$$

where E and V_s denote hydropower generation and sediment that deposited in the effective capacity of reservoir for all scheduling intervals; T is the total number of intervals over the scheduling horizon; $f_1(\boldsymbol{X})$ and $f_2(\boldsymbol{X})$ are functions of the objectives E and V_s; $\boldsymbol{X}$ is an independent variable decided by reservoir operation mode; η represents power coefficient; Δt is duration of an scheduling interval; H_t and Q_t are hydraulic head and release passing turbines of hydropower plant in the tth scheduling interval; V_{st} is sediment deposited in a reservoir in the tth scheduling interval.

Reservoir sedimentation calculation

In the present study, the V_{st} is calculated by a one-dimensional mathematical model for unsteady flow and nonuniform sediment transport in a reservoir (Peng and Zhang, 2006). The governing equations are described as follows:

Flow continuity equation (van Rijn, 1993)

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = q_i \tag{Eq.3}$$

Flow momentum equation (van Rijn, 1993)

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\alpha_1 \frac{Q^2}{A}\right) + gA\frac{\partial Z}{\partial x} + g\frac{n^2 Q|Q|}{AR^{4/3}} = 0 \tag{Eq.4}$$

Nonequilibrium transport equation for suspended load (Han and He, 2015)

$$\frac{\partial\left(AS_k\right)}{\partial t} + \frac{\partial\left(QS_k\right)}{\partial x} = \alpha_k \omega_k B\left(S_{*k} - S_k\right) \tag{Eq.5}$$

Sediment carrying capacity equation for suspended load (Yang, 1993)

$$S_* = S_*(U, H, \omega, \cdots) \tag{Eq.6}$$

Sediment transport rate equation for bed load (Yang, 1993)

$$g_{bk} = g_{b*k}(U, H, d, \cdots) \tag{Eq.7}$$

Total bed deformation equation (Yang, 1993)

$$\rho' \frac{\partial Z_d}{\partial t} = \sum_{k=1}^{N_s} \alpha_k \omega_k (S_k - S_{*k}) - \sum_{k=N_s+1}^{N} \frac{\partial g_{bk}}{\partial x} \tag{Eq.8}$$

where t and x are temporal and spatial axes; Q is streamflow discharge; q_l is discharge of lateral flow; A, B and H are cross-sectional area, width and depth of streamflow, respectively; Z is water level; U is cross-sectional average velocity of streamflow; R is hydraulic radius; n is Manning's roughness coefficient; S_k is suspended sediment concentration of size group k, written as $S_k = p_k S$ in which p_k and S are size distribution and sediment concentration of suspended load; S_{*k} is suspended sediment-carrying capacity of the kth group, expressed as $S_{*k} = p_{*k} S_*$ in which S_* sediment-carrying capacity of suspended load, and p_{*k} is size distribution of S_*; α_k is saturation recovery coefficient for nonequilibrium suspended load transportation of kth size group; ω_k is settling velocity of kth size group of suspended load; g_{b*k} is the actual bed load transport rate of kth size group, written as $g_{b*k} = g_b\left(d_k\right) * p_k$ in which $g_b(d_k)$ is bed load transport capacity of kth size group, and d_k is particle size of kth size group; Z_k is depth of bed deposition; g is gravitational acceleration; ρ_s' is dry density of deposits or bed material; d is particle diameter of bed material, N_s is number of size groups of suspended load; N is number of size groups of total load.

Equations 3–8 can be solved by finite difference method. Solution process consists of two steps: Flow *Equations 3* and *4* were solved firstly, and sediment *Equations* 5 and *8* secondly. For details of the solution process, the readers are referred to (Peng and Zhang, 2006). The V_{st} in *Equation 2* can be calculated based on the computed result of Z_d in *Equation 8.*

Constraints

Water volume balance constraint

The water volume balance equation is expressed as

$$V_{t+1} - V_t = \left(Q_t - q_t - S_t\right)\Delta t \qquad \text{(Eq.9)}$$

where V_{t+1} and V_t are reservoir storages at the end and beginning of the tth interval, respectively; Q_t and q_t are inflow and outflow discharges during the tth interval; S_t is discharge of loss water in a reservoir during the tth interval.

Flood control constraints of reservoir

The flood control constraints of reservoir are expressed as

$$\mathrm{Z}_t \le Z_{\max}\ ,\ q_t \le q_{\max} \qquad \text{(Eq.10)}$$

where Z_t is water level of reservoir at the end of the tth interval; Z_{max} is reservoir flood limiting water level; q_{max} is permissible discharge of streamflow at flood control point downstream of reservoir.

Navigation constraints

The navigation constraints of reservoir are expressed as

$$Z_t \geq Z_{\min}\, ,\, q_t \geq q_{t\min} \tag{Eq.11}$$

where Z_{min} is navigable water stage in reservoir area; q_{tmin} is minimum discharge flow that meets the needs of navigation downstream of reservoir during interval *t*.

Water release ability limit of dam

The outflow discharging from reservoir is restrained by the maximum discharge capacity of dam discharge structure, which can be expressed as

$$q_t \leq q(Z_t) \tag{Eq.12}$$

where $q(Z_t)$ is flood peak discharge ability limit of dam at level Z_t.

Power constraints of station

The power constraints of station are expressed as

$$N_{t,\min} \leq \eta H_t Q_t \leq N_{t,\max} \tag{Eq.13}$$

where $N_{t,\min}$ and $N_{t,\max}$ are the upper and lower limits of plant power during interval *t*, respectively.

