Applied Soft Computing 34 (2015) 526–538

Contents lists available at ScienceDirect

Applied Soft Computing

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

A hybrid multi-objective PSO–EDA algorithm for reservoir flood control operation

Jungang Luo[a,b,*], Yutao Qi[c], Jiancang Xie[a,b], Xiao Zhang[b]

[a] State Key Laboratory Base of Eco-hydraulic Engineering in Arid Area, Xi'an University of Technology, Xi'an, China
[b] Institute of Water Resources and Hydro-Electric Engineering, Xi'an University of Technology, Xi'an, China
[c] School of Computer Science and Technology, Xidian University, Xi'an, China

ARTICLE INFO

Article history:
Received 13 August 2013
Received in revised form 26 March 2015
Accepted 25 May 2015
Available online 2 June 2015

Keywords:
Multi-objective optimization
Hybrid algorithm
Particle swarm optimization algorithm
Estimation of distribution algorithm
Reservoir flood control operation

ABSTRACT

Reservoir flood control operation (RFCO) is a complex multi-objective optimization problem (MOP) with interdependent decision variables. Traditionally, RFCO is modeled as a single optimization problem by using a certain scalar method. Few works have been done for solving multi-objective RFCO (MO-RFCO) problems. In this paper, a hybrid multi-objective optimization approach named MO-PSO–EDA which combines the particle swarm optimization (PSO) algorithm and the estimation of distribution algorithm (EDA) is developed for solving the MO-RFCO problem. MO-PSO–EDA divides the particle population into several sub-populations and builds probability models for each of them. Based on the probability model, each sub-population reproduces new offspring by using PSO based and EDA methods. In the PSO based method, a novel global best position selection method is designed. With the help of the EDA based reproduction, the algorithm can lean linkage between decision variables and hence have a good capability of solving complex multi-objective optimization problems, such as the MO-RFCO problem. Experimental studies on six benchmark problems and two typical multi-objective flood control operation problems of Ankang reservoir have indicated that the proposed MO-PSO–EDA performs as well as or superior to the other three competitive multi-objective optimization algorithms. MO-PSO–EDA is suitable for solving MO-RFCO problems.

1. Introduction

Flood disaster is one of the most serious, frequent and wide-spread natural disasters, especially in China. Influenced by monsoons from both Pacific and India oceans, 60% to 80% rainfalls in China are concentrated in the flood season, which increases the frequency and destruction strength of flood disaster. Reservoir plays an important role in flood management during flood seasons, it helps to minify flood peaks, reduce flood damages, and reserve flood [1].

Reservoir flood control operation (RFCO) is an important area of research in water resource management. As it involves more than one conflicting tasks, such as minimizing downstream damage and keeping dam safety within reasonable limits, the RFCO can be modeled as a multi-objective optimization problem (MOP) [2]. Further more, RFCO is a complicated decision problem with multi-objective, multi-scales, multiple constraints and multi-stage, it is a MOP with continuous and interdependent decision variables.

Traditionally, RFCO was treated as a single objective optimization problem by dealing with each optimization task sequentially or converting the MOP into single optimization problem with a mixed target function [3]. Many optimization techniques were also employed to solve this single objective RFCO problem. Some of them are heuristic algorithms, such as linear programming [4], dynamic programming [5] and non-linear programming [6]. With the development of computational intelligence, some nature inspired optimization techniques have been gradually applied to reservoir operation [7]. Genetic algorithm (GA) [8–10] is the representative approach of this type. These works have shown the significant potential of GA in water resources management and clearly demonstrated the advantages of GA over traditional techniques in terms of computational requirements. Ant colony optimization (ACO) algorithm which is inspired by the foraging behavior of some ant species was presented to solve continuous four reservoir operation problem [11]. Most recently, particle swarm optimization was also used for solving the reservoir operation problem [12].

[*] Corresponding author at: Institute of Water Resources and Hydro-Electric Engineering, Xi'an University of Technology, Xi'an, Shaanxi 710048, China. Tel.: +86 029 82312494; fax: +86 029 82312494.

E-mail addresses: jgluo@xaut.edu.cn (J. Luo), ytqi@xidian.edu.cn (Y. Qi), jcxie@xaut.edu.cn (J. Xie).

http://dx.doi.org/10.1016/j.asoc.2015.05.036
1568-4946/© 2015 Elsevier B.V. All rights reserved.

As RFCO is a MOP, it is not possible to find a single scheme simultaneously optimizing all objectives. Instead, the solution to RFCO becomes a set of good trade-offs between the multiple objectives. The trade-offs between the conflicting objectives are known as Pareto optimal schemes, for which any single objective cannot be improved without compromising at least one of the others. All the Pareto optimal schemes form the Pareto set (PS) of the RFCO and the Pareto optimal front (PF) is thus defined as the corresponding objective vectors of the schemes in Pareto optimal set. For solving RFCO, the above mentioned single-objective optimization algorithms have a drawback in common, they are usually sensitive to the shape or continuity of the PF of RFCO [13]. For example, non-convex parts of the Pareto front cannot be recovered by optimizing convex combinations of the objective functions. Superior to single-objective optimization algorithms, a multi-objective optimization algorithm can obtain a set of Pareto optimal schemes within a single run, which provides more information to decision makers.

In recent years, a variety of newly developed methods have been proposed to solve MOP. Multi-objective optimization has been one of the hottest research areas in the field of meta-heuristic and swarm intelligence techniques [14]. These multi-objective optimization techniques have also been employed to solve RFCO with multiple objectives and achieved various degrees of success. Chen et al. [15,16] summarized decision making problems of flood control and proposed multi-objective decision making theory, model and methods. Based on the fuzzy optimum model, Hou [17], Yu [18] and Fu [19] developed fuzzy decision-making methods concerning multiple objectives. Zhou et al. [20] introduced the theory of information entropy into fuzzy optimum model and proposed an entropy weights based multi-objective decision making approach for RFCO problem. Qin et al. proposed a multi-objective optimization model for RFCO problem and solved the optimization model using multi-objective differential evolution algorithm [21] and multi-objective cultured differential evolution algorithm [2] respectively. These two approaches were population based multi-objective optimization algorithms which can provide a set of Pareto optimal solutions with good diversity in a single run.

RFCO problem is a complex MOP because its decision variables are interdependent of one another [22]. So far, all of the above mentioned multi-objective optimization algorithms for solving RFCO problem simply employ the conventional reproduction operators designed for single-objective optimization algorithms. Few works have been done to improve the searching efficiency of the algorithm according to the characteristic of MOP, especially when decision variables of the target MOP are interdependent.

Particle swarm optimization (PSO) is a population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling which aim to find food. The process in PSO involves both social interaction and intelligence so that particles learn from their own experience and from experiences of other particles around them [23]. In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods. Recently, there has been a growing interest in multi-objective particle swarm optimization (MOPSO) which investigates PSO techniques for handling MOPs [24].

In this work, a hybrid multi-objective optimization algorithm combining PSO and estimation of distribution algorithm (EDA), simply MO-PSO–EDA for short, is developed to solve multi-objective RFCO problem. Although PSO converges fast, it is easy to fall into local optimum. Thus PSO is not suitable for solving complex optimization problems like RFCO problem whose decision variables are interdependent. In order to overcome the shortcoming of PSO, an EDA based reproduction method is introduced into MOPSO to form the proposed MO-PSO–EDA. EDA is an evolutionary computation optimization paradigm based on probabilistic modeling of promising solutions [25]. EDA aims to extract the distribution model of the population and discover the variable linkage information to benefit offspring generation. It has been proved that if the variable interaction structure of the probability model used in EDA is properly chosen, EDA could converge to global optimal solutions [26]. By taking the advantage of EDA [27], MO-PSO–EDA is expected to be suitable for solving multi-objective RFCO problem.

The remainder of this paper is organized as follows. Section 2 introduces some backgrounds include the MOP model and the workflow of the PSO algorithm. Section 3 gives the multi-objective optimization model for reservoir flood control operation. Section 4 presents the details of the proposed hybrid multi-objective PSO–EDA algorithm for solving RFCO problem. Section 5 briefly presents and analyzes the experimental results to validate our proposed approach. Section 5 concludes this paper and outlines future research work.

2. Related backgrounds

This section gives a brief introduction of the background of MOP. A literature review of the multi-objective PSO algorithm and its applications to the reservoir operation problems has been conducted.

2.1. The model of multi-objective optimization problem

A multi-objective optimization problem (MOP) with n decision variables and m objectives can be mathematically formulated as following:

$$\begin{array}{ll} \text{Minimize} & \mathbf{F}(x) = \left\{ f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}) \right\} \\ \text{Subject to} & \mathbf{x} \in \Omega \end{array} \qquad (1)$$

where Ω is the feasible region of the n-dimensional decision space, $\mathbf{x} = \{x_1, x_2, \ldots, x_n\} \in \Omega$ is the decision variable vector. The objective function $\mathbf{F}(\mathbf{x})$ consists of m objective functions $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$ and $\mathbf{R}^m$ is the objective space.

