

文章编号: 1000-6788(2006)09-0129-06

## 基于 PSO 的水库泄洪风险计算

席秋义<sup>1</sup>, 谢小平<sup>1,2</sup>, 黄强<sup>1</sup>, 王义民<sup>1</sup>, 张雯怡<sup>1</sup>

(1. 西安理工大学水利水电学院, 西安 710048; 2. 黄河上游水电开发有限责任公司, 西宁 810003)

**摘要:** 基于结构可靠度指标的物理含义, 建立了水库泄洪风险计算优化模型, 并引进了粒子群全局优化算法对该模型进行求解. 通过对该模型在具体水库泄洪风险计算的应用举例, 表明了该水库泄洪风险计算理论不仅能有效地避免求偏导运算, 且能获得高精度的运算结果, 求解速度也有一定的优势.

**关键词:** 泄洪风险; 可靠度指标; 粒子群算法; 约束优化

**中图分类号:** TV698.2

**文献标志码:** A

## The Calculation of Flood Discharging Risk Based on Particle Swarm Optimization

XI Qiu-yi<sup>1</sup>, XIE Xiao-ping<sup>1,2</sup>, HUANG Qiang<sup>1</sup>, WANG Yi-min<sup>1</sup>, ZHANG Wen-yi<sup>1</sup>

(1. Institute of Water Resources and Hydroelectric Engineering, Xi'an University of Technology, Xi'an 710048, China; 2. Hydroelectric Exploitation Ltd of the Upper Yellow River, Xi'ning 810003, China)

**Abstract:** The risk analysis of reservoir flood discharging as an important research subject has been focused on widely by hydraulic engineering people during past days. Based on the physical meanings of structural reliability index, the optimization model of flood discharging risk is established in this paper, and the global PSO is introduced to solve the model. Combining with the listed computation example on flood discharging risk of reservoir, it is showed that Partial difference operation can not only be avoided effectively by means of the theory, but also the better solution and a quick operation speed can also be gained in the way.

**Key words:** flood discharging risk; reliability index; Particle Swarm Optimization; optimization with constraints

### 1 泄洪风险

水库泄洪风险分析关系着泄洪建筑物规模和布置方案的确定, 关系着大坝的安全和投资, 是水利工程人员历来重视的问题之一. 传统的防洪设计方法是根据不同设计标准的洪水过程线和水库的调度规则, 通过水库调洪演算, 推求各个时刻的库水位, 以最终确定水库的各防洪特征水位和库容以及相应的泄洪建筑物形式、尺寸、高程、布置等. 但由于水库调洪演算本身所依据的洪水过程线、水位库容关系和出库泄流能力等都存在某些难以预料和控制的不确定性因素, 导致了库水位和下泄流量过程的随机性, 从而影响了水库自身的防洪安全和下游的防洪安全.

本文提到的泄洪风险, 范围仅限于水库本身的防洪任务, 并不考虑对下游的防洪任务, 故水库泄洪风险定义为: 在设计泄洪建筑物规模下, 大坝遭遇设计洪水和校核洪水时, 水库不能安全渡汛, 发生洪水漫坝事件的概率<sup>[1]</sup>, 用极限状态方程表示, 则为

$$Z_d - Z = 0, \quad (1)$$

其中,  $Z_d$  表示坝顶高程(相当于抗力);  $Z$  表示调洪最高水位(相当于荷载). 若  $Z$  大于  $Z_d$ , 则表示洪水漫顶,  $Z$  小于  $Z_d$ , 漫顶事件不发生.

泄洪风险则可通过下式来计算:

收稿日期: 2005-07-11

资助项目: 国家自然科学基金(50479024); 陕西省教育厅专项科研计划(04JK233); 青海省科技厅项目(2004-G-158); 陕西省教育厅省级重点实验室项目(02JSS37)

作者简介: 席秋义(1978-), 男, 西安理工大学博士生.

$$P_f = P(Z > Z_d) = P(Z > 0) = \int_D f(z) dz, \quad (2)$$

其中,  $Z = Z - Z_d$  为超过坝顶的洪水位;  $f(z)$  为超过坝顶洪水位的概率密度函数;  $D$  表示  $z$  的失事域。

坝顶高程  $Z_d$  由于量测、施工误差, 呈随机分布, 一般变化不大, 通常认为服从均值为设计坝顶高程, 均方差较小的正态分布<sup>[2]</sup>; 设计洪水过程的不确定性, 水位库容关系的不确定性以及泄流水力条件的不确定性等, 都对调洪最高水位  $Z$  产生影响, 使其在某一值附近随机变动, 同样看作是服从某种分布的随机变量。