The proposed catfish-effect MOPSO algorithm

Catfish effect mechanism

The catfish effect originates from an effect observed by Norwegian fishermen when they introduced catfish into a holding tank for caught sardines. The added catfishes would stimulate the movement of sardines, thus making the sardines alive and fresh longer (Hu, 2004).

The catfish effect in this study is applied to the MOPSO algorithm and this method is called catfish-effect MOPSO algorithm (CE-MOPSO). At the early evolutionary stage of particle group, particles are treated as sardine’s particles, and the Sigma method (Mostaghim and Teich, 2003) is used to search the space of solution with a fast convergence rate. When sardines group are trapped in a local optimum because of poor diversity, the catfish particles will be produced from the non-inferior solutions and be put into sardine particles. The generated catfish particles will trigger the driven influence on sardine particles, and guide sardine particles towards a new search region to improve the diversity in the population.

Implementation of CE-MOPSO

The detailed steps of the proposed CE-MOPSO algorithm are given below.

Step 1: Set the parameters including swarm population size *N*, maximum number of iteration *K*, and external archive size *M*. The external archive is used to store the generated non-inferior solutions that can be used to guide the search. The size of external archive determines the number of non-inferior solutions.

Step 2: Initialize the population. Treat particles as sardine particles, and initialize particles in the population with random positions and velocities that meet the constrains. Initialize *M/2* non-inferior solutions and set them as the individuals in external archive.

Step 3: Calculate the fitness of each particle, and maintain the external archive based on the crowding distance of each non-inferior solution.

The equation used for computing the fitness of each particle with two-objective fitness function was expressed as follows:

$$\begin{cases} \mathrm{Fit}(f_1(\mathbf{x})) = \sum_{t=1}^{T} \eta H_t Q_t \Delta t / 3600 - M \sum_{l=1}^{L} W_l \\ \mathrm{Fit}(f_2(\mathbf{x})) = 1/\{\sum_{t=1}^{T} V_{st} - M \sum_{l=1}^{L} W_l\} \end{cases} \tag{Eq.14}$$

where *M* is penalty factor; W_l is the value that the *t*th constraint is violated; *L* is total number of constraints.

Generate non-inferior solutions according to the calculated fitness values of particles, and store the generated non-inferior solutions per iteration in external archive. If the number of non-inferior solutions in external archive exceeds the *M*, retain the individuals with larger crowding distance, thus maintain the diversity of non-inferior solutions. The crowding distance (d_j) of individual *j* in external archive was computed as follows (Reddy and Kumar, 2007):

$$d_j = \sum_{ob=1}^{O} \frac{|f_{ob}(j+1) - f_{ob}(j-1)|}{f_{ob}^{\max} - f_{ob}^{\min}} \tag{Eq.15}$$

where *ob* denotes the objective index; *O* is the total number of objectives; $f_{ob}^{\max}$ and $f_{ob}^{\min}$ denote the max and min values of objective function, respectively; $f_{ob}(j + 1)$ and $f_{ob}(j\text{-}1)$ are the values of objective function for individuals *j*-1 and *j* + 1, which are the two nearest to individual *j*.

Step 4: Determine the best value of each particle in population, and calculate the Sigma value of each individual in both population and external file as follows (Mostaghim and Teich, 2003):

$$\delta = \frac{f_1^2 - f_2^2}{f_1^2 + f_2^2} \tag{Eq.16}$$

where f_1 and f_2 denote the two objective functions' values of particle.

Step 5: Calculate the diversity of particle swarm at the *k*th iteration ($\xi(k)$) as follows:

$$\begin{cases} \xi(k) = \dfrac{div(k)}{div(0)} \\ div(k) = \sqrt{\dfrac{1}{Nm} \sum_{j=1}^{N} \sum_{i=1}^{m} \left[\dfrac{(x_i^j(k) - x_i^{j,gbest}(k))}{(Bu_i - Bd_i)} \right]^2} \end{cases} \tag{Eq.17}$$

DOI: http://dx.doi.org/10.15666/aeer/1705_1166911686

where k is iteration number; j is serial number of particle; x_i^j is position in the *i*th dimension of particle *j*; $x_i^{j,gbest}$ is global best position in the *i*th dimension of particle *j* which exists in external archive, and its Sigma value is the second closest to the Sigma value of particle *j*; Bu_i and Bd_i denote the upper and lower limits of x_i^j .

Check whether the $\xi(k)$ is less than the scheduled threshold ξ_0. If $\xi(k) > \xi_0$, go to step 6, otherwise go to step 7.

Step 6: Determine the global optimal value of each particle using Sigma method (Mostaghim and Teich, 2003). If the sigma value of a particle in population is closest to the sigma value of an individual in external archive, the individual in external archive will be selected as the particle's global best position. Update the velocity and position of the *j*th particle as follows (Kennedy and Eberhart, 1995):

$$v_i^j(k+1) = \omega \times v_i^j(k) + c_1 \times r_1(x_i^{j,pbest}(k) - x_i^j(k)) + c_2 \times r_2(x_i^{gbest}(k) - x_i^j(k)) \quad \text{(Eq.18)}$$

$$x_i^j(k+1) = x_i^j(k) + v_i^j(k+1) \quad \text{(Eq.19)}$$

where ω is coefficient of inertia weight, which controls the influence of the particle previous velocity on its current one; c_1 and c_2 are learning factors; r_1 and r_2 are independent random numbers uniformly distributed in the interval [0,1]; v_i^j is the *i*th dimension velocity of particle *j*; $x_i^{j,pbest}$ is the *j*th particle's best position in the *i*th dimension.

