

演化算法中一种基于单分类的预选择策略

张晋媛 周爱民 张桂戌

(华东师范大学计算机科学与技术学院 上海 200062)

摘 要 演化算法中,预选择算子用于为后续的环境选择过程筛选出好的潜在候选后代解. 现有预选择算子大多基于适应值评估、代理模型或分类模型. 由于预选择过程本质上是一个分类过程,因此基于分类的预选择过程天然适用于演化算法. 先前研究工作采用二分类或多分类模型进行预选择,需预先准备“好”和“差”两组或具有区分性的多组训练样本来构建分类模型,而随着演化算法的执行,“好”解和“差”解之间的界限将愈加模糊,因此准备具有区分性的两组或多组训练样本将变得具有挑战性. 为解决该问题,本文提出了一种基于单分类的预选择策略(One-class Classification based PreSelection, OCPS),首先将当前种群中的解均视为“好”类样本,之后只利用该类“好”样本构建单分类模型,然后利用构建的模型对产生的多个候选解进行标记与选择. 提出的策略应用在三个代表性演化算法中,数值实验结果表明,提出的策略能够提升现有演化算法的收敛速度.

关键词 全局优化;演化算法;分类;单分类;预选择

中图法分类号 TP18 DOI号 10.11897/SP.J.1016.2020.00233

A Pre-Selection Based on One-Class Classification in Evolutionary Algorithms

ZHANG Jin-Yuan ZHOU Ai-Min ZHANG Gui-Xu

(School of Computer Science and Technology, East China Normal University, Shanghai 200062)

Abstract In evolutionary algorithms, a preselection operator is a part of the reproduction procedure that aims to choose some promising candidate offspring solutions for further environmental selection. The preselection operator can help to improve the algorithm performance significantly if it is correctly utilized. In preselection, the unpromising candidate solutions can be discarded before the real function evaluations, and thus, the computational resources can be saved. Also, using the preselection operator in evolutionary algorithms will help the algorithms to generate more potentially good solutions in one generation without adding more function evaluations. Most existing preselection operators are based on fitness evaluations, surrogate models, and classification models. Since a preselection operator can be regarded as a classification procedure, where selected solutions can be treated as ‘positive’ ones and the discarded solutions are ‘negative’ ones. In terms of this situation, using the classification model to assist preselection is a natural choice for evolutionary algorithms. Previous research uses binary and/or multi-class classification models to guide preselection, in which ‘positive’ and ‘negative’ training samples or more classes of samples should be prepared to build the classification model. However, after several generations, for some of the evolutionary algorithms, almost all of the solutions in the current population are relatively ‘positive’ ones. Thus, the gap between ‘positive’ and ‘negative’ solutions is not easy to be defined. Furthermore, for these kinds of evolutionary algorithms, preparing ‘negative’ training samples has three disadvantages: (1) to reduce the accuracy of the classification model on predic-

tion, (2) to improve the cost on model building, (3) to increase the complexity of algorithms. For this reason, it is not trivial to prepare ‘positive’ and ‘negative’ training samples. To deal with this problem and avoid the above disadvantages, we consider employing the one-class classification (OCC) model, which only needs one class of ‘positive’ training samples in the classification model-building procedure to guide the preselection. Thus, the model-building procedure can become simple by only defining the ‘positive’ training samples. Based on this idea, this paper proposes a one-class classification based preselection (OCPS) scheme that uses the OCC model for the preselection. The proposed OCPS scheme mainly has three components: at first, solutions in the current population are all labeled as ‘positive’ ones with label +1. Then, the labeled solutions are used to build an OCC model. Thirdly, for each parent solution, a set of candidate offspring solutions are generated. The built model is employed to label the newly generated offspring solutions. And only the one with the ‘positive’ label will be selected as the real offspring solution of its parent for the following environmental selection. If there are more than one candidate solutions are labeled as ‘positive’ or there are not any solutions labeled as ‘positive’, the real offspring solution will be randomly selected from these candidates. The proposed OCPS scheme is applied to three state-of-the-art evolutionary algorithms and studied on a test suite. The experimental results show the potential of OCPS on improving the performance of some existing evolutionary algorithms and reducing the number of function evaluations. Also, the OCPS performs better than the other models in most situations.

Keywords global optimization; evolutionary algorithm; classification; one-class classification; preselection

1 引 言

优化是科学研究和工业应用中广泛存在的一类挑战性问题,依据优化目标的不同,优化问题有多种表示形式,本文考虑以下连续优化问题:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x})$$

其中 $\mathbf{x} = (x^1, x^2, \dots, x^n)$ 是一个决策向量, Ω 是搜索空间的可行域, f 是目标函数.

若上述目标函数是凸函数,首选求解方法是基于梯度的经典优化方法^[1]. 然而很多实际优化问题都是非凸函数,因而需要启发式方法来求解. 其中,借鉴自然界物种进化思想发展而来的演化算法(Evolutionary Algorithms, EA)^[2-5]吸引了众多研究者和应用者的目光,在理论分析、算法设计与应用等方面对算法进行了研究^[6-7]. 目前针对不同类型的优化问题也提出了不同的 EA 算法框架^[5, 8-10].

一般的 EA 算法框架如图 1 所示,主要包含初始化、后代产生、环境选择、结束判断等四个核心组成部分. 算法首先在初始化过程中产生包含一组解的初始种群;之后,在后代产生过程中利用当前种群

中的解产生后代种群;然后,在环境选择过程中从当前种群与后代种群中选择出一组好解组成新的种群;最后,判断当前算法执行是否满足结束条件,若满足结束条件则算法终止,否则,继续执行后代产生、选择和结束判断过程.

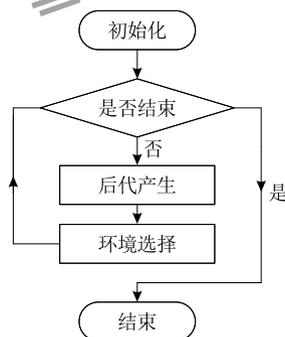


图 1 EA 算法框架

在上述四个组成部分中,后代产生和环境选择是关键,在过去的几十年中 EA 研究工作大部分是围绕这两部分展开的. 需要特别指出的是,EA 中的选择过程实际包括: 交配选择、环境选择和预选择. 其中,交配选择从当前种群中选择好的父代解来产生后代解;环境选择从当前种群和后代种群中挑选

好解组成新的种群; 预选择因不同的应用场景, 有多种定义^[11-12], 本文将预选择过程定义为从一组候选后代解中选出好解的过程. 近几年, 后代解选择策略受到了越来越多研究者们的关注, 并通过理论分析对后代解选择策略中需要关注的问题进行了论证, 文献[13]验证了在一些情况下, 演化算法中使用非精英选择策略会比精英选择策略选出更好的后代解; 文献[14]验证了对于噪声环境中的优化问题, 使用保守选择策略选出的后代解具有更好的抗噪性能.

因为预选择过程并非 EA 的一个必要组成部分, 所以在许多 EA 实现过程中预选择过程总是被忽略. 然而, 好的预选择过程能够有效帮助提升 EA 算法性能, 这是因为预选择过程中差的候选解可以在进行真正的函数值评估前直接被筛选掉, 从而避免了计算资源的浪费并提高了算法搜索效率. 图 2 显示了一个带有预选择过程的 EA 算法框架. 预选择的关键是如何评估候选解的质量, 即判断一个候选解为“好”还是“差”. 据此, 现有预选择过程可以分为以下三类:

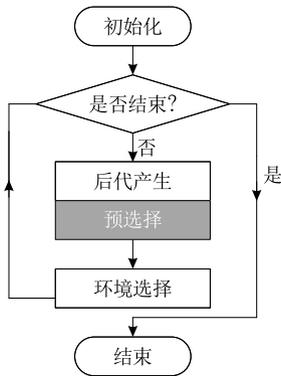


图 2 预选择过程定义

(1) 基于适应值评估的过程^[15-16]. 这类方法直接使用目标函数评估候选解, 并选出适应值最优的一个解. 这一过程能够准确的评估每个候选解的质量, 然而, 由于许多评估过的解只在挑选“好”解时使用一次, 因此造成了适应值评估的大量浪费.

(2) 基于代理模型的过程^[17-21]. 这类方法首先构建一个代理模型来近似解和函数值之间的关系, 其次根据模型得到候选后代解的近似函数值, 最后根据近似值选出好的后代解. 虽然这一过程会因构建代理模型而带来附加的计算代价, 但是却能够显著地降低适应值评估次数.

(3) 基于分类的过程^[22-25]. 这类方法是上述方法的一个特例, 主要的贡献在于将预选择视为一个分类过程, 其中选中的后代解属于“好”类, 丢弃的解是“差”类. 这类方法使用当前解构建分类模型并预

测候选后代解的质量, 进而选出其中的“好”解. 因此, 使用这类方法进行预选择过程也是很自然的一种策略.

文献[22]中提出了一个基于二分类的预选择策略(Classification based Preselection, CPS), 使用二分类模型对候选后代解的质量进行评估. 该方法中, 为了构建分类模型, 首先需要定义训练样本, 然而大多数的 EA 算法在迭代多次后, 当前种群中的解有很大概率都是相对“好”解. 因此, 对这类算法而言, 定义“差”样本可能会影响模型的准确率、增加模型构建成本、增加算法复杂性; 其次, EA 的优化过程更加关心的是“好”解, 大多数“差”解的信息会被忽略. 因此, 为了简化算法设计流程、降低算法复杂度并提升分类模型在 EA 中的准确率, 我们考虑使用只需要一类训练样本进行模型构建的单分类(One-Class Classification, OCC)^[26-27]模型来辅助预选择过程. 为此, 针对演化全局优化算法以及文献[22]提出的 CPS 策略的样本定义过程中存在的不足, 文献[28]初步提出了基于单分类的预选择(One-class Classification based Preselection, OCPS)策略. 本文对文献[28]中的研究工作进行了深入和拓展, 主要贡献如下:

(1) 提出了通用的 OCPS 策略: 与文献[28]中的工作相比, 本文提出的策略能够应用在不同类型的 EA 中;

(2) 进行了系统的实验和分析: 与文献[28]相比, 本文对 OCPS 策略的性能进行了系统的分析、对比, 验证了 OCPS 策略的有效性;

(3) 与其他预选择策略对比分析: 与文献[28]相比, 本文分别将 OCPS 策略与基于二分类的预选择策略、基于代理模型的预选择策略等进行了系统性对比分析, 进一步分析了 OCPS 策略的优势, 验证了 OCPS 具有更好的优化性能.

2 单分类预选择策略

本节对提出的 OCPS 策略的实现过程及算法框架进行了详细的介绍.

2.1 训练样本定义

为了在 EA 中应用 OCPS 策略, 在模型的构建过程中, 首先需要进行训练样本定义. 不同于 CPS^[22]需要将种群分为两类作为“好”和“差”样本来构建模型, OCPS 策略直接使用当前种群中的解作为唯一“好”样本构建模型.

在 OCPS 中,解 x 为一个特征向量或“好”类样本点,使用解种群 P 作为“好”样本集来构建 OCC 模型.

2.2 模型构建

OCPS 策略的实现过程中,其次需要考虑模型的构建.

与传统二类、多分类模型相比,单分类(One-Class Classification, OCC)^[26]模型的主要特点是:仅使用一类训练样本进行模型构建.因此,为了提升模型准确率,大多数的 OCC 算法通过估计训练样本的密度来构建模型.训练样本的单一,使得 OCC 模型训练样本的定义得到了简化,进而分类算法的复杂度也进一步降低.

与传统二分类模型相似,在 EA 中 OCC 模型的目标是找到“好”解与异常解之间的边界.然而,由于异常解自身类别信息的缺失,因此, OCC 模型的构建过程需要尽可能地用一个最小的区域来覆盖所有“好”解以降低误分类的概率.以基于支持向量机(Support Vector Machine, SVM)的 OCC 模型^[27]为例,这类模型通过寻找到一个能将所有“好”解划分在同一边的超平面,且超平面到“好”类的距离尽可能的小,从而实现分类.