通过以上分析可以看出, 对泄洪风险的分析, 最后归结为如何求解(1)式或(2)式。在各变量分布已知的条件下, 显然, 蒙特卡洛模拟法(MC)和JC法(拉克维茨-菲斯莱法)都能用来求解泄洪风险<sup>[3]</sup>。JC法是通过可靠度指标的求解, 进而利用可靠度与风险的关系计算出泄洪风险, 它与MC法相比, 显著的优点是求解效率高, 缺点是当极限状态方程是高次非线性时, 将其线性化时因略去二阶和更高阶项, 会带来线性化误差<sup>[4]</sup>。

本文对(1)式的求解, 采取的基本思想类似于JC法, 同样通过计算可靠度指标来计算风险, 不同之处是本文基于可靠度指标的物理意义, 建立带有约束条件的可靠度指标计算优化模型, 通过对该模型的求解来实现对风险的计算。模型求解方法采用了“零次”粒子群优化(PSO)算法, 很快能得到泄洪风险的满意解和结构概念上的设计点。

## 2 风险计算优化模型

本节从结构可靠度指标的计算, 以及可靠度与失效风险之间的关系方面来说明水库泄洪风险计算优化模型的建立。

首先, 设随机向量  $X$  由  $n$  个相互独立且与结构安全有关的随机变量  $X_i (i = 1, 2, \dots, n)$  组成, 失事域定义为  $D$ ,  $f_X(x)$  是它的联合概率密度函数, 结构失事状态用极限状态方程  $G$  表示, 则存在以下性质:

$$\begin{aligned} G: \mathbf{R}^n & \rightarrow \mathbf{R} \\ X & \rightarrow G(X) \end{aligned} \quad (3)$$

其中, 当

- 1)  $G(X) < 0$ , 系统处于失事状态;
- 2)  $G(X) = 0$ , 系统处于极限状态(此状态也认为处于失事状态);
- 3)  $G(X) > 0$ , 系统处于安全状态。

如果知道  $X$  的联合概率密度函数  $f_X(x)$ , 计算结构的失事概率  $P_f$ , 只需对其在失事域  $D$  求积分, 即计算

$$P_f = \int_D f_X(x) dx, \quad (4)$$

但准确的  $f_X(x)$  通常难以推求, 失事域边界 ( $D$ ) 是非线性和随机向量  $X$  的分量互相相关等原因, 以至于上式很难求解。

风险  $P_f$  与可靠度  $P_R$  存在以下关系:

$$P_f = 1 - P_R, \quad (5)$$

利用上式的关系, 可以通过求解结构的可靠度进而得到结构失事风险。可靠度指标 用于度量结构的可靠度, 其物理意义是指在标准正态坐标系中, 原点到极限状态曲面的最短距离。设 为标准高斯空间(向量各分量之间互不相关),  $U$  是此空间的任意向量, 是由极限状态函数  $G$  定义的物理空间 的  $n$  维曲面, 并且  $T(\cdot)$  是该曲面在标准高斯空间上的映射面。按照其物理意义, 则有下式:

$$= \min(d(O, P)) = d(O, P^*), \quad (6)$$

其中,  $O$  是空间 的中心或坐标原点;  $P$  是 曲面上的任意一点;  $P^*$  是标准正态坐标系中, 原点到极限状态面的最短距离对应的极限状态面最大失事概率的点, 即设计点, 如图1所示。

若向量  $X$  的各分量服从正态分布且互不相关,  $m_i$  和  $\sigma_i$  分别是向量  $X$  各分量  $X_i$  的均值和标准差, 则

通过  $T_i(X) = U_i (i = 1, 2, \dots, n)$  和  $U_i = (X_i - m_i) / \sigma_i$  即可实现如  $T(X) = U$  的空间转换关系.