The objectives of MOP often conflict with each other. In other words, improvement of one objective may lead to deterioration of another. Therefore, a single solution that optimizes all objectives simultaneously does not exist. Suppose x_A and x_B are two solutions of a MOP, x_B is called to be dominated by x_A, or x_A dominates x_B, noted as $x_A \prec x_B$, if and only if $f_i(x_A) \leq f_i(x_B)$, $\forall\, i = 1, 2, \ldots, m$, and there exist a $j \in \left\{1, 2, \ldots, m\right\}$, which makes $f_j(x_A) < f_j(x_B)$. A solution $\boldsymbol{x}^*$ is a Pareto optimal solution, if there is no other feasible solution that dominates it. The set of all Pareto optimal solutions is called the Pareto optimal set (PS), which can be mathematically formulated as $PS \triangleq \{x^* | \neg \exists x \in \Omega, x \prec x^*\}$. The collection of corresponding objective vectors of the solutions in PS is called the Pareto optimal front (PF), that is $PF \triangleq \{F(x), x \in PS\}$.

Science it is impossible to find the whole PS of continuous MOPs, we aim at finding a finite set of Pareto optimal vectors which are uniformly scattered along the true PF, and thus good representatives of the entire PF.

2.2. The multi-objective particle swarm optimization algorithm

Particle swarm optimization (PSO) is a population-based stochastic optimization method which is inspired by the social behavior of bird flocking [23]. Due to its good performance and low computational cost, PSO has attracted significant researching attentions. In PSO algorithm, solutions are regarded as particles which are endowed with fitness values and velocities. A particle in the PSO system flies through the decision space of the target optimization problem by adjusting its velocity and position according to its own experience, represented by its personal best (pbest), as well as the experience of neighboring particles, represented by the global best (gbest).

Moore et al. [28] made the first attempt to extend the PSO algorithm to solve MOP. In their algorithm, Pareto dominance is used to generate a list of leaders that guide the search. Coello et al. [29] proposed a proposal for multiple objective particle swarm optimization which uses an external memory and a geographically based approach to maintain diversity. They first made comparisons between MOPSO with highly competitive multi-objective evolutionary algorithms. After these two pioneering works, a variety of PSO algorithms for handling multi-objective optimization problems have been developed in the last few decades [24]. Different from the PSO for single objective problems, the goal of multi-objective PSO (MOPSO) algorithm is to obtain a good representative of the entire PF of the target MOP. Therefore, MOPSO follows a set of best leaders in order to find more than one optimal solution. How to select the global and local best particles to guide the search of a particle is a key issue to be considered. Tripathi et al. [30] selected the global best particle from the non-dominated solutions by using roulette wheel selection. Liu et al. [31] utilized a tournament niche method to select the global best particle, and the local best particle is updated by the Pareto dominance. Wang et al. [32] developed a preference order to rank all the particles and thus to identified the global best particle. Elhossini et al. [33] combined PSO with evolutionary algorithm and selected the global best particle from the external archive by a tournament selection. The personal best particle is selected as the one with lowest strength Pareto fitness. Yazdani et al. [34] proposed a multi-objective PSO algorithm enhanced with a fuzzy logic-based controller. The algorithm used a number of fuzzy rules as well as dynamic membership functions to evaluate search spaces.

In order to prevent PSO algorithms from falling into local optimal, diversity preserving techniques are usually employed. Being a considerable technique for maintaining the swarm diversity in PSO, multi-swarm technique has attracted increasing attention. In recent years, many PSO algorithms with multiple swarms have been developed for MOPs. Coello et al. [35] adopted various clustering techniques to divide the population of particles into several swarms, with the aim of maintaining a better diversity. Janson et al. [36] clustered the particles into groups by using *k*-means approach and developed a multi-swam PSO for molecular docking problem. Leong et al. [37] developed a dynamic population strategy and integrated it with the multiple-swarm PSO to form an efficient algorithm for MOP. Several tricks, such as cell-based rank density estimation, population growing and declining, as well as adaptive local search, are designed to improve the algorithmic performance. Zhang et al. [38] developed a cooperative multi-objective particle swarm algorithm. The algorithm consists of multiple sub-swarms which connect each other with the ring topology. Liang et al. [39] proposed a dynamic multi-swarm PSO for MOPs. In this algorithm, the population is dynamically divided into several small sized swarms and regrouped randomly every other a few generations. They also applied this algorithm to large-scale portfolio optimization problem and achieved competitive performances [40]. Hu et al. [41] proposed a cooperative particle swarm optimization algorithm which employs multiple sub-swarms for solving routing recovery problem of wireless sensor networks with multiple mobile sinks. Britto et al. [42] developed an iterated multi-swarm algorithm based on archive for many-objective problems. In their algorithm, different swarms communicate with each other using the ring topology and have different achieving methods. Based on the iterated multi-swarm algorithm, Castro et al. [43] proposed a hybrid competent multi-swarm approach for many-objective problem. In this algorithm, the estimation of distribution algorithm is combined with multi-swarm PSO which has similar idea with our work. However, rather than taking advantage of the regularity of MOPs, they clustered the population into sub-swarms using a *k*-means algorithm in the decision space and built a Bayesian factorization graph to model the dependence between variables. In this work, the multi-swarm technique is employed to enhance the algorithm's performance on MOPs. Different from existing research works, we cluster the population into sub-swarms by using the local principal component analysis algorithm which takes advantage of the regularity of MOPs. Moreover, two complementary reproduction methods are used to generate new offspring which makes the algorithm efficient and robust.

2.3. Applications of PSO algorithm in reservoir operation

PSO algorithm has also been applied to the water resources optimization problems [44]. Nagesh Kumar et al. [45] implemented the PSO algorithm to multi-objective reservoir operation of the Bhadra reservoir system in India which serves irrigation and hydropower generation. To handle the multi-objective reservoir operation problem, a weighted approach was adopted. Baltar et al. [46] developed a multi-objective PSO algorithm to solve multipurpose reservoir operation problem with up to four objectives and problem of selective withdrawal from a thermally stratified reservoir with three objectives. This algorithm evaluates alternative solutions based on Pareto dominance and uses an external repository to store non-dominated solutions. A fitness sharing approach is employed to promote diversity and a mutation operator is used to improve global search. Wang et al. [47] developed a PSO algorithm with adaptive random inertia weight for optimizing reservoir operation. In this algorithm, the probability distribution of random inertia can be adjusted adaptively. Afshar [48] presented a partially constrained and a constrained PSO algorithm for the efficient solution of large scale reservoir operation problem which solves two problems of water supply and hydropower operation of Dez reservoir in Iran. Ostadrahimi et al. [49] presented and tested a set of operation rules for a multi-reservoir system using a multi-swarm PSO algorithm. The performance of the algorithm was examined by using three-reservoir system of Mica, Libby, and Grand Coulee. Guo et al. [50] developed a multi-objective PSO algorithm with multiple populations and non-dominated sorting for solving multi-reservoir operation problem.

3. Multi-objective optimization model for reservoir flood control operation

In this paper, a hybrid multi-objective PSO algorithm is developed to solve the following reservoir flood control operation problem which takes release volumes as the decision variables and serves two optimization goals. The multi-objective optimization model for RFCO problem (MO-RFCO) concerned in this paper can be mathematically formulated as following:

$$\begin{aligned}
&\text{Minimize}\quad F(Q) = (f_1(Q), f_2(Q))\\
&\qquad f_1(Q) = \max(Z_t)\\
&\qquad f_1(Q) = \max(Q_t)\\
&\qquad t = 1, 2, \ldots, T\\
&\text{Subject to :}\\
&\qquad (1)\ \ Z_{\min} \le Z_t \le Z_{\max}\\
&\qquad (2)\ \ 0 \le Q_t \le Q_{\max}\\
&\qquad (3)\ \ V_t = V_{t-1} + I_t - Q_t
\end{aligned} \tag{2}$$

In the above model, T is the total scheduling periods of the MO-RFCO problem. The decision vector $Q = (Q_1, Q_2, \ldots, Q_T)$ contains the release volumes of each scheduling periods. Z_t means the upstream water level of the tth scheduling period. Hence, the first optimization goal is to minimize the highest upstream water level during the T scheduling periods, taking release volumes as variables. This optimization goal expects the reservoir to store no large flood water volume to ensure the safety of the dam and decrease loss of upstream area. The second optimization goal is to minimize the largest release volume during scheduling periods, which expects the reservoir store as much flood water volume as possible to protect the downstream areas [51].

The MO-RFCO model has three constrains. Constraint (1) means the upstream water level Z_t must be a value between its lower bound $Z_{\min}$ and upper bound $Z_{\max}$. Constraint (2) defines that the release volume Q_t must be a nonnegative number no larger than its upper bound $Q_{\max}$. Constraint (3) is the water balance equation, in which V_t and V_{t-1} the reservoir storages of the t-th and the $(t-1)$-th scheduling period. I_t is the reservoir inflow volume of the t-th scheduling period. Using the water balance equation, the algorithm can calculate the upstream water level Z_t according to the reservoir's water volume, inflow volume and release volume Q_t of the t-th scheduling period.

4. A hybrid multi-objective PSO–EDA algorithm

By introducing an EDA based reproduction method into MOPSO, a hybrid algorithm named MO-PSO–EDA is developed in this work. The proposed MO-PSO–EDA has two types of reproduction methods to generate new solutions. One is the PSO method which performs a local search around the parent population and develops new searching areas. The other is the EDA based method which learns the variable linkages and promotes the algorithm's capability of solving complex problems such as the MO-RFCO problem.