如果 $\langle x, lx \rangle$ 是一个训练样本,其中 x 是一个特征向量, lx 是样本的类标记.则在 OCC 模型中, lx 的取值有且仅有一个即 $lx = +1$,因此, OCC 模型中的训练样本实际为 $\langle x, +1 \rangle$,则构建 OCC 模型的目的是找到关系: $lx = OCC(x)$.

以图 3 为例,图 3(a)中五角星表示“好”的训练样本,虚线框表示仅根据“好”的样本点构建的 OCC 模型的边界;图 3(b)给出了 OCC 模型的预测结果,即六角形表示的数据点为“好”类,十字星表示的数据点为界外点即属于“差”类.

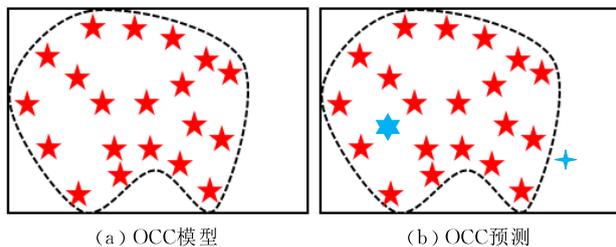


图 3 OCC 模型的实现过程示例图

本文使用了文献[29]中提出的基于 SVM 的 OCC 算法 OCSVM.算法模型定义如下,其中算法的问题定义为

$$\begin{aligned} \min_{\omega, \delta, \rho} & \frac{1}{2} \omega^T \omega - \rho + \frac{1}{\nu N} \sum_{i=1}^N \delta^i \\ \text{s. t.} & \omega^T \varphi(x^i) \geq \rho - \delta^i \\ & \delta^i \geq 0, i = 1, \dots, N \end{aligned}$$

决策函数定义为

$$\begin{aligned} f(x) &= \text{sgn}(\omega^T \varphi(x^i) - \rho) \\ &= \text{sgn}\left(\sum_{i=1}^N \alpha^i K(x^i, x) - \rho\right) \end{aligned}$$

其中, x 是训练数据, $\nu \in (0, 1]$ 定义了异常值分数的上限以及训练数据个数的下限, α 是拉格朗日乘子, ω 和 ρ 定义了分类超平面.算法通过求解上述问题,得到模型相关参数.然后,使用决策函数预测新数据的类别.

2.3 标记后代解与选择

OCPS 策略实现过程中,第三需要考虑后代标记与选择过程.

本文提出的 OCPS 策略主要包括 3 个步骤:(1)为每个父代解产生 M 个后代解;(2)使用 OCC 模型对 M 个后代解的类别进行标记;(3)根据标记结果,从标记为“好”的解中随机选出一个作为父代真正的后代解,如果没有解被标记为“好”,则从 M 个候选解中随机选出一个作为真正的后代解.

2.4 OCPS 算法框架

如图 4 所示,将基于 OCPS 的演化算法框架标记为 OCPS-EA.其算法细节如算法 1 所示.

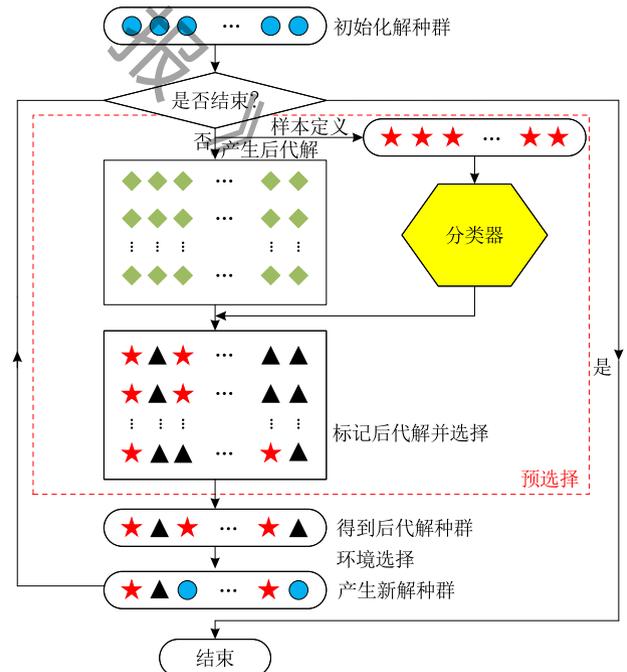


图 4 OCPS-EA 算法框架

算法 1. OCPS-EA 算法.

1. 初始化解种群 $P = \{x^1, x^2, \dots, x^N\}$;

2. 主循环

- 2.1. 将每个解 $x \in P$ 标记为 $lx = +1$;
- 2.2. 使用训练样本 $\{\langle x, lx \mid x \in P \rangle\}$ 构建 OCC 模型 $lx = OCC(x)$;
- 2.3. 对每个解 $x \in P$
 - 2.3.1. 产生 M 个候选后代解 $Y = \{y^1, y^2, \dots, y^M\}$;
 - 2.3.2. 令 $S = \{y \in Y \mid ly = 1\}$, 若 $S = \emptyset$, 令 $S = Y$;
 - 2.3.3. 随机选出一个 $y \in S$ 作为 x 的后代解;
 - 2.3.4. 评估 y 的函数适应值, 如果 $f(y) < f(x)$, 则 $x = y$;
 - 2.3.5. 若未满足终止条件, 则返回步 2.1;
3. 输出当前种群中的最优解.

算法 1 的具体细节解释如下:

- 步 1: 种群初始化, 即从搜索可行域 Ω 中随机产生 N 个解 x 组成初始解种群 P .

- 步 2.1: 训练样本定义, 即将当前解种群中的所有解的类别均标记为“+1”.

- 步 2.2: 模型构建, 即使用训练样本来构建 OCC 模型.

- 步 2.3.1: 产生后代解, 即为当前种群中的每个解 x 都产生 M 个候选后代解.

- 步 2.3.2: 标记后代解, 即使用 OCC 模型对候选后代解进行标记以构建候选解集合.

- 步 2.3.3: 选择后代解, 即从候选解集合中随机选出一个解作为父代真正的后代解.

- 步 2.3.4: 环境选择, 即首先评估后代解的函数适应值, 通过比较父代和后代解的适应值, 选出其中更优的一个作为新解进入下一代.

3 实验验证

本节通过实验对提出的 OCPS 策略的性能进行验证. 实验使用了文献[30]提出的 YLL 测试集中的前 13 个测试函数作为测试样例, 13 个题目中, $f_1 \sim f_4$ 是单模问题; f_5 在维数 $n=2$ 和 $n=3$ 时, 是单模问题, 当 $n>5$ 时, 是多模问题; f_6 是阶梯函数; f_7 是白噪声函数; $f_8 \sim f_{13}$ 是多模函数. 在后续几节中, 首先, 节 3.2、节 3.3、节 3.4 分析了训练样本规模、产生的后代解数目 M 、OCC 模型中的核函数等参数对 OCPS 策略性能的影响; 其次, 节 3.5、节 3.6 对 OCPS 在 DE/EDA^[31], JADE^[32], EDA/LS^[33] 等 3 个 EA 框架、不同变量维度中的优化性能进行了实验验证; 之后, 节 3.7、节 3.8 将基于单分类与二分类的预选择策略、基于单分类与代理模型的预选择策略分别进行了实验对比分析; 最后, 节 3.9 对基于

单分类的预选择策略、基于代理模型的预选择策略和原算法的算法运行时间进行了对比分析.

3.1 实验设置

对所有测试样例, 变量的维数 $n=30$; 对 DE/EDA^[31] 及其变形算法, 种群规模 $N=100$; 对 JADE^[32] 及其变形算法, 种群规模 $N=50$; 对 EDA/LS^[33] 及其变形算法, 种群规模 $N=150$; 所有算法运行过程中的最大函数适应值评估次数 $FES=300\,000$; 每个算法在每个测试题上独立运行 30 次; 对 OCPS 策略, 每个父代解产生的后代解数目 $M=3$. 其它的参数均与原始算法相同^[31-33].

实验使用文献[29]中提出的 OCSVM 算法构建 OCC 模型, 使用径向基函数(Radial Basis Function, RBF)作为核函数, OCC 模型通过 SVM 库函数 libsvm^[34-35] 实现, 模型的控制参数均为 libsvm 中设置的默认值.

实验中将 OCPS 与文献[22]中提出的 CPS 策略进行了对比分析, 为了保证对比实验中模型的一致性、减少参数对算法性能的影响, 使用基于 SVM 模型的 CPS 策略进行实验对比, SVM 模型同样通过 libsvm 实现. 为了更好地进行区分, 将文献[22]中基于二分类的预选择策略标记为 BCPS.

在后续几部分实验中, 将对比分析 DE/EDA、JADE、EDA/LS 和它们基于 OCPS 策略的变形算法 OCPS-DE/EDA、OCPS-JADE、OCPS-EDA/LS, 以及它们基于 BCPS 策略的变形算法 BCPS-DE/EDA、BCPS-JADE、BCPS-EDA/LS.

本文使用 Wilcoxon^[36] 排序对比方法对实验结果进行比较, 表中“+”、“-”、“~”表示使用当前算法得到的最优解优于、差于、相近于其它算法.

3.2 训练样本规模分析

本节对训练样本的规模对 OCPS 策略性能的影响进行实验分析, OCPS-DE/EDA 分别在样本规模为 $1N$ 、 $2N$ 、 $3N$ 、 $4N$ 、 $5N$ 时进行实验 (N 为种群规模). 图 5 统计了在 13 个测试题中得到相同最优解时算法需要的函数适应值评估次数(FES)的均值.

图 5 中结果表明在 $f_1 \sim f_3$ 、 f_6 、 f_{10} 、 f_{13} 中, 基于 $1N$ 的 OCPS-DE/EDA 需要的适应值评估次数最少; 在 f_5 、 f_7 、 f_8 、 f_{11} 、 f_{12} 中, $1N$ 和 $2N$ 均在得到部分最优解时需要的函数适应值评估次数最少; 在 f_4 、 f_9 中基于 $2N$ 、 $3N$ 的算法分别需要的适应值评估次数最少. 总体而言, 种群规模为 $1N$ 时 OCPS 优化性能最好.