Step 7: Generate catfish particles to promote the diversity of swarm particles.

Let C be the maximal number of catfish particles. If the number of non-inferior solutions is less than C, set all the non-inferior solutions as catfish particles. Otherwise, select C non-inferior solutions with larger crowding distance as catfish particles. The catfish particles are put into the sardine particle swarm as external competitive individuals. The added catfish particles, which act only at the current iteration and will be "dead" and removed from the particle swarm at next iteration, will trigger the driven influence on sardine particles. Thus, the search of solutions is affected by not only the guidance of global optimal value and individual optimal value, but also the driving of catfish particle. The renewed velocity of the *j*th particle is updated as follows:

$$\begin{gathered} v_i^j(k+1) = \omega \times v_i^j(k) + c_1 \times r_1(x_i^{j,pbest}(k) - x_i^j(k)) + c_2 \times r_2(x_i^{j,gbest}(k) - x_i^j(k)) - \\ Sat \times P \times sign(x_{Ci}^I(k) - x_i^j(k)) \times \exp\left\{-Q \times \left|x_{Ci}^I(k) - x_i^j(k)\right|\right\} \end{gathered} \quad \text{(Eq.20)}$$

where subscript C denotes catfish particle index; superscript I is a serial number of catfish particle which is closest to sardine particle *j* in dimension *i*, and is calculated as:

$$I = \left\{ l \mid \min_l \left(\left| x_{Ci}^I(k) - x_i^j(k) \right| \right) \right\} \quad \text{(Eq.21)}$$

Thus, the catfish particle nearest to each sardine particle in each dimension can be chosen at each iteration.

Note that the fourth item in *Equation 20* is called the driven effect item. *Sat* is a binary variable, 0 or 1 stochastically that decides if sardine particles are driven-influenced by catfish particles, which can be expressed as:

$$Sat = \begin{cases} 0 & F_I^{k+1} > F_I^k \\ 1 & F_I^{k+1} \le F_I^k \end{cases} \tag{Eq.22}$$

where F_I^k is the fitness of catfish particle *I* at the *k*th iteration.

Parameters *P* and *Q* in *Equation 20* decide how hard catfish particles drive sardine particles. The nearer the distance between catfish particle and sardine particle is, the greater driven effect of catfish particle on sardine particle is. Consequently, sardine particles keep escaping from catfish particles, and catfish particles occupy around global best position rapidly.

It can be seen that the catfish effect mechanism described in *Equation 20* has two key points: (1) Global optimal value of sardine particle is decided by the Sigma value of catfish particle, which is the second close to Sigma value of sardine particle instead of the closest one. It prevents sardine particles from trapping into the local optimal solutions caused by the Sigma method and helps sardine particles find the global optimal solution with local depth search and increased convergence speed. (2) Including the guidance of global optimal value and individual optimal value, the search of solution was also affected by the driven effect of catfish particles. This makes sardine particles escape from catfish particles generated from non-inferior solutions, which can avoid the solution search concentrating near non-inferior solutions and falling into a "locked" status. Thus, the searching space is broadened and the diversity of solution is improved.

Step 8: Generate the newer particle according to the computed velocity and position.

Step 9: Take the maximum number of iteration as terminal condition of the algorithm. If the number of iteration is less than the maximum iteration, then go to step 3; otherwise end the search and output the non-dominated solution set from external files.

The flowchart of the CE-MOPSO is exhibited in *Figure 2*.

Results and discussion

Efficiency of CE-MOPSO algorithm

Before applying to a real case study, we first demonstrated the efficiency of the proposed CE-MOPSO algorithm through a bi-objective test function (ZDT3). We compared this algorithm with MOPSO (Coello et al., 2004) and Sigma-MOPSO algorithms (Mostaghim and Teich, 2003), and used two performance metrics to assess their performance.

Performance measures

Convergence index (*γ*) measures the distance of the obtained non-dominated solutions to the true Pareto front, which is written as (Tripathi, 2007):

$$\gamma = \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \qquad \text{(Eq.23)}$$

where *n* denotes the number of members in non-dominated solutions; d_i is the Euclidean distance between the *i*th member in non-dominated solutions and its nearest member in the true Pareto front. A smaller value of γ reflects better convergence toward the true Pareto front.

