

# 说明书

---

## 一种软件缺陷预测的改进排序学习方法

### 技术领域

本发明涉及软件工程中的软件分析及缺陷预测领域，具体为一种软件缺陷预测的改进排序学习方法。

### 背景技术

软件缺陷预测起始于 20 世纪 70 年代，是指利用统计学习技术，根据历史数据以及已经发现的缺陷等软件度量数据预测软件系统的缺陷数目及其类型。缺陷预测技术的目的在于统计计算软件系统的缺陷数、没有发现但还可能存在的缺陷数，以决定系统是否可以交付使用。缺陷预测技术促进了软件质量的提高，同时，也使软件工程技术向前大大地发展了一步。

于 2016 年 12 月 7 日公开的发明专利“基于代价敏感半监督的软件缺陷预测方法”（公开号 CN106201871A），提出了一种整体代价最小化的半监督预测方法，解决了软件缺陷预测中缺陷数据难以获取和类不平衡的问题。于 2016 年 5 月 18 日公开的“一种基于 SMOTE+Boosting 算法的软件缺陷倾向预测方法”（公开号 CN105589806A），使用 SMOTE 算法合成人工少数类样本，根据原样本分布的密度信息计算人工少数类样本的惩罚因子以调整人工样本的权重，使基本分类器区别原样本和人工少数类样本的学习而更偏重原样本和可信度较高的人工样本的学习，提高了少数类和多数类的分类精度。于 2019 年 1 月 8 日公开的发明专利“基于核主成分分析算法的软件缺陷预测模型设计方法”（公开号 CN109165160A），使用核主成分分析算法来对数据集进行降维处理，解决了软件缺陷的度量元中存在冗余数据的问题，提高了机器学习算法的准确率。

软件缺陷预测技术大体上分为静态缺陷预测方法和动态缺陷预测方法。其中，静态缺陷预测方法基于缺陷相关的度量数据对程序模块的缺陷倾向性、缺陷密度或缺陷数进行预测。而动态缺陷预测方法则是基于缺陷或失效产生的时间对系统缺陷随时间的分布进行预测，以发现软件缺陷随其生命周期或其中某些阶段的时间关系的分布规律。

基于度量元的软件缺陷预测是静态软件缺陷预测的一个主要分支，其利用软件模块的各种特征（如代码长度、代码变化等）构造模型，用于预测新的软件模块的缺陷相关信息。模型预测的软件模块的缺陷相关信息不但能反映软件模块的质量，还能用于指导测试资源的分配。最为常见的预测目标包括预测软件模块是否含有缺陷，以及预测软件模块含有缺陷的个数。根据这两种预测目标，目前基于度量元的软件缺陷预测大致分为两类：分类任务的软件缺陷预测和排序任务的软件缺陷预测。

排序任务的软件缺陷预测模型适用于测试资源未知的情况——测试资源少时，只对含缺

陷个数很多的软件模块进行测试；当测试资源多时，含缺陷个数较少的软件模块也能得到测试。现有的排序任务的软件缺陷预测模型的主要构造算法是回归算法，主要由最小二乘法或者最大似然法得到，以拟合度为目标。这样构造排序任务的软件缺陷预测模型存在的问题是，拟合度好的模型给出的模块排序不一定好。因此，有份文献（X. Yang, K. Tang, and X. Yao, A Learning-to-Rank Approach to Software Defect Prediction, IEEE Transaction on Reliability, Vol. 64, No.1, pp.234-246, 2015.）提出排序学习算法，即通过直接优化软件缺陷预测模型的排序性能来构造模型；实验结果表明，与最小二乘法、最大似然法相比，直接优化模型的排序性能得到的预测模型能给出更好的模块排序。

## 发明内容

本发明为了更好地构造符合多种实际需求下的排序任务的软件缺陷预测模型，提出一种软件缺陷预测的改进排序学习方法，优化模型的排序性能、回归性能以及模型复杂度，以更好地适应不同应用场景下的不同需求。