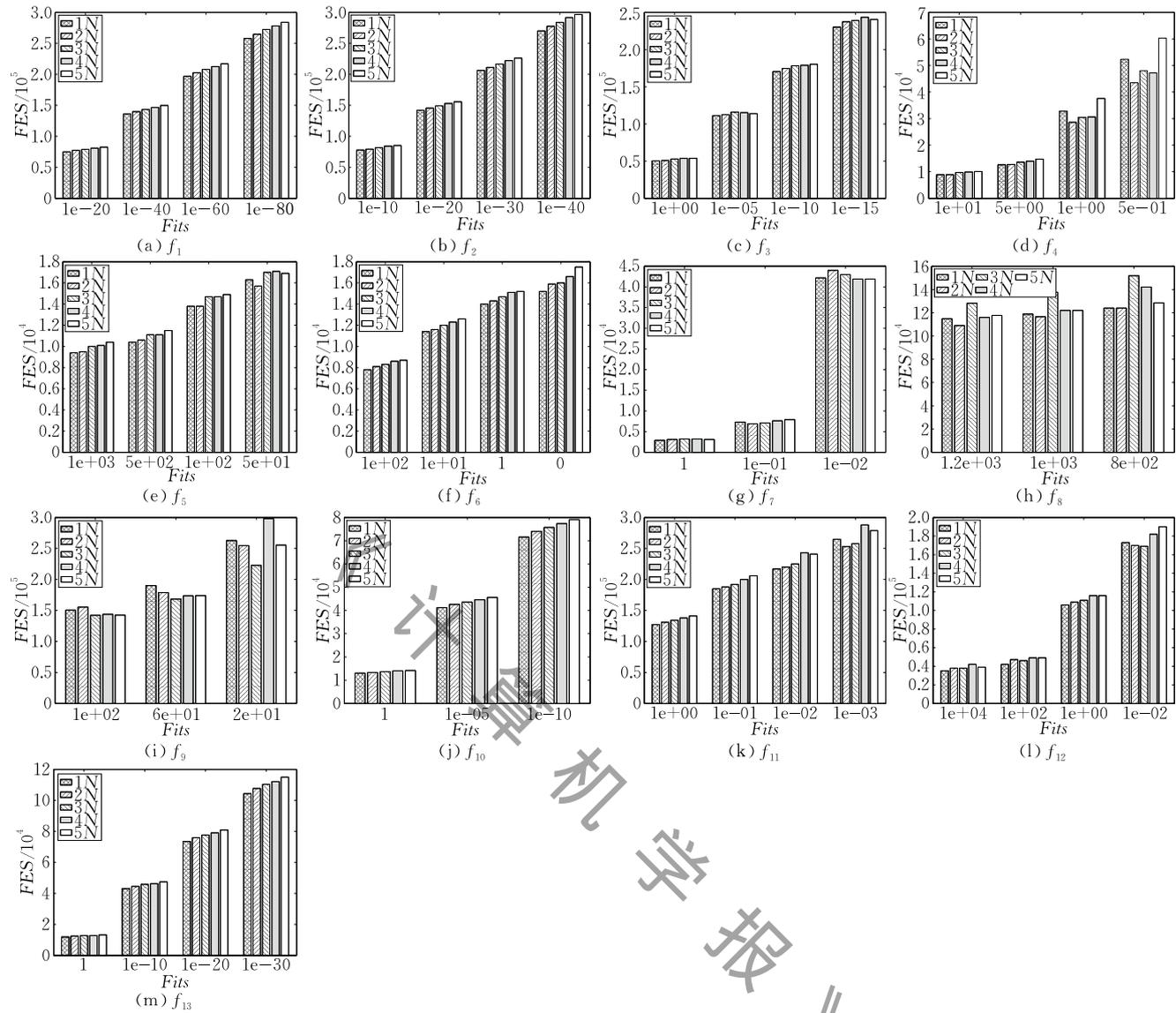


图 5 训练样本规模为 $1N, 2N, 3N, 4N, 5N$ 时, OCPS-DE/EDA 在 13 个测试题中得到相同最优解时需要的适应值评估次数的均值统计图(其中纵坐标表示 FES , 横坐标表示最优解适应值 $Fits$)

3.3 产生的候选后代解数目分析

本节对产生的候选后代解数目对 OCPS 优化性能的影响进行了实验分析. 当 $M=2, 3, 4, 5, 6$ 时, 分别对 OCPS-DE/EDA 算法进行了实验验证. 图 6 统计了在得到相同最优解时, $M=2, 3, 4, 5, 6$ 分别需要的平均函数适应值评估次数.

图 6 中结果表明, 当 $M=2$ 时, OCPS-DE/EDA 的优化性能总体最差; 当 $M \geq 3$ 时, 在 $f_1 \sim f_3, f_5 \sim f_7, f_{10} \sim f_{13}$ 中, 基于不同 M 值的 OCPS-DE/EDA 算法的优化性能总体相似; 在 f_4 中, $M=2$ 和 $M=6$ 时 OCPS-DE/EDA 的优化性能分别为最好和最差; 在 f_8 中, $M=3$ 时算法优化性能最好, $M=5$ 时算法优化性能最差; 在 f_9 中, $M=5$ 和 $M=3$ 时算法优化性能分别为最好和最差. 综合考虑 M 取值不同时算

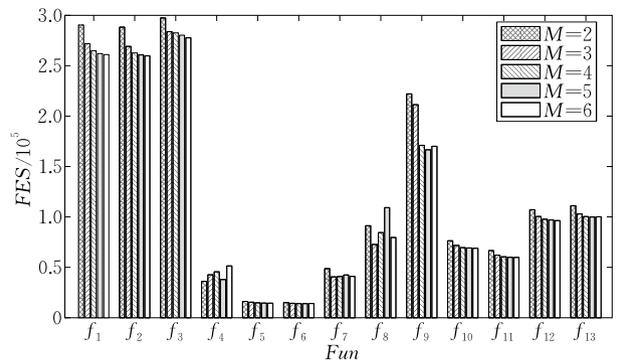


图 6 $M=2, 3, 4, 5, 6$ 时, OCPS-DE/EDA 在 13 个测试题中得到相同最优解时, 需要的适应值评估次数的均值统计图(其中纵坐标表示 FES , 横坐标表示测试题目 Fun)

法的优化性能以及后代解产生过程中需要的时间复杂度, 选择 $M=3$ 进行后续实验更为合适.

3.4 OCC 模型中核函数分析

本节对实验中使用的 OCSVM 模型中的核函数对算法优化性能的影响进行了实验分析. OCPS-DE/EDA 分别使用线性核函数 (Linear Kernel, $t=0$)、多项式核函数 (Polynomial Kernel, $t=1$)、径向基核函数 (Radial Basis Function Kernel, $t=2$)、Sigmoid 核函数 (Sigmoid Kernel, $t=3$) 进行了实验. 图 7 是算法在 12 个测试题中最优解函数适应值

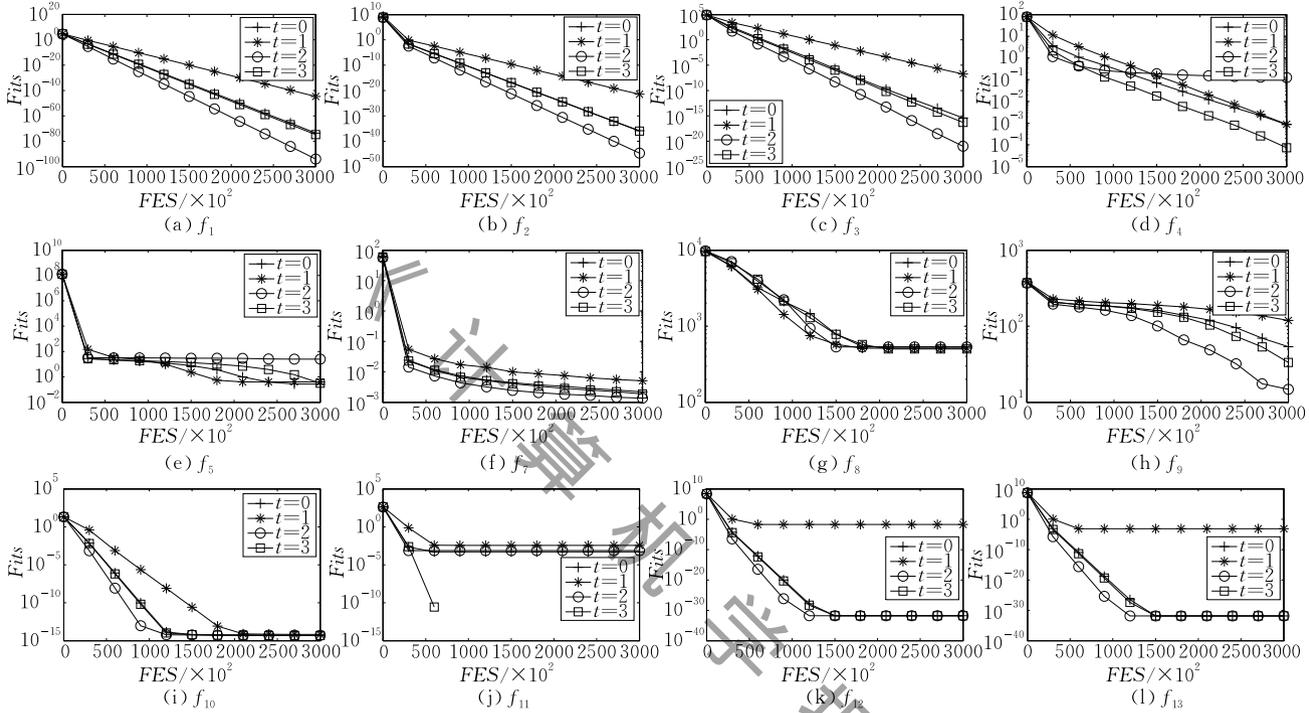


图 7 基于不同核函数的 OCPS-DE/EDA 在 12 个测试题中的运行过程统计图(其中纵坐标为适应值 $Fits$, 横坐标为 FES)

3.5 不同算法框架中的实验结果

本节对基于 OCPS、BCPS 的 DE/EDA、JADE、EDA/LS 以及原始 DE/EDA、JADE、EDA/LS 算法在 13 个测试题上进行实验对比分析. 表 1 是对几种算法得到的最优解的函数适应值的中值、均值和方差的统计对比. 表 2 是对表 1 中的最优解适应值的排序结果的统计. 实验结果总结如下.

对于 DE/EDA 及其变形算法, 表 1 中的结果表明: 在 f_6 、 f_8 、 $f_{10} \sim f_{13}$ 中, OCPS-DE/EDA、BCPS-DE/EDA、DE/EDA 得到的最优解相近; 在 $f_1 \sim f_3$ 、 f_7 、 f_9 中, OCPS-DE/EDA 得到的解最优; 在 f_4 、 f_5 中, BCPS-DE/EDA、DE/EDA 得到的解分别最优, 而 OCPS-DE/EDA 最差.

对于 JADE 及其变形算法, 表 1 中的实验结果表明: 在 f_2 、 $f_6 \sim f_{10}$ 、 $f_{12} \sim f_{13}$ 中, OCPS-JADE、BCPS-JADE、JADE 得到的最优解相近; 在 f_5 中, OCPS-JADE 最优; 在 f_1 、 f_3 、 f_{11} 中, OCPS-JADE

的均值 $Fits$ 随 FES 变化的曲线图.

图 7 中结果表明, 在 f_4 、 f_{11} 中, 基于 Sigmoid 核函数的算法收敛最快; 在 f_5 、 f_8 中, 基于多项式核函数的算法收敛最快; 而在其它 8 个测试题中, 基于 RBF 的 OCPS-DE/EDA 的优化性能均明显好于基于其它 3 种核函数的算法. 与此同时, 考虑到实验测试题均为高维问题, 因此在 OCPS 策略中使用径向基核函数更合适.

分别优于 JADE、BCPS-JADE; 在 f_3 、 f_4 中, BCPS-JADE 最优, JADE、OCPS-JADE 分别最差.

对于 EDA/LS 及其变形算法, 表 1 中的实验结果表明: 在 f_3 、 f_5 、 f_6 、 $f_8 \sim f_{13}$ 中, 3 个算法优化性能相似; 在 f_1 、 f_2 、 f_7 中, OCPS-EDA/LS 的优化性能最好; 在 f_4 中, BCPS-EDA/LS 的优化性能最好, OCPS-EDA/LS 的性能最差.

Wilcoxon 排序统计测试表明, 在 3 个算法中, 基于 OCPS 的算法的优化性能不差于, 且在更多情况下好于原算法及相应的基于 BCPS 的算法.

表 2 的结果表明, OCPS-DE/EDA、OCPS-JADE 和 OCPS-EDA/LS 的排序结果始终最优.