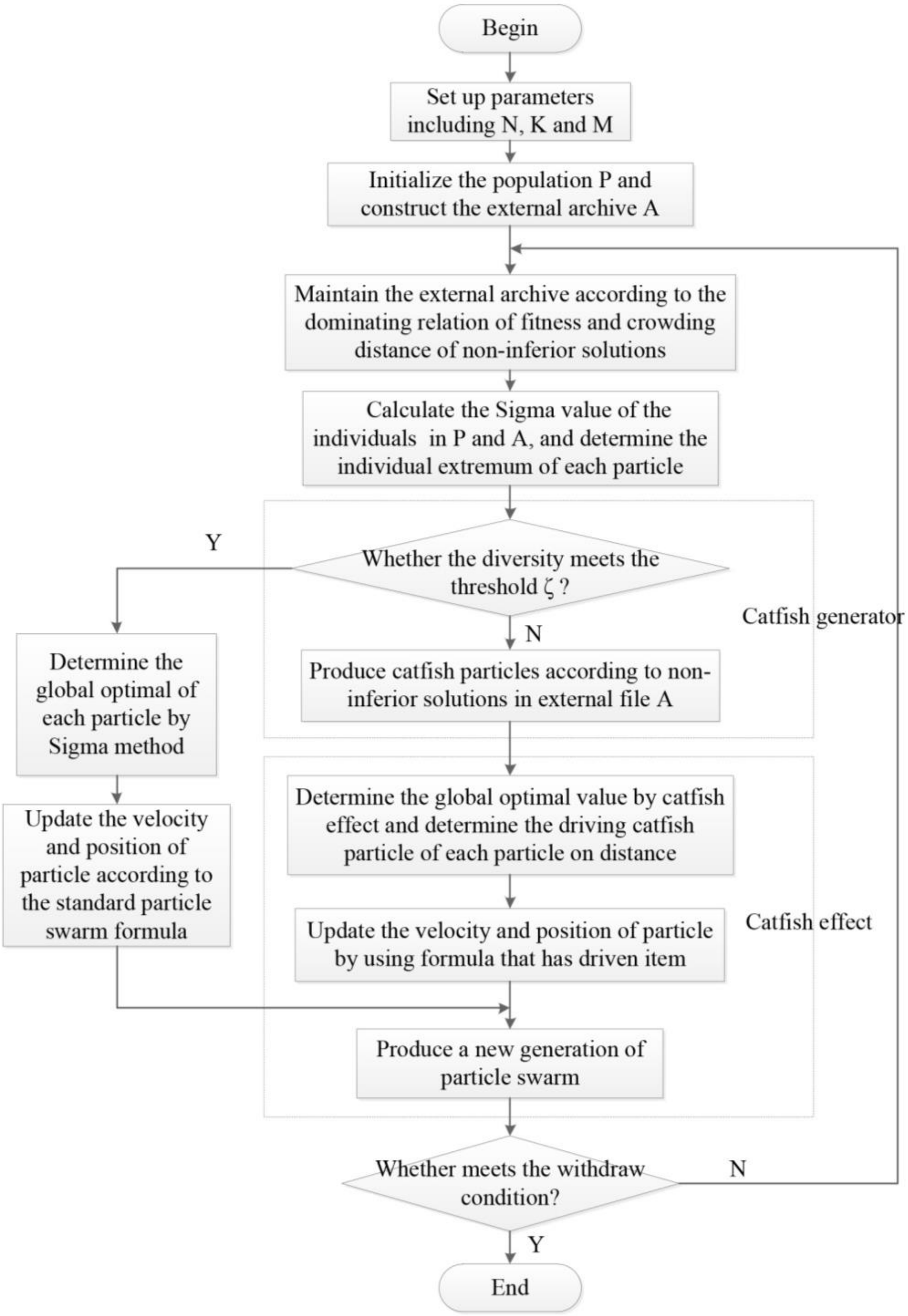
Begin
Set up parameters including N, K and M
Initialize the population P and construct the external archive A
Maintain the external archive according to the dominating relation of fitness and crowding distance of non-inferior solutions
Calculate the Sigma value of the individuals in P and A, and determine the individual extremum of each particle
Whether the diversity meets the threshold ζ ?
Y
N
Catfish generator
Produce catfish particles according to non-inferior solutions in external file A
Determine the global optimal of each particle by Sigma method
Update the velocity and position of particle according to the standard particle swarm formula
Determine the global optimal value by catfish effect and determine the driving catfish particle of each particle on distance
Update the velocity and position of particle by using formula that has driven item
Catfish effect
Produce a new generation of particle swarm
Whether meets the withdraw condition?
N
Y
End

Figure 2. *The flowchart of the catfish-effect multi-objective particle swarm algorithm*

Diversity index (*Δ*) indicates the spread along the non-dominated solutions, which is written as (Tripathi, 2007):

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (n-1) \times \bar{d}} \qquad \text{(Eq.24)}$$

where d_i is the Euclidean distance between adjacent solutions in the obtained non-dominated solutions set, and $\bar{d}$ is the average of all the d_i, which is expressed as $\bar{d} = \sum_{i=1}^{n-1} d_i / (n-1)$; d_f and d_l are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained non-dominated set. The smaller the value of *Δ*, the more uniform the distribution of the non-dominated solutions.

Discussion of results

The following parameters were chosen for the three algorithms: population size $N = 50$, archive size $M = 100$, maximum number of iteration $K = 500$, learning factors $c_1 = c_2 = 2$, inertia weight $\omega = 0.9\text{-}0.5\ k/K$, in which *k* is the current iteration number. Relative diversity threshold of CE-MOPSO algorithm was set to $\xi_0 = 0.45$ (The calculated results showed that the evolution of the population stabilized in the vicinity of this value and the diversity of the population needed to be promoted).

Twenty independent runs are performed for each algorithm on ZDT3, and their average and variance values of the two metrics (*γ* and *Δ*) are presented in *Table 2*. It can be seen that the proposed CE-MOPSO algorithm outperforms MOPSO and Sigma-MOPSO algorithms for ZDT3 test problem, with the lowest average and variance values of the *γ*. It indicates that the CE-MOPSO achieves a faster convergence rate to the true Pareto optimal fronts than MOPSO and Sigma-MOPSO algorithms. Also, our proposed CE-MOPSO algorithm obtained the best results with respect to the *Δ*, with the smallest average and variance values of the *Δ*. This shows that the proposed CE-MOPSO can to attain a better distribution of solutions than the other two algorithms for ZDT3 test problem. Therefore, it can be concluded that the solutions obtained by CE-MOPSO have better performance in convergence and diversity. It should be mentioned that the CE-MOPSO consumes much more computation time than MOPSO and Sigma-MOPSO algorithms, because the catfish's driven effect makes sardine particles explore a deeper search space to promote the diversity in population, which will consume more time.