本发明采用以下技术方案来实现：一种软件缺陷预测的改进排序学习方法，包括以下步骤：

步骤 1、从已知缺陷个数的源代码文件中提取度量元和相应的缺陷个数，作为训练数据；

步骤 2、对训练数据进行预处理，包括对重复样本、不一致样本、缺失数据的处理；

步骤 3、利用预处理过的训练数据构造预测模型，使用多目标优化算法同时优化模型的排序性能、回归性能以及模型复杂度，得到一组模型参数，然后根据实际需要选择合适的模型参数；或者先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值，然后使用单目标优化算法求得模型参数；

步骤 4、输入测试数据，利用训练得到的模型分析测试数据，得到相应软件模块的缺陷信息。

在优选的实施例，所述步骤 3 中，使用线性预测模型 $f(x) = \sum_{i=1}^d \alpha_i x_i$ ，使用演化算法得到模型参数，具体过程如下：

S31、使用平均缺陷百分比（Fault-Percentile-Average, FPA）衡量模型的排序性能，平均绝对误差（Average Absolute Errors, AAE）作为模型的回归性能度量，非零参数的个数作为模型复杂度的衡量指标；

S32、利用训练数据构造预测模型：

使用多目标优化算法（这里使用带精英策略的非支配排序的遗传算法 Non-dominated Sorting Genetic Algorithm II，NSGA-II，或者是基于分解的多目标进化算法 Multi-Objective Evolutionary Algorithm based on Decomposition, MOEA/D）同时优化模

型的排序性能、回归性能以及模型复杂度，得到 Pareto 最优解的集合，即一组在排序性能、回归性能以及模型复杂度各有优势的模型参数；然后根据实际需要选择合适的模型参数 $\alpha_i$ ， $i=1\cdots d$ ；

或者：先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值；然后使用单目标优化算法（这里使用组合差分进化 Composite Differential Evolution, CoDE）直接求解得一组模型参数 $\alpha_i$ ，（ $i=1\cdots d$ ）。

本发明相对于现有技术具有如下的优点及效果：

由于本发明同时优化模型的排序性能、回归性能以及模型复杂度，本发明可以根据不同的实际需求构造相应的排序任务的软件缺陷预测模型，即不但可以构造只考虑排序性能的软件预测模型，还可以构造要求模型复杂度较低的软件预测模型，或者考虑回归性能的软件缺陷预测模型等，能很好地适应不同应用场景下的不同需求。

## 附图说明

图 1 是本发明的方法流程图；

图 2 是本发明使用组合差分进化 CoDE 求解的排序学习方法流程图；

图 3 是本发明使用带精英策略非支配排序遗传算法 NSGA-II 求解的排序学习方法流程图。

## 具体实施方式

本发明属于静态缺陷预测技术中的基于度量元的软件缺陷预测，所解决的主要是排序任务的软件缺陷预测问题。下面结合实施例和附图对本发明做进一步详细的描述，但本发明的实施方式不限于此。

本实施例中，软件缺陷预测的改进排序学习方法主要包括以下步骤：

步骤 1：收集训练数据，从已知缺陷个数的源代码文件中提取度量元和相应的缺陷个数作为训练数据。

提取度量元时可从以下方面提取：过程度量、以往缺陷、源代码度量、变更熵、源代码度量的变化及源代码度量的熵等。每一个模块对应一个度量元向量，定义 $x_i$ 表示第  $i$  个模块的度量元向量， $x_i = \{x_{ij} | j = 1, 2, \dots, d\}$ ，其中  $d$  是提取的度量元的总数。度量元的提取是对源代码进行静态分析统计的过程，现在有很多提取度量元的工具，如：Dependency Finder，OOMeter 等。