图 8 是 OCPS-DE/EDA、BCPS-DE/EDA、DE/EDA 在 13 个测试题中, 在得到相同最优解时, 需要的函数适应值评估次数的中值统计. 图 8 中结果表明, 对于除 f_4 、 f_8 以外的 11 个测试题, 在得到相同最优解时, OCPS-DE/EDA 需要的函数适应值评估

表 1 DE/EDA, JADE, EDA/LS 及其变形算法在 13 个 YLL 测试题上独立运行 30 次, 每次函数适应值评估次数均达到 300000, 得到的最优解适应值的中值、均值和方差统计表

	OCPS-DE/EDA				BCPS-DE/EDA				DE/EDA			
	median		mean(std)		median		mean(std)		median		mean(std)	
f_1	3.61e-95[1]	7.81e-95(1.02e-94)[1]	1.11e-85[2]	5.05e-85(9.50e-85)[2](-)	4.95e-70[3]	8.82e-70(1.33e-69)[3](-)						
f_2	1.32e-45[1]	2.04e-45(1.76e-45)[1]	8.72e-41[2]	1.27e-40(1.10e-40)[2](-)	9.73e-34[3]	1.24e-33(1.04e-33)[3](-)						
f_3	4.73e-22[1]	1.02e-21(1.40e-21)[1]	6.39e-21[2]	2.84e-20(8.36e-20)[2](-)	9.87e-16[3]	1.34e-15(1.28e-15)[3](-)						
f_4	1.29e-02[3]	1.25e-01(3.46e-01)[3]	5.58e-09[1]	2.65e-08(4.87e-08)[1](+)	3.70e-08[2]	1.30e-07(4.32e-07)[2](+)						
f_5	2.34e+01[3]	2.51e+01(9.54e+00)[3]	1.59e-12[2]	5.32e-01(1.38e+00)[2](+)	4.07e-18[1]	2.66e-01(1.01e+00)[1](+)						
f_6	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00+00)[1](~)	0.00e+00[1]	0.00e+00(0.00+00)[1](~)						
f_7	1.32e-03[1]	1.36e-03(3.39e-04)[1]	1.76e-03[2]	1.69e-03(5.18e-04)[2](-)	2.04e-03[3]	2.05e-03(4.61e-04)[3](-)						
f_8	5.92e+02[2]	5.34e+02(2.05e+02)[2]	6.51e+02[3]	6.44e+02(2.30e+02)[3](-)	4.15e+02[1]	4.50e+02(2.19e+02)[1](-)						
f_9	1.49e+01[1]	1.48e+01(4.49e+00)[1]	2.24e+01[2]	3.51e+01(3.49e+01)[2](-)	2.79e+01[3]	4.19e+01(4.72e+01)[3](-)						
f_{10}	4.44e-15[1]	4.91e-15(1.23e-15)[2]	4.44e-15[1]	5.03e-15(1.35e-15)[3](-)	4.44e-15[1]	4.68e-15(9.01e-16)[1](-)						
f_{11}	0.00e+00[1]	7.40e-04(2.26e-03)[2]	0.00e+00[1]	1.64e-03(3.85e-03)[3](-)	0.00e+00[1]	3.29e-04(1.80e-03)[1](-)						
f_{12}	1.57e-32[1]	1.57e-32(5.57e-48)[1]	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)						
f_{13}	1.35e-32[1]	1.35e-32(5.57e-48)[1]	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)						
+/-/~				2/5/6				2/5/6				
	OCPS-JADE				BCPS-JADE				JADE			
f_1	4.96e-239[1]	4.67e-121(2.45e-120)[2]	5.19e-239[2]	6.45e-134(3.53e-133)[1](-)	5.39e-230[3]	1.31e-115(6.15e-115)[3](-)						
f_2	1.19e-66[1]	2.70e-53(1.30e-52)[1]	2.64e-62[3]	7.22e-52(3.10e-51)[2](-)	2.43e-63[2]	3.70e-51(1.43e-50)[3](-)						
f_3	1.51e-42[2]	2.74e-40(1.23e-39)[2]	5.37e-46[1]	1.33e-44(4.05e-44)[1](+)	2.98e-40[3]	1.00e-38(2.35e-38)[3](-)						
f_4	1.46e+00[3]	1.76e+00(8.94e-01)[3]	3.62e-03[1]	8.05e-03(9.42e-03)[1](+)	1.24e-02[2]	2.42e-02(2.56e-02)[2](+)						
f_5	4.07e-06[3]	5.32e-01(1.38e+00)[1]	9.11e-29[1]	7.97e-01(1.62e+00)[2](-)	1.70e-28[2]	9.30e-01(1.71e+00)[3](-)						
f_6	0.00e+00[1]	1.33e-01(4.34e-01)[3]	0.00e+00[1]	0.00e+00(0.00+00)[1](-)	0.00e+00[1]	0.00e+00(0.00+00)[1](-)						
f_7	9.30e-04[1]	1.04e-03(3.75e-04)[1]	1.10e-03[3]	1.18e-03(5.18e-04)[3](-)	1.07e-03[2]	1.14e-03(4.51e-04)[2](-)						
f_8	0.00e+00[1]	5.53e+01(6.77e+01)[1]	5.92e+01[3]	8.69e+01(1.28e+02)[3](-)	0.00e+00[1]	5.53e+01(1.11e+02)[1](-)						
f_9	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)	0.00e+00[1]	0.00e+00(0.00+00)[1](-)						
f_{10}	4.44e-15[1]	5.39e-15(1.60e-15)[2]	4.44e-15[1]	5.98e-15(1.79e-15)[3](-)	4.44e-15[1]	5.27e-15(1.53e-15)[1](-)						
f_{11}	0.00e+00[1]	7.39e-04(2.83e-03)[1]	0.00e+00[1]	4.51e-03(8.72e-03)[3](-)	0.00e+00[1]	2.30e-03(5.56e-03)[2](-)						
f_{12}	1.57e-32[1]	1.57e-32(5.57e-48)[1]	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)						
f_{13}	1.35e-32[1]	1.35e-32(5.57e-48)[1]	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)						
+/-/~				2/2/9				2/3/9				
	OCPS-EDA/LS				BCPS-EDA/LS				EDA/LS			
f_1	9.10e-147[1]	1.16e-146(8.86e-147)[1]	1.48e-132[2]	1.45e-131(6.62e-131)[2](-)	4.44e-130[3]	5.46e-130(5.51e-130)[3](-)						
f_2	3.69e-75[1]	4.07e-75(2.13e-75)[1]	6.24e-65[2]	6.93e-65(3.55e-65)[2](-)	9.84e-65[3]	1.08e-64(8.10e-65)[3](-)						
f_3	1.54e-37[2]	1.18e-36(3.85e-36)[1]	7.41e-38[1]	8.24e-36(2.19e-35)[2](-)	5.28e-37[3]	3.10e-35(8.12e-35)[3](-)						
f_4	5.92e-47[1]	9.13e-17(5.00e-16)[3]	1.41e-42[2]	1.50e-42(6.71e-43)[1](+)	1.99e-42[3]	6.64e-40(3.62e-39)[2](+)						
f_5	1.78e-29[2]	1.33e-01(7.28e-01)[3]	2.77e-29[3]	3.93e-29(4.76e-29)[2](-)	1.74e-29[1]	2.65e-29(3.12e-29)[1](-)						
f_6	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00+00)[1](-)	0.00e+00[1]	0.00e+00(0.00+00)[1](-)						
f_7	1.80e-03[1]	1.80e-03(5.35e-04)[1]	2.39e-03[3]	2.35e-03(6.39e-04)[2](-)	2.26e-03[2]	2.35e-03(6.88e-04)[2](-)						
f_8	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)						
f_9	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)						
f_{10}	4.44e-15[1]	4.44e-15(0.00e+00)[1]	4.44e-15[1]	4.44e-15(0.00e+00)[1](-)	4.44e-15[1]	4.44e-15(0.00e+00)[1](-)						
f_{11}	0.00e+00[1]	0.00e+00(0.00e+00)[1]	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)	0.00e+00[1]	0.00e+00(0.00e+00)[1](-)						
f_{12}	1.57e-32[1]	1.57e-32(5.57e-48)[1]	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)	1.57e-32[1]	1.57e-32(5.57e-48)[1](-)						
f_{13}	1.35e-32[1]	1.35e-32(5.57e-48)[1]	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)	1.35e-32[1]	1.35e-32(5.57e-48)[1](-)						
+/-/~				1/3/9				1/3/9				

表 2 DE/EDA, JADE, EDA/LS 及其变形算法在 13 个测试题上得到的最优解函数适应值的排序结果统计

	median				mean				
	Rank	1	2	3	mean	1	2	3	mean
OCPS-DE/EDA	10	1	2	1.38	8	3	2	1.54	
BCPS-DE/EDA	6	6	1	1.62	4	6	3	1.92	
DE/EDA	7	1	5	1.85	7	1	5	1.85	
OCPS-JADE	10	1	2	1.38	8	3	2	1.54	
BCPS-JADE	9	1	3	1.54	6	3	4	1.85	
JADE	7	4	2	1.62	5	4	4	1.92	
OCPS-EDA/LS	11	2	0	1.15	11	0	2	1.31	
BCPS-EDA/LS	8	3	2	1.54	8	5	0	1.38	
EDA/LS	8	1	4	1.69	8	2	3	1.62	

次数始终最少. 图 9 是 OCPS-JADE、BCPS-JADE、JADE 在 12 个测试题中的运行过程统计图. 图 9 中结果表明, 在 $f_1, f_2, f_7, f_8 \sim f_{13}$ 中, OCPS-JADE 的收敛速度均快于 BCPS-JADE 和 JADE; 在 f_3, f_4, f_5 中, BCPS-JADE 收敛最快; 在 f_3 中, OCPS-JADE 收敛速度次快; 在 f_4, f_5 中, OCPS-JADE 收敛最慢.

以上实验结果表明, 作为一个通用策略, OCPS 能在不同类型的 EA 算法中应用, 并且在大多数测试题中, 基于 OCPS 的算法优化性能不差于且在更多情况下好于原算法和基于 BCPS 的算法. 由此可

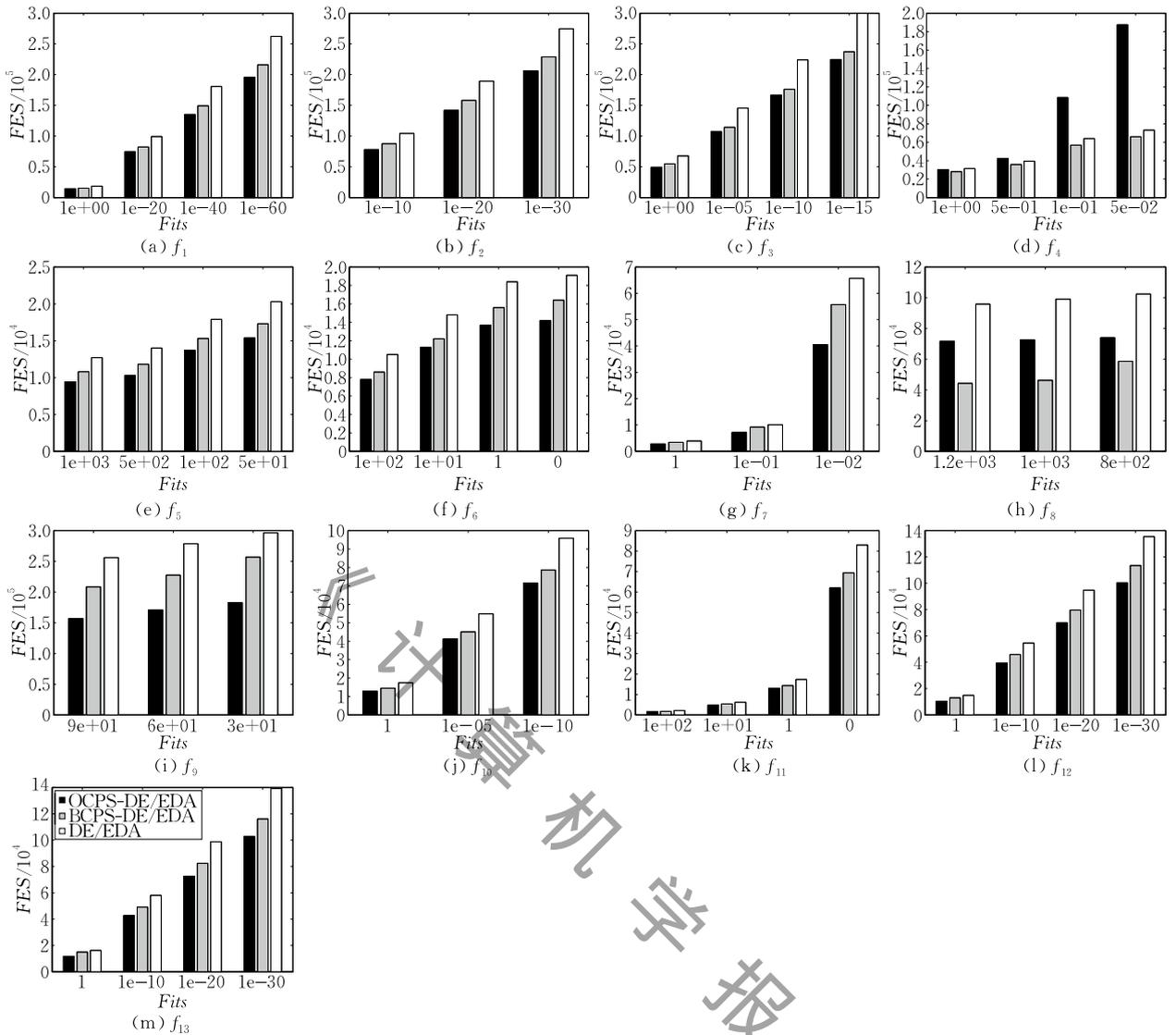


图 8 OCPS-DE/EDA、BCPS-DE/EDA、DE/EDA 在 13 个测试题中,得到相同最优解时,使用的函数适应值评估次数的中值统计图(其中纵坐标表示 FES,横坐标表示最优解适应值 Fits)

