

# 基于多样性分类和距离回归的进化算法\*

孙哲人, 黄玉划, 陈志远



(南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106)

通信作者: 孙哲人, E-mail: szheren2k@163.com

**摘要:** 提出了一个基于多样性分类和距离回归的进化算法(DCDREA), 以解决昂贵超多目标优化问题(MaOPs). DCDREA 采用随机森林(RF)作为全局分类代理模型, 它把种群中所有解作为训练样本, 并根据是否为最小相关解, 把训练样本分类为正负样本, 使模型学习到训练样本中含有的分类标准. 全局分类代理模型主要用来筛选新产生的候选解, 以得到一组有希望的候选解. 此外, 它采用 Kriging 作为局部回归代理模型, 其选择种群中距离当前候选解最近的解作为训练样本, 拟合训练样本与理想点的距离. 然后, 通过  $K$ -means 方法把候选解划分为  $\mu$  个簇, 并从每个簇中选择一个用于真实评估的候选解. 在实验部分, 使用大规模 3、4、6、8、10 目标的 DTLZ 测试问题集, 把 DCDREA 与目前流行的代理辅助进化算法进行对比实验. 对于不同测试问题, 每个算法独立运行 20 次, 然后统计反向迭代距离(IGD)和算法运行时间. 最后, 使用秩和检验来分析结果. 实验对比结果表明, DCDREA 在大多数情况下表现更好. 由此可见, DCDREA 具有较好的有效性和可行性.

**关键词:** 代理模型; 进化算法; 超多目标优化; 昂贵问题; 随机森林; Kriging

**中图法分类号:** TP301

中文引用格式: 孙哲人, 黄玉划, 陈志远. 基于多样性分类和距离回归的进化算法. 软件学报, 2022, 33(10): 3700–3716. <http://www.jos.org.cn/1000-9825/6310.htm>

英文引用格式: Sun ZR, Huang YH, Chen ZY. Diversity Classification and Distance Regression Assisted Evolutionary Algorithm. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3700–3716 (in Chinese). <http://www.jos.org.cn/1000-9825/6310.htm>

## Diversity Classification and Distance Regression Assisted Evolutionary Algorithm

SUN Zhe-Ren, HUANG Yu-Hua, CHEN Zhi-Yuan

(College of Computer Science and Technology (Nanjing University of Aeronautics and Astronautics), Nanjing 211106, China)

**Abstract:** This study proposed a diversity classification and distance regression assisted evolutionary algorithm (DCDREA) to solve expensive many-objective optimization problems (MaOPs). In DCDREA, the random forest (RF) is adopted as the global classification surrogate model. All the solutions in the population are as the training samples and classified into positive or negative samples according to whether they are minimum correlative solutions, so that the model can learn the classification criteria contained in the training samples. The global classification surrogate model is mainly used to filter the newly generated candidates to obtain a group of promising candidates. In addition, Kriging is adopted as the local regression surrogate model, where the solutions closest to the current candidates in the population are selected as the training samples, and the distance between the training samples and the ideal point is approximated by the model. Then, by the  $K$ -means method, the candidates are divided into  $\mu$  clusters, and from each cluster, one candidate is selected for real function evaluation. In the experimental part, the DTLZ suite with large scale 3, 4, 6, 8, and 10 objectives was used to compare DCDREA with the current popular surrogate-assisted evolutionary algorithms. For different test problems, each algorithm was run independently for 20 times. Then the inverted generational distance (IGD) and algorithm running time were counted. At last, the Wilcoxon rank sum test was used to analyze the results. The result comparison shows that DCDREA performs better in most cases, indicating that DCDREA has sound effectiveness and feasibility.

**Key words:** surrogate; evolutionary algorithm; many-objective optimization; expensive problem; random forest; Kriging

\* 基金项目: 江苏省科技支撑计划(BE2013879)

收稿时间: 2020-08-25; 修改时间: 2020-11-07; 采用时间: 2021-01-26

多目标优化问题是指包含两个或两个以上目标的优化问题, 其目标之间往往相互冲突且难以相互比较. 因进化算法能够较好地权衡多个目标之间的关系, 且不受问题特性的约束, 其逐渐成为了很多领域中解决多目标优化问题的一种流行的工具<sup>[1]</sup>, 如经济、医疗、物联网<sup>[2-4]</sup>.

多目标优化问题(multi-objective optimization problems, MOPs)<sup>[5]</sup>的一般形式如下:

$$\begin{aligned} \min \mathbf{F}(\mathbf{x}) &= \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T, \\ \text{subject to } \mathbf{x} &\in R^d, \end{aligned}$$

其中,  $\mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\}^T$  为  $m$  个目标组成的目标向量,  $\mathbf{x} = \{x_1, \dots, x_d\}^T$  为  $d$  个决策变量组成的决策向量,  $R^d$  为  $d$  维决策空间,  $\mathbf{F}$  表示从决策空间映射到目标空间. 在求解最小化问题时, 当且仅当  $\forall i = \{1, \dots, m\}$ ,  $f_i(\mathbf{x}^k) \leq f_i(\mathbf{x}^l)$  和  $\exists j = \{1, \dots, m\}$ ,  $f_j(\mathbf{x}^k) < f_j(\mathbf{x}^l)$ , 则称  $\mathbf{x}^k$  支配  $\mathbf{x}^l$ . 如果在解集中没有解能够支配  $\mathbf{x}$ , 则称  $\mathbf{x}$  为非支配解; 反之, 则称  $\mathbf{x}$  为支配解. 所有非支配解的决策向量构成的集合叫作帕累托解集(Pareto set, PS), 对应目标向量构成的集合叫作帕累托前沿(Pareto front, PF).

在过去的几十年里, 专家学者们提出了许多有效的多目标进化算法, 如 SPEA2<sup>[6]</sup>、NSGA-II<sup>[7]</sup>、MOEA/D<sup>[8]</sup>、IBEA<sup>[9]</sup>和 SMS-EMOA<sup>[10]</sup>, 它们被证明可以较为有效地解决 2 目标和 3 目标优化问题. 但是随着目标数的增加, 这些算法, 尤其是基于支配关系的算法, 其表现会快速衰减<sup>[11]</sup>. 目标数超过 3 的优化问题被称为超多目标优化问题, 针对此类问题而提出的算法有 NSGA-III<sup>[11]</sup>、KnEA<sup>[12]</sup>、HypE<sup>[13]</sup>和 RVEA<sup>[14]</sup>.