4.1. Algorithm framework

The proposed MO-PSO–EDA maintains an evolving particle population with size N and an external elite population with size M. At each generation, the evolving particle population is divided into K sub-populations which perform reproduction steps separately. All the new generated individuals of each sub-population are merged together to update the external elite population. The workflow of the proposed MO-PSO–EDA can be described as the following Algorithm 1. Some of its details are illustrated in later subsections.

Algorithm 1 *(The main framework of the proposed MO-PSO–EDA.).*

```
Step 1: Initialization. Initialize the evolving particle population size $N$, the external elite population size $M$ and the sub-population number $K$. Set a stopping criterion. Set the iteration time $t = 0$.
    Step 1.1: Generate initial particle population $P(t) = \left\{Q_1(t), \ldots, Q_N(t)\right\}$ with size $N$ at random and evaluate all the individuals in $P(t)$.
    Step 1.2: Initialize the velocity of each particle in $P(t)$ at random to get the velocity vector $V(t) = \left\{V_1(t), \cdots, V_N(t)\right\}$.
    Step 1.3: Initialize the personal best position for each particle to its current position, that is pbest$(t) = P(t)$.
    Step 1.4: Set the initial elite population $E(t)$ as the non-dominated solutions in $P(t)$.
Step 2: Termination. If stopping condition is met, stop and output individuals in current $E(t)$ as the resulting approximate Pareto-optimal set; otherwise, continue.
Step 3: Population division and modeling. Cluster $P(t)$ into $K$ sub-populations $S_1, S_2, \ldots, S_K$ using local principal component analysis (PCA) algorithm [52], build probability model and get the centroids $C_1, C_2, \ldots, C_K$ for all clusters.
Step 4: Reproduction.
    Step 4.1: Reproduction by using EDA, giving rise to offspring set $P_{EDA}^1, P_{EDA}^2, \ldots, P_{EDA}^K$.
    Step 4.2: Reproduction by using PSO, giving rise to offspring set $P_{PSO}^1, P_{PSO}^2, \ldots, P_{PSO}^K$.
Step 5: Selection. Let $P(t+1) = P_{PSO}^1 \cup P_{PSO}^2 \cup \ldots \cup P_{PSO}^K$, $P_{EDA} = P_{EDA}^1 \cup P_{EDA}^2 \cup \ldots \cup P_{EDA}^K$. Update the elite population, let $E(t+1) = E(t) \cup P(t+1) \cup P_{EDA}$. Remove dominated solutions in $E(t+1)$. If the size of $E(t+1)$ is larger than $M$, calculate the crowding-distance [53] values of all elite individuals, sort them in descending order of crowding-distance, and keep the first M individuals in $E(t+1)$.
Step 6: Set $t = t + 1$, go to Step 2.
```

In the following, the implementation of Step 3 and two reproduction methods in Step 4 will be described in detail.

4.2. Population division and modeling

In the Step 3 of the proposed MO-PSO–EDA, the modeling method proposed by Zhang et al. [54] in RMMEDA algorithm is employed to build statistical model of current particle population and learn the linkages between decision variables.

Fig. 1 illustrates the basic idea of the division and modeling of current particle population. Given the sub-population number K ($K = 3$ in this example), the current particle population is divided into K clusters by using the local PCA algorithm. After that, probability models with linear centroid and zero-mean noises are built for each sub-population.

The probability model for the i-th sub-population ($i = 1, 2, \ldots, K$) can be described by the following equation:

$$\Psi_i = C_i + \varepsilon_i \tag{3}$$

Given the i-th sub-population S_i, note mean of S_i as $\bar{x}_i$. Let U_i^j be the j-th principal component which is a unity eigenvector associated with the j-th largest eigenvalue of the covariance matrix of the points in S_i. C_i in Eq. (3) can be described as follows:

$$C_i = \left\{ x \in R^n \,\middle|\, x = \bar{x}_i + \sum_{j=1}^{m-1} \alpha^j U_i^j, \quad \min_{x \in S_i} (x - \bar{x}_i)^T U_i^j \le \alpha^j \le \max_{x \in S_i} (x - \bar{x}_i)^T U_i^j \right\} \tag{4}$$

Suppose the target MOP has n decision variables and m objectives, C_i in Eq. (3) is a $(m-1)$-dimensional hyper-rectangle in the n-dimensional decision space. ε_i is an n-dimensional zero mean noise vector. Particularly, when the target MOP is a bi-objective problem, each C_i is a line segment in the n-dimensional decision space, as shown in Fig. 1. When the target MOP is a tri-objective problem, each C_i is a 2-D rectangle in the n-dimensional decision space.

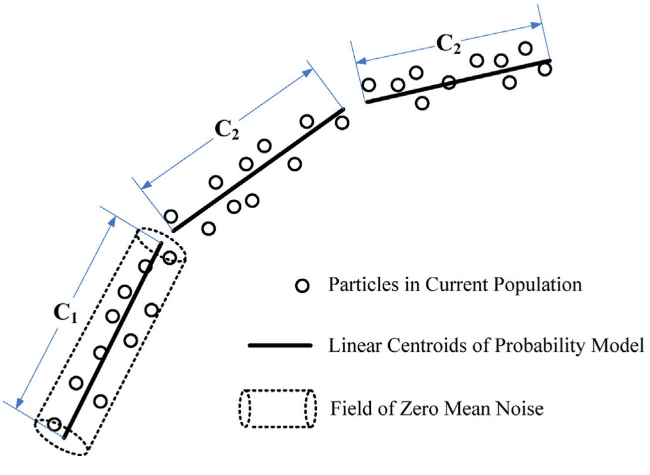

C2
C2
C1
Particles in Current Population
Linear Centroids of Probability Model
Field of Zero Mean Noise

Fig. 1. Illustration of the division and modeling of current particle population.

4.3. Reproduction by using EDA

The step of reproduction by using EDA is applied on each sub-population $S_i (i=1, 2, \ldots, K)$, giving rise to offspring set P_{EDA}^i. In order to improve population diversity, probability model in Eq. (3) is extended by 50% along each of the directions $U_i^1, U_i^2, \ldots, U_i^{m-1}$ to form the new probability model $Q_i^j(t)$, as is described in Eqs. (5)–(8), which is used to generate offspring individuals.

$$\hat{\Psi}_i = \hat{C}_i + \varepsilon_i \tag{5}$$

$$\hat{C}_i = \left\{ x \in R^n \,\middle|\, x = \bar{x}_i + \sum_{j=1}^{m-1} \alpha^j U_i^j, \quad \alpha_{\min} - 0.25\,(\alpha_{\max} - \alpha_{\min}) \right.$$
$$\left. \leq \alpha^j \leq \alpha_{\max} + 0.25\,(\alpha_{\max} - \alpha_{\min}) \right\} \tag{6}$$

$$\alpha_{\min} = \min_{x \in S_i} (x - \bar{x}_i)^T U_i^j \tag{7}$$

$$\alpha_{\max} = \max_{x \in S_i} (x - \bar{x}_i)^T U_i^j \tag{8}$$

Note $|S_i|$ as the particle number of the sub-population S_i, generate $|S_i|$ offspring particles by sampling the probability model $\hat{\Psi}_i$ uniformly and randomly, giving rise to the offspring set P_{EDA}^i.

The details of how to sample the probability model can be summarized as following Algorithm 2. It returns one sample vector from the probability model at each call.

Algorithm 2 *(Probability model sampling.).*

```
Step 1: Generate an integer $q \in \{1, 2, \ldots, K\}$ at random with the following
    probability, in which vol$(\Psi_i)$ denotes the $(m-1)$-dimensional volume of
    $\Psi_i$.
probability$(q = i) = \frac{\mathrm{vol}(\Psi_i)}{\sum_{j=1}^{K} \Psi_j}$
Step 2: Uniformly randomly generate a point $x'$ from $\hat{C}_q$ as defined in Eq. (6).
Step 3: Generate a $n$-dimensional zero mean noise vector $\varepsilon'$ from $N(0, \sigma_q I)$, in
    which $\sigma_q$ can be calculated by the following expression.
$\sigma_q = \frac{1}{n-m+1} \sum_{i=1}^{n} \lambda_i^q$
where $\lambda_i^q$ is the $i$-th largest eigenvalue of the covariance matrix of the points
    in sub-population $S_q$.
Step 4: Return the sample vector $x = x' + \varepsilon'$
```

4.4. Reproduction by using PSO

The step of reproduction by using PSO is applied on each sub-population $S_i (i=1, 2, \ldots, K)$, giving rise to offspring set P_{PSO}^i who has the same particle numbers as S_i. At the t-th iteration, we note $S_i = \left\{ Q_i^1(t), Q_i^2(t), \ldots, Q_i^{|S_i|}(t) \right\}$ and $P_{PSO}^i = \left\{ Q_i^1(t+1), Q_i^2(t+1), \ldots, Q_i^{|S_i|}(t+1) \right\}$. Each particle $Q_i^j(t) (j = 1, 2, \ldots, |S_i|)$ in S_i has its velocity $V_i^j(t)$, personal best position pbest$_i^j(t)$ and global best position gbest$_i^j(t)$. $V_i^j(t)$ and pbest$_i^j(t)$ are initialized in the Step 1 of Algorithm 1 and updated in the Step 2.2 at each iteration. gbest$_i^j(t)$ for $Q_i^j(t)$ is selected from the elite population $E(t)$ by using a novel designed method. The implementation details of reproduction by using PSO can be described as following Algorithm 3.