按(6)式建立可靠度指标优化模型,即

$$\begin{cases} \text{O. b.} & \min_u f = \sum_{i=1}^n u_i^2 \\ \text{s. t.} & G(T^{-1}(u)) = 0 \end{cases} \quad (7)$$

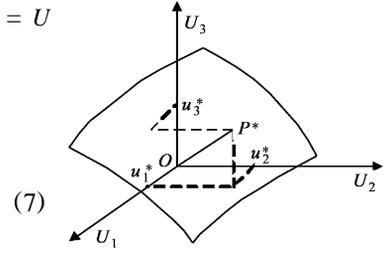


图 1 极限状态曲面

求解上述约束优化问题等价于求解非约束优化问题:

$$\min_u f = \sum_{i=1}^n u_i^2 + (G(T^{-1}(u))), \quad (8)$$

其中,  $f$  是惩罚函数;  $r$  是惩罚系数 ( $r > 0$ ). 上式采用的是外部惩罚, 解的搜索是从解空间的外部进行.

若  $u^*$  是优化模型(7)式或(8)式的解, 按下式即可得到结构可靠度指标:

$$\beta = \frac{1}{\sqrt{u_1^2 + u_2^2 + u_3^2}} \quad (9)$$

得到结构可靠度指标后, 结构失事风险可通过下式计算:

$$P_f = \Phi(-\beta) \quad (10)$$

惩罚系数  $r$  决定空间解的搜索是否朝着收敛的方向进行, 太大容易导致病态问题<sup>[5]</sup>. 本文采用迭代法来确定惩罚系数  $r$ , 可以有效解决这个问题. 迭代公式:

$$r_{i+1} = 2 \cdot r_i \quad (11)$$

$r_i$  由具体问题决定. 如果惩罚函数  $f(u)$  足够小时 (例如小于  $10^{-4}$ ), 可考虑停止迭代, 此时  $r_i$  即是所选定的惩罚系数.

同理, 水库泄洪风险计算优化模型可按上述过程建立, 本文采用如下形式:

$$\min_u f = u_d^2 + u^2 + |u_d - u| \quad (12)$$

式中,  $u_d$ 、 $u$  分别是标准正态空间的  $Z_d$  和  $Z$  换算变量.

粒子群算法是一种有效解决连续变量的全局优化算法, 其优点在于简单、易于实现并且很快能得到准精确解, 本文选其作为上述优化模型的求解算法.

### 3 粒子群优化算法

#### 3.1 算法概述

粒子群算法 (Particle Swarm Optimization, PSO) 起源于对鸟群捕食社会行为的模拟, 同蚁群算法 (Ant Colony Optimization, ACO) 一样, 均属于基于群智能的算法, 最初由 Ebert 和 Kenndy 博士提出.

PSO 处理优化问题的思路是, 每个优化问题的解看作是搜索空间中的一只鸟, 我们称之为“粒子”, 所有的粒子都有一个由被优化的函数决定的适应值 (fitness value) 和一个决定他们飞行方向和距离的速度, 它们各自按照自己的飞行经历和同伴的飞行经历调节自己的飞行, 对解空间进行搜索, 最后得到优化问题的精确解或满意解<sup>[6,7]</sup>.

PSO 同遗传算法 (GA) 类似, 是一种基于迭代的优化算法, 但它并没有 GA 的交叉 (crossover) 以及变异 (mutation) 操作, 而是按粒子在解空间追随最优的粒子进行搜索. 同 GA 相比, PSO 的优势在于简单、容易实现并且没有许多参数需要调整.

#### 3.2 PSO 基本原理

如上所述, PSO 算法中的每个粒子可以看作是  $D$  维解空间中的一个点, 如果粒子的群体大小为  $N$ , 第  $i (i = 1, 2, \dots, N)$  个粒子的位置表示为  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , 它所经历过的最好的位置 (适应值最好) 记为  $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$ , 群体中最好粒子用符号  $g$  表示, 粒子的位置变化率 (速度) 表示为  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ , 则粒子  $i$  根据 (13) 和 (14) 式来更新自己的速度和位置:

$$v_{id} = w \times v_{id} + c_1 \times \text{rand}() \times (P_{id} - x_{id}) + c_2 \times \text{Rand}() \times (P_{gd} - x_{id}), \quad (13)$$

$$x_{id} = x_{id} + v_{id}, \quad (14)$$

其中,  $c_1$  和  $c_2$  是正常数,称为学习因子;  $\text{rand}()$  和  $\text{Rand}()$  分别是  $[0,1]$  区间上的随机数;  $w$  为惯性权重.