Table 2. *Results for ZDT3: γ, Δ, and the time required per iteration*

Test function	Index	MOPSO	Sigma-MOPSO	CE-MOPSO
ZDT3	γ (average)	0.00418	0.10205	0.00311
	γ (variance)	0.00000	0.00238	0.00000
	Δ (average)	0.83195	0.76016	0.33004
	Δ (variance)	0.00892	0.00349	0.00007
	Time (sec)	16.78	19.35	71.49

The resulting Pareto fronts produced by the three algorithms for ZDT3 function are presented in *Figure 3*. It is seen that compared with MOPSO algorithm, the solutions of

DOI: http://dx.doi.org/10.15666/aeer/1705_1166911686

Sigma-MOPSO had better convergence at two ends, but the distribution in the middle was rather dispersed with several gaps (*Fig. 3b*). The solutions of MOPSO algorithm were scattered at two ends, but the distribution in the middle was relatively continuous and uniform (*Fig. 3a*). Thus, the Sigma-MOPSO algorithm paid more attention to the convergence at two ends, but ignored the convergence quality and diversity of solution. Compared with the MOPSO and Sigma-MOPSO algorithms, the CE-MOPSO in *Figure 3c* can converge faster towards the true Pareto front, and produce the true Pareto front with better spread both at two ends and in the middle on this function. This also demonstrates that the proposed CE-MOPSO has obvious advantages in convergence and spread of solutions for ZDT3 test problem.

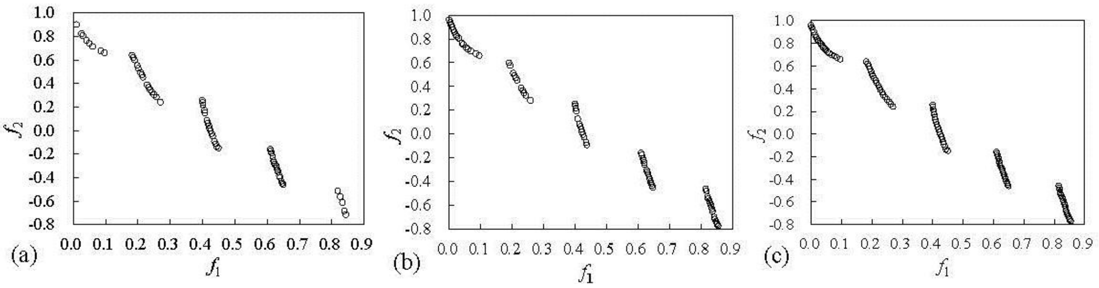

Figure 3. *Obtained Pareto fronts using (a) MOPSO, (b) Sigma-MOPSO and (c) CE-MOPSO on test function ZDT3*

Application

Case study

The one-dimensional numerical model for reservoir sedimentation was validated with other numerical results. The comparison of the computed sediment deposition for 10 years operation of the TGR was presented in *Table 3*. It is shown that the calculated sedimentation in TGR agreed well with the results calculated by Yangtze River Scientific Research Institute, the maximum relative error is 8.584% in the reach of Cuntan-Jiangjin. This indicated that the one-dimensional numerical model was accurate enough to model the sedimentation in the TGR.

Table 3*. Computed sediment deposition for 10 years operation of the TGR*

River reach	V_1 (10^8 m^3)	V_2 (10^8 m^3)	Relative error (%)
From the dam site to Fengdu	26.32	26.3	0.076
From Fengdu to Fuling	2.291	2.31	0.822
From Fuling to Changshou	1.146	1.2	4.500
From Changshou to Cuntan	0.327	0.337	2.967
From Cuntan to Jiangjin	0.213	0.233	8.584
From the dam site to Jiangjin	30.297	30.38	0.273

V_1 denotes the computed results by using the one-dimensional numerical model in the paper; V_2 denotes the computed results of Yangtze River Scientific Research Institute

Based on the designed operation scheduling of GTR, the main constraints were set as following: (1) Requirements of sediment flushing and constraints of flood control:

considering improvement of flood forecast techniques and requirement of sediment flushing, limited water level of flood control is varied between 140 m and 145 m, maximum release discharge reservoir is 55,000 m^3/s. (2) Restrictions of minimum power and navigation: minimum power of plants in dry season is 4.99 million kilowatts (kW), water level controlled by navigation upstream of TGR in dry season is 155 m.

Select water level of TGR at each time step as decision variable, and one particle represents one of operation scheduling strategies of reservoir. Thus, each particle is expressed as:

$$X^j = (x_1^j, x_2^j, \cdots, x_t^j, \cdots, x_T^j) \tag{Eq.25}$$

in which x_t^j is the water level of reservoir during period t; superscript j denotes the serial number of particles; T denote total number of time step.