步骤 2：为了得到更好的预测性能，首先对训练数据进行预处理，包括对重复样本、不一致样本、缺失数据的处理。具体如下：

a) 删除重复样本；

b) 对不一致样本（即相同的度量元的值、不同的缺陷个数），如果存在其中一个样本的

## 说明书

数量最多，则保留该样本，否则删除所有不一致样本；

c) 对缺失数据，采用中位数替代缺失值。

步骤 3：利用预处理过的训练数据构造预测模型，使用多目标优化算法同时优化模型的排序性能、回归性能以及模型复杂度，得到一组模型参数，然后根据实际需要选择合适的模型参数；或者先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值，然后使用单目标优化算法求得模型参数。上述两种构造方法具体分别如下：

方法一，如图 2 所示，假设线性预测模型为  $f(x) = \sum_{i=1}^d \alpha_i x_i$ ，然后使用组合差分进化 (Composite Differential Evolution, CoDE) 算法求得模型参数  $\alpha_i$ ，具体优化过程如下：

a)、输入  $M$  个训练向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ,  $i = 1, 2, \dots, M$ ，以及最小化目标函数  $G(x) = -\lambda_1 G_1(x) + \lambda_2 G_2(x) + \lambda_3 G_3(x)$ 。其中  $G_1(x) = \frac{1}{k} \sum_{m=1}^k \frac{1}{n} \sum_{i=k-m+1}^k n_i$ ，即平均缺陷百分比

(Fault-Percentile-Average, FPA)，用于衡量模型的排序性能； $G_2(x) = \frac{1}{k} \sum_{m=1}^k |h(x^{(m)}) - y^{(m)}|$ ，

即平均绝对误差 (Average Absolute Errors, AAE)，用来衡量模型的回归性能； $G_3(x)$  为非零参数的个数，是模型复杂度的衡量指标。 $\lambda_1$ 、 $\lambda_2$ 、 $\lambda_3$  分别是模型的排序性能、回归性能及模型复杂度的权重。

b)、初始化  $N$  为每一次迭代的种群数量，随机生成  $N$  个解，组成初始种群  $P_0$ ，设置为第 0 代种群  $t = 0$ 。

c)、设  $t_{\max}$  为设定的最大种群代数，即最大迭代次数，当  $t < t_{\max}$  时，循环以下步骤：

(1) 对于  $P_t$  中的每一个解  $v_i$ ，通过杂交或变异方法根据以下三种策略生成三个新解：

Rand-1-bin:

$$u_{i,j,t} = \begin{cases} \alpha_{r1,j,t} + F \cdot (\alpha_{r2,j,t} - \alpha_{r3,j,t}), & \text{if } rand < C_r, \text{ or } i = j_{rand} \\ \alpha_{i,j,t}, & \text{otherwise} \end{cases}$$

Rand-2-bin:

$$u_{i,j,t} = \begin{cases} \alpha_{r1,j,t} + F1 \cdot (\alpha_{r2,j,t} - \alpha_{r3,j,t}) + F \cdot (\alpha_{r4,j,t} - \alpha_{r5,j,t}), & \text{if } rand < C_r, \text{ or } i = j_{rand} \\ \alpha_{i,j,t}, & \text{otherwise} \end{cases}$$

Current-to-rand-1:

$$\vec{u}_{i,t} = \vec{a}_{i,t} + rand \cdot (\vec{a}_{r1,t} - \vec{a}_{i,t}) + F \cdot (\vec{a}_{r2,t} - \vec{a}_{r3,t})$$

每一种策略创造出一个新解  $\vec{u}_{i,t}$ ，其中  $F$  和杂交控制参数  $C_r$  是从  $[F=1.0, C_r = 0.1]$ ， $[F =$

## 说明书

1.0,  $C_r = 0.9$ ],  $[F=0.8, C_r = 0.2]$ 中随机选择的,  $r_1, r_2, r_3, r_4, r_5$  是从 $[1, N]$ 中随机选择出来的不相同的整数,  $\text{rand}$  和  $F_1$  是  $(0, 1)$  中均匀分布的随机数。  $j_{\text{rand}}$  是从 $[1, d]$ 中随机选择的整数。根据目标函数, 从三个新解以及原始解  $v_i$  中选择出最优的解进入到下一代种群中。

