

# 基于 C&S-PSO 的软件成本估算类比法特征权重优化<sup>①</sup>

杨 抒<sup>1</sup>, 王 业<sup>2</sup>, 乌尔柯西<sup>1</sup>, 游香霏<sup>1</sup>

<sup>1</sup>(新疆农业大学 计算机与信息工程学院, 乌鲁木齐 830052)

<sup>2</sup>(新疆农业大学 现代教育技术中心, 乌鲁木齐 830052)

**摘要:** 运用软件成本估算技术可以在软件项目实施过程中有效控制进度、降低风险从而保证所开发软件的质量。本文对成本估算技术中加权类比估算模型的特征属性的权重, 使用优化的粒子群算法(C&S-PSO)进行优化后, 利用 MMRE 和 Pred(0.25)两个标准与非加权类比模型、支持回归和模糊神经网络等模型进行估算的精度比较。另外, 采用非参数自助法对优化的 PSO 加权类比估算模型稳定性进行评估。研究结果表明, 从估算精度来看, 采用本文的估算模型比上述估算模型估算精度高, 同时该模型具有较好的稳定性。

**关键词:** 软件成本估算; 粒子群算法; 类比方法; 自助法; 非线性惯性属性权重

## Weight of Software Cost Estimation Analogy Model Using C&S-PSO

YANG Shu<sup>1</sup>, WANG Ye<sup>2</sup>, WU Erkexi<sup>1</sup>, YOU Xiang-Ru<sup>1</sup>

<sup>1</sup>(Computer and Information Engineering College, Xinjiang Agricultural University, Urumqi 830052, China)

<sup>2</sup>(Modern Education Technology Center, Xinjiang Agricultural University, Urumqi 830052, China)

**Abstract:** Software cost estimation technology has become an important measure of software project management aspects of the control software schedule, reducing software risks and ensure software quality. In this paper, we use the model of C&S-PSO and weighted analogy estimates characteristic attributes to the weights. According to both MMRE and Pred (0.25) standards, we compare un-weighted analogy model, SRV, FNN and CART to estimate accuracy. Additionally, we use the non-parametric bootstrap method for raw data sampling, analysis PSO weighted analogy estimation model of stability. The results show that from the view of estimation accuracy the model of PSO has higher precision in weighted analogy estimates, and the model has also better stability.

**Key words:** software cost estimation; PSO (Particle Swarm Optimization); analogy method; bootstrapping nonlinear inertia weight

从 20 世纪 60 年代开始, 软件工程领域出现了大量进度延期、预算超支和项目质量缺陷等典型特征, 这些我们称之为软件危机的特征目前仍然频繁发生。著名咨询公司 The Standish Group 于 2011 年公布的数据显示, 只有 32% 的软件项目是“成功的(succeeded)”; 24% 是“失败的(failed)”; 其余 44% 是“被质疑的(challenged)”<sup>[1]</sup>。Bannerman 认为对软件需求不稳定以及对成本估算工作不足是软件项目失控的两大主因<sup>[2]</sup>。因此, 软件的质量和成本可以通过准确的估算软件成本进行保证。作为降低软件风险和控制软件实施进度

的主要手段之一, 如何提高软件估算精确度已成为学术界积极研究的内容。Boehm 认为通过准确的估算软件成本可以帮助我们开发出更多更好的软件<sup>[3]</sup>。因此, 研究有效软件成本估算模型对项目进行预测, 进而提高软件成本估算的精确度是有必要的。

## 1 类比估算的相关研究

专家判断、算法模型、回归分析以及类比估算是最常用的 4 种软件成本估算的方法<sup>[4]</sup>。其中, 类比(analogy)估算分析是 CBR(case-based reasoning)基于实

① 基金项目:新疆农业大学前期资助项目(XJAU201116);新疆维吾尔自治区自然科学基金(2014211B023)