Results analysis

Ten independent runs were performed for optimizing the coordinative flow-sediment scheduling operation in the TGR by using the proposed CE-MOPSO. *Figure 4* shows the obtained optimal frontier of Pareto and its evolution process through 50, 100, 300, and 500 iterations of CE-MOPSO algorithm. It is shown that the non-inferior solution set is renewed during the process of evolution, and the obtained Pareto set effectively approximate to the true Pareto frontier. Comparing the obtained Pareto frontier result of 500 iterations with that of 300 iterations, it is basically the same in the central section of B-C and A-B area, but the distribution of the non-inferior solutions near A point of 500 iterative times are better than the result of 300 iterative times. The non-inferior solutions near B point has formed a more continuous uniform distribution than those of 300 iterative times. This indicates that the diversity of non-inferior solution set is improved with iterations by catfish effect mechanism, especially the non-inferior solutions are well-distributed at each end and inflection point of the optimal frontier.

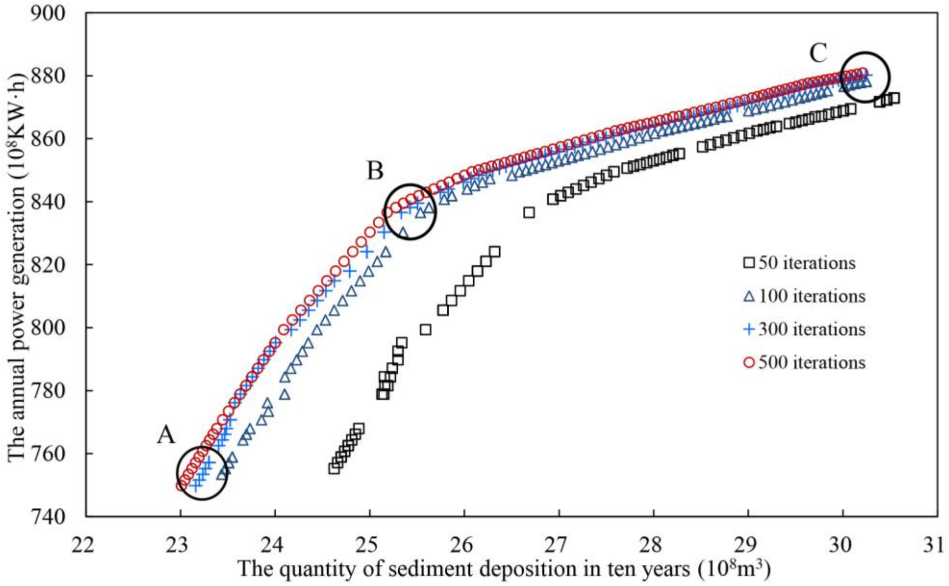

***Figure 4.** The optimal frontier of Pareto and evolution process of the sedimentation in ten year and the annual average power generation*

The Pareto optimal frontier shown in *Figure 4* also reflected the relationship between annual average power generation and sediment deposition in ten years. An increased hydropower generation resulted in an increased deposition in reservoir. The targets of increasing hydropower generation and decreasing reservoir sediment deposition are in contradictory ways. B point is an inflection point of the relationship between hydropower generation and sediment deposition of the TGR. In the section of A-B area, the rate of increasing power generation is greater than the rate of increasing siltation, while the rate of increasing power generation is smaller than the rate of increasing siltation in the section of B-C area. The inflection point B is very important for the selection of a non-inferior scheme in the TGR operation. For the schemes is in A-B area, increasing 1 × 10^8 kW·h power generation results in 0.025 × 10^8 m^3 sediment silting in the reservoir. For the schemes is in B-C area, increasing 1 × 10^8 kW·h power generation results in 1.08 × 10^8 m^3 sediment silting in the reservoir.

Figure 5 shows the non-inferior reservoir operation water level processes of the schemes corresponding to points A, B and C. In A scheme, since reservoir water level was lowered in flood season and did not reach the normal storage water level at the end of flood season that affects the power generation dispatch in non-flood season, both sediment deposition and power generation is the minimum in A scheme. In C scheme, due to the high water level operation in flood season, its power generation and sediment deposition is the maximum. The operation water level of B scheme in flood season is between A and C, which not only makes full use of lowering water level for sediment reduction in flood season, but also raising the water level to the normal storage level to ensure the power generation dispatch in non-flood season, so its power generation and sediment deposition is in the middle. This operation mode is different from traditional single-objective dispatch, which belongs to of multi-objective optimization of the coordinative water-sediment dispatch.