(2) 令  $t = t+1$  (即  $t++$ ), 当  $t < t_{\max}$  时返回步骤 (1), 否则转入步骤 d)。

d)、根据目标函数  $G(x)$ , 求得种群  $P_{t+1}$  中的最优解, 即从第  $t$  代种群中选择目标函数值最大的作为最终解, 最终得到预测模型  $f(x) = \sum_{i=1}^d \alpha_i x_i$ 。

方法二, 如图 3 所示, 假设线性预测模型为  $f(x) = \sum_{i=1}^d \alpha_i x_i$ , 然后使用多目标优化算法 (这里使用带精英策略的非支配排序的遗传算法 Non-dominated Sorting Genetic Algorithm II, NSGA-II) 求得模型参数  $\alpha_i$ , 具体优化过程如下:

a)、输入  $M$  个训练向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ,  $i = 1, 2, \dots, M$ , 以及相应的最小化目标函数  $G(x) = (G_1(x), G_2(x), G_3(x))$ 。其中  $G_1(x) = -\frac{1}{k} \sum_{m=1}^k \frac{1}{n} \sum_{i=k-m+1}^k n_i$ , 即负的平均缺陷百分比

(Fault-Percentile-Average, FPA), 用于衡量模型的排序性能;  $G_2(x) = \frac{1}{k} \sum_{m=1}^k |h(x^{(m)}) - y^{(m)}|$ ,

即平均绝对误差 (Average Absolute Errors, AAE), 用来衡量模型的回归性能;  $G_3(x)$  = 非零参数的个数, 是模型复杂度的衡量指标。

b)、初始化  $N$  为每一次迭代的种群数量, 随机生成  $N$  个解, 组成初始种群  $P_0$ , 设置第 0 代种群  $t = 0$ , 并计算种群  $P_0$  的每个解的目标值。

c)、设  $t_{\max}$  为设定的最大种群代数, 即最大迭代次数, 当  $t < t_{\max}$  时, 循环下列步骤:

(1) 使用选择、交叉或变异方法从第  $t$  代种群 (父代) 得到  $N$  个新解, 得到子代, 将父代和子代合并, 进行非支配排序 (对于种群中的每一个个体  $p$  都有两个参数  $n_p, S_p$ , 其中  $n_p$  是种群中支配个体  $p$  得到的个体数,  $S_p$  是种群中被个体  $p$  支配的个体集合), 对每个非支配层的个体进行拥挤度计算, 并根据非支配关系以及拥挤度, 选择种群中合适的个体组成新的父代种群;

本实施例中, 根据 Pareto 等级进行上述非支配排序, 采用 Pareto 支配关系定义 Pareto 等级中的支配。其中, Pareto 支配关系: 对于最小化多目标优化问题, 对于 3 个目标分量  $G_i(x), i=1, \dots, n, n=3$ , 任意给定两个决策变量  $X_a, X_b$ , 如果有以下两个条件成立, 则称  $X_a$  支配  $X_b$ ;

对于  $\forall i=1, \dots, n, n=3$ , 都有  $G_i(X_a) \leq G_i(X_b)$  成立;

$\exists i \in 1, \dots, n, n=3$ , 使得的  $G_i(X_a) < G_i(X_b)$  成立。

如果对于一个决策变量，不存在其他决策变量能够支配它，那么就称该决策变量为非支配解。

Pareto 等级：在一组解中，非支配解 Pareto 等级定义为 1，将非支配解从解的集合中删除，剩下解的 Pareto 等级定义为 2，以此类推，可以得到该解集中所有解的 Pareto 等级。

(2) 令  $t=t+1$  (即  $t++$ )，当  $t < t_{\max}$  时返回步骤 (1)，否则转入步骤 d)。

d)、根据实际的需求，从第  $t$  代种群中选择合适的解作为最终解，得到预测模型  $f(x) = \sum_{i=1}^d \alpha_i x_i$ 。

步骤 4：输入测试数据，即相应软件模块度量元的值，利用训练得到的模型分析测试数据，得到相应软件模块的缺陷信息；根据模型预测结果分配测试资源。具体过程为：

S41、输入测试数据；

S42、根据步骤 3 得到的模型参数  $\alpha_i$ ，( $i=1 \cdots d$ )，得到预测模型  $f(x) = \sum_{i=1}^d \alpha_i x_i$ ；