进化算法在解决多目标优化问题, 特别是超多目标优化问题时, 往往需要大量的真实评估, 即使用原始的目标函数评估. 然而, 现实中存在一类被称为昂贵问题的优化问题, 这类问题往往需要通过计算机仿真、真实实验或数据驱动的方式来真实评估, 其评估过程需要付出很高的代价<sup>[15]</sup>, 其高昂的评估代价导致真实评估次数受到了一定的限制. 代理辅助进化算法可以通过廉价的模型评估来减少真实评估的使用, 在降低评估代价的同时, 得到较好的结果<sup>[1,15]</sup>.

代理辅助进化算法中常用的代理模型有多项式响应模型<sup>[16]</sup>、Kriging 模型<sup>[17]</sup>、神经网络模型<sup>[18]</sup>、径向基函数网络模型<sup>[19]</sup>、支持向量回归模型<sup>[20]</sup>和集成学习模型<sup>[21]</sup>. 目前来看, 对于超多目标优化问题来说, 很多代理辅助进化算法被证明在减少真实评估次数和加速优化方面是十分有效的<sup>[22]</sup>, 如 ParEGO<sup>[23]</sup>、SMS-EGO<sup>[24]</sup>、MOEA/D-EGO<sup>[25]</sup>、K-RVEA<sup>[26]</sup>、GCS-MOE<sup>[27]</sup>、CPS-MOEA<sup>[28]</sup>和 CSEA<sup>[29]</sup>.

本文提出了一个基于多样性分类和距离回归的进化算法(DCDREA)来解决昂贵超多目标优化问题. 分类模型作为全局代理模型, 用来从候选解中选出一组有希望的解. 回归模型作为局部代理模型, 用来在有希望的解中选出用于真实评估的解. 其中, 使用分类模型作为全局代理模型主要因为以下两点: 一方面, 在解决超多目标优化问题时, 不同于分类模型, 回归模型的构建时间会随目标数的增加而成倍增加<sup>[14]</sup>; 另一方面, 对于进化算法来说, 无论是利用适应度值还是支配关系, 环境选择的本质是确定解之间的优劣关系以保留好解<sup>[30]</sup>. 因此, 使用分类模型作为全局代理模型能够控制模型构建时间, 同时也更符合进化算法的思想. 而使用回归模型作为局部代理模型主要原因有以下 3 点.

- 第一, 由于训练样本非常分散且有限, 如果将回归模型作为全局代理模型, 就有很大可能选择伪好解, 即在真实评估下并不是好解的解<sup>[27]</sup>;
- 第二, 对于回归模型来说, 训练样本在决策空间越靠近需要评估的解, 同时多样性越好, 则模型评估会相对越准确<sup>[30]</sup>. 即在同等条件下, 局部回归模型会比全局回归模型准确;
- 第三, 由于样本数量较少, 对所有样本进行学习反而会降低模型的性能<sup>[31]</sup>. 因此, 把回归模型作为局部代理模型, 可以得到在局部空间内相对准确的模型评估.

## 1 背景

本节主要简单介绍目前流行的代理辅助进化算法以及本文中全局代理模型所使用的随机森林<sup>[32]</sup>分类模型和局部代理模型所使用的 Kriging<sup>[17,33]</sup>回归模型的构建.

## 1.1 代理辅助进化算法

目前有很多关于多目标昂贵问题的代理辅助进化算法的研究,大体上可以分成使用回归模型的算法、使用分类模型的算法以及使用多模型混合的算法。

- 使用回归模型的算法一般是使用代理模型去拟合目标函数或者某个聚合函数. Knowles 提出的 ParEGO<sup>[23]</sup>把原问题分解为多个子问题,每次迭代随机选择一个子问题,使用 Kriging 模型拟合这个子问题的聚合函数,并把期望改善(EI)<sup>[17]</sup>作为其选解依据. Ponweiser 等人提出的 SMS-EGO<sup>[24]</sup>使用 Kriging 模型拟合超体积 HV<sup>[34]</sup>函数,将解对种群的 HV 贡献值,即 S-metric,作为选解依据. Zhang 等人提出的 MOEA/D-EGO<sup>[25]</sup>使用 Kriging 模型拟合每个子问题的聚合函数,并把 EI 作为其选解依据. Chugh 等人提出的 K-RVEA<sup>[26]</sup>使用 Kriging 模型分别拟合每个目标函数,并把角惩罚距离(APD)作为选解依据. Luo 等人提出的 GCS-MOE<sup>[27]</sup>使用 Kriging 模型拟合每个任务的中心子问题的聚合函数,通过任务组之间的协作寻找好解;
- 使用分类模型的算法一般需要对训练样本进行预分类,从而使分类模型可以学习到期望的分类标准. Zhang 等人提出的 CPS-MOEA<sup>[28]</sup>选用分类回归树(CART)与  $K$  近邻算法(KNN)作为代理模型,其采用非支配排序的方法把训练样本分为好解和坏解. Pan 等人提出的 CSEA<sup>[29]</sup>使用前馈神经网络(FNN)作为代理模型,其首先采用径向基投影的方法选出参考点,然后据此对训练样本进行分类;
- 除此之外,不少研究者采用多模型混合来辅助进化算法,以期获得更好的效果. Lu 等人提出的 CRADE<sup>[32]</sup>采用分类模型或回归模型处理候选解,从而分成不保留至下一代的解、需要真实评估的解和使用模型评估的解。

## 1.2 随机森林

集成学习是将多个独立基学习器融合在一起,其中,单个基学习器的性能可能较差,但是多个基学习器结合起来得到的学习器的性能会明显优于单个基学习器. Bagging<sup>[21]</sup>是集成学习中的一个代表算法,它通过对样本进行 Bootstrap 采样,即允许取出后放回的采样方式获得多个样本集,然后用每个样本集训练一个基学习器,再将这些基学习器结合起来. 随机森林是以决策树为基学习器的一种集成学习,它在 Bagging 的基础上引进了随机属性选择,进一步增加了采样获得的样本集的随机性. 相对其他算法来说,随机森林(RF)具有很好的泛化性能和准确性,同时还有学习迅速和易于实现等优点。