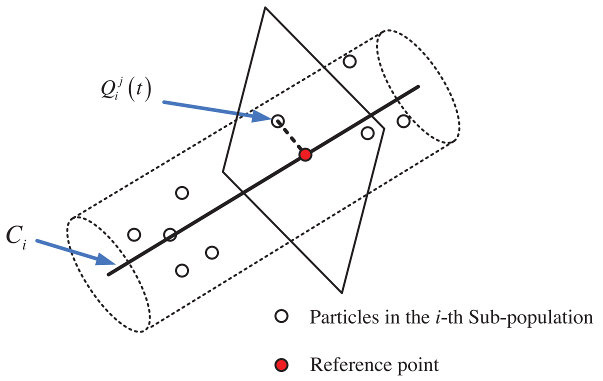


Fig. 2. Illustration of the reference point selection.

Algorithm 3 *(Reproduction by using PSO.).*

```
Step 1: Determine the global best position. Select a global best position
    gbest$_i^j(t)$ from current elite population $E(t)$ for each $Q_i^j(t) (j = 1, 2, \ldots, |S_i|)$
    in $S_i$, according to the reference point located at the centroid $C_i$ of the
    probability model $\Psi_i$.
Step 2: Update velocity. For each $Q_i^j(t)$, update the velocity $V_i^j(t)$ according
    to its personal best position pbest$_i^j(t)$ and global best position gbest$_i^j(t)$ by
    using
$V_i^j(t+1) = \omega V_i^j(t) + c_1 r_1(\mathrm{pbest}_i^j(t) - Q_i^j(t)) + c_2 r_2(\mathrm{gbest}_i^j(t) - Q_i^j(t))$
where $\omega$ is the inertia weight, it decreases linearly from 1.4 to 0.7 with the
    iterations. Acceleration coefficient values of $c_1$ and $c_2$ are set as 1.5 and 2.5
    respectively. $r_1$ and $r_2$ are two random numbers between 0 and 1.
Step 3: Update location. Calculate new position of each particle $Q_i^j(t)$ by
    using
$Q_i^j(t+1) = Q_i^j(t) + V_i^j(t+1)$
Step 4: Repairing. Note $Q_i^j(t+1) = (q_1, q_2, \ldots, q_n)$, if any $q_s (s = 1, 2, \ldots, n)$
    has a value outside the range $[q_{\min}, q_{\max}]$, let $q_s = r_3 q_{\max}$ and
    $V_i^j(t+1) = -r_4 V_i^j(t+1)$, where $r_3$ and $r_4$ are two random numbers
    between 0 and 1.
Step 5: Update personal best position. Evaluate $Q_i^j(t+1)$, if $Q_i^j(t+1)$
    dominates pbest$_i^j(t)$ then replace pbest$_i^j(t)$ with $Q_i^j(t+1)$, otherwise, keep
    pbest$_i^j(t)$ unchanged.
Step 6. Return results. Let $P_{PSO}^i = \left\{ Q_i^1(t+1), Q_i^2(t+1), \ldots, Q_i^{|S_i|}(t+1) \right\}$
    and return $P_{PSO}^i$.
```

In Algorithm 3, the global best position gbest$_i^j(t)$ for each particle $Q_i^j(t)$ is determined by a novel designed global best position selection method. The developed gbest selection method consists of two steps. First, determine a reference point which is the projection point of $Q_i^j(t)$ on the centroid C_i of the probability model Ψ_i. For bi-objective RFCO problem, the centroid of the probability model is a line segment in the decision space. The reference point of $Q_i^j(t)$ is the intersection of C_i with the hyper-plane who takes C_i as normal vector and with $Q_i^j(t)$ on it, as shown in Fig. 2.

Second, select the particle, who is the nearest to the reference point in the decision space, from current elite population $E(t)$, and let it be the gbest$_i^j(t)$ of $Q_i^j(t)$.

5. Experimental study

In the experimental studies, the proposed MO-PSO–EDA is first tested on some benchmark problems with different characteristics in Section 5.2. Then in Section 5.3, MO-PSO–EDA is applied to flood control operation on Ankang reservoir in Shanxi province of China.

5.1. General experimental setting and performance metric

In MO-PSO–EDA, the evolving particle population size N and the external elite population size M are both set as 100. The sub-population number K is set as 5.

Table 1
Multi-objective optimization test problems.

Prob.	Var.	Dim.	Objective functions (minimized)
ZDT1	$x_i \in [0, 1]$ $i = 1, \ldots, n$	10	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT3	$x_i \in [0, 1]$ $i = 1, \ldots, n$	10	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)} - x_1/g(x)\sin(10\pi x_1)]$ $g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$
ZDT4	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$ $i = 2, \ldots, n$	10	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4\pi x_i)]$
ZDT6	$x_i \in [0, 1]$ $i = 1, \ldots, n$	10	$f_1(x) = 1 - e^{(-4x_1)\times[\sin(6\pi x_1)]^6}$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9\left[\left(\sum_{i=2}^{n} x_i\right)/(n-1)\right]^{0.25}$
F1	$x_i \in [0, 1]$ $i = 1, \ldots, n$	10	$f_1(x) = 1 - e^{(-4x_1)\times[\sin(6\pi x_1)]^6}$ $f_2(x) = g(x)\left[1 - (f_1(x)/g(x))^2\right]$ $g(x) = 1 + 9\left[\sum_{i=2}^{n}(x_i^2 - x_1)^2/(n-1)\right]^{0.25}$
F2	$x_1 \in [0, 1]$ $x_i \in [0, 10]$ $i = 2, \ldots, n$	10	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 1/4000\sum_{i=2}^{n}(x_i^2 - x_1)^2 - \prod_{i=2}^{n}\cos(x_i^2 - x_1/\sqrt{i-1}) + 2$

In order to evaluate the quality of non-dominated solution sets found by the comparing algorithms, the hyper-volume (HV) performance measure proposed by Zitzler et al. [55] is employed in this work. The HV value corresponds to the non-overlapped volume of all the hyper-cubes formed by a reference point R. In this work, the reference point R is set by maximum values of non-dominated solutions obtained by the compared algorithms. Note p as a non-dominated solution in the Pareto approximation set PF_m of an m-dimensional MOP. The hyper-volume performance measure can be mathematically defined as follows:

$$\mathbf{HV} = \Lambda\left(\bigcup_{p \in PF_m}\left\{x \,\middle|\, p \prec x \prec R\right\}\right) \tag{9}$$

In which, Λ is the Lebesgue measure, $R \in \mathbf{R}^m$ is the user-defined reference point in the m-dimensional objective space, which is dominated by all valid candidate solutions in PF_m.

5.2. Performance comparisons on benchmark problems

In this part of experimental studies, the proposed MO-PSO–EDA algorithm is compared with three other competitive multi-objective optimization algorithms, NSGAII [53], RMMEDA [54] and ClustMPSO [36], on solving 8 well-known multi-objective benchmark problems with different characteristics. Among the compared algorithms, NSGAII is known as effective and robust evolutionary multi-objective algorithm, ClustMPSO is a multi-objective PSO algorithm which clusters the particles into groups just like MO-PSO–EDA, RMMEDA is an effective EDA based algorithm for solving MOP.

Four ZDT problems developed by Zitzler [56] and two complex problems proposed by Zhang et al. [54] are selected as test problems to validate the effectiveness of the propose approach. All these test instances are minimization bi-objective problems, their expressions are listed in detail in the following Table 1. Among these six test problems, ZDT1 and ZDT3 have convex and discontinuous PF respectively, ZDT4 has many local PFs, and ZDT6 has low density of solutions near the Pareto front. F1 and F2 have nonlinear interdependent decision variables. The PS of problem F1 is a bounded continuous curve in the decision space. The F2 problem has many local PFs which makes it hard to solve.

Fig. 3 illustrates the final non-dominated fronts with the largest HV values found by MO-PSO–EDA and the other three comparing algorithm NSGAII, RMMEDA, ClustMPSO respectively on six test problems over 30 independent runs. The stop criteria of all the compared algorithms are set as follows, each simulation continues until the total function evaluation number reaches the maximum value 10,000.

It can be seen from Fig. 3 that the proposed MO-PSO–EDA performs well on the six benchmark problems both in term of coverage and uniformity. For ZDT problems, RMMEDA performs not as well as MO-PSO–EDA, NSGAII and ClustMPSO. MO-PSO–EDA performs better than NSGAII and ClustMPSO in term of uniformity. For F1 and F2 problems, MO-PSO–EDA and RMMEDA perform better than NSGAII and ClustMPSO, which implies that the EDA based reproduction method is suitable for solving complex MOPs with nonlinearly correlated decision variables.

Table 2 compares the HV values obtained by the compared algorithms on six benchmark problems. For hyper volume calculation, we used the upper bound reference point containing the maximum of ideal PF for each objective. The experimental data in Table 2

Table 2
Performances on benchmark problems: mean HV value (standard deviation within parenthesis, the best value in each row is marked in bold and italics).