S43、使用预测模型得到测试数据的缺陷信息。

上述实施例为本发明较佳的实施方式，但本发明的实施方式并不受上述实施例的限制，其他的任何未背离本发明的精神实质与原理下所作的改变、修饰、替代、组合、简化，均应为等效的置换方式，都包含在本发明的保护范围之内。

# 权 利 要 求 书

1、一种软件缺陷预测的改进排序学习方法，其特征在于，包括以下步骤：

步骤 1、从已知缺陷个数的源代码文件中提取度量元和相应的缺陷个数，作为训练数据；

步骤 2、对训练数据进行预处理，包括对重复样本、不一致样本、缺失数据的处理；

步骤 3、利用预处理过的训练数据构造预测模型，使用多目标优化算法同时优化模型的排序性能、回归性能以及模型复杂度，得到一组模型参数，然后根据实际需要选择合适的模型参数；或者先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值，然后使用单目标优化算法求得模型参数；

步骤 4、输入测试数据，利用训练得到的模型分析测试数据，得到相应软件模块的缺陷信息。

2、根据权利要求 1 所述的软件缺陷预测的改进排序学习方法，其特征在于，所述步骤 3 使用线性预测模型，根据演化算法得到模型参数：

S31、使用平均缺陷百分比 FPA 衡量模型的排序性能，平均绝对误差 AAE 作为模型的回归性能度量，非零参数的个数作为模型复杂度的衡量指标；

S32、利用预处理过的训练数据构造预测模型，使用多目标优化算法同时优化模型的排序性能、回归性能以及模型复杂度，得到一组模型参数，然后根据实际需要选择合适的模型参数；或者先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值，然后使用单目标优化算法求得模型参数。

3、根据权利要求 2 所述的软件缺陷预测的改进排序学习方法，其特征在于，步骤 3 中，假设线性预测模型为  $f(x) = \sum_{i=1}^d \alpha_i x_i$ ，使用组合差分进化算法求得模型参数  $\alpha_i$ ：

a)、输入  $M$  个训练向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ,  $i = 1, 2, \dots, M$ ，以及最小化目标函数  $G(x) = -\lambda_1 G_1(x) + \lambda_2 G_2(x) + \lambda_3 G_3(x)$ ；其中  $G_1(x) = \frac{1}{k} \sum_{m=1}^k \frac{1}{n} \sum_{i=k-m+1}^k n_i$ ，即平均缺陷百分比 FPA，

用于衡量模型的排序性能； $G_2(x) = \frac{1}{k} \sum_{m=1}^k |h(x^{(m)}) - y^{(m)}|$ ，即平均绝对误差 AAE，用来衡量模型的

回归性能； $G_3(x)$  为非零参数的个数，是模型复杂度的衡量指标； $\lambda_1$ 、 $\lambda_2$ 、 $\lambda_3$  分别是模型的排序性能、回归性能及模型复杂度的权重；

b)、初始化  $N$  为每一次迭代的种群数量，随机生成  $N$  个解，组成初始种群  $P_0$ ，设置为第 0 代种群  $t = 0$ ；

c)、设  $t_{\max}$  为设定的最大种群代数，当  $t < t_{\max}$  时，循环以下步骤：

(1) 对于  $P_t$  中的每一个解  $v_i$ ，通过杂交或变异方法根据以下三种策略生成三个新解：

Rand-1-bin:

$$u_{i,j,t} = \begin{cases} \alpha_{r1,j,t} + F \cdot (\alpha_{r2,j,t} - \alpha_{r3,j,t}), & \text{if } rand < C_r, \text{ or } i = j_{rand} \\ \alpha_{i,j,t}, & \text{otherwise} \end{cases}$$