随机森林<sup>[33]</sup>的示意图如图 1 所示,其中的决策树首先从当前节点的属性集中选择包含  $k$  个属性的子集,然后从子集中选择最优属性进行划分. 参数  $k$  基本上控制着样本集的随机程度,文献[33]建议  $k=\log_2 d$ , 其中,  $d$  为属性的总个数。

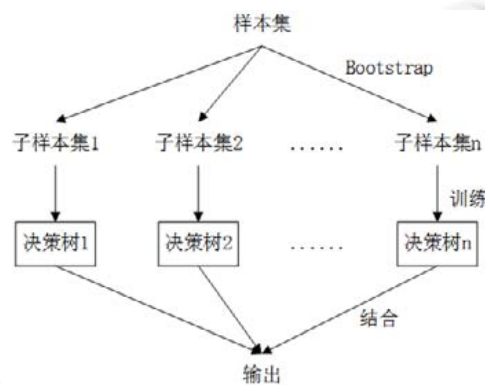


图 1 随机森林示意图

对于一个具有  $m_{train}$  个样本的数据集,通过 Bootstrap 采样法,我们可以得到  $n_t$  个样本集,每个样本集中都含有  $m_{train}$  个样本. 每一个样本集都可以用来训练一个决策树,这样就可以训练出  $n_t$  个决策树. 对预测结果进

行结合时, 分类任务一般采用投票法.

其中, 本文中随机森林的基学习器个数为 10, 即  $n_t=10$ , 决策树采用基尼指数来划分属性. 数据集  $D$  的纯度可以用基尼值来计算:

$$Gini(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} \quad (1)$$

其中,  $|y|$  为样本中总共含有的类别数量, 即可分为多少类;  $p_k$  为  $k$ th 类样本所占的比例.

属性  $a$  的基尼指数为

$$Gini\_index(D, a) = \sum_{v=1}^V \frac{|D^v|}{|D|} Gini(D^v) \quad (2)$$

其中,  $V$  为  $a$  属性可能的取值, 即样本集中  $a$  属性取值为  $\{a^1, a^2, \dots, a^V\}$ ;  $D^v$  为  $a$  属性取  $a^v$  时所包含的样本集. 对于当前节点, 最优划分属性为  $a^* = \arg_{a \in A} \min Gini\_index(D, a)$ .

### 1.3 Kriging

Kriging<sup>[17,34]</sup> 是一个无参数的机器学习算法, 具有较高的准确性, 并且可以获得解的不确定性. 为了建立一个廉价的 Kriging 模型, 我们需要两点假设.

- 第一, 假设对于任意  $\mathbf{x}$ , 其目标函数值  $y$  都满足:

$$y = \mu + \varepsilon \quad (3)$$

其中,  $\mu$  是回归模型的预测值,  $\varepsilon \sim N(0, \sigma^2)$ ;

- 第二, 假设对于任意两个  $\mathbf{x}^i$  和  $\mathbf{x}^j$ , 其高斯随机过程的协方差与它们之间的距离相关, 表示如下:

$$c[\varepsilon(\mathbf{x}^i), \varepsilon(\mathbf{x}^j)] = \sigma^2 R(\mathbf{x}^i, \mathbf{x}^j) \quad (4)$$

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\sum_{k=1}^d \theta_k |x_k^i - x_k^j|^{p_k}\right) \quad (5)$$

其中,  $\theta_k$  和  $p_k$  为超参数,  $p_k \in [1, 2]$ .

基于以上假设, 对于  $N_t$  个训练样本, 我们可以最大似然, 由公式(4)来估计参数  $\mu$  和  $\sigma^2$ :

$$\frac{1}{(2\pi\sigma^2)^{N_t/2} \sqrt{\det(\mathbf{R})}} \exp\left[-\frac{(\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right] \quad (6)$$

其中,  $\mathbf{y}$  为训练样本的目标函数值,  $\mathbf{1}$  为长度为  $N_t$  的单位列向量,  $\det(\mathbf{R})$  为相关矩阵  $\mathbf{R}$  的行列式.  $\mathbf{R}$  表示如下:

$$\mathbf{R} = \begin{bmatrix} R(\mathbf{x}^1, \mathbf{x}^2) & \dots & R(\mathbf{x}^1, \mathbf{x}^{N_t}) \\ \vdots & \ddots & \vdots \\ R(\mathbf{x}^{N_t}, \mathbf{x}^1) & \dots & R(\mathbf{x}^{N_t}, \mathbf{x}^{N_t}) \end{bmatrix} \quad (7)$$

为了最大化公式(4), 可以得到  $\mu$  和  $\sigma^2$ :

$$\mu = (\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1})^{-1} \mathbf{1}^T \mathbf{R}^{-1} \mathbf{y} \quad (8)$$

$$\sigma^2 = \frac{1}{N_t} (\mathbf{y} - \mathbf{1}\mu)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu) \quad (9)$$

至此, 对于输入  $\hat{\mathbf{x}}$ , 可以估计其近似目标值  $\hat{y}$  和方差  $\hat{\sigma}^2$ , 即不确定性:

$$\hat{y} = \mu + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\mu) \quad (10)$$

$$\hat{\sigma}^2 = \sigma^2 [1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + (\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1})^{-1} (1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2] \quad (11)$$

其中,  $\mathbf{r}^T$  是样本与输入  $\bar{\mathbf{x}}$  的相关矩阵:

$$\mathbf{r}^T = [R(\bar{\mathbf{x}}, \mathbf{x}^1), \dots, R(\bar{\mathbf{x}}, \mathbf{x}^{N_t})]^T \quad (12)$$

## 2 基于多样性分类和距离回归的进化算法(DCDREA)

本文主要提出了 DCDREA, 其算法流程图如图 2 所示, 算法框架如算法 1 所示.