Problem	Algorithms			
	MO-PSO–EDA	NSGAII	RMMEDA	ClustMPSO
ZDT1	6.512e−01 (6.900e−03)	***6.522e−01 (7.100e−03)***	6.315e−01 (2.860e−02)	6.386e−01 (2.260e−02)
ZDT3	***7.613e−01 (1.772e−02)***	7.432e−01 (2.913e−02)	7.226e−01 (8.520e−03)	7.318e−01 (4.669e−02)
ZDT4	***6.537e−01 (4.118e−01)***	6.512e−01 (3.922e−01)	0.000e+00 (0.000e+00)	6.144e−01 (3.903e−01)
ZDT6	2.431e−01 (6.490e−03)	***2.518e−01 (1.174e−02)***	2.204e−01 (1.033e−02)	2.315e−01 (3.146e−02)
F1	***2.507e−01 (1.067e−02)***	2.311e−01 (1.304e−02)	2.455e−01 (8.700e−04)	2.054e−01 (2.302e−02)
F2	***6.445e−01 (1.305e−02)***	6.312e−01 (2.632e−02)	6.471e−01 (2.120e−03)	6.155e−01 (3.582e−02)

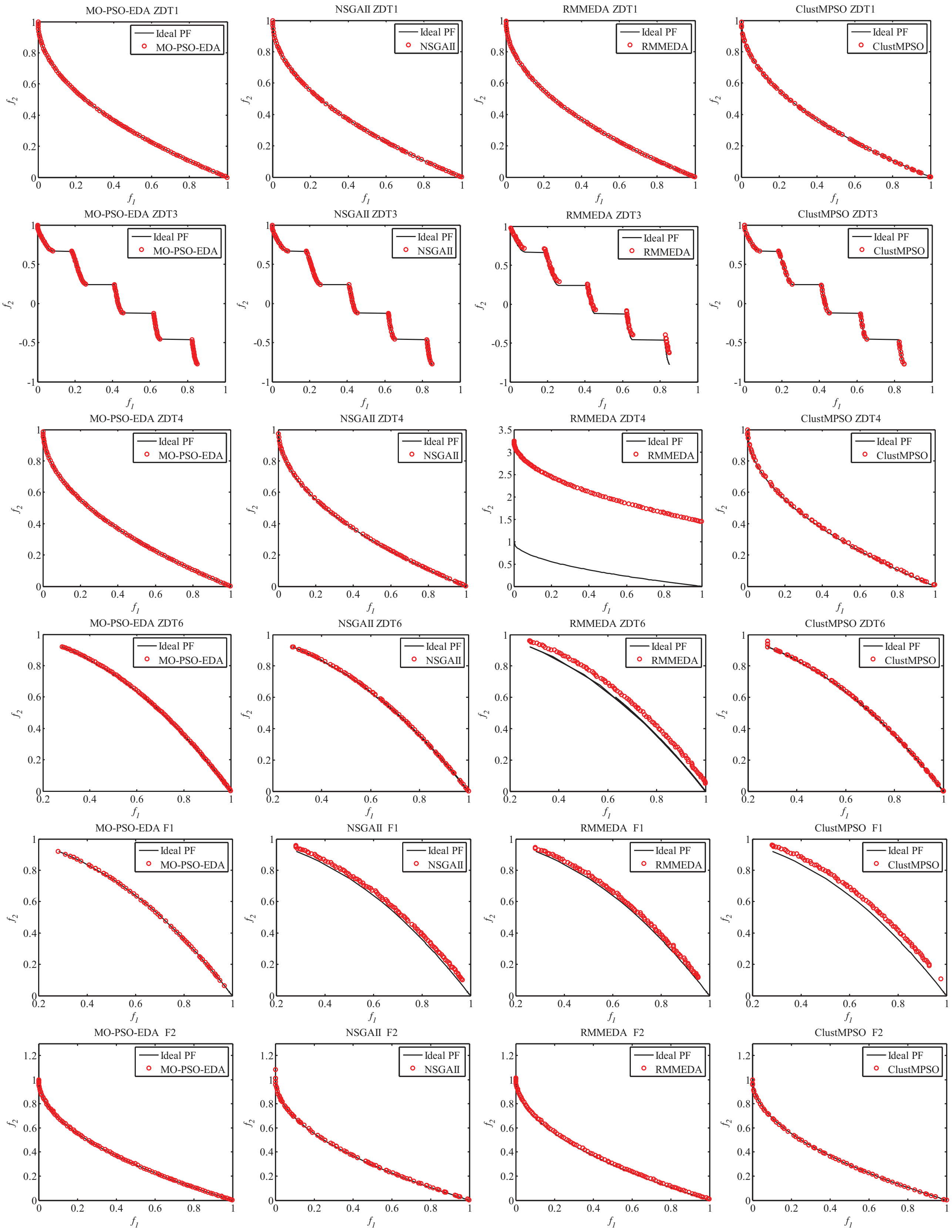


Fig. 3. Performance comparisons on the six benchmark problems.

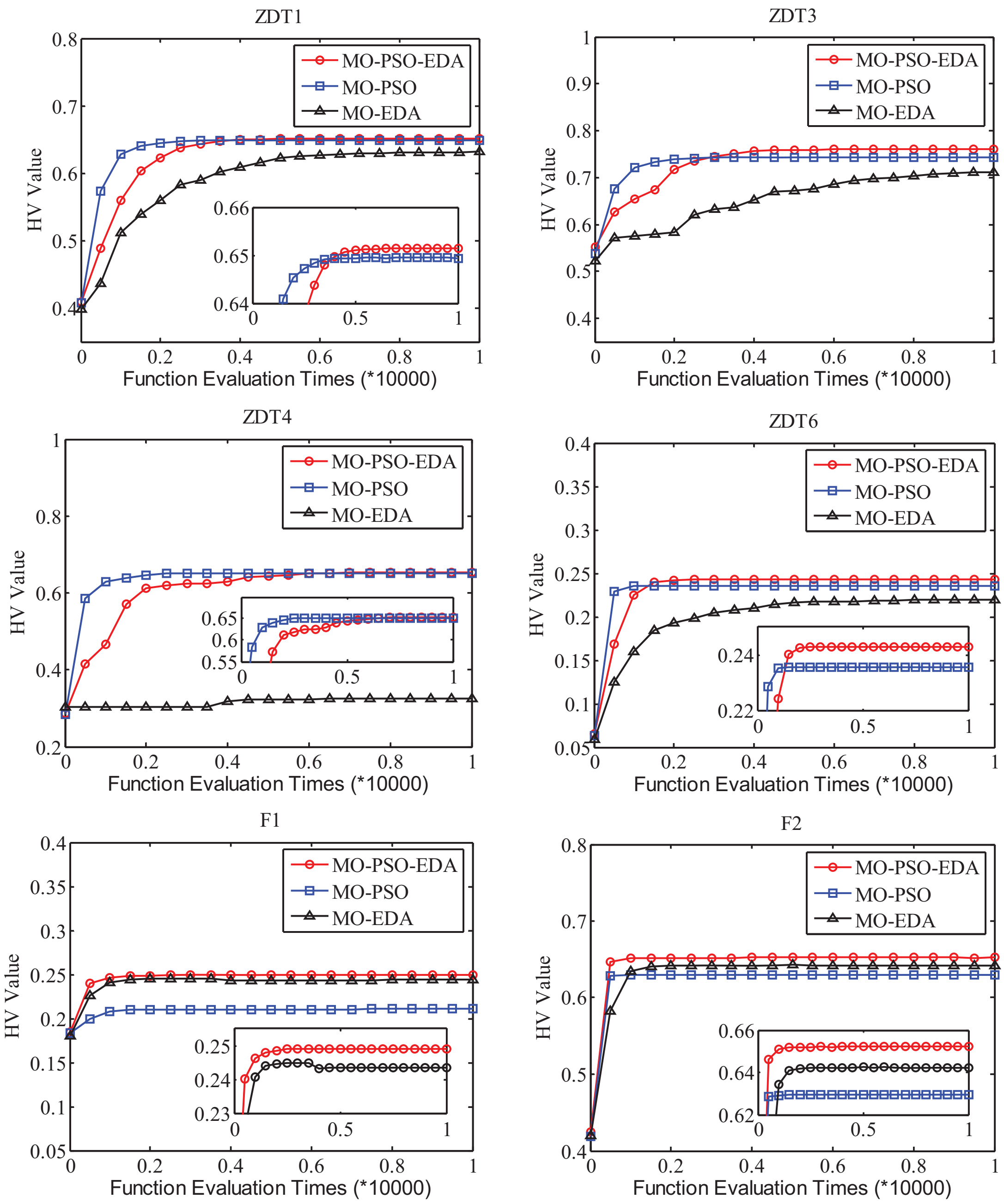


Fig. 4. Comparisons of the HV curves between MO-PSO–EDA with its two variations.