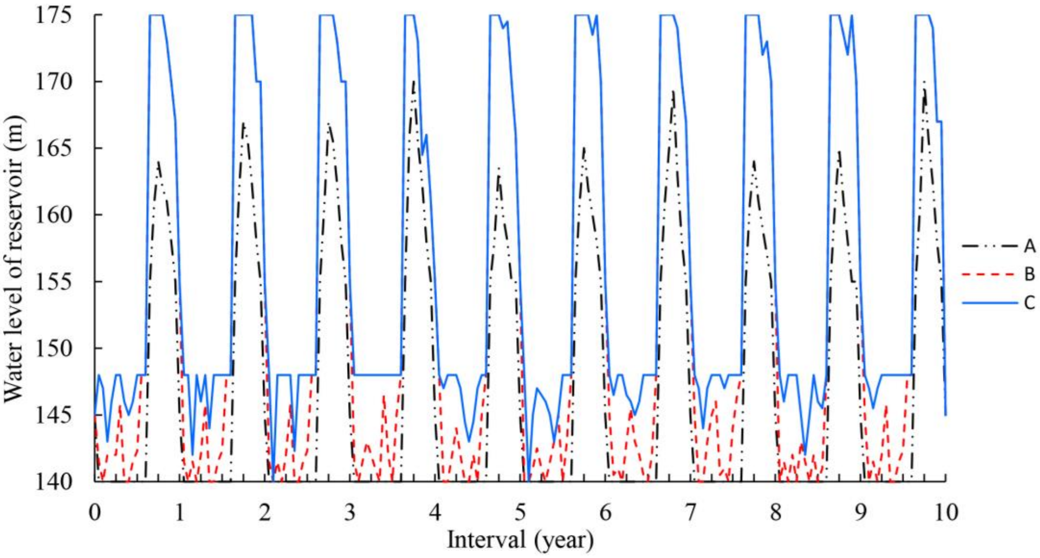

Figure 5. *Non-inferior reservoir operation level process of A, B and C schemes*

The mean annual power benefits and sedimentation represent short-term benefit and long-term benefits of the reservoir, respectively. If reservoir managers or decision makers focus on the life of reservoir that could play a long-term comprehensive utilization benefit, the corresponding reservoir water level need to be lowed, such as scheme A. If decision makers focus on the short-term benefits of reservoir, the corresponding reservoir water level need to be raised, such as scheme C. The optimal scheme has better balance between power generation and reservoir sedimentation, and the obtained the non-inferior solutions can be used to formulate reservoir operation policies in practice. These policies can also be used to plan sediment trapping and flow operations.

Conclusion

In this paper, a catfish-effect multi-objective particle swarm optimization (CE-MOPSO) algorithm is proposed for optimizing the coordinated dispatch of water and sediment in a reservoir. Catfish particles, generated from the individuals in external archive, have greater vigor and more competitiveness than sardine particles. When added into sardine particles, the catfish particles have driven effect on sardine particles, which makes the sardine particles escape from them. By this, the sardine particles may jump out of the “locked” status, explore the search region, and improve the diversity of population. This helps the MOPSO algorithm to effectively guide the search towards the true Pareto front in further generations. In addition, the global optimal value of particle is second close to its Sigma value instead of the closest one. This prevents sardine particles from trapping into the local optimal solutions caused by the Sigma method, and helps sardine particles find the global optimal solution with local deeper search and increased convergence speed. The performance of CE-MOPSO in nonlinear numerical function optimization was first investigated with a test of classical bi-objective test function (ZDT3). The results were encouraging and promising in both computational efficiency and search efficiency when compared with the MOPSO and Sigma-MOPSO. The proposed CE-MOPSO was then applied to a multi-objective optimization of the coordinative flow-sediment regulation in the Three Gorges Reservoir. The obtained Pareto set effectively approximated the true Pareto frontier during the process of evolution. The scheduling results of CE-MOPSO reflected the tradeoff between power generation and sediment deposition in ten years, which is useful for making reservoir operation policies of sediment trapping and flow operations in practice. These results demonstrate that the proposed CE-MOPSO can provide efficient and effective solutions for the optimization of the coordinative flow-sediment scheduling in a reservoir.

However, in CE-MOPSO, we need to define the maximal number of catfish particles, which has some subjective. In the future, a better approach for selecting the maximal number of catfish particles is desired to make a balance between computation time and performance of the algorithm. In addition, we would like to compare CE-MOPSO with some other evolutionary algorithms and apply CE-MOPSO for solving higher dimensional optimization problems.

Acknowledgments. This work is funded by the Project of National Key Research and Development Program (2016YFC0402308) and the National Natural Science Foundation of China (51679088).

DOI: http://dx.doi.org/10.15666/aeer/1705_1166911686

REFERENCES

[1] Abido, M. A. (2008): Multiobjective particle swarm optimization for optimal power flow problem. – 12th International Middle-East Power System Conf. IEEE Industrial Electronic Society, Aswan, pp. 392-396.

[2] Azadnia, A., Zahraie, B. (2010): Application of multi-objective particle swarm optimization in operation management of reservoirs with sediment problems. – World Environmental and Water Resources Congress: Challenges of Change, American Society of Civil Engineers, Rhode Island, pp. 2260-2268.

[3] Branke, J., Mostaghim, S. (2006): About selecting the personal best in multi-objective particle swarm optimization. – Parallel Problem Solving from Nature-PPSN IX, Reykjavik, pp. 523-532.

[4] Chuang, L. Y., Tsai, S. W., Yang, C. H. (2008): Catfish particle swarm optimization. – Swarm Intelligence Symposium IEEE, St. Louis, MO.

[5] Chuang, L. Y., Tsai, S. W., Yang, C. H. (2011): Improved binary particle swarm optimization using catfish effect for feature selection. – Expert Systems with Applications 38(10): 12699-12707.

[6] Coello, C. A. C., Pulido, G. T., Lechuga, M. S. (2004): Handling multiple objectives with particle swarm optimization. – IEEE Transactions on evolutionary computation 8(3): 256-79.

[7] De Souza, L., Ricardo, S., Prudêncio, B. C., Barros, F. de A. (2014): Multi-Objective Test Case Selection: A study of the influence of the Catfish effect on PSO based strategies. – Anais do XV Workshop de Testes e Tolerância a Falhas-WTF, pp. 3-16.

[8] Duan, Q., Ajami, N. K., Gao, X., Sorooshian, S. (2007): Multi-model ensemble hydrologic prediction using Bayesian model averaging. – Advance Water Resource 30(5): 1371-1386.

[9] Han, Q. (2003): Reservoir Sedimentation. – Science Press, Beijing (in Chinese).