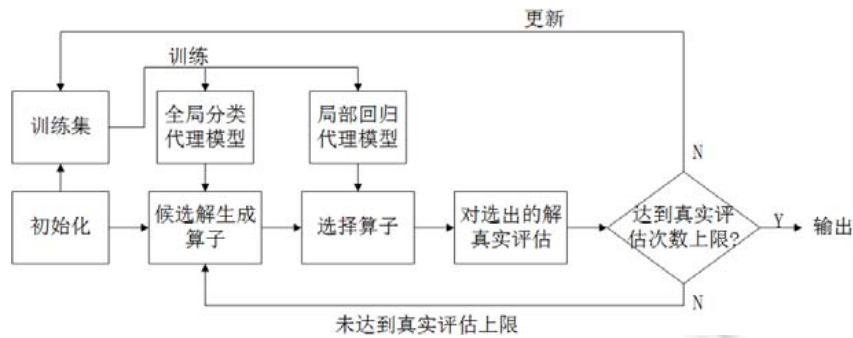


图 2 DCDREA 流程示意图

算法 1. DCDREA 算法框架.

输入: 训练样本中好解的期望数量  $k$ ; 选出解的数量  $\mu$ ; 最大真实评估次数  $FE_{\max}$ ;

输出: 种群  $P$  中的非支配解.

1. 均匀生成  $k$  个参考向量, 记为  $W$ ;
2. 采样  $11d-1$  个初始解, 真实评估后加入  $P$ ;
3. **while**  $FE_{\max} > 0$  **do**
4.    $[trainX, trainY] = sampleClassifier(P, W)$ ;                   //对全局代理模型的训练样本分类, 算法 2
5.    $globalModel = globalModelBuilder(trainX, trainY)$ ;       //全局代理模型构建
6.    $X_g = P(trainY == 1)$ ;
7.    $X = candidateProducer(X_g, globalModel, w_{\max})$ ;       //候选解生成算子, 算法 5
8.    $[trainX', trainY'] = sampleSelector(P, X)$ ;           //选择局部代理模型的训练样本, 算法 4
9.    $localModel = localModelBuilder(trainX', trainY')$ ;     //局部代理模型构建
10.    $X_f = selector(X, \mu, localModel)$ ;               //选择算子, 算法 6
11.   真实评估  $X_f$  并添加到  $P$ ;
12.    $FE_{\max} = FE_{\max} - |X_f|$ ;
13. **end**

算法大体可以分为如下 6 个部分.

- 初始化(步骤 1、步骤 2): 均匀生成  $k$  个参考向量  $W$ . 另外, 随机采样  $11d-1$  个初始解, 真实评估后加入  $P$ ;
- 训练全局代理模型(步骤 4、步骤 5): 把种群  $P$  中所有解作为训练样本, 其中, 最小相关解集作为正样本, 标记为 1; 其余作为负样本, 标记为 0. 输入空间  $trainX$  为样本的决策向量, 而输出空间  $trainY$  为样本的分类标记.  $trainX$  与  $trainY$  构成的训练样本集用于训练全局代理模型  $globalModel$ . 全局代理模型可以使用任何分类模型, 本文采用随机森林分类模型;
- 候选解生成(步骤 6、步骤 7): 候选解生成算子把  $P$  中分类为正样本的解集  $X_g$  作为初始解, 并使用  $globalModel$  来分类筛选生成的子代, 最后得到候选解集  $X$ . 在候选解生成的迭代过程中,  $globalModel$  评估上限为  $w_{\max}$  次;
- 训练局部代理模型(步骤 8、步骤 9): 局部代理模型把  $P$  中在决策空间距离  $X$  最近的所有解作为训练样本集, 把其决策向量作为输入空间  $trainX'$ , 其到理想点的距离作为输出空间  $trainY'$ .  $trainX'$  与  $trainY'$  构成的训练样本集用于训练局部代理模型  $localModel$ . 局部代理模型可以使用任何回归模型, 本文采用 Kriging 回归模型;
- 选择候选解(步骤 10): 使用  $K$ -means 方法对候选解集  $X$  聚类, 划分为  $\mu$  个簇. 使用  $localModel$  在每个簇中选出一个最优解, 把  $\mu$  个解构成的解集记为  $X_f$ ;

- 在对  $X_j$  进行真实评估并添加到  $P$  后, 如果达到真实评估次数上限, 算法终止并输出最终解, 否则, 跳回步骤 2.

其中, 初始化、训练样本的处理(包括全局代理模型训练样本的分类和局部代理模型样本的选择)、候选解生成算子和选择算子为 DCDREA 的关键组成部分, 下面分别给出详细说明.

### 2.1 初始化

初始化部分主要为均匀生成参考向量和生成初始解.

- (1) 均匀生成参考向量: 利用正则单形点阵设计方法<sup>[35,36]</sup>在单位超平面上生成一组均匀分布的参考向量. 对于每个参考向量  $w^i = \{w_1^i, \dots, w_m^i\}$ ,  $w^i \in W$ ,  $i = 1, \dots, N$ , 从下面集合中均匀取值:

$$\left\{ \frac{0}{H}, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\},$$

且满足  $\sum_{j=1}^m w_j^i = 1$ . 其中,  $H$  用于控制参考向量生成的间隔,  $m$  为目标数量. 生成参考向量的数量为  $N = C_{H+m-1}^{m-1}$ .

- (2) 生成初始解: 利用拉丁超立方随机采样(LHS)<sup>[37]</sup>方法对决策空间均匀采样初始解. 较为合适的随机采样次数为  $11d-1$ <sup>[23]</sup>, 其中,  $d$  为决策变量的数量.

### 2.2 训练样本的处理

本文采用随机森林作为全局代理模型, Kriging 模型作为局部代理模型, 相关介绍详见第 1.2 节和第 1.3 节. 这里主要介绍其训练样本的处理, 包括全局代理模型的训练样本分类以及局部代理模型的训练样本的选择.

- (1) 全局代理模型训练样本的分类

算法 2 为全局代理模型训练样本分类的过程. 由于使用的是分类模型, 所以需要对每个训练样本预先进行分类, 标记为正或负样本. 这里, 我们把  $P$  的所有解作为训练样本, 其中, 最小相关解集分类为正样本, 其余分类为负样本. 其中, 最小相关解集较好地平衡了收敛性和多样性, 在种群中具有分布性较广和收敛性较高的特点. 这样的训练样本有利于全局代理模型的学习. 另外, 随着算法的迭代, 最小相关解集会越来越接近  $PF$ , 即分类模型学习到的分类标准也会随之提高, 这会给进化算法带来更多的压力, 促使新产生的候选解向  $PF$  靠拢. 下面将对最小相关解集的计算进行详细说明.