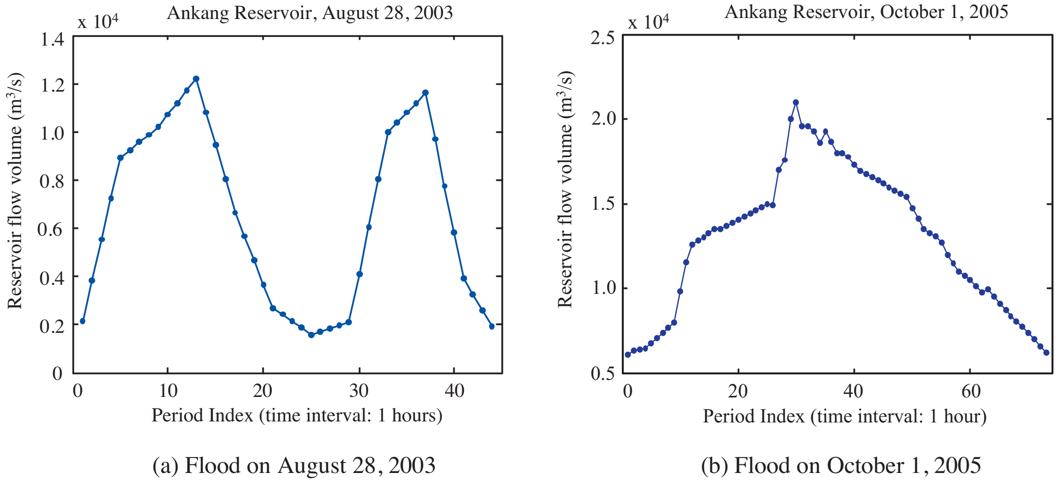
are statistic values over 30 independent runs. The largest mean HV value in each row is highlighted in bold and italic type, which means the corresponding algorithm performs best on the benchmark problem.

It can be seen from Fig. 3 that the proposed MO-PSO–EDA can obtain non-dominated fronts with good coverage and uniformity. It performs as well as or superior to the compared algorithm on the six benchmark problems. From Table 2 we can see that the proposed MO-PSO–EDA performs as well as NSGAII in solving ZDT problems. Both of them are superior to RMMEDA and ClustMPSO. For F1 and F2 problems, MO-PSO–EDA performs the best. Moreover, RMMEDA is superior to NSGAII in solving these two problems.

From this part experimental study we can come to the conclusion that EDA based algorithms perform well on F1 and F2 problems whose decision variables are interdependent. The algorithm ClustMPSO performs well in average, it has a good capability of escaping from local optimal. However, ClustMPSO has a large variance of HV values which mean the algorithm is not stable. By taking advantage of the two types of algorithms, the proposed MO-PSO–EDA outperforms the other three competitive multi-objective optimization algorithms.

The proposed MO-PSO–EDA combines the PSO based and the EDA based reproduction methods which cooperate with each other. In order to demonstrate the performance of the combination, MO-PSO–EDA is compared with its two variations MO-PSO and MO-EDA in which only one of the two reproduction methods is applied solely. More specifically, MO-PSO employs the PSO based reproduction only, while MO-EDA only uses the EDA based method to generate new offspring.

Fig. 4 compares the HV curves of the proposed MO-PSO–EDA with its two variations MO-PSO and MO-EDA. As shown in Fig. 4, MO-PSO–EDA converges to higher HV values on the six benchmark



(a) Flood on August 28, 2003

(b) Flood on October 1, 2005

Fig. 5. Water inflow volumes of the two typical floods. (a) Flood on August 28, 2003 (b) Flood on October 1, 2005.

problems. These experimental results imply that the combination of the two reproduction methods make MO-PSO–EDA performs better than MO-PSO and MO-EDA. It can be also seen that MO-PSO converges faster than MO-PSO–EDA and MO-EDA on the four ZDT problems. However, when dealing with F1 and F2 problems whose decision variables are nonlinearly interdependent, MO-PSO–EDA and MO-EDA perform significantly better than MO-PSO. From these experimental results we can come to conclusion that the PSO based reproduction method helps the algorithm to converge fast on ZDT problems which are relatively simple. When dealing with complex problem with correlated decision variables, like F1 and F2 problems, the EDA based reproduction method plays an important role on enhancing the performance of the algorithm. By combing these two complementary reproduction methods, the proposed MO-PSO–EDA performs well and stably on the investigated problems.

5.3. *Multi-objective flood control operation of Ankang reservoir*

In this subsection, the proposed MO-PSO–EDA is applied to solve real-world MO-RFCO problems on Ankang reservoir in Shanxi Province of China. Ankang reservoir on the Hanjiang river is a large reservoir with the purpose of flood protection, power generation, navigation, water supply, and so on. The Ankang reservoir was first built in October 1982 with an area of 77.5 km^2, and maximum water capacity of 2.585 bn. m^3. It has a normal water level of 330 m, a flood control limit level of 325 m and a dead water level of 300 m. The Designed discharge capacity of Ankang reservoir is 37,474 m^3 s^{-1}.

In this work, we use discharge volume as the decision variable to encode the particles in the MO-PSO–EDA algorithm. Each particle in the population can be coded as a series of discharge volumes during T scheduling periods, that is $Q = (Q_1, Q_2, \ldots, Q_T)$ where $Q_t(t = 1, 2, \ldots, T)$ is the discharge volume of the individual in the t-th period. The RFCO model is a multi-objective model which has been defined in Section 3.

Two typical floods which were occurred on August 28, 2003 and October 1, 2005 respectively are investigated. Fig. 5(a) and (b) shows the reservoir inflow volumes of these two floods. As shown in Fig. 5, the flood on August 28, 2003 has two flood peaks with maximum inflow volume of 12,200 m^3 s^{-1}. The flood on October 1, 2005 has one flood peaks with maximum inflow volume of 21,000 m^3 s^{-1}.

Fig. 6 shows the non-dominated solution sets with the largest HV values obtained by the proposed MO-PSO–EDA and the comparing algorithm NSGAII, RMMEDA, ClustMPSO respectively on the MO-RFCO problems over 30 independent runs. The stop criteria of all the compared algorithms are set as follows, each simulation continues until the total function evaluation number reaches the maximum value 300,000. The dispatching time interval of the MO-RFCO problems on August 28, 2003 and October 1, 2005 are 3 h and 4 h respectively.

Considering the demand of irrigation and water supply, scheduling schemes with water level of 325 m at the end of the flood should be of interest to decision makers. It can be seen from Fig. 6 that the proposed MO-PSO–EDA can obtain more solutions with highest upstream water level between 320 m and 320 m than the other three comparing algorithm. In other words, MO-PSO–EDA can provide more effective decision-making supporting information for dispatchers.

Table 3 illustrates the HV values obtained by the compared algorithms on two MO-RFCO problems. For hyper volume calculation, we used the upper bound reference point containing the maximum of non-dominated solution found by all the comparing algorithms for each objective. The experimental results in Table 3 are statistic values over 30 independent runs. As shown in Table 3, MO-PSO–EDA obtains higher mean HV values on both of the two typical floods, which means MO-PSO–EDA performs better than the other two compared algorithm in solving the real world MO-PFCO problem.

Given the non-dominated solution set with the largest HV values obtained by the proposed MO-PSO–EDA, Fig. 7 shows the changes of upstream water level during dispatching periods of the two typical floods. It can be seen form Fig. 7 that MO-PSO–EDA provides a set of dispatching schemes with good diversity. All the dispatching schemes keep the upstream water level lower than 360 meters, which ensures the safety of the upstream of the reservoir. In order to store water, scheduling schemes with water level of 325 m at the end of the flood are preferred. Table 4 lists the details of the top 10 preferred scheduling schemes which are marked with asterisks in Fig. 7.

Fig. 8 shows discharge volumes of the preferred scheduling schemes obtained by the proposed MO-PSO–EDA. From Fig. 8 we can see that all the scheduling schemes have stable and low discharging volumes during dispatching periods, which ensures the safety of the downstream of the reservoir.

To conclude, the proposed MO-PSO–EDA performs well on solving MO-RFCO problems both in view of multi-objective optimization algorithm and the demands of the application itself.