收稿时间:2014-11-03;收到修改稿时间:2014-12-10

例推理的一种形式<sup>[5]</sup>,它是通过对一个或多个已完成的项目(历史项目)与新项目(新的类似项目)对比以后,来预测拟估算项目的成本与进度<sup>[6]</sup>.该方法具有直观、可理解性好、样本学习能力强等符合人们习惯思维的特点,因此,类比估算在软件成本估算技术中有着比较广泛的应用<sup>[7]</sup>.在类比估算方法中,欧式距离可以衡量已有项目与拟估算项目之间的相似程度.目前,欧式距离的度量分别采用非加权的欧式距离(unweighted Euclidean distance)<sup>[5]</sup>和加权的欧式距离(weighted Euclidean distance)两种方式<sup>[8]</sup>.尽管对于相似函数的定义在形式上有所不同,但它们本质是一致的<sup>[9]</sup>.现有研究表明,如果在加权类比模型中选择合适的权重系数,软件成本估算准确度较高.通常,类比估算模型中权重系数的选择有两种,一种是由专家人为确定权重;另一种方法是采用启发式算法计算出最优估算结果的权重.一般来说,当选取的特征不够全面时,可以选择专家来确定权重这种常用的方式<sup>[10]</sup>.相关学者对类比估算的贡献有很多,比如, Huang 在改进类比法时采用了灰色关联度方法和模糊神经网络,他通过这两种方法评判不同项目间的相似度<sup>[11]</sup>.

类比方法在软件成本估算应用的过程中要注意选择的已完成项目与拟估算项目应该类似.一般做法是将案例集中已经完成的若干个类似的历史软件项目形成案例集和.有了历史项目的案例集合后,再选择进行相似度计算的属性权重对新项目估算是类别估算的关键.由于类比方法使用专家判断权重或者随意选择算法估算属性权重都不能保证模型的精确度.因此,相关学者使用遗传算法优化类比方法中特征属性权重<sup>[12]</sup>.基于遗传算法的估算模型有一定的精确度,但是由于遗传算法是具有一定复杂性计算方法,所以这种模型在实现的效率方面表现不佳.吴登生等学者尝试用 PSO 算法优化类比模型中权重的选择,但是该方法在惯性属性权重的选择采用了线性递减的方法,这种方法在局部求解时容易陷入局部结果早熟的情况<sup>[13]</sup>.

本研究在对类比估算模型的权重进行优化过程中采用一种惯性权重属性非线性递减的粒子群算法(C&S-PSO, C&S particle swarm optimization)对各特征属性进行权重拟合后,对 Desharnais 数据集中项目进行估算,以该模型的计算结果为基础,与其它常用估算模型比较估算结果的精确度;最后采用 Bootstrap 法验证该模型的表现是否稳定.

## 2 软件成本估算的类比法

### 2.1 计算项目之间的相似度

欧式距离可以用于拟估算项目与已有项目的相似度,如公式(1)所示.

$$D(C_x, C_y) = \sqrt{\sum_{i=1}^p Dis(C_{xi}, C_{yi})} \quad (1)$$

$C_{xi}$ 和 $C_{yi}$ 代表两个参与比较的对象. $C_{xi}$ 与 $C_{yi}$ 之间计算非加权欧式距离具体方法如公式(2)所示.

$$Dis(C_{xi}, C_{yi}) = \begin{cases} (C_{xi} - C_{yi})^2 & C_{xi} \text{ 为数值} \\ 1 & C_{xi} \text{ 为名义值且 } C_{xi} \neq C_{yi} \\ 0 & C_{xi} \text{ 为名义值且 } C_{xi} = C_{yi} \end{cases} \quad (2)$$

$C_{xi}$ 与 $C_{yi}$ 之间采用加权的方式计算欧式距离,如公式(3)所示

$$Dis(C_{xi}, C_{yi}) = \begin{cases} W_i (C_{xi} - C_{yi})^2 & C_{xi} \text{ 为数值} \\ W_i & C_{xi} \text{ 为名义值且 } C_{xi} \neq C_{yi} \\ 0 & C_{xi} \text{ 为名义值且 } C_{xi} = C_{yi} \end{cases} \quad (3)$$

从公式(3)中可以看出,如果拟估算项目和历史项目在比较时, $W_i$ 作为项目属性特征权重带入计算,这些权重用于度量类比估算模型中属性的不同的重要程度,那么度量后结果会比非加权的欧式距离更加准确.