### 3.3 参数选择

#### 1) 惯性权 $w$ 和允许最大速度 $V_{\max}$

在 PSO 中,全局与局部搜索能力主要由惯性权重  $w$  控制,试验发现,随时间递减的惯性权重比固定的惯性权重值有比较好的性能.开始时较大的惯性权重有助于很快找到比较好的解,而在找到较好的解之后,较小的惯性权重又促进了局部搜索能力,从而全局和局部解的搜索都得到了兼顾.

全局搜索能力除受惯性权  $w$  控制外,还受允许最大速度  $V_{\max}$  的控制.由(13)式计算出的速度,即受允许最大速度的约束.允许最大速度若太小,则限制了最大全局搜索能力,这时不论惯性权的大小,PSO 总是支持局部搜索能力.允许最大速度若设定太大,则 PSO 通过选择合适的惯性权具有大范围的搜索能力.由于允许最大速度不直接影响 PSO 的全局搜索能力,而惯性权直接影响,并且允许最大速度的确定依赖于实际问题,一般很难确定,通常的作法是仅通过惯性权对 PSO 的全局搜索能力加以控制.但这样做又意味着系统总是渴望搜索新的区域,最终导致系统缺乏局部搜索能力,甚至找不到解.设定惯性权随循环次数的变化逐步递减,可以有效地解决这个问题.

综合上述分析,本文设定  $V_{\max} = X_{\max}$ ,即不设允许最大速度限制.采用的惯性权重在开始的循环次数内从 0.95 递减到 0.4,而在最后的一些循环内保持 0.4 不变<sup>[8]</sup>.

#### 2) 参数 $c_1$ 和 $c_2$

本文设定参数  $c_1 = 2, c_2 = 2$ ,表示粒子以当前位置到目标的距离长度进行“超飞”.

#### 3) 群体大小 $N$

群体大小关系着计算所耗时间的大小,文中取  $N = 20$ .

### 3.4 PSO 算法流程

PSO 算法流程如下:

**步骤 1** 初始化粒子群:给定群体规模  $N$ ,随机产生每个粒子的位置  $X_i$ ,速度  $V_i$ .

**步骤 2** 用目标函数  $f(x)$  计算每个粒子的当前适应度值:

$$\text{Fitness}(i) = f(X_i), \quad i = 1, 2, \dots, N.$$

**步骤 3** 求每个粒子的个体极值  $\text{Pbest}[i], i = 1, 2, \dots, N$ :分别比较每个粒子的当前适应度值和  $\text{Pbest}[i]$ ,若当前适应度值较好,则指定  $\text{Pbest}[i] = \text{Fitness}[i]$ ,个体极值位置  $P[i] = X[i]$ ,否则,保持  $\text{Pbest}[i]$  和  $P[i]$  不变.  $\text{Pbest}[i]$  存储第  $i$  粒子经历过的最好适应度值;  $P[i]$  存储第  $i$  粒子经历过的最好适应度值所对应的位置.

**步骤 4** 求全局极值  $\text{Gbest}$ :变量  $\text{Gbest}$  存储群体所有粒子当前为止最好的适应度值,  $P[g]$  存储全局最好粒子的位置.比较群体所有个体极值  $\text{Pbest}[i], i = 1, 2, \dots, n$  和  $\text{Gbest}$ ,若个体极值较好,则指定

$$\text{Gbest} = \text{Best}(\text{Pbest}[i], i = 1, 2, \dots, N), P[g] = \text{Best}(X[i], i = 1, 2, \dots, N)$$

否则,保持  $\text{Gbest}$  和  $P[g]$  不变.

**步骤 5** 计算更新速度  $V_i$ :根据(13)式计算各粒子的速度更新分量,若  $v_{id} > V_{\max}$ ,则指定  $v_{id} = V_{\max}$ ;若  $v_{id} < -V_{\max}$ ,则指定  $v_{id} = -V_{\max}, d = 1, 2, \dots, D$ .

**步骤 6** 计算更新位置  $X_i$ :根据(14)式计算各粒子的更新位置,若  $x_{id} > X_{up}[d]$ ,则指定  $x_{id} = X_{up}[d]$ ;若  $x_{id} < X_{low}[d]$ ,则指定  $x_{id} = X_{low}[d]$ .其中,  $X_{up}[d]$  和  $X_{low}[d]$  分别是位置区间第  $d$  维上界和下界,  $d = 1, 2, \dots, D$ .