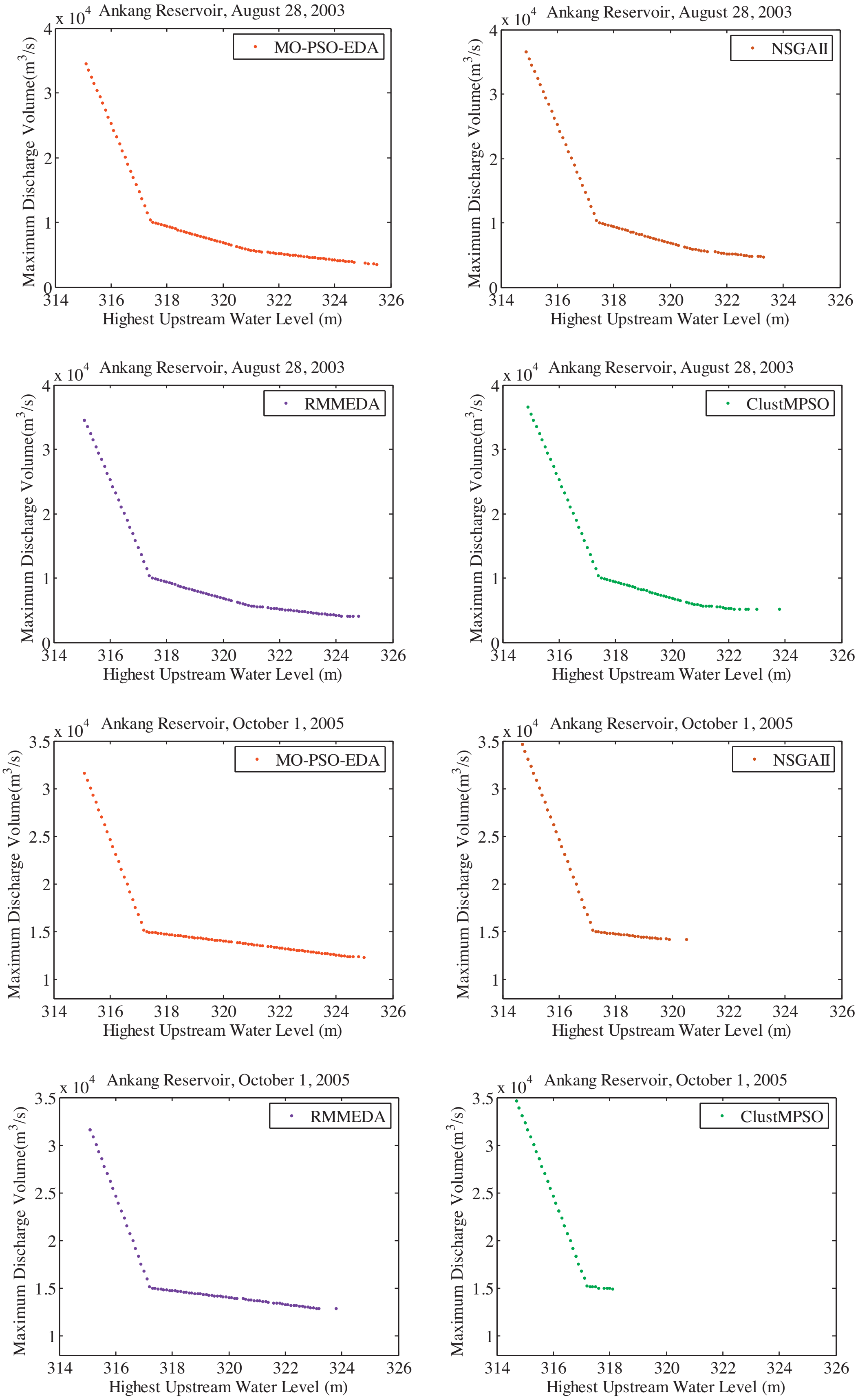


Fig. 6. Performance comparisons on the two typical floods.

Table 3
Performances on MO-RFCO problems: mean HV value (standard deviation within parenthesis, the best value in each row is marked in bold and italics).

Problem	Algorithms			
	MO-PSO–EDA	NSGAII	RMMEDA	ClustMPSO
MO-RFCO 2003.08.28	***1.0625e+009 (4.03e+005)***	1.0222e+009 (3.18e+005)	1.0428e+009 (3.76e+005)	1.0093e+009 (1.72e+006)
MO-RFCO 2005.10.01	***8.0685e+008 (3.55e+007)***	7.4502e+008 (2.68e+007)	7.8905e+008 (3.14e+007)	7.2157e+008 (6.88+007)

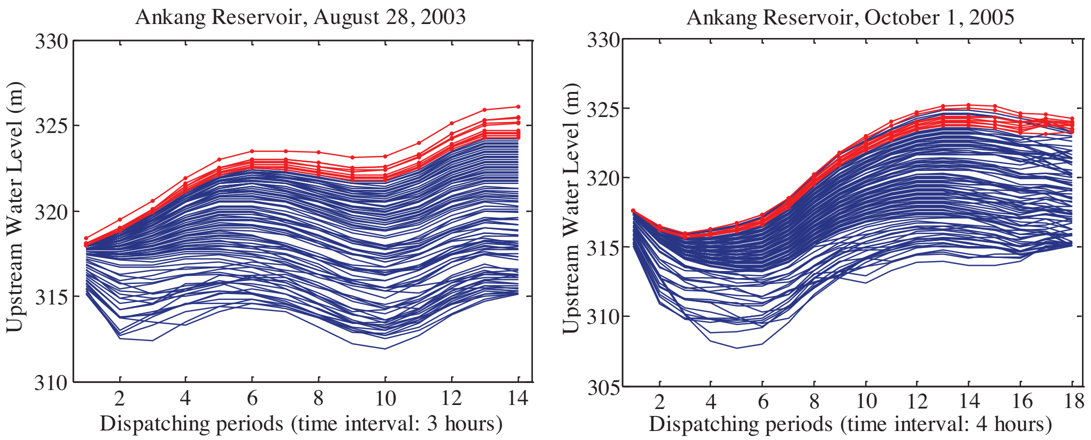


Fig. 7. Changes of upstream water level during dispatching periods (non-dominated solution set with the largest HV values obtained by the proposed MO-PSO–EDA).

Table 4
Details of the top 10 preferred scheduling schemes (f_1 value: highest upstream water level, f_2 value: maximum discharge volume, FWL: final water level after dispatching).

Ankang reservoir, August 28, 2003				Ankang reservoir, October 1, 2005			
Scheme num.	f_1 value (m)	f_2 value (m^3 s^{-1})	FWL (m)	Scheme num.	f_1 value (m)	f_2 value (m^3 s^{-1})	FWL (m)
1	324.3	4017.9	324.3	1	323.7	12,681	323.3
2	324.4	3982.3	324.4	2	325	12,314	323.4
3	324.5	3926.9	324.5	3	323.9	12,604	323.5
4	324.6	3889.6	324.6	4	323.8	12,637	323.7
5	324.7	3803.9	324.7	5	324	12,560	323.8
6	325.1	3746.9	325.1	6	324.4	12,403	323.8
7	325.2	3596.1	325.2	7	324.1	12,524	323.9
8	325.4	3539.2	325.4	8	324.3	12,458	323.9
9	325.5	3499.2	325.5	9	324.2	12,474	324.0
10	326.1	3472.5	326.1	10	325.2	12,314	324.2

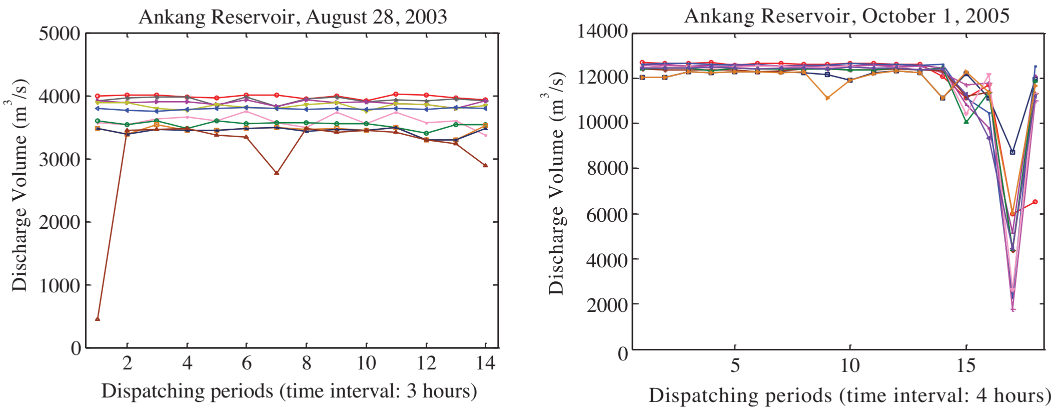


Fig. 8. Changes of discharge volume during dispatching periods. (The top 10 preferred scheduling schemes obtained by the proposed MO-PSO–EDA.)

6. Conclusions and future works

By introducing an EDA based reproduction method into MOPSO, a hybrid algorithm named MO-PSO–EDA is developed in this work. In MO-PSO–EDA, the particle population is divided into several sub-populations and probability models with linear centroid and zero-mean noises are built for each sub-population. Based on the probability model, each sub-population reproduces new offspring by using two types of method. One is the PSO based method, in which a novel global best position selection method is designed according to the reference point on the centroid of the probability model of the sub-population. The other one is EDA based method, which learns the variable linkages and promotes the algorithm's capability of solving complex multi-objective optimization problems.

Experimental studies on six benchmark problems and two typical multi-objective flood control operation problems of Ankang reservoir have indicated that the proposed MO-PSO–EDA performs as well as or superior to the other three competitive multi-objective optimization algorithms, especially when decision variables of the target problem are interdependent. As the multi-objective reservoir flood control operation problem is a multi-objective optimization problem with interdependent decision variables, the proposed MO-PSO–EDA is suitable for solving this problem.

Further research efforts may focus on providing more comprehensive and effective decision-making supporting information for dispatchers by considering the demand of power generation and using more accurate estimation of flood damages.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 51109175 and 61303119, the Science and Technology Program of Shaanxi Province under Grant Nos. 2014K09-07 and 2015KJXX-30, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20126118110011.

Appendix. List of abbreviations

RFCO: reservoir flood control operation
MOPs: multi-objective optimization problems
MO-RFCO: multi-objective reservoir flood control operation
PSO: particle swarm optimization
EDA: estimation of distribution algorithm
GA: genetic algorithm
ACO: ant colony optimization
MOPSO: multi-objective particle swarm optimization
PS: Pareto optimal set
PF: Pareto optimal front
PCA: principal component analysis
RMMEDA: regularity model-based multi-objective estimation of distribution algorithm
NSGAII: non-dominated sorting genetic algorithm II
ClustMPSO: a multiple swarm multi-objective particle swarm optimization
HV: hyper-volume

References

[1] P. Hlavinek, Hazards, vulnerability and mitigation measures of water supply and sewerage systems, in: P. Hlavinek, C. Popovska, J. Marsalek, I. Mahrikova, T. Kukharchyk (Eds.), Risk Management of Water Supply and Sanitation Systems, Springer, Netherlands, 2009, pp. 3–12.