### 2.2 项目拟和数目和计算方式

在项目估算时的第一步是需要确定几个相似度较高的历史项目作为新项目成本估算的依据,计算历史项目与新项目之间的相似度,通常使用平均值法或中位数法.对多个项目成本估算,计算公式为:

$$E_{new} = median(E_1, \dots, E_k) \quad (4)$$

其中, $E_{new}$ 是计算出的拟估算项目的成本; $k$ 是已有项目个数, $k$ 取值的范围是从1到 $n-1$ ; $(E_1, \dots, E_k)$ 是已有项目的实际成本的集合; $median$ 是 $k$ 个已有项目成本中位数; $mean$ 是 $k$ 个已有项目成本的均值.本研究在非加权类比和C&S-PSO优化成本类比法进行准确性对比时, $k$ 值取范围是1到3.

### 2.3 估算结果的评价标准

相关研究中,通常会采用两个评价标准衡量评估模型的准确度,一个是建立在相对误差值的基础上的平均相对误差(mean magnitude related error, MMRE),计算方法如公式(5)所示.

$$MRE_i = \frac{|E_i - \hat{E}_i|}{E_i} \quad (5)$$

公式(5)中的  $MRE_i$  为数据集中项目  $i$  的估算结果与实际结果之间的误差;  $E_i$  是实验数据集中项目  $i$  的实际成本,  $E_i^*$  是数据集中项目  $i$  的估算成本. 在此基础上, 模型在数据集中的整体误差用平均相对误差标准来衡量, 这里要注意的是, 不考虑每个项目的估算误差进行权重加权, 计算过程为公式(6)所示.

$$MMRE = \frac{1}{n} \sum_{i=1}^n MRE_i \quad (6)$$

公式(6)中  $n$  代表数据集中已完成项目的总数.

在相关研究中, 另一个评价标准是度量结果准确度误差的  $Pred(0.25)$ <sup>[14]</sup>. 具体计算公式如公式(7)所示

$$pred(0.25) = \frac{m}{n} \quad (7)$$

公式(7)中,  $m$  是数据集中相对误差小于 0.25 的项目个数.

### 3 使用优化的 PSO 的类比法优化权重

#### 3.1 优化的 PSO(C&S-PSO)算法介绍

由于, 类比法中的特征权重属性的权重对类比估算模型的精确度非常重要, 因此, 相关学者选择 PSO 方法优化类比法的属性权重<sup>[15]</sup>. 该算法在保持了种群全局搜索策略的基础上以速度和位置为参数作为搜索依据进行迭代搜索. 它的优点之一是可以避免类似遗传算法那样复杂的遗传步骤, 其被认为是一种更高效的并行算法<sup>[16]</sup>. 其具体算法表述如下, 粒子群中的  $i$  粒子在  $D$  维解空间位置为  $X_i$ ,  $X_i = (X_{i1}, X_{i2}, \dots, X_{iD})^T$ , 速度是  $V_i$ ,  $V_i = (V_{i1}, V_{i2}, \dots, V_{iD})^T$ .  $P_{ibest}$  是当前个体极值,  $G_{best}$  是当前全局极值. 每次迭代过程中, 粒子通过跟踪  $P_{ibest}$ 、 $G_{best}$  自己前一时刻的状态调整自己当前时刻的位置和速度, 其迭代过程如公式(9)

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot rand() \cdot [p_{best} - X_i(t)] + c_2 \cdot rand() \cdot [g_{best} - X_i(t)] \quad (8)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (9)$$

公式(9)中,  $V(t)$  和  $X(t)$  分别是当前一个粒子的速度和位置; 同理,  $V(t+1)$  和  $X(t+1)$  是下个时刻粒子的速度与位置;  $rand()$  是随机数(其范围是  $0 < rand < 1$ ).

为了在全局搜索和局部搜索的搜索能力得到有效地平衡, 同时在高维度空间中加快其收敛的速度, 我们采用了惯性权重非线性递减的方法加快算法收敛速度<sup>[17]</sup>. 具体定义为公式(10):

$$w(k) = \left[ \frac{(k_{max} - k)^N}{(k_{max})^N} \right] (w_{initial} - w_{final}) + w_{final} \quad (10)$$

本研究采用的优化的 PSO 方法选择了非线性递减的惯性属性权重, 首先可以对种群的全局搜索策略进行保留, 其次可以保持遗传步骤的复杂性, 最后在保留模型计算简洁的程度上比遗传算法的估算精度和效率有一定的提高<sup>[18]</sup>.