[10] Han, Q., He, M. (2015): Mathematic Modelling of Non-Equilibrium Suspended Load Transport, Reservoir Sedimentation, and Fluvial Processes. – In: Yang, C. T., Wang, L. K. Advances in Water Resources Engineering. Springer, New York.

[11] Hu, B. (2004): Breaking Grounds. – Homa & Sekey Books, Paramus, NJ.

[12] Ji, C., Liu, F., Zhang, X. (2011): Particle swarm optimization based on catfish effect for flood optimal operation of reservoir. – Seventh International Conference on Natural Computation, IEEE, Shanghai, pp. 1197-1201.

[13] Kennedy, J., Eberhart, R. (1995): Particle swarm optimization. – Proceedings of the IEEE International Conference on Neural Networks 4: 1942-1948.

[14] Leong, W. F., Yen, G. G. (2006): Dynamic population size in PSO-based multiobjective optimization. – 2006 IEEE Congress on Evolutionary Computation, IEEE Industrial Electronic Society, Vancouver, pp. 6182-6189.

[15] Li, H., Lian, J. (2008): Multi-objective optimization of water-sedimentation-power in reservoir based on Pareto-optimal solution. – Transactions of Tianjin University 14: 282-288.

[16] Lian, J., Hu, M., Liu, Y. (2004): Research of multi-objective operation of water and sand in reservoir on sandy river. – Journal of Hydroelectric Engineering 23(2): 12-16 (in Chinese).

[17] Mostaghim, S., Teich, J. (2003): Strategies for finding good local guides in multi-objective particle swarm optimization. – Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Swarm Intelligence Symposium Institute of Electrical and Electronics Engineers, Indianapolis, pp. 26-33.

[18] Peng, Y., Zhang, H. (2006): 1-D numerical simulation of unsteady flow and sedimentation transport at the Three Gorges Reservoir (TGR). – Journal of Hydrodynamics A 21(3): 285-292 (in Chinese).

DOI: http://dx.doi.org/10.15666/aeer/1705_1166911686

[19] Peng, Y., Li, Y. T., Zhang, H. W. (2004): Multi-objective decision-making model for coordinative dispatch of water and sediment in reservoir. – Journal of Hydraulic Engineering 4: 1-7 (in Chinese).

[20] Peng, Y., Ji, C., Gu, R. (2014): A multi-objective optimization model for coordinated regulation of flow and sediment in cascade reservoirs. – Water Resources Management 28: 4019-4033.

[21] Reddy, M. J., Kumar, D. N. (2007): Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. – Hydrological Processes 21(21): 2897-909.

[22] Reddy, M. J., Kumar, D. N. (2009): Performance evaluation of elitist-mutated multi-objective particle swarm optimization for integrated water resources management. – Journal of Hydroinformatics 11(1): 79-88.

[23] Reyes-Sierra, M., Coello, C. A. C. (2005): Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. – Proc. of Evolutionary Multi-Criterion Optimization Conf. Guanajuato, pp. 505-519.

[24] Reyes-Sierra, M., Coello, C. A. C. (2006): Multi-objective particle swarm optimizers: a survey of the state-of-the-art. – International Journal of Computational Intelligence Research 2(3): 297-308.

[25] Tripathi, P. K., Bandyopadhyay, S., Pal, S. K. (2007): Adaptive multi-objective particle swarm optimization algorithm. – Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Congress on Evolutionary Computation, Singapore, pp. 2281-2288.

[26] van Rijn, L. C. (1993): Principles of Sediment Transport in Rivers, Estuaries and Coastal Seas. – Aqua Publication, Amsterdam, Netherland.

[27] Xiang, B., Ji, C. M., Peng, Y., Zhou, T. (2010): Study of water-sediment operating model based on immune particle swarm algorithm. – Journal of Hydroelectric Engineering 29(1): 97-101 (in Chinese).

[28] Xiao, Y., Peng, Y., Wang. T. (2013): Water-sediment coordinated optimized dispatch model of reservoir based on genetic algorithm and neural network. – Advances in Science and Technology of Water Resources 33(2): 9-13 (in Chinese).

[29] Yan, H., Moradkhani, H. (2014): Bayesian model averaging for flood frequency analysis. – World Environmental and Water Resources Congress, American Society of Civil Engineers, Portland, pp. 1886-1895.

[30] Yan, H., Moradkhani, H. (2016): Toward more robust extreme flood prediction by Bayesian hierarchical and multimodeling. – Natural Hazards 81: 203-225.

[31] Yang, G. L. (1993): River Mathematic Modelling. – China Ocean Press, Beijing (in Chinese).

[32] Yang, X. (2003): Manual on Sediment Management and Measurement. – Secretariat of the World Meteorological Organization, Geneva.

[33] Yoo, J. H. (2009): Maximization of hydropower generation through the application of a linear programming model. – Journal of Hydrology 376(1-2): 182-187.

[34] Yoon, Y. N. (1992): The state and the perspective of the direct sediment removal methods from reservoirs. – International Journal of Sediment Research 7(20): 99-115.

[35] Zhang, Y. X., Feng, S. Y. (1988): Multi-objective programming model in reservoir operation and its application. – Journal of Hydraulic Engineering 9: 19-27 (in Chinese).

[36] Zhu, J. (1997): Main measures that control the sedimentation in a reservoir: sediment operation. – Sichuan Water Conservancy 18(3): 6-10 (in Chinese).