以得出结论,OCPS 策略能够提升不同类型的 EA 算法的优化性能。

3.6 不同变量维度中的实验结果

本节对不同变量维度中,OCPS策略对算法优

化性能的影响进行了实验分析。表 3 是 OCPS-DE/EDA 和 DE/EDA 算法分别在变量维度 $n=10、20、30$ 时,在 YLL 测试集上运行得到的最优解的函数适应值中值的统计,实验参数同节 3.1 设置。

表 3 算法在不同变量维度下得到的最优解中值统计

	n	OCPS-DE/EDA	DE/EDA		n	OCPS-DE/EDA	DE/EDA
f_1	10	1.02e-292(+)	2.33e-215	f_8	10	1.18e+02(-)	0.00e+00
	20	9.35e-140(+)	6.75e-103		20	2.37e+02(-)	1.18e+02
	30	3.61e-95(+)	4.95e-70		30	5.92e+02(-)	4.15e+02
f_2	10	1.38e-138(+)	3.39e-102	f_9	10	1.99e+00(+)	2.98e+00
	20	7.84e-66(+)	1.48e-48		20	7.96e+00(+)	1.09e+01
	30	1.32e-45(+)	9.73e-34		30	1.49e+01(+)	2.79e+01
f_3	10	5.28e-174(+)	5.48e-129	f_{10}	10	4.44e-15(~)	4.44e-15
	20	2.79e-55(+)	1.17e-39		20	4.44e-15(~)	4.44e-15
	30	4.73e-22(+)	9.87e-16		30	4.44e-15(~)	4.44e-15
f_4	10	1.12e-115(+)	5.20e-84	f_{11}	10	1.85e-02(+)	2.95e-02
	20	5.23e-09(-)	5.71e-31		20	0.00e+00(~)	0.00e+00
	30	1.29e-02(-)	3.70e-08		30	0.00e+00(~)	0.00e+00
f_5	10	7.00e+00(-)	0.00e+00	f_{12}	10	4.71e-32(~)	4.71e-32
	20	1.65e+01(-)	0.00e+00		20	2.36e-32(~)	2.36e-32
	30	2.34e+01(-)	4.07e-18		30	1.57e-32(~)	1.57e-32
f_6	10	0.00e+00(~)	0.00e+00	f_{13}	10	1.35e-32(~)	1.35e-32
	20	0.00e+00(~)	0.00e+00		20	1.35e-32(~)	1.35e-32
	30	0.00e+00(~)	0.00e+00		30	1.35e-32(~)	1.35e-32
f_7	10	1.26e-04(+)	2.12e-04	+/-/~	10	7/2/4	
	20	6.43e-04(+)	9.51e-04		20	5/3/5	
	30	1.32e-03(+)	2.04e-03		30	5/3/5	

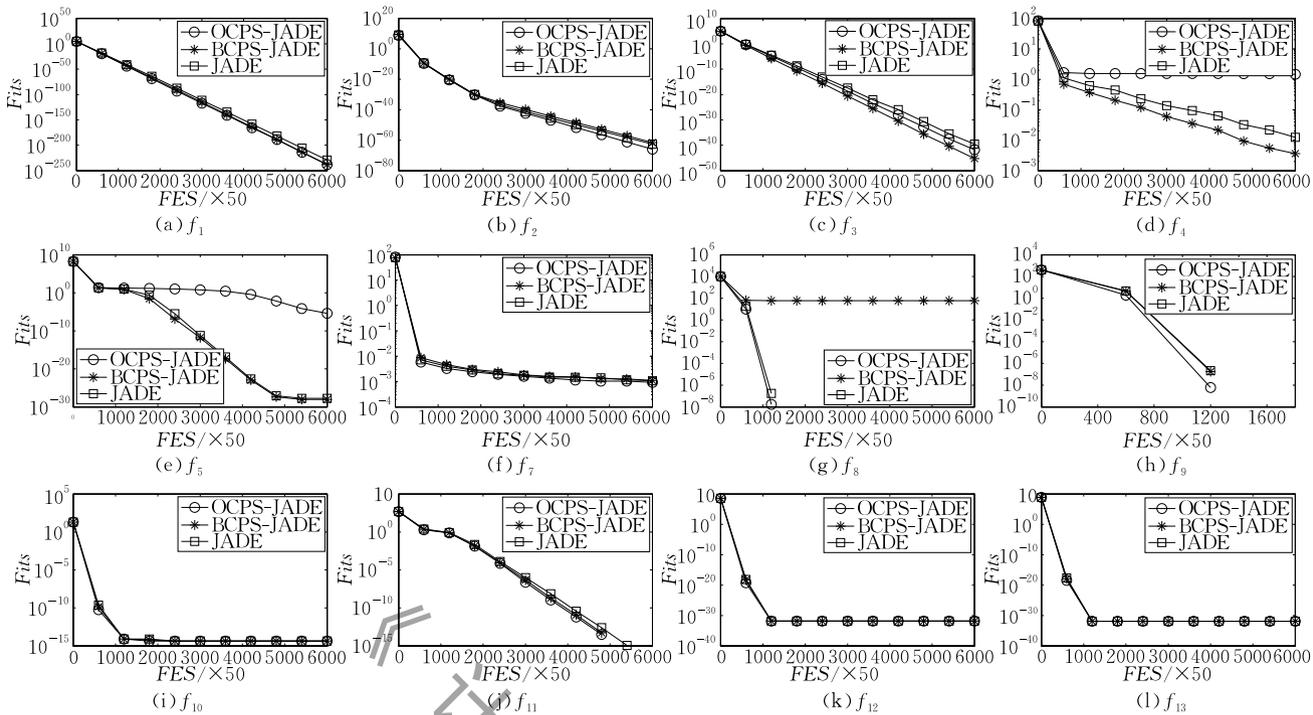


图 9 OCPS-JADE,BCPS-JADE,JADE在 12 个测试题中的运行过程统计图(其中横坐标为适应值 $Fits$,纵坐标为 FES)

实验结果表明在不同变量维度下,OCPS-DE/EDA 均能在大多数题目中比 DE/EDA 拥有更好的优化性能。

3.7 单分类与二分类模型对比分析

在节 3.5 的对比实验基础上,本节对 EA 中,OCPS 策略中单分类模型(OCC)与 BCPS 策略中二分类模型(SVM)的性能进行对比分析.在算法优化过程的每代运算中均进行如下实验:(1)使用当前解种群分别构建 OCC 模型和 SVM 模型;(2)用 OCC 模型和 SVM 模型分别对每个父代解产生的多个候选后代解进行类标记;(3)根据预选择过程中的后代选择规则,使用 OCC 模型和 SVM 模型分别选出一个真正的后代解;(4)分别计算这两个后代解的函数适应值并进行比较,统计使用 OCC 模型选出的解更

优的数目.使用 DE/EDA 作为实验算法,在 13 个测试题中算法执行 3000 代,种群大小为 100.表 4 为实验结果统计。

表 4 中的统计结果表明,在算法执行的每一代中,使用 OCC 模型选出的后代解中有 34.00%~100.00% 优于使用 SVM 模型选出的后代解;反之,使用 SVM 模型选出的后代解中只有 00.00%~66.00% 优于使用 OCC 模型选出的解.中值和均值的统计结果也基本一致,根据均值,使用 OCC 模型选出的解有 51.00%~93.00% 好于使用 SVM 模型选出的解;此外,在 $f_1 \sim f_6$ 、 $f_8 \sim f_{13}$ 等 12 个测试题中,在超过 98.17% 的迭代次数中,使用 OCC 模型选出的后代解有超过 50.00% 好于使用 SVM 模型选出的解。

表 4 每一代使用 OCC 模型选出的后代解优于使用 SVM 模型选出的后代解的数目统计

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
mean	60	59	57	71	55	81	51	79	69	85	77	72	92
median	59	59	57	71	54	81	50	82	70	89	79	73	100
max	79	75	72	87	78	93	70	95	90	99	93	88	100
min	39	42	40	52	37	48	33	43	41	44	43	45	42
win generations	2919	2892	2793	3000	2401	2999	1640	2970	2974	2984	2988	2984	2971

图 10 为在 $f_1 \sim f_4$ 等 4 个测试题目上,在算法运行的每一代,使用 OCC 模型选出的解优于使用 SVM 模型选出的解的个数的盒型统计图,图 10 中

结果更加清楚地表明 OCC 模型能够选出更多的好解.从而进一步验证了 OCPS 策略的优化性能优于 BCPS 策略。

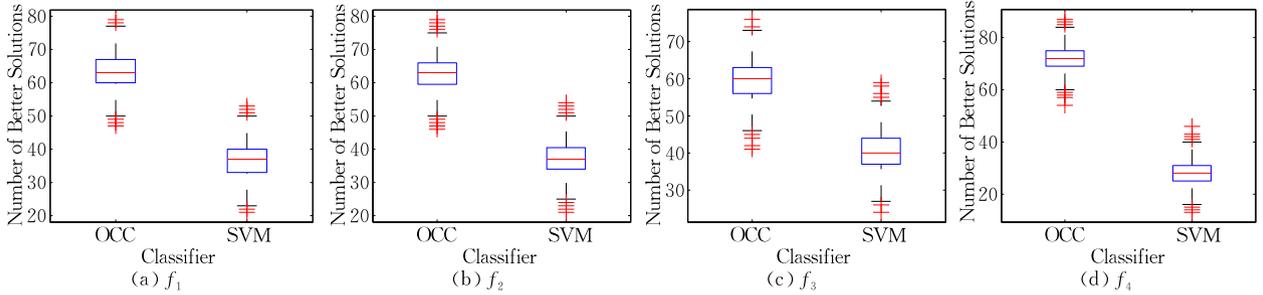


图 10 $f_1 \sim f_4$ 中 OCC 模型选出的后代解优于 SVM 模型选出的后代解的数目统计图(其中纵坐标表示选出的解的数目)

为了进一步分析 OCC 模型的优化性能好于 SVM 模型的原因,图 11 描绘了在 $x_1 \sim x_2$ 维子空间中,在一次算法运行过程中,在算法执行的第 100、1000、2000、3000 代,使用 SVM 模型得到的“好”解和“差”解的空间分布情况.图 12 描绘了在不同的子空间,在一次运行过程中,在 1500 代,使用 SVM 得

到的“好”解和“差”解的分布情况.图 11 和图 12 表明,在解空间中,“好”解和“差”解之间相互叠加、区分不明显.因此,对 SVM 模型而言,构建一个高质量的模型来评估新产生的候选后代解的质量是不容易的,这也进一步说明了 OCPS 策略比 BCPS 策略优化性能更好的原因.