#### 3.2 优化的 PSO 算法模型

本研究采用 C&S-PSO 算法优化类比方法中的特征属性权重. 优化的基本过程是将加权类比法中的特征权重定义为一个粒子. 通过 PSO 算法不断迭代计算出最佳权重的拟合, 从而达到提高估算模型精确度的目标. 在这个过程中的适应度函数(fitness)的值使用 MMRE 值. 模型计算步骤如图 1 所示.

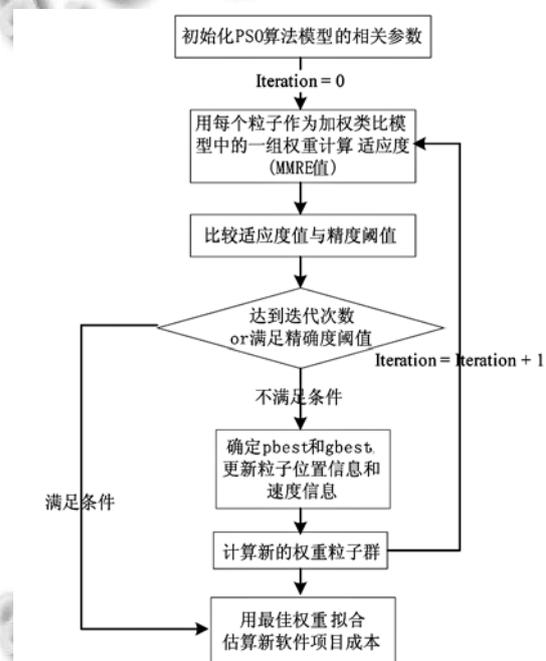


图 1 C&S-PSO 算法优化类比方法中的属性权重图

如图 1 所示, 首先, 设置 C&S-PSO 算法模型计算过程中的参数, 其中, 迭代次数设置为 0. 这个过程主要完成定义粒子群算法中计算作为特征属性的每个粒子的速度和位置; 然后, 计算适应度函数是否满足精度阈值, 在这个过程中需要计算适应度函数与精度阈值的差; 在迭代过程中如果已经满足阈值精度要求(计算适应度值小于设定的阈值)或已经达到设定的迭代次数, 那么执行流程图的最后一步, 也就是利用获得的新权重计算拟估算的软件项目成本, 以及另一项评价标准  $Pred(0.25)$ ; 如果在第 4 步不满足迭代次数和精度阈值的要求, 还要计算下一时刻的  $pBest$  和  $gBest$ ,

从而得到新的粒子群, 另外, 迭代次数加 1.

#### 4 判断估算模型的稳定性

1979 年 Efron 提出了 Bootstrap 方法<sup>[19]</sup>. 本研究使用非参数 bootstrap 方法验证本文提出的估算模型的稳定性. 这样做的好处是一方面可以避免增大样本含量重新搜集资料所耗费的时间、人、财等资源; 另一方面可以解决原样本含量分解以后各样本含量不够大等问题. Angelis 和 Stamelos 已经在软件成本估算过程中已经运用 Bootstrap 的方法对成本估算中的相关变量和计算结果的不确定性进行过分析.

非参数 Bootstrap 过程基本的思路是原始样本数据范围  $n$  不变的情况下, 在样本范围  $n$  内重复进行抽样. 已经被抽样的数据样本需要放回原始样本当中, 保持原始数据中每个样本数据被抽样的概率均为  $\frac{1}{n}$ , 这个过程重复得到的样本就是自助法中的 bootstrap 样本. 通过上述方法可以计算出估计值参数  $\theta$ , 通过  $B$  次重复可以计算出  $\theta$  参数的  $B$  个估计值. 通过此时参数  $\theta$  标准误的 bootstrap 估计值为<sup>[20]</sup>:

$$\hat{S} = \sqrt{\sum_{b=1}^B \frac{\hat{\theta}(b) - \theta}{B} / (B-1)} \quad (11)$$

我们可以通过公式(11)中的  $\hat{\theta} = \sum_{b=1}^B \hat{\theta}(b)$  获得参数  $\theta$  的部分性质<sup>[20]</sup>.

#### 5 实验结果分析

本研究使用 Desharnais 数据集检验构建模型的有效性. Desharnais 数据集分别包括 TeamExp、ManageExp、YearEnd、Transaction、Entities、PointsAdjust、Envergure、DevEnv、PointsNonAjust、Effort 等 12 个属性且都是数值型的值. 在 Desharnais 数据集中选择其中特征属性完整的 70 个样本数据进行实验. 在样本中最大的项目成本值为 13860, 最小的项目成本值为 651.