**算法 2.** 全局代理模型训练样本的分类 *sampleClassifier*.

输入: 种群  $P$ ; 参考向量集  $W$ ;

输出: 训练样本的输入 *trainX*; 训练样本的输出 *trainY*.

1.  $X_{mcs} = MCS(P, W)$ ; //计算  $P$  的最小相关解集, 见后文算法 3
2.  $indexes = find(X_{mcs}, P)$ ; //找到  $X_{mcs}$  对应应在  $P$  中的索引
3.  $trainX = P.decs$ ; //trainX 为样本的决策向量
4.  $trainY(indexes) = 1$ ; // $X_{mcs}$  对应的样本标记设为 1
5.  $trainY(\sim indexes) = 0$ ; //其余样本的标记设为 0

本文中, 解与参考向量相关性的判定是根据它们之间的角度大小. 解  $x$  与参考向量  $w$  的角度  $\theta$  的计算如下:

$$\theta = \arccos\left(\frac{y \cdot w}{\|y\| \cdot \|w\|}\right) \quad (13)$$

其中,  $y$  是目标函数值,  $w \in W$  为参考向量,  $\|\cdot\|$  是向量的模.

最小相关解集的计算具体见算法 3, 其中,  $d(x, z^*)$  为解  $x$  到理想点  $z^*$  的距离, 如公式(14);  $C_w$  为  $w$  的最小相关解. 对于解集  $X$  中的每个解  $x$ , 其相关参考向量是与  $x$  相差角度最小的参考向量. 而对于参考向量  $w$ , 其相关解为所有相关参考向量为  $w$  的解. 参考向量  $w$  的最小相关解为其相关解中距离理想点  $z^*$  最近的解. 所有参考向量的最小相关解组成的集合, 即为  $X$  的最小相关解集. 其中,  $z^* = (\min(f_1), \dots, \min(f_m))$ ,  $\min(f_i)$  为  $X$  在第  $i$  个目标上的最小值.

$$dis(x, z^*) = \|x - z^*\| \quad (14)$$

**算法 3.** 最小相关解集 MCS.

输入: 解集  $X$ ; 参考向量集  $W$ ;

输出: 最小相关解集  $C$ .

1. **for** 每个  $x \in X$  **do**
2. 找到与  $x$  角度相差最小参考向量  $w \in W$ ;
3. **if**  $isempty(C_w) \vee d(x, z^*) < d(C_w, z^*)$  **then**
4.  $C_w = x$ ;
5. **end**
6. **end**

(2) 局部代理模型训练样本的选择

算法4为局部代理模型训练样本选择的过程. 一般来说, 随着训练样本数量的增加, 模型的训练时间会快速上升, 其中, Kriging 模型尤为明显, 其训练时间对样本数量非常敏感. 为了在不明显降低模型有效性的同时缩短建模时间, 我们只保留  $P$  中与候选解在决策空间内距离最近的解作为训练样本. 考虑到超多目标优化问题的目标数量较多, 直接拟合所有的目标函数易造成较大的误差, 难以较为准确地选解. 另外, 超多目标优化中, 根据支配关系难以直接判断解的优劣. 因此, 这里用局部代理模型去拟合解到理想点的距离.

**算法 4.** 局部代理模型训练样本的选择 sampleSelector.

输入: 种群  $P$ ; 候选解集  $X$ ;

输出: 训练样本的输入  $trainX'$ ; 训练样本的输出  $trainY'$ .

1.  $trainX', trainY' = \emptyset$ ;
2. **for** 每个  $x \in X$  **do**
3. 找到  $P$  中与  $x$  在决策空间距离最小的解  $x_s$ ;
4.  $trainX' = trainX' \cup x_s$ ;
5.  $trainY' = trainY' \cup dis(x_s, z^*)$ ;
6.  $P = P - x_s$ ;
7. **end**

### 2.3 候选解生成算子

候选解生成算子是在全局代理模型评估下迭代优化候选解, 以得到表现较为优秀的候选解, 具体见算法5. 为了充分利用现有条件, 候选解生成算子使用  $P$  的最小相关解集  $X_g$  作为初始解集. 这里可以采用任意子代生成的方法, 本文中使用了模拟二进制交叉(simulated binary crossover)<sup>[38]</sup>和多项式突变(polynomial mutation)<sup>[39]</sup>生成子代  $off$ . 最后, 使用  $globalModel$  评估  $off$ , 将其分类为好解与坏解, 并保留好解.

**算法 5.** 候选解生成算子 candidateProducer.

输入: 初始解  $X_g$ ; 全局代理模型  $globalModel$ ; 模型评估次数  $w_{max}$ ;

输出: 候选解集  $X$ .

1.  $X = X_g$ ;
2. **while**  $w_{max} > 0$  **do**
3.  $X$  作为父代, 生成子代  $off$ ;
4. 使用  $globalModel$  评估分类  $off$ , 结果记为  $label$ ;
5.  $X = X \cup off(label == 1)$ ; //保留  $label$  为 1 的子代
6.  $w_{max} = w_{max} - |off|$ ;
7. **end**

## 2.4 选择算子

选择算子是在候选解集中选择部分解进行真实评估, 具体见算法 4. 其先是使用 *localModel* 评估每个解  $x \in X$ , 得到评估值  $y_{hat}$  和不确定性  $s$ , 据此计算出下置信界(LCD)<sup>[40]</sup>值  $y_{LCD}$ . 然后对解集  $X$  使用 *K-means* 方法聚类, 划分成  $\mu$  个簇, 并在每个簇中选出一个  $y_{LCD}$  最小的解, 即选出最有可能与理想点  $z^*$  距离最小的解:

$$y_{LCD} = y_{hat} - \alpha \times s \quad (15)$$

其中,  $y_{hat}$  为解的模型评估值;  $s$  为解的不确定性;  $\alpha$  为系数, 这里取 2.

**算法 6.** 选择算子 *selector*.

输入: 候选解集  $X$ ; 选出解的数量  $\mu$ ; *localModel*: 局部代理模型;

输出: 将用于真实评估的解  $X_f$ .