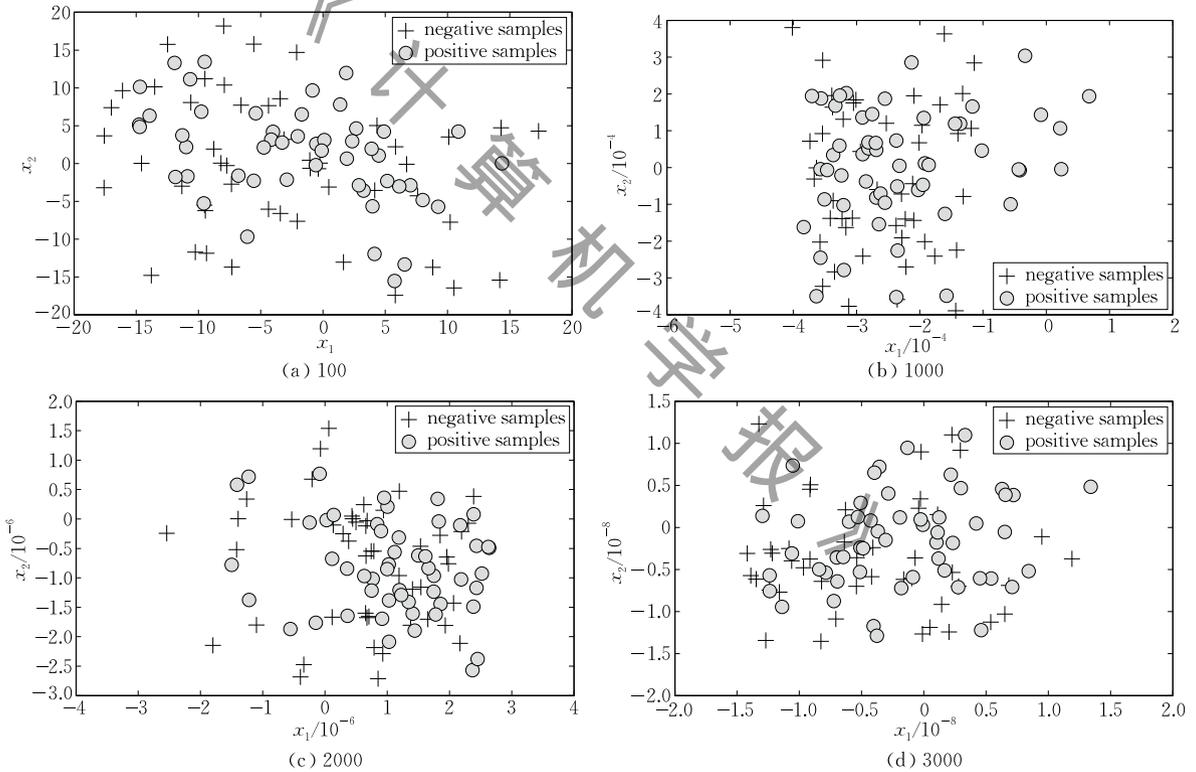


图 11 在不同代中,构建 SVM 模型时“好”解和“差”解的分布

3.8 单分类与代理模型对比分析

本节对基于单分类的预选择策略与基于代理模型的预选择策略进行了实验对比分析.

基于代理模型的预选择策略具体流程为:(1)使用当前种群中的解作为模型训练样本,根据真实目标函数计算训练样本的函数适应值;(2)使用训练样本及其真实函数适应值构建代理模型;(3)使用构建的代理模型,得到每个父代解产生的多个候选后代解的近似函数适应值;(4)选取近似适应值最

小的解作为真正的后代解.

本节在 DE/EDA 中分别使用二叉回归树(Binary Tree, Btree)^[37]、高斯过程(Gaussian Process, GP)^[38]、径向基函数(Radial Basis Function, RBF)^[39]、支持向量机(Support Vector Machines, SVR)^[40]等代理模型进行预选择过程,每个算法在 13 个测试题中独立运行 30 次,每次适应值评估次数 $FES=300\,000$,将优化结果与 OCPS-DE/EDA 的实验结果进行对比分析.表 5 是 OCPS 与基于代理模型 Btree、

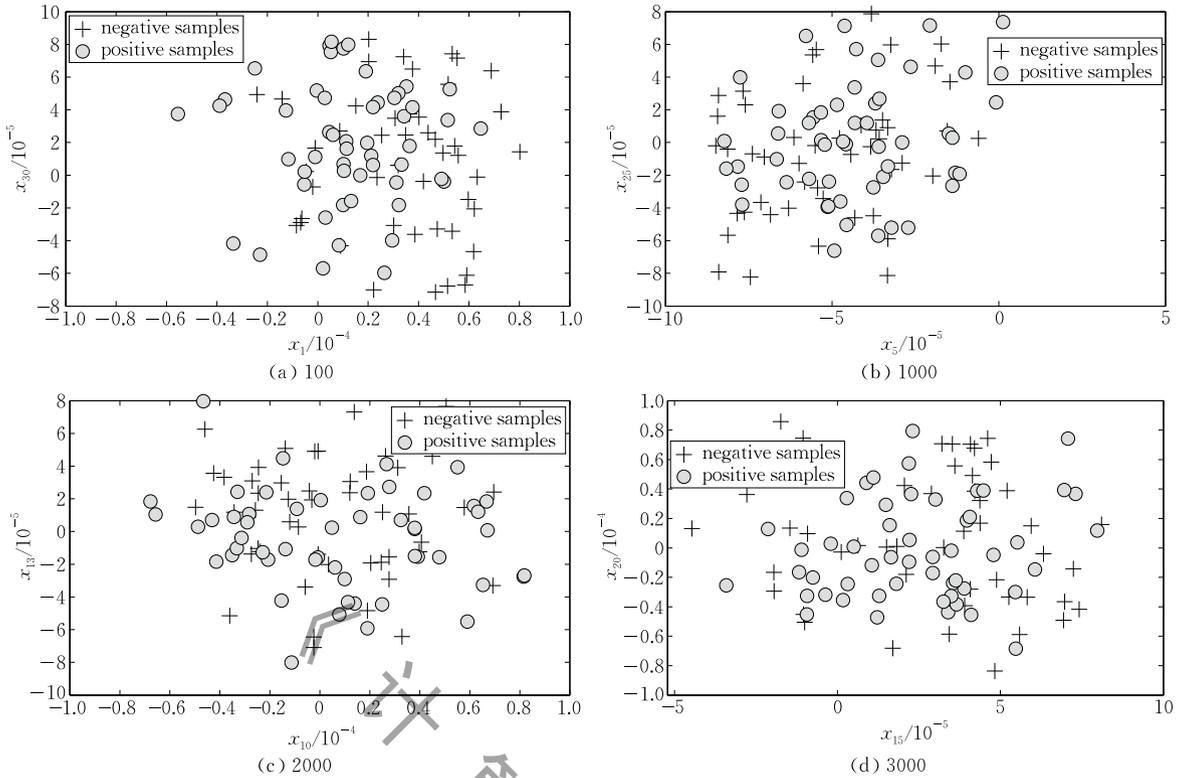


图 12 在不同子空间中, SVM 模型构建时“好”解和“差”解分布

表 5 OCPS 与基于代理模型 Btree、GP、RBF、SVR 的预选择策略在 13 个测试题中的实验结果统计

	OCPS	Btree	GP	RBF	SVR
	mean _{std} [rank]				
f_1	7.81e-95 _{1.02e-94} [2]	9.03e-96 _{1.81e-95} [1]	5.17e-75 _{1.15e-74} [4]	6.34e-76 _{1.09e-75} [3]	5.07e-73 _{1.10e-72} [5]
f_2	2.04e-45 _{1.76e-45} [1]	3.27e-45 _{4.16e-45} [2]	7.18e-36 _{1.01e-35} [5]	2.37e-36 _{1.67e-36} [4]	1.18e-36 _{1.15e-36} [3]
f_3	1.02e-21 _{1.40e-21} [2]	3.62e-24 _{9.70e-24} [1]	2.16e-21 _{6.02e-21} [3]	4.32e-21 _{8.56e-21} [4]	2.16e-19 _{2.98e-19} [5]
f_4	1.25e-01 _{3.46e-01} [5]	1.71e-12 _{3.04e-12} [1]	8.75e-07 _{2.99e-06} [4]	1.47e-10 _{3.88e-10} [3]	6.57e-11 _{9.42e-11} [2]
f_5	2.51e+01 _{9.54e+00} [5]	3.36e-01 _{1.05e+00} [1]	2.05e+01 _{8.63e+00} [4]	1.60e+01 _{1.53e+00} [3]	1.12e+00 _{1.81e+00} [2]
f_6	0.00e+00 _{0.00e+00} [1]				
f_7	1.36e-03 _{3.39e-04} [3]	1.67e-03 _{6.16e-04} [5]	9.45e-04 _{3.00e-04} [1]	1.07e-03 _{3.64e-04} [2]	1.64e-03 _{6.56e-04} [4]
f_8	5.34e+02 _{2.05e+02} [1]	6.96e+02 _{2.53e+02} [3]	1.59e+03 _{4.81e+02} [5]	6.11e+02 _{2.10e+02} [2]	6.97e+02 _{2.32e+02} [4]
f_9	1.48e+01 _{4.49e+00} [1]	2.20e+01 _{5.57e+00} [2]	2.26e+01 _{6.79e+00} [3]	4.80e+01 _{4.80e+01} [5]	3.28e+01 _{2.48e+01} [4]
f_{10}	4.91e-15 _{1.23e-15} [3]	5.63e-15 _{1.70e-15} [5]	4.68e-15 _{9.01e-16} [1]	4.80e-15 _{1.08e-15} [2]	4.91e-15 _{1.23e-15} [3]
f_{11}	7.40e-04 _{2.26e-03} [2]	4.60e-03 _{7.16e-03} [5]	3.29e-04 _{1.80e-03} [1]	9.04e-04 _{2.78e-03} [3]	2.96e-03 _{4.90e-03} [4]
f_{12}	1.57e-32 _{5.57e-48} [1]	1.57e-32 _{5.57e-48} [1]	1.57e-32 _{5.57e-48} [1]	1.57e-32 _{5.57e-48} [1]	6.91e-03 _{2.63e-02} [5]
f_{13}	1.35e-32 _{5.57e-48} [1]	1.35e-32 _{5.57e-48} [1]	1.35e-32 _{5.57e-48} [1]	3.66e-04 _{2.01e-03} [4]	3.66e-04 _{2.01e-03} [4]
mean rank	2.15	2.23	2.62	2.85	3.54

GP、RBF、SVR 的预选择策略在 13 个测试题中的实验结果统计。

表 5 中的结果表明,从获得的解最优的数目而言,基于 OCPS 和 4 种代理模型的 DE/EDA 分别获得了 6、7、6、2、1 个最优解,从而表明基于 OCPS、Btree、GP 的算法优化性能好于其它两种方法,最优适应值排序结果的均值表明 OCPS-DE/EDA 算法的优化性能最好,且明显优于其余 4 种算法,Btree-DE/EDA 的优化性能为次优。

图 13 是基于 OCPS 和 4 种代理模型的 DE/EDA 在 12 个测试题中最优解适应值的均值随 FES

变化的曲线图。

图 13 中的结果表明,在 f_1 、 f_2 、 f_{10} 、 f_{12} 、 f_{13} 等 5 个测试题中,OCPS-DE/EDA 与 Btree-DE/EDA 在优化过程中的收敛速度相似且明显好于其它 3 种代理模型;在 f_8 、 f_9 中,OCPS-DE/EDA 得到的最优解优于其它 4 种算法;在 f_7 、 f_{11} 中,OCPS-DE/EDA 次优;在 f_4 中,OCPS-DE/EDA 最差;Btree-DE/EDA 在 f_3 、 f_4 、 f_5 中最优,在 f_7 、 f_{11} 中最差;GP-DE/EDA 在 f_{11} 最优, f_2 、 f_8 最差;RBF-DE/EDA 在 f_9 、 f_{13} 中最差;SVR-DE/EDA 在 f_1 、 f_3 、 f_7 、 f_{12} 、 f_{13} 中最差。总体而言,OCPS-DE/EDA 和 Btree-DE/EDA