##### 5.1 该模型的估算结果分析

基于 C&S-PSO 算法的类比权重选择通过 MATLAB7.0 提供的 PSOOTB 工具箱中的 PSO 算法再修改权重的计算方式后编程实现类比估算的计算. 算法参数设置如表 1.

表 1 中参数的迭代次数的设置依据是在 C&S-PSO 方法中, 如果  $N$  在 0.8 和 1.2 之间 300 次迭代就可以得到比较精确的收敛结果.

表 1 优化的 PSO 算法参数设置表

模型计算参数	参数值
粒子的个数	20
迭代的次数	300
精确度阈值(MMRE)	0.4
学习因子 c1	2
学习因子 c2	2

从表 2 的实验数据可以看出, 在对 Desharnais 数据集实验的三个类比项目中, 使用 PSO 对特征属性加权后的估算结果都比传统类比法具有更低的 MMRE 值以及更高的 Pred(0.25)值, 说明这种方法精确度有了较大提高.

表 2 不同类比法估算模型对 Desharnais 数据集实验结果

估算模型	类比项目数(K)	MMRE 值	Pred(0.25)值
普通类比(非加权)	1	0.732	0.225
	2	0.501	0.268
	3	0.663	0.213
C&S-PSO 类比 (惯性权重非线性递减)	1	<b>0.425</b>	<b>0.513</b>
	2	0.522	0.222
	3	0.514	0.243
支持向量回归机(SVR)		0.471	0.383
模糊神经网络(FNN)		0.438	0.470
分类回归树(CART)		0.553	0.347

从表 1 可以看出, 当  $k=1$  时,  $MMRE=0.425$ . PSO 类比中方法比非加权类比中  $k=1$  时,  $MMRE=0.73$  增加了 25%; 就 Pred(0.25)而言, 当  $k=1$  时, Pred(0.25)为 0.513 比非加权类比中最优值高 0.245, 说明优化的 PSO 加权类比的 MMRE 值要低于非加权类比. 本研究在 Desharnais 数据集中, 使用了 3 种不同  $k$  值情况下的计算结果, 显示当  $k=1$  时, PSO 加权类比法得到最小的 MMRE 值为 0.48. 另外, 表 1 中结果显示, 除了当  $k=2$  时, Pred(0.25) PSO 加权类比 PSO 非加权类比低, 其它情况下的指标都要高. 我们可以得出, 在  $k=1$  时可以取得最优估算精度. 因此, 在选择 1 个类比项目时的两个指标(MMRE 和 Pred(0.25))估算结果误差最小而且精度最高. 因此可以得出结论, 优化的 PSO 对特征属性加权的方法精度比非类比算法高, 另外, 当 PSO 类比模型中  $k$  为 1 时, 得到的估算精度最高.

将 PSO 类比模型中最佳的估算精度结果与支持回归机、BP 神经网络、分类回归树三个模型进行比较发现 PSO 类比模型 MMRE 值比三者要小, Pred(0.25)值比其他三个模型的 Pred(0.25)值要高, 这说明该模型估算精确有一定的优势.

##### 5.2 模型的稳定性分析

本研究采用基于原始数据的模拟抽样方法 bootstrap 方法验证模型估计值的稳定性. 本研究中将抽样次数 B 设为 10 后对原始数据抽样, 表 2 是 10 次非参数 bootstrap 抽样分析结果.

表 3 10 次非参数 bootstrap 抽样比较分析结果

抽样次数	原始估计值( $\hat{U}$ )	$\hat{q}$	$\hat{S}_q$	$ \hat{q} - \hat{U} $	$ \hat{q} - \hat{U}  / \hat{S}_q$ (%)
1	0.425	0.471	0.214	0.046	21.45
2		0.392	0.182	0.033	18.17
3		0.413	0.110	0.012	10.95
4		0.436	0.105	0.011	10.49
5		0.412	0.114	0.013	11.40
6		0.461	0.190	0.036	18.97
7		0.487	0.249	0.062	24.90
8		0.402	0.152	0.023	15.17
9		0.469	0.210	0.044	20.98
10		0.382	0.207	0.043	20.74

表 2 中参数估计值( $\hat{U}$ )为 PSO 加权类比估算模型当项目数等于 1 时估计值的 MMRE,  $\hat{q}$  是非参数 bootstrap 法计算得到的样本估计值的 MMRE. 从计算结果我们发现二者相差不大. 估算值的偏倚量在当二者之差小于 bootstrap 抽样标准误的 25%(<25%)时可以忽略<sup>[16]</sup>. 因为表 2 给出的 10 次抽样参数估计值结果均小于 25%, 所以可以得出本研究提出的估算模型有较强的稳定性.