Rand-2-bin:

$$u_{i,j,t} = \begin{cases} \alpha_{r1,j,t} + F1 \cdot (\alpha_{r2,j,t} - \alpha_{r3,j,t}) + F \cdot (\alpha_{r4,j,t} - \alpha_{r5,j,t}), & \text{if } rand < C_r, \text{ or } i = j_{rand} \\ \alpha_{i,j,t}, & \text{otherwise} \end{cases}$$

Current-to-rand-1:

$$\vec{u}_{i,t} = \vec{\alpha}_{i,t} + rand \cdot (\vec{\alpha}_{r1,t} - \vec{\alpha}_{i,t}) + F \cdot (\vec{\alpha}_{r2,t} - \vec{\alpha}_{r3,t})$$

每一种策略创造出一个新解 $\vec{u}_{i,t}$ , 其中 F 和杂交控制参数  $C_r$  是从  $[F=1.0, C_r = 0.1], [F = 1.0, C_r = 0.9], [F=0.8, C_r = 0.2]$  中随机选择的,  $r1$ 、 $r2$ 、 $r3$ 、 $r4$ 、 $r5$  是从  $[1, N]$  中随机选择的不相同整数,  $rand$  和  $F1$  是  $(0, 1)$  中均匀分布的随机数,  $j_{rand}$  是从  $[1, d]$  中随机选择的整数; 根据目标函数, 从三个新解以及原始解  $v_i$  中选出最优的解进入到下一代种群中;

(2) 令  $t = t+1$ , 当  $t < t_{max}$  时返回步骤 (1), 否则转入步骤 d);

d)、根据目标函数  $G(x)$ , 求得种群  $P_{t+1}$  中的最优解, 即从第  $t$  代种群中选择目标函数值最大的作为最终解, 最终得到预测模型  $f(x) = \sum_{i=1}^d \alpha_i x_i$ 。

4、根据权利要求 2 所述的软件缺陷预测的改进排序学习方法, 其特征在于, 步骤 3 中, 假设线性预测模型为  $f(x) = \sum_{i=1}^d \alpha_i x_i$ , 使用多目标优化算法求得模型参数  $\alpha_i$ :

a)、输入  $M$  个训练向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ ,  $i = 1, 2, \dots, M$ , 以及相应的最小化目标函数  $G(x) = (G_1(x), G_2(x), G_3(x))$ ; 其中  $G_1(x) = -\frac{1}{k} \sum_{m=1}^k \frac{1}{n} \sum_{i=k-m+1}^k n_i$ , 即负的平均缺陷百分比 FPA,

用于衡量模型的排序性能;  $G_2(x) = \frac{1}{k} \sum_{m=1}^k |h(x^{(m)}) - y^{(m)}|$ , 即平均绝对误差 AAE, 用来衡量模型的

回归性能;  $G_3(x)$  = 非零参数的个数, 是模型复杂度的衡量指标;

b)、初始化  $N$  为每一次迭代的种群数量, 随机生成  $N$  个解, 组成初始种群  $P_0$ , 设置第 0 代种群  $t = 0$ , 并计算种群  $P_0$  的每个解的目标值;

c)、设  $t_{max}$  为设定的最大种群代数, 当  $t < t_{max}$  时, 循环下列步骤:

(1) 使用选择、交叉或变异方法从第  $t$  代种群父代得到  $N$  个新解, 得到子代, 将父代和子代合并, 进行非支配排序, 对每个非支配层的个体进行拥挤度计算, 并根据非支配关系以及拥挤度, 选择种群中合适的个体组成新的父代种群;

(2) 令  $t=t+1$ , 当  $t < t_{max}$  时返回步骤 (1), 否则转入步骤 d);

d)、根据实际的需求, 从第  $t$  代种群中选择合适的解作为最终解, 得到预测模型  $f(x) =$



$\sum_{i=1}^d \alpha_i x_i$ 。

5、根据权利要求1所述的软件缺陷预测的改进排序学习方法，其特征在于，步骤1提取度量元时，从以下方面提取：过程度量、以往缺陷、源代码度量、变更熵、源代码度量的变化和源代码度量的熵。