1.  $X_f = \emptyset$ ;
2. 使用 *localModel* 评估  $X$  中的解, 计算出每个解的  $y_{LCD}$  值;
3. 使用 *K-means* 聚类把  $X$  划分为  $\mu$  个簇;
4. 对每个簇, 选出  $y_{LCD}$  最小的解加入  $X_f$ ;

## 3 实验

由于使用原昂贵问题评估需要花费大量的时间, 若采用真实的昂贵问题, 那么对比实验将会花费巨量的时间, 实验的时间花费将会令人难以接受. 因此, 我们限制测试问题的真实评估次数来模拟解决昂贵多目标优化问题的场景, 即测试问题只会被有限次地用于真实评估解.

本文通过在大规模 3、4、6、8、10 目标优化问题上的对比实验来证明 DCDREA 的有效性. 实验把目前较为流行且性能较佳的代理辅助进化算法作为对比算法, 如采用分类模型的算法 CPS-MOEA、CSEA 以及采用回归模型的算法 K-RVEA、MOEA/D-EGO、ParEGO.

### 3.1 测试问题集

实验选用 DTLZ1–DTLZ7<sup>[41]</sup>测试问题作为基准测试问题.

### 3.2 度量指标

实验选用反向迭代距离(IGD)<sup>[29]</sup>和超体积(HV)<sup>[42]</sup>作为度量指标.

(1) IGD 计算公式如下:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|},$$

其中,  $P$  是对  $PF$  均匀采样后得到的目标向量集合,  $P^*$  是算法计算出的目标向量集合,  $d(v, P)$  是一个目标向量  $v \in P^*$  到  $P$  中最近的向量的距离. 本文实验在  $PF$  上均匀取 10 000 个点作为参考点来计算 IGD.

(2) HV 计算公式如下:

$$HV = \text{volume} \left( \bigcup_{v \in P'} v \right),$$

其中,  $P^*$  是算法计算得到的目标向量集合,  $P'$  是  $P^*$  中非支配解的集合, *volume* 是计算向量到参考点的超体积. 其参考点的选择对 HV 结果会有较大影响, 本文实验参考点设置如下: 设  $fr = \max(f_1), \dots, \max(f_m)$ ,  $f_1, \dots, f_m \in PF$ , DTLZ2、DTLZ4、DTLZ5、DTLZ7 和 10 目标 DTLZ1、DTLZ3 的参考点为  $1.1fr$ , DTLZ6 的参考点为  $3fr$ , 其余 DTLZ1、DTLZ3 的参考点为  $120fr$ .

### 3.3 参数设置

(1) 一般设置

- 决策变量数量  $d$ : 10;



- 目标变量数量  $m$ : 3、4、6、8、10;
- 最大真实评估次数  $FE_{max}$ : 300;
- 初始采样: LHS 采样  $11d-1$  个, 即采样 109 个初始解;
- 交叉算子: 模拟二进制交叉, 设置概率为 1.0, 分布指数为 20;
- 变异算子: 多项式突变, 设置概率为  $1/d$ , 分布指数为 20;
- 候选解生成算子的模型评估次数  $w_{max}$ :  $20 \times (11d-1)$ ;
- 运行次数: 每个算法在每个测试问题上独立运行 20 次;
- 种群大小: 每个算法的种群大小由参数  $(H_1, H_2)$  和目标数决定<sup>[11,43]</sup>, 设置见表 1. DCDREA 中的参数  $k$  与表 1 中种群大小的设置一致.

表 1 不同目标下,  $(H_1, H_2)$  的设置和对应的种群大小

目标数	$(H_1, H_2)$	种群大小
3	(5,0)	21
4	(3,1)	24
6	(2,1)	27
8	(2,1)	44
10	(2,1)	65

(2) 其他设置

本文实验的  $\mu$  设置为 5, 即选择算子每次选出 5 个解. CPS-MOEA 使用 KNN 作为其分类模型, 其中,  $K=5$ . 其他未说明的参数设置与其原文中相一致.

3.4 实验结果

本节主要对实验结果在算法的收敛性和多样性表现以及运行时间方面进行分析.

图3-图7是各算法的收敛曲线. 表2和表3分别是各算法在 DTLZ1-DTLZ7 测试问题上的 IGD 和 HV 结果. 其中, 数字表示该指标的平均值, 括号内的数字表示该指标的标准差, 加粗表示在此测试问题上该项平均值为所有算法中的最优. 另外, 我们使用秩和检验在显著性水平为 0.05 下对 20 次独立运行结果进行分析. + 表示此算法与 DCDREA 相比在此测试问题上表现更好, - 表示此算法与 DCDREA 相比在此测试问题上表现更差,  $\approx$  表示此算法与 DCDREA 相比在此测试问题上的表现区别不大. 每个表格的最后一行还给出了各个算法在 35 个测试用例下 +/~/ $\approx$  的汇总.