**步骤 7** 重复执行步骤 2~6,直至满足终止条件(到达一定的循环数或两次解相差不大).

## 4 水库泄洪风险计算优化模型应用算例

某水库大坝是土石坝,漫顶一定失事.坝顶高程 123m,相对变化不大,可认为其服从均值为 123m,方差为  $0.05^2$  的正态分布.因该水库运行时间不长,尚未遭遇过大的洪水,年调洪最高水位实测资料系列太短,不具备代表性,故利用随机水文学模拟的方法(典型解集模型),生成大容量的校核入库洪水过程,再通

过调洪演算得到该水库调洪最高水位系列,以此作为年调洪最高水位的样本,求得该样本系列的统计均值为 122.2m,统计方差为 0.2<sup>2</sup>,分析调洪最高水位服从下面两种分布时的泄洪风险:1) 正态分布;2) 对数正态分布.

下面是基于 PSO 算法的泄洪风险求解过程:

对于正态分布,解的搜索区间限定如下:

$$\begin{bmatrix} Z_d^{low} & Z_d^{up} \\ Z^{low} & Z^{up} \end{bmatrix} = \begin{bmatrix} 122 & 124 \\ 121.2 & 123.2 \end{bmatrix} \begin{bmatrix} -20 & 20 \\ -5 & 5 \end{bmatrix}. \tag{15}$$

箭头后矩阵是根据(16)和(17)式,进行解空间转换求得的.

$$U_d = (Z_d - 123)/0.05, \tag{16}$$

$$U = (Z - 122.2)/0.2. \tag{17}$$

接着在解搜索区间初始化粒子群,并依照 PSO 算法流程求解(12)式,最后通过(10)式计算泄洪风险.计算结果如表 1 所示.表 1 同时列出了 JC 法和 MC 法(模拟运行了 10000000 次)的泄洪风险计算结果.图 2 是适应度的变化过程.

表 1 正态分布设计点与风险率计算结果

	标准高斯空间设计点	原空间设计点	可靠度	风险率
PSO	- 0.9548, 3.7613	122.952, 122.952	3.8806	0.5211 ×10 <sup>-4</sup>
JC 法	-	122.953, 122.953	3.8806	0.5211 ×10 <sup>-4</sup>
MC 法	-	-	-	0.5240 ×10 <sup>-4</sup>

注:1) - 表示没有数据;2) PSO 中 取 200

由于泄洪风险极限状态方程是线性的,JC 法的结果可看作是精确解.从表 1 结果可以看出,采用 PSO 算法求解泄洪风险优化模型的结果与采用 JC 法的结果基本相同,而 MC 法在模拟运行了 10000000 次后,仍与精确解有一定差距.

当调洪最高水位服从对数正态分布时,由于其不能直接用 PSO 算法求解,故需进行空间转换.设  $Y = \ln Z$ ,则  $Y$  服从正态分布,极限状态方程转换后为

$$Z_d - e^Y = 0. \tag{18}$$

此后的求解过程同前正态分布, $Y$  的均值和方差根据文献[4]提供的公式计算,分别为 4.8056 和 0.001637<sup>2</sup>,泄洪风险计算结果见表 2.同理,表 2 给出了 JC 法和 MC 法的计算结果,图 3 是适应度的变化过程.

表 2 对数正态分布设计点与风险率计算结果

	标准高斯空间设计点	原空间设计点	可靠度	风险率
PSO	- 0.9483, 3.7867	122.953, 122.953	3.9036	0.4740 ×10 <sup>-4</sup>
JC 法	-	122.953, 122.953	3.8814	0.5193 ×10 <sup>-4</sup>
MC 法	-	-	-	0.4880 ×10 <sup>-4</sup>

注:1) - 表示没有数据;2) PSO 中 取 200

表 2 中 JC 法计算时采用了当量正态变换,从中可以看出,三种算法的计算结果都在同一数量级,相差不大.表 1 和表 2 的风险值也基本接近,这与文献[4]提到的当  $P_f < 10^{-5}$  ( $> 4.26$ ),分布类型对  $P_f$  的计算结果就相当敏感相一致.

图 2 在大约 300 代后即找到了满意解;图 3 在大约 240 代后即找到了满意解,计算耗时远小于 MC 法,这也说明 PSO 算法的求解效率比较高,相比 MC 法具有一定优势.