6、根据权利要求1所述的软件缺陷预测的改进排序学习方法，其特征在于，步骤2中预处理过程如下：

- a) 删除重复样本；
- b) 对不一致样本，如果存在其中一个样本的数量最多，则保留该样本，否则删除所有不一致样本；
- c) 对缺失数据，采用中位数替代缺失值。

## 说明书摘要

---

本发明为软件缺陷预测的改进排序学习方法，包括步骤：从已知缺陷个数的源代码文件中提取度量元和相应的缺陷个数，作为训练数据；对训练数据进行预处理；利用预处理过的训练数据构造预测模型，使用多目标优化算法同时优化模型的排序性能、回归性能以及模型复杂度，得到一组模型参数，然后根据实际需要选择合适的模型参数，或者先根据实际需求对模型的排序性能、回归性能以及模型复杂度赋予相应的权值，然后使用单目标优化算法求得模型参数；利用训练得到的模型分析测试数据，得到相应软件模块的缺陷信息。本发明能优化模型的排序性能、回归性能以及模型复杂度，更好适应不同应用场景下的不同需求。

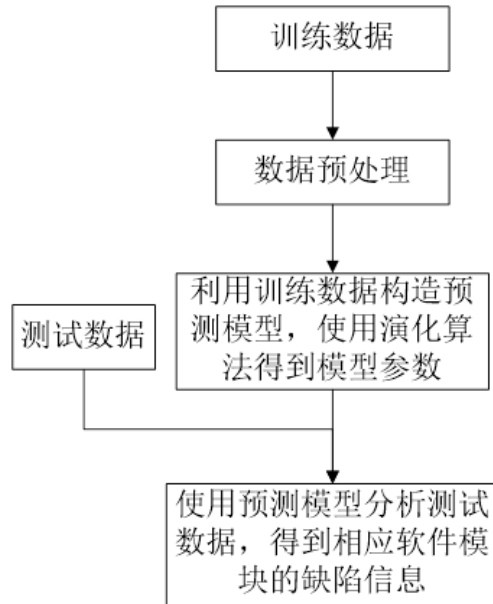


图 1

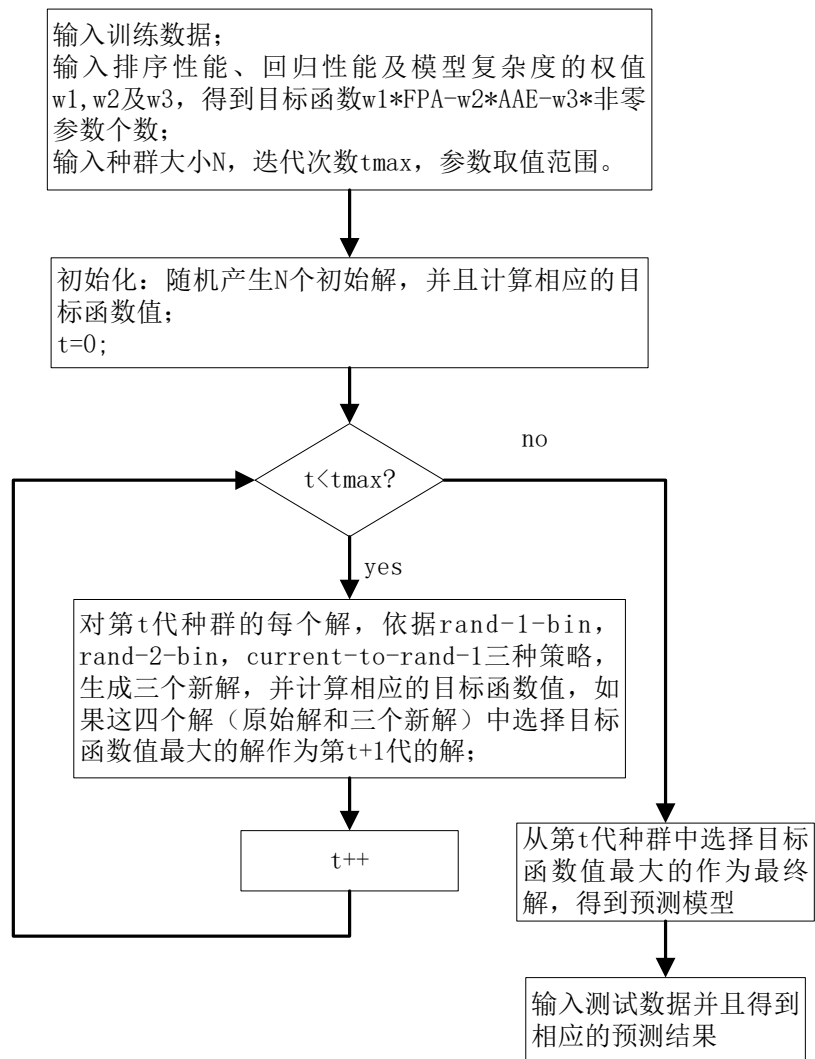


图 2

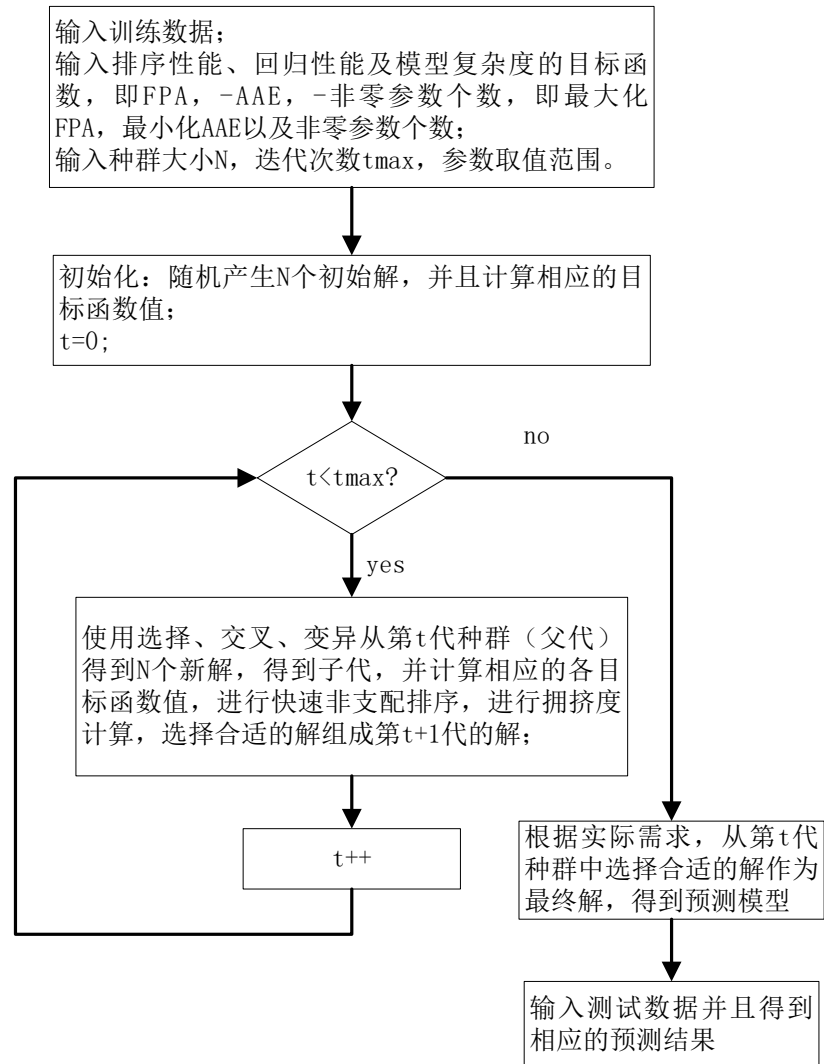


图 3