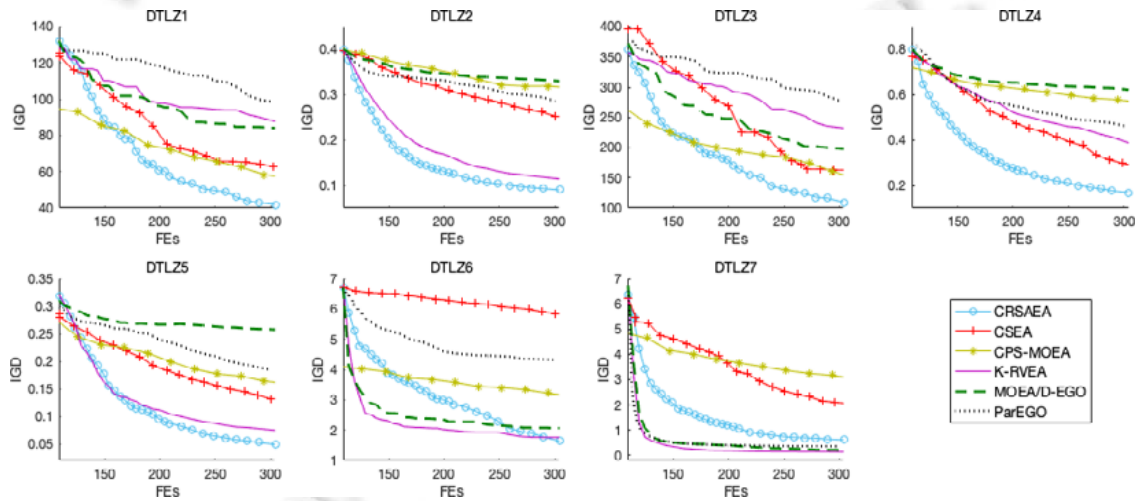


图 3 各个算法在 3 目标下的收敛曲线

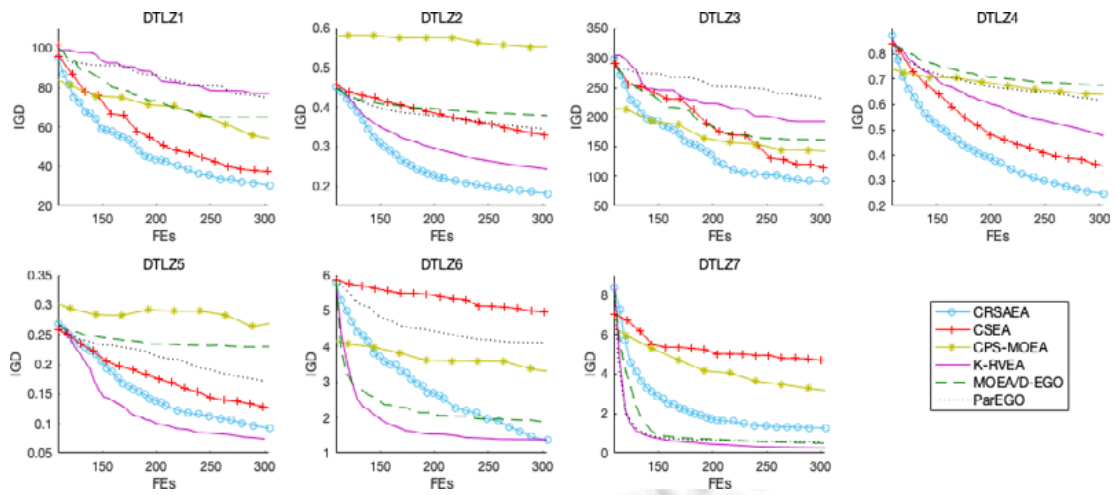


图 4 各个算法在 4 目标下的收敛曲线

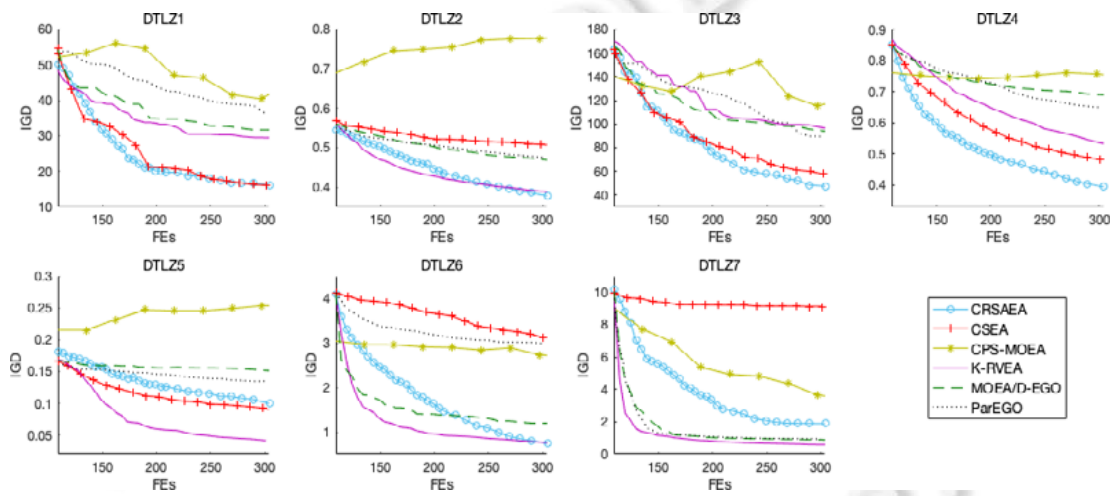


图 5 各个算法在 6 目标下的收敛曲线

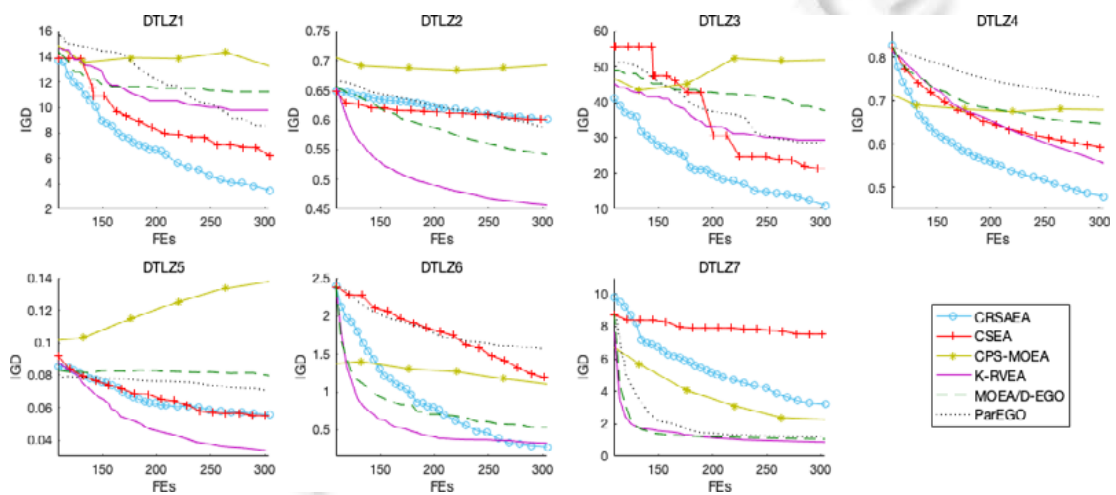


图 6 各个算法在 8 目标下的收敛曲线

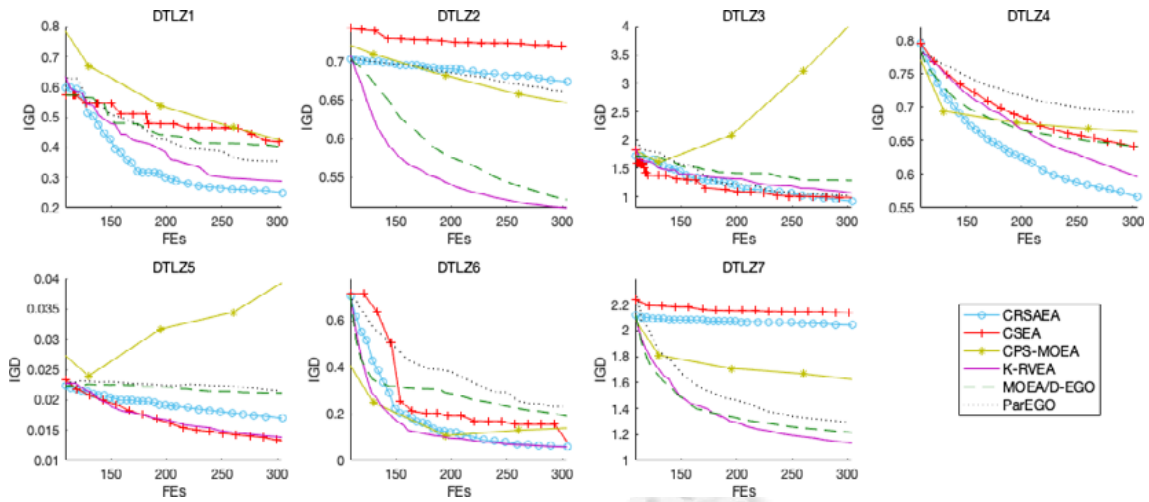


图 7 各个算法在 10 目标下的收敛曲线

表 2 反向迭代距离(IGD)指标结果

Problem	$m$	DCDREA	CSEA	CPS-MOEA
DTLZ1	3	<b>4.1940e+1 (1.04e+1)</b>	6.0729e+1 (1.47e+1)-	5.6566e+1 (1.54e+1)-
	4	<b>3.0333e+1 (8.12e+0)</b>	3.7365e+1 (1.05e+1)-	5.3392e+1 (1.07e+1)-
	6	<b>1.6227e+1 (4.69e+0)</b>	1.6271e+1 (4.03e+0)≈	4.5673e+1 (1.69e+1)-
	8	<b>3.4697e+0 (1.96e+0)</b>	5.3275e+0 (2.57e+0)-	1.3159e+1 (5.03e+0)-
	10	<b>2.5069e-1 (2.42e-2)</b>	3.0320e-1 (7.17e-2)-	4.0143e-1 (1.19e-1)-
DTLZ2	3	<b>8.9727e-2 (8.06e-3)</b>	2.5091e-1 (3.37e-2)-	3.1416e-1 (5.60e-2)-
	4	<b>1.8211e-1 (1.34e-2)</b>	3.2459e-1 (2.44e-2)-	5.5280e-1 (5.46e-2)-
	6	<b>3.7861e-1 (2.48e-2)</b>	4.8831e-1 (3.02e-2)-	7.8147e-1 (6.59e-2)-
	8	6.0093e-1 (3.76e-2)	6.1842e-1 (3.20e-2)≈	6.9348e-1 (2.94e-2)-
	10	6.7392e-1 (1.71e-2)	6.7737e-1 (1.99e-2)≈	6.4153e-1 (2.99e-2)+
DTLZ3	3	<b>1.0947e+2 (2.92e+1)</b>	1.6383e+2 (4.95e+1)-	1.4808e+2 (5.58e+1)-
	4	<b>9.1753e+1 (1.79e+1)</b>	1.1383e+2 (2.74e+1)-	1.4143e+2 (4.48e+1)-