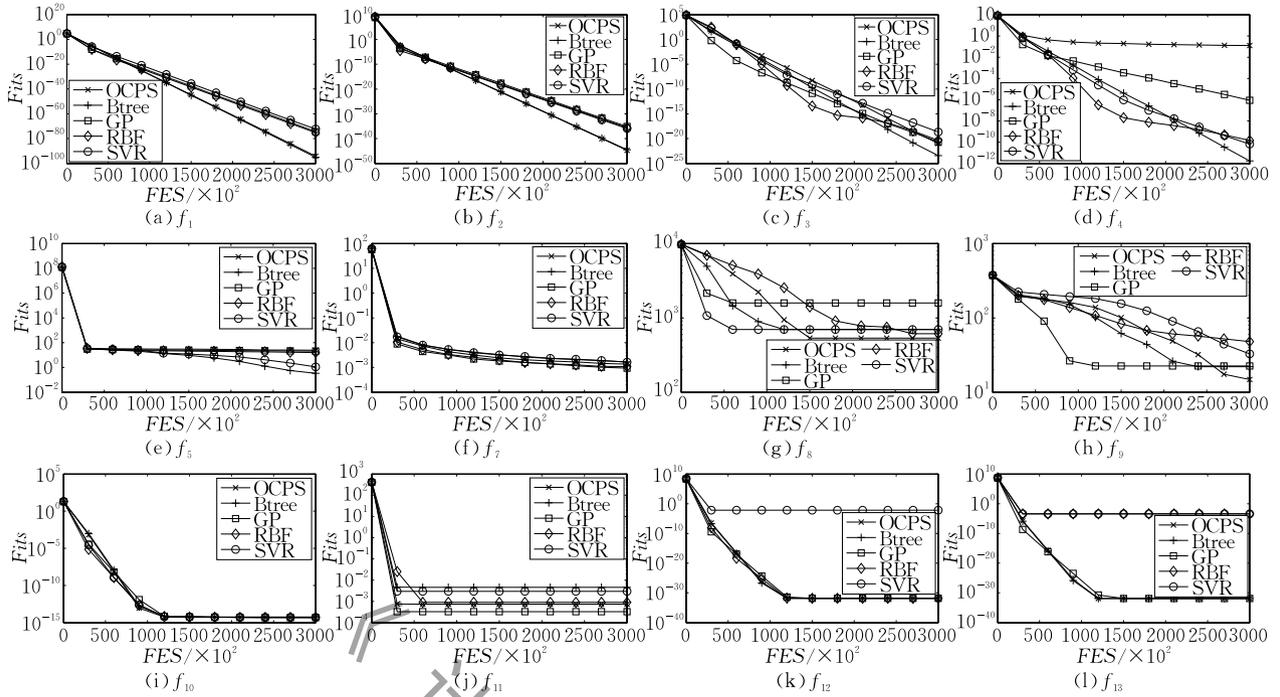


图 13 基于 OCPS 和 4 种代理模型的 DE/EDA 在 12 个测试题中的运行过程统计图(其中横坐标为适应值 $Fits$, 纵坐标为 FES)

的优化性能好于其它 3 个算法, 综合考虑结果排序, OCPS-DE/EDA 优化性能比 Btree-DE/EDA 更优。

为了进一步分析 OCC 模型与代理模型在预选策略中的性能, 根据上述实验结果, 选择代理模型 Btree 与 OCC 模型进行对比分析。在算法优化过程的每代运算中均进行如下实验: (1) 使用当前解种群分别构建 OCC 模型和 Btree 模型; (2) 用 OCC 模型对每个父代解产生的多个候选后代解进行类标

记, 使用 Btree 模型评估多个候选后代解的近似函数适应值; (3) 根据预选过程中的后代选择规则, 使用 OCC 模型和 Btree 模型分别选出一个真正的后代解; (4) 分别计算这两个后代解的真实函数适应值并进行比较, 统计使用 OCC 模型选出的解更优的数目。使用 DE/EDA 作为实验算法, 在 13 个测试题中算法执行 3000 代, 种群大小为 100。表 6 为实验结果统计。

表 6 每一代使用 OCC 模型选出的后代解优于使用 Btree 模型选出的后代解的数目统计

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}
mean	64	63	60	72	62	80	51	77	72	86	78	73	93
median	63	63	60	72	61	80	50	80	75	89	79	74	100
max	79	79	76	87	78	92	70	94	96	99	92	90	100
min	47	46	41	54	44	51	34	42	43	50	48	48	48
win generations	2992	2982	2945	3000	2969	3000	1746	2970	2981	3000	2995	2998	2998

表 6 中的结果表明, 在算法执行的每一代中, 使用 OCC 模型选出的后代解有 33.00%~100.00% 优于使用 Btree 模型选出的后代解; 反之, 有 0.00%~67.00% 通过 Btree 模型选出的解优于使用 OCC 模型选出的解。均值统计表明, 使用 OCC 模型选出的解有 51.00%~92.00% 优于使用 Btree 模型选出的解。在除了 f_5, f_7 外的所有测试题中, 在超过 93.10% 的迭代次数中, 使用 OCC 模型选出的后代解有超过 50.00% 好于使用 Btree 模型选出的解。由此可以说

明在预选过程中, 使用 OCC 模型能够比使用 Btree 模型选出更多“好”解。

图 14 为在 f_1, f_5, f_7, f_{11} 等 4 个测试题目上, 在算法运行每一代, 使用 OCC 模型选出的解优于 Btree 模型选出的解的数目的盒型统计图, 图 14 中结果更加表明 OCC 模型能够选出更多的好解。

由此, 进一步证实了在预选过程中, 使用 OCC 模型比使用代理模型, 更能提升算法的优化性能。

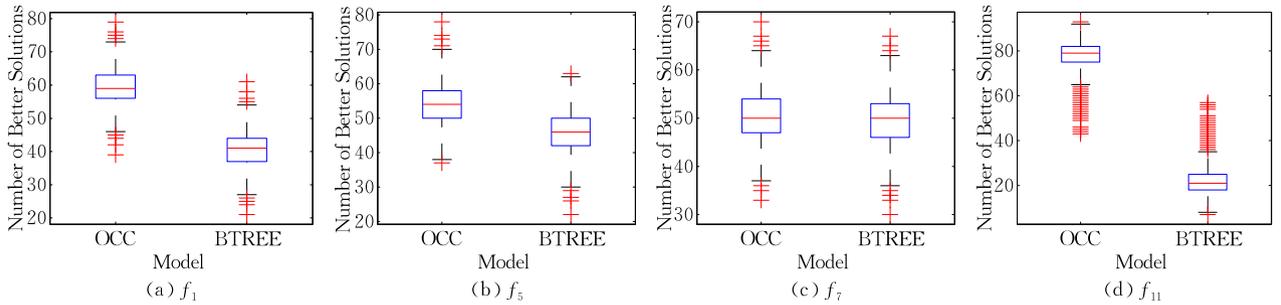


图 14 f_1 、 f_5 、 f_7 、 f_{11} 中,使用 OCC 模型选出的后代解优于使用 Btree 模型选出的后代解的数目统计图(其中纵坐标表示选出的解的数目)

3.9 算法运行时间对比分析

本节对基于单分类的预选择策略、基于代理模型的预选择策略以及原算法的运行时间进行对比分析.在前几节研究内容的基础上,本节实验分别将 OCPS-DE/EDA、Btree-DE/EDA、DE/EDA 在 13 个测试题中独立运行 30 次,每次适应值评估次数均达到 300 000 次,将得到的运行时间的统计结果进行对比.算法在操作系统为 Windows 10、硬件条件为 i7-3470 QM CPU @ 2.70 GHz, 8.00 GB RAM 的 Lenovo Thinkpad W530 中的 MATLAB R2018a 中运行.

表 7 中的运行时间统计结果表明,在 3 个算法分别达到 300 000 次适应值评估次数时,OCPS-DE/EDA 和 Btree-DE/EDA 的运行时间均高于 DE/EDA,而与 Btree-DE/EDA 相比,OCPS-DE/EDA 的运行时间更少.由此可以得出结论,在本文实验中未使用模型的原算法需要的运行时间最少,基于单分类的预选择策略的运行时间居中,而基于代理模型的预选择策略的运行时间最多.需要注意,算法的运行时间与算法的具体实现有关,此处给出的运行时间统计仅供参考.

表 7 OCPS-DE/EDA、Btree-DE/EDA、DE/EDA 在 13 个测试题中平均运行时间统计 /s

	OCPS-DE/EDA	Btree-DE/EDA	DE/EDA
f_1	13.11	26.73	2.04
f_2	12.47	26.79	2.16
f_3	10.89	26.73	2.14
f_4	7.92	26.70	1.66
f_5	12.28	26.37	2.18
f_6	13.08	16.00	2.30
f_7	14.12	26.18	3.62
f_8	11.74	21.96	2.42
f_9	15.24	26.23	2.26
f_{10}	11.34	21.18	2.06
f_{11}	12.69	16.91	2.31
f_{12}	16.54	21.61	4.39
f_{13}	14.14	22.73	3.36

均是在演化算法中引入了机器学习模型,而模型的训练、预测均需要时间,因此毫无疑问模型的引入会增加算法的运行时间.而在众多实际应用问题中,函数适应值的评估需要很多的时间,与之相比,模型需要的时间要少很多,因此在这种情况下,引入模型来降低函数的适应值评估次数,能够有效降低算法运行需要的时间.

4 总 结

本文在文献[22]和文献[28]中提出的预选择策略的基础上提出了演化算法中一种通用的基于单分类的预选择策略(One-class Classification based Preselection, OCPS),OCPS 使用单分类模型来辅助 EA 中的预选择过程,在不增加额外的函数适应值评估次数的情况下,使得 EA 算法在优化过程的每一代中能够得到更多的潜在“好”解.同时,因模型训练样本只需要一类,所以简化了算法设计流程、降低了算法复杂度并提升了分类模型在 EA 中的准确率,进而提升了演化算法的优化性能.OCPS 策略主要包括以下 3 个步骤:(1)训练样本定义,将当前种群中的解均标记为“好”解,作为训练样本;(2)模型构建,使用训练样本构建 OCC 模型;(3)后代解标记与选择,使用构建的 OCC 模型对每个父代解产生的多个候选后代解的类别进行标记,选出一个标记为“好”的候选后代解,作为父代真正的后代解进行后续的函数适应值评估和环境选择过程.

为了验证提出的 OCPS 策略的性能,将 OCPS 应用在 DE/EDA^[31]、JADE^[32]、EDA/LS^[33] 等 3 个经典 EA 算法和 13 个 YLL^[30] 测试题中进行实验验证,为了保证模型的一致性、减少参数对算法性能的影响,选择文献[22]中基于 SVM 的 BCPS 策略用作对比实验,实验结果表明,在大多数情况下,基于 OCPS 的算法的优化性能不差于且在更多情况下好

鉴于基于单分类和基于代理模型的预选择策略

于基于 BCPS 的算法和原算法. 同时, 实验也对 OCC 模型的优化性能好于 SVM 模型的原因进行了实验分析, 进一步验证了 OCC 模型的优势. 最后, 将 OCPS 策略与基于代理模型的预选择策略进行了实验对比, 实验结果也证明了 OCPS 策略具有更好的优化性能. 此外, 文章也分析了 OCC 模型优化性能好于代理模型的原因, 进一步证明了 OCPS 策略的优势.

未来的研究工作中我们将考虑单分类预选择策略在多目标演化算法^[41-43]、约束优化算法^[8,44]以及实际应用问题^[45]中的应用.