[2] H. Qin, J.Z. Zhou, Y.L. Lu, Y.H. Li, Y.C. Zhang, Multi-objective cultured differential evolution for generating optimal trade-offs in reservoir flood control operation, Water Resour. Manag. 24 (2010) 2611–2632.

[3] S. Hajkowicz, K. Collins, A review of multiple criteria analysis for water resource planning and management, Water Resour. Manag. 21 (2007) 1553–1566.

[4] J. Needham, D. Watkins Jr., J. Lund, S. Nanda, Linear programming for flood control in the Iowa and Des Moines rivers, J. Water Resour. Plan. Manag. 126 (2000) 118–127.

[5] S. Yakowitz, Dynamic programming applications in water resources, Water Resour. Res. 18 (1982) 673–696.

[6] O. Unver, L. Mays, Model for real-time optimal flood control operation of a reservoir system, Water Resour. Manag. 4 (1990) 21–46.

[7] M.S. Hossain, A. El-shafie, Intelligent systems in optimizing reservoir operation policy: a review, Water Resour. Manag. 27 (2013) 3387–3407.

[8] S.B. Wang, X.J. Cang, K. Kong, Application of adaptive genetic algorithm in optimization of reservoir operation, J. Hydraul. Eng. 37 (2006) 480–485.

[9] M. Karamouz, O. Abesi, A. Moridi, A. Ahmadi, Development of optimization schemes for floodplain management: a case study, Water Resour. Manag. 23 (2009) 1743–1761.

[10] B. Malekmohammadi, B. Zahraie, R. Kerachian, A real-time operation optimization model for flood management in river–reservoir systems, Nat. Hazards 53 (2010) 459–482.

[11] M.R. Jalali, A. Afshar, M.A. Mariño, Multi-colony ant algorithm for continuous multi-reservoir operation optimization problem, Water Resour. Manag. 21 (2007) 1429–1447.

[12] M.H. Afshar, Extension of the constrained particle swarm optimization algorithm to optimal operation of multi-reservoirs system, Int. J. Electr. Power Energy Syst. 51 (2013) 71–81.

[13] C.A.C. Coello, G.B. Lamont, D.A. Van Veldhuisen, Evolutionary Algorithms for Solving Multi-objective Problems, Springer, 2007.

[14] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, Q.F. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (2011) 32–49.

[15] S.Y. Chen, Multiobjective decision making theory and model for floodcontrol operation, Eng. Sci. 2 (2000) 6.

[16] S.Y. Chen, Z.C. Hou, Multicriterion decision making for flood control operations: theory and applications, J. Am. Water Resour. Assoc. 40 (2004) 67–76.

[17] Z.C. Hou, S.Y. Chen, Multi-objective fuzzy group decision-making method for reservoir flood control operation, J. Hydraul. Eng. 35 (2004) 106–111, 119.

[18] Y.B. Yu, B.D. Wang, G.L. Wang, W. Li, Multi-person multiobjective fuzzy decision-making model for reservoir flood control operation, Water Resour. Manag. 18 (2004) 111–124.

[19] G.T. Fu, A fuzzy optimization method for multicriteria decision making: an application to reservoir flood control operation, Exp. Syst. Appl. 34 (2008) 145–149.

[20] H. Zhou, G. Zhang, G. Wang, Multi-objective decision making approach based on entropy weights for reservoir flood control operation, J. Hydraul. Eng. 38 (2007) 100–106.

[21] H. Qin, J.Z. Zhou, G.Q. Wang, Y.C. Zhang, Multi-objective optimization of reservoir flood dispatch based on multi-objective differential evolution algorithm, J. Hydraul. Eng. 40 (2009) 513–519.

[22] M.J. Reddy, D.N. Kumar, Multiobjective differential evolution with application to reservoir system optimization, J. Comput. Civ. Eng. 21 (2007) 136–146.

[23] M. Hu, T. Wu, J.D. Weir, An intelligent augmentation of particle swarm optimization with multiple adaptive methods, Inf. Sci. 213 (2012) 68–83.

[24] S. Mohankrishna, D. Maheshwari, P. Satyanarayana, S.C. Satapathy, A comprehensive study of particle swarm based multi-objective optimization, in: S.C. Satapathy, P.S. Avadhani, A. Abraham (Eds.), Proceedings of the International Conference on Information Systems Design and Intelligent Applications, 2012, pp. 689–701.

[25] P. Larrañaga, J.A. Lozano, Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation, Springer, 2002.

[26] Q.F. Zhang, H. Muhlenbein, On the convergence of a class of estimation of distribution algorithms, IEEE Trans. Evol. Comput. 8 (2004) 127–136.

[27] Y.T. Qi, F. Liu, M.Y. Liu, M.G. Gong, L.C. Jiao, Multi-objective immune algorithm with Baldwinian learning, Appl. Soft Comput. 12 (2012) 2654–2674.

[28] J. Moore, R. Chapman, Application of Particle Swarm to Multiobjective Optimization, Department of Computer Science and Software Engineering, Auburn University, 1999.

[29] C.A.C. Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, IEEE Press, 2002, pp. 1051–1056.

[30] P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients, Inf. Sci. 177 (2007) 5033–5049.

[31] D.S. Liu, K.C. Tan, S. Huang, C.K. Goh, W.K. Ho, On solving multiobjective bin packing problems using evolutionary particle swarm optimization, Eur. J. Oper. Res. 190 (2008) 357–382.

[32] Y.J. Wang, Y.P. Yang, Particle swarm optimization with preference order ranking for multi-objective optimization, Inf. Sci. 179 (2009) 1944–1959.

[33] A. Elhossini, S. Areibi, R. Dony, Strength pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization, Evol. Comput. 18 (2010) 127–156.

[34] H. Yazdani, H. Kwasnicka, D. Ortiz-Arroyo, Multiobjective particle swarm optimization using fuzzy logic, in: Computational Collective Intelligence. Technologies and Applications, Springer, 2011, pp. 224–233.

[35] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (2004) 256–279.

[36] S. Janson, D. Merkle, M. Middendorf, Molecular docking with multi-objective particle swarm optimization, Appl. Soft Comput. 8 (2008) 666–675.

[37] W.F. Leong, G.G. Yen, PSO-based multiobjective optimization with dynamic population size and adaptive local archives, IEEE Trans. Syst. Man Cybern. 38 (2008) 1270–1293.

[38] Y. Zhang, D.W. Gong, C.L. Qi, Vector evolved multiobjective particle swarm optimization algorithm, in: 2011 International Conference in Electrics, Communication and Automatic Control Proceedings, Springer, 2012, pp. 295–301.

[39] J.J. Liang, B.Y. Qu, P.N. Suganthan, B. Niu, IEEE, dynamic multi-swarm particle swarm optimization for multi-objective optimization problems, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.

[40] J.J. Liang, B.Y. Qu, Large-scale portfolio optimization using multiobjective dynamic mutli-swarm particle swarm optimizer, in: 2013 IEEE Symposium on Swarm Intelligence, 2013, pp. 1–6.

[41] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, H. Han, An endocrine cooperative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks, Inf. Sci. 300 (2015) 100–113.

[42] A. Britto, S. Mostaghim, A. Pozo, Archive based multi-swarm algorithm for many-objective problems, in: 2014 Brazilian Conference on Intelligent Systems, 2014, pp. 79–84.

[43] O.R. Castro, A. Pozo, A hybrid competent multi-swarm approach for many-objective problems, in: 2014 Brazilian Conference on Intelligent Systems, 2014, pp. 426–431.

[44] R. Cyriac, A.K. Rastogi, An overview of the applications of particle swarm in water resources optimization, in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Springer, 2013, pp. 41–52.

[45] D. Nagesh Kumar, M. Janga Reddy, Multipurpose reservoir operation using particle swarm optimization, J. Water Resour. Plan. Manag. 133 (2007) 192–201.

[46] A.M. Baltar, D.G. Fontane, Use of multiobjective particle swarm optimization in water resources management, J. Water Resour. Plan. Manag. 134 (2008) 257–265.

[47] W.C. Wang, L. Qiu, Optimal reservoir operation using PSO with adaptive random inertia weight, in: 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI), IEEE Press, 2010, pp. 377–381.

[48] M.H. Afshar, Large scale reservoir operation by constrained particle swarm optimization algorithms, J. Hydro-environ. Res. 6 (2012) 75–87.

[49] L. Ostadrahimi, M.A. Mariño, A. Afshar, Multi-reservoir operation rules: multi-swarm PSO-based optimization approach, Water Resour. Manag. 26 (2012) 407–427.

[50] X.N. Guo, T.S. Hu, C.L. Wu, T. Zhang, Y.B. Lv, Multi-objective optimization of the proposed multi-reservoir operating policy using improved NSPSO, Water Resour. Manag. 27 (2013) 1–17.

[51] Y. Qi, F. Liu, M. Liu, M. Gong, L. Jiao, Multi-objective immune algorithm with Baldwinian learning, Appl. Soft Comput. 12 (2012) 2654–2674.

[52] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, Neural Comput. 9 (1997) 1493–1516.

[53] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.

[54] Q.F. Zhang, A.M. Zhou, Y.C. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (2008) 41–63.

[55] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3 (1999) 257–271.

[56] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Shaker Ithaca, 1999.