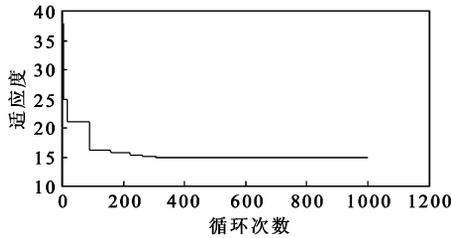


图2 适应度变化(正态分布)

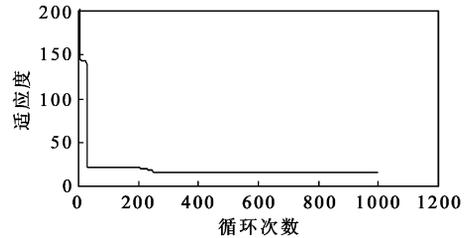


图3 适应度变化(对数正态分布)

## 5 结语

本文提出的理论仅着重于对变量分布和统计特征已知的极限状态方程的求解.不同于JC法,该方法不需进行求偏导运算,在经过一定次数的搜索循环后,即可得到满足要求精度的解;求解时且充分利用PSO全局优化算法搜索能力强和求解速度快的优点,提高了泄洪风险计算精度,并且求解效率明显要高于MC法.

本文在算例中假设设计或校核情况下调洪最高水位服从正态分布或对数正态分布.正态分布是最常应用的一种分布形式,从经验上可认为在设计 and 校核标准下,调洪最高水位服从于该分布;对数正态分布是结构可靠度计算抗力和荷载经常采用的一种分布,它的密度函数具有正偏斜度,向左偏倚.当调节水库可能遭遇的所有洪水时,调洪最高水位可能符合此类分布.当然,调洪最高水位服从的确切分布应该是通过其样本序列的分布检验得到,本文仅通过算例给定分布的形式来说明其泄洪风险的另一种求解方法,对其分布如何确定不作详细探讨.

本文的研究同样为大坝防洪安全风险分析提供了一种新方法,大坝防洪安全风险考虑的洪水由于包括该坝可能遭遇的所有量级的洪水,调洪最高水位服从的分布很可能是偏态分布,例如极值I型分布等.其它概率分布形式下如何与优化模型结合起来计算泄洪风险,需要进一步研究.

## 参考文献:

- [1] 姜树海. 随机微分方程在泄洪风险分析中的应用[J]. 水利学报, 1994, (3): 1 - 9.  
Jiang Shuhai. Application of stochastic differential equations in risk analysis for flood relief[J]. Journal of Hydraulic Engineering, 1994, (3): 1 - 9.
- [2] 梅亚东, 谈广鸣. 大坝防洪安全的风险分析[J]. 武汉大学学报(工学版), 2002, 35(6): 11 - 15.  
Mei Yadong, Tan Guangming. Risk analysis for flood prevention and safety of dam [J]. Engineering Journal of Wuhan University, 2002, 35(6): 11 - 15.
- [3] 杨百银, 王锐琛, 安占刚. 单一水库泄洪风险分析模式和计算方法[J]. 水文, 1999, (4): 5 - 12.  
Yang Baiyin, Wang Ruichen, An Zhangang. Risk analysis of flood relief of single reservoir [J]. Hydrology, 1994, (4): 5 - 12.
- [4] 赵国藩, 曹居易, 张宽权. 工程结构可靠度[M]. 北京: 水利电力出版社, 1984, 12.  
Zhao Guofan, Cao Juyi, Zhang Kuanquan. Structural Reliability Analysis In Engineering [M]. Beijing: Water Resources and Electric Power Press, 1984, 12.
- [5] Charles Elegbebe. Structural reliability assessment based on particles swarm optimization [J]. Structural Safety, 2005, (27): 171 - 186.
- [6] Kenny j, Eberhart R. Particle swarm optimization [A]. Proc IEEE International Conf on Neural Networks (Perth, Australia) [C]. IEEE Service Center, Piscataway, NJ, 1995, 1942 - 1948.
- [7] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory [A]. Proc Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan) [C]. IEEE Service Center, Piscataway, NJ, 1995, 39 - 43.
- [8] Shi Y, Eberhart R C. Parameter selection in particle swarm optimization [A]. In Evolutionary Programming VII: Proc EP98 [C]. New York: Springer-Verlag, 1998, 591 - 600.