参 考 文 献

- [1] Polak E. Optimization: Algorithms and Consistent Approximations. New York, USA: Springer-Verlag, 1997
- [2] Back T, Schwefel H-P. Evolutionary computation: An overview //Proceedings of the IEEE International Congress on Evolutionary Computation. Nagoya, Japan, 1996: 20-29
- [3] Back T. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. New York, USA: Oxford University Press, 1996
- [4] Back T, Hammel U, Schwefel H-P. Evolutionary computation: Comments on the history and current state. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 3-17
- [5] Simon D. Evolutionary Optimization Algorithms. Hoboken, USA: John Wiley & Sons, 2013
- [6] Yang Hai-Jun, Li Min-Qiang. Schemata theorem and building blocks in evolutionary algorithms. Chinese Journal of Computers, 2003, 26(11): 1550-1554(in Chinese)
(杨海军, 李敏强. 进化算法中的模式定理及建筑块. 计算机学报, 2003, 26(11): 1550-1554)
- [7] Huang Han, Lin Zhi-Yong, Hao Zhi-Feng, et al. Convergence analysis and comparison of evolutionary algorithms based on relation model. Chinese Journal of Computers, 2011, 34(5): 801-811(in Chinese)
(黄翰, 林智勇, 郝志峰等. 基于关系模型的进化算法收敛性分析与对比. 计算机学报, 2011, 34(5): 801-811)
- [8] Li Zhi-Yong, Huang Tao, Chen Shao-Miao, Li Ren-Fa. Overview of constrained optimization evolutionary algorithms. Journal of Software, 2017, 28(6): 1529-1546(in Chinese)
(李智勇, 黄滔, 陈少淼, 李仁发. 约束优化进化算法综述. 软件学报, 2017, 28(6): 1529-1546)
- [9] Wang Xi-Zhao, He Yi-Chao. Evolutionary algorithms for knapsack problems. Journal of Software, 2017, 28(1): 1-16 (in Chinese)
(王熙照, 贺毅朝. 求解背包问题的演化算法. 软件学报, 2017, 28(1): 1-16)
- [10] Pan Xiao-Ying, Jiao Li-Cheng. A multi-agent social evolutionary algorithm for project optimization scheduling. Journal of Computer Research and Development, 2008, 45(6): 998-1003(in Chinese)
(潘晓英, 焦李成. 项目优化调度的多智能体社会进化算法. 计算机研究与发展, 2008, 45(6): 998-1003)
- [11] Cavicchio D J. Adaptive Search Using Simulated Evolution [Ph. D. dissertation]. University of Michigan, Ann Arbor, USA, 1970
- [12] Mahfoud S W. Crowding and preselection revisited// Proceedings of the 2nd Conference on Parallel Problem Solving from Nature. Brussels, Belgium, 1992: 27-36
- [13] Oliveto P S, Paixao T, Heredia J P, et al. How to escape local optima in black box optimisation: When non-elitism outperforms elitism. Algorithmica, 2018, 80(5): 1604-1633
- [14] Qian Chao, Yu Yang, Zhou Zhi-Hua. Analyzing evolutionary optimization in noisy environments. Evolutionary Computation, 2018, 26(1): 1-41
- [15] Wang Yong, Cai Zi-Xing, Zhang Qing-Fu. Differential evolution with composite trial vector generation strategies and control parameters. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55-66
- [16] Li Yang, Zhou Ai-Min, Zhang Gui-Xu. An MOEA/D with multiple differential evolution mutation operators//Proceedings of the IEEE Congress on Evolutionary Computation. Beijing, China, 2014: 397-404
- [17] Jin Yao-Chu. A comprehensive survey of fitness approximation in evolutionary computation. Soft Computing, 2003, 9(1): 3-12
- [18] Jin Yao-Chu. Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation, 2011, 1(2): 61-70
- [19] Tabatabaei M, Hakanen J, Hartikainen M, et al. A survey on handling computationally expensive multiobjective optimization problems using surrogates: Non-nature inspired methods. Structural & Multidisciplinary Optimization, 2015, 52(1): 1-25
- [20] Zhou Z, Ong Y S, Nair P B, Keane A J. Combining global and local surrogate models to accelerate evolutionary optimization. IEEE Transactions on Systems Man & Cybernetics Part C, 2007, 37(1): 66-76
- [21] Kattan A, Ong Y S. Surrogate genetic programming: A semantic aware evolutionary search. Information Sciences, 2015, 296: 345-359
- [22] Zhang Jin-Yuan, Zhou Ai-Min, Zhang Gui-Xu. Preselection via classification: A case study on global optimization. International Journal of Bio-Inspired Computation, 2018, 11(4): 267-281
- [23] Lu Xiao-Fen, Tang Ke, Yao Xin. Classification-assisted differential evolution for computationally expensive problems// Proceedings of the IEEE Congress on Evolutionary Computation. New Orleans, USA, 2011: 1986-1993
- [24] Lu Xiao-Fen, Tang Ke. Classification- and regression- assisted differential evolution for computationally expensive problems. Journal of Computer Science and Technology, 2012, 27(5): 1024-1034
- [25] Lu Xiao-Fen, Tang Ke, Sendhoff Bernhard, Yao Xin. A new self-adaptation scheme for differential evolution. Neurocomputing, 2014, 146: 2-16

- [26] Minter T. C. Single-class classification//Proceedings of the Symposium on Machine Processing of Remotely Sensed Data. West Lafayette, USA, 1975; 2A12-2A15
- [27] Schoelkopf B, Williamson R C, Smola A J, et al. Support vector method for novelty detection//Proceedings of the Advances in Neural Information Processing Systems. Denver, USA, 2000; 582-588
- [28] Zhang Jin-Yuan, Zhou Ai-Min, Zhang Gui-Xu. Preselection via one-class classification for evolutionary optimization//Proceedings of the Genetic and Evolutionary Computation Conference Companion. Kyoto, Japan, 2018; 259
- [29] Scholkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the support of a high dimensional distribution. *Neural Computation*, 2001, 13(7): 1443-1471
- [30] Yao Xin, Liu Yong, Lin Guang-Ming. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82-102
- [31] Sun Jian-Yong, Zhang Qing-Fu, Tsang Edward P K. DE/EDA: A new evolutionary algorithm for global optimization. *Information Sciences*, 2005, 169(3/4): 249-262
- [32] Zhang Jing-Qiao, Sanderson A C. JADE. Adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945-958
- [33] Zhou Ai-Min, Sun Jian-Yong, Zhang Qing-Fu. An estimation of distribution algorithm with cheap and expensive local search methods. *IEEE Transactions on Evolutionary Computation*, 2015, 19(6): 807-822
- [34] Fan R-E, Chen P-H, Lin C-J. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 2005, 6(4): 1889-1918
- [35] Chang C-C, Lin C-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011, 2(3): 389-396
- [36] Wilcoxon F, Katti S K, Wilcox R A. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics*, 1970, 1: 171-259
- [37] Breiman L, Friedman J, Olshen R, Stone C J. Classification and regression trees. *Biometrics*, 1984, 40(3): 17-23
- [38] MacKay D J. Introduction to Gaussian processes. NATO ASI Series F Computer and Systems Sciences, 1998, 168: 133-166
- [39] Sun Chao-Li, Ding Jin-Liang, Zeng Jian-Chao, Jin Yao-Chu. A fitness approximation assisted competitive swarm optimizer for large scale expensive optimization problems. *Memetic Computing*, 2018, 10(2): 123-134
- [40] Burges C J. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998, 2(2): 121-167
- [41] Zheng Jin-Hua, Li Ke, Li Mi-Qing, Wen Shi-Hua. Adaptive neighbor multi-objective evolutionary algorithm based on Hypervolume indicator. *Journal of Computer Research and Development*, 2012, 49(2): 312-326(in Chinese)
(郑金华, 李珂, 李密青, 文诗华. 一种基于 Hypervolume 指标的自适应域多目标进化算法. *计算机研究与发展*, 2012, 49(2): 312-326)
- [42] Zheng Jin-Hua, Shen Rui-Min, Li Mi-Qing, et al. An evolutionary environmental model for multiobjective optimization and its realization. *Chinese Journal of Computers*, 2014, 37(12): 2530-2547(in Chinese)
(郑金华, 申瑞珉, 李密青等. 多目标优化的进化环境模型及实现. *计算机学报*, 2014, 37(12): 2530-2547)
- [43] Zhou Ai-Min, Zhang Qing-Fu, Zhang Gui-Xu. Multiobjective evolutionary algorithm based on mixture Gaussian models. *Journal of Software*, 2014(5): 913-928(in Chinese)
(周爱民, 张青富, 张桂戌. 一种基于混合高斯模型的多目标进化算. *软件学报*, 2014(5): 913-928)
- [44] Cai Zi-Xing, Jiang Zhong-Yang, Wang Yong, Luo Yi-Dan. A novel constrained optimization evolutionary algorithm based on orthogonal experimental design. *Chinese Journal of Computers*, 2010, 33(5): 855-864(in Chinese)
(蔡自兴, 江中央, 王勇, 罗一丹. 一种新的基于正交实验设计的约束优化进化算法. *计算机学报*, 2010, 33(5): 855-864)
- [45] Pan Xiao-Ying, Jiao Li-Cheng, Liu Fang. A multi-agent social evolutionary algorithm for SAT problem. *Chinese Journal of Computers*, 2014, 37(9): 2011-2020(in Chinese)
(潘晓英, 焦李成, 刘芳. 求解 SAT 问题的多智能体社会进化算法. *计算机学报*, 2014, 37(9): 2011-2020)



ZHANG Jin-Yuan, Ph. D. Her main research interests include evolutionary computation and optimization.

ZHOU Ai-Min, Ph. D., professor, Ph. D. supervisor. His main research interests include evolutionary computation and optimization, machine learning and image processing.

ZHANG Gui-Xu, Ph. D., professor, Ph. D. supervisor. His main research interests include image processing and machine learning.

Background

In EAs, the preselection has different meanings. In most situations, it refers to selecting promising ones from a set of candidate offspring solutions. Preselection can help to improve the algorithm performance significantly if correctly utilized, in which case the unpromising candidate solutions can be discarded before the real function evaluations and thus the computational resources can be saved. A key issue with the preselection is on how to measure the qualities of candidate solutions, i. e. , to judge whether a candidate solution is promising or not. According to this issue, strategies for preselection can be roughly classified into three categories: fitness based approaches, surrogate model based approaches and classification based approaches.

For the classification based approach, the preselection is always regarded as a classification procedure, where the chosen offspring solutions belong to the ‘good’ class, and the discarded ones are with the ‘bad’ class. In this approach, the visited solutions are used to build a classification model and to predict the qualities, i. e. , ‘good’ or ‘bad’, of the candidate offspring solutions and the promising ones are then chosen out. In this category, most work focus on the binary-class classification based approach.

In some EAs, it is believed that after generations, all left solutions in the current population are good ones. Therefore,

it is not trivial to maintain proper training data sets. Actually, there may not exist a clear gap between ‘good’ and ‘bad’ solutions. In the community of classification, there exists a type of so-called one-class classification (OCC), which usually needs only one class of training data points. Thus applying the OCC model to assist the preselection procedure should be more suitable than the binary-class models. Following this idea, this paper proposes a one-class classification based preselection (OCPS) strategy for evolutionary global optimization. For an OCPS based EAs, an OCC model is built firstly according to the current population in each generation. Then, for each parent solution, a set of candidate offspring solutions are generated by the variation operator. Finally, a solution labeled as ‘good’ according to the OCC model is chosen as the offspring solution. The experiments show that the OCPS performs better than the other models and reduces the number of function evaluations significantly.

The work is supported by the National Natural Science Foundation of China (61731009, 61673180, 61703382). The project is intended to study the machine learning model assisted evolutionary optimization and its application. Up to now, the project team has proposed some model based evolutionary algorithms and has applied these algorithms to some real-world problems.