## 6 总结

实验表明, 通过优化的粒子群算法(C&S-PSO)对类比模型中的特征属性权重优化后的估算结果比传统的非加权方法精确, 也比其它常用方法在 MMRE 值和 Pred(0.25)值的方面表现要好, 说明 PSO 优化加权类比模型可以忽略一些属性的冗余信息. 另外通过非参数 Bootstrap 判断估算模型的稳定性后发现该模型稳定性较好. 除此之外, 当类比项目数为 1 时, 估算结果精度最高, 因此使用该模型时应该找到与估算项目最类似一个已知项目. 本研究使用估算模型来估算项目成本的方法可以为降低软件项目风险、保证软件质量提供一定借鉴.

### 参考文献

- 1 The Standish Group. <http://www.standishgroup.com>, 2011.
- 2 Bannerman PL. Risk and Risk Management in software projects: A reassessment. *Journal of Systems and Software*, 2008, 81(12): 2118–2133.
- 3 Boehm BW. Understanding and controlling software costs. *IEEE Trans. on Software Engineering*, 1988, 14(10): 1462–1477.
- 4 Magne J, Martin S. A review of software development cost estimation studies. *IEEE Trans. Software Engineering*, 2007, 33(1): 33–53.

- 5 范柳斌, 李路, 陈妮娜, 胡昱, 张子厚. 基于改进的粒子群和遗传算法的混合优化算法. *上海工程技术大学学报*, 2011, 25(3): 262–265.
- 6 Delany SJ, Cunningham P, Wilke W. The limits of CBR in software project estimation. *Proc. of the 6th German Workshop on Case-Based-Reasoning*. Berlin. Springer-Verlag. 1998.
- 7 Mittasa N, Athanasiadesa M, Angelis L. Improving analogy-based software cost estimation by a resampling method. *Information and Software Technology*, 2008, 50(3): 221–230.
- 8 Cost S, Salzberg S. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 1993, 10(1): 57–78.
- 9 Briand LC, Wiecek I. Resource estimation in software engineering. *Encyclopedia of Software Engineering*. New York. John Wiley & Sons. 2002. 1160–1196.
- 10 Shepperd M, Schofield C, Kitchenham B. Effort estimation using analogy. *Proc. of the 18th Int'l Conf. on Software Engineering IEEE CS Press*, 1996. 170–178.
- 11 Huang SJ, Chiu NH, Chen LW. Integration of the grey relational analysis with genetic algorithm for software effort estimation. *European Journal of Operational Research*, 2008, 188(3): 898–909.
- 12 付雅芳, 刘晓东, 李延杰, 邢晓莉. 基于遗传算法和案例推理的软件费用估算方法. *计算机工程与应用*, 2012, 48(8): 86–91.
- 13 吴登生, 李建平, 蔡晨. 件成本估算的粒子群算法类比模型及自助法推断. *管理学报*, 2010, 23(3): 113–120.
- 14 张旭辉, 林海军, 刘明珠, 高豹江. 基于蚁群粒子群优化的卡尔曼滤波算法模型参数辨识. *电力系统自动化*, 2014, 38(4): 44–50.
- 15 Stamelos, Ingelis L. Managing uncertainty in project portfolio cost estimation. *Information and Software Technology*, 2001, 43(13): 759–768.
- 16 Chiu NH. An early software-quality classification based on improved grey relational classifier. *Expert Systems with Applications*, 2009, 36(7): 10727–10734.
- 17 Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaption in particle swarm optimization. *Computers & Operations Research*. 2006, 33: 859–871.
- 18 Medeiros CC, Schirru R. Identification of nuclear power plant transients using the particle swarm optimization algorithm. *Annals of Nuclear Energy*, 2008, 35(4): 576–582.
- 19 Efron B. Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 1979, 7(1): 1–26.
- 20 刘光逊. 基于流域空间属性的水环境响应关系研究[博士学位论文]. 天津: 南开大学, 2012.