	6	<b>4.7585e+1 (1.69e+1)</b>	5.4629e+1 (1.82e+1)≈	1.2185e+2 (4.82e+1)-
	8	<b>1.0946e+1 (6.23e+0)</b>	1.5586e+1 (8.94e+0)-	5.1963e+1 (1.71e+1)-
	10	<b>9.2328e-1 (1.45e-1)</b>	1.0812e+0 (2.89e-1)-	4.4706e+0 (3.99e+0)-
DTLZ4	3	<b>1.6774e-1 (3.23e-2)</b>	3.3389e-1 (1.62e-1)-	5.6645e-1 (1.11e-1)-
	4	<b>2.4874e-1 (3.11e-2)</b>	3.3815e-1 (5.98e-2)-	6.3953e-1 (5.54e-2)-
	6	<b>3.9464e-1 (4.32e-2)</b>	4.8195e-1 (5.21e-2)-	7.4981e-1 (4.35e-2)-
	8	<b>4.7957e-1 (2.04e-2)</b>	5.9261e-1 (3.27e-2)-	6.7973e-1 (1.79e-2)-
	10	<b>5.6650e-1 (1.33e-2)</b>	6.4122e-1 (2.99e-2)-	6.6101e-1 (1.18e-2)-
DTLZ5	3	<b>4.8970e-2 (1.44e-2)</b>	1.2649e-1 (3.22e-2)-	1.5943e-1 (3.91e-2)-
	4	9.2050e-2 (2.04e-2)	1.2385e-1 (2.10e-2)-	2.7119e-1 (4.85e-2)-
	6	9.9738e-2 (2.22e-2)	9.0433e-2 (1.89e-2)≈	2.5365e-1 (6.94e-2)-
	8	5.5616e-2 (1.13e-2)	5.0851e-2 (7.82e-3)≈	1.3833e-1 (3.25e-2)-
	10	1.6952e-2 (1.80e-3)	<b>1.3763e-2 (1.49e-3)+</b>	4.1612e-2 (8.38e-3)-
DTLZ6	3	<b>1.6392e+0 (4.94e-1)</b>	5.8099e+0 (3.73e-1)-	3.1075e+0 (9.12e-1)-
	4	<b>1.3786e+0 (4.57e-1)</b>	4.9426e+0 (3.81e-1)-	3.2676e+0 (1.01e+0)-
	6	<b>7.4126e-1 (4.17e-1)</b>	3.0858e+0 (6.03e-1)-	2.6801e+0 (5.76e-1)-
	8	<b>2.7087e-1 (2.10e-1)</b>	1.2206e+0 (4.69e-1)-	1.0972e+0 (5.62e-1)-
	10	6.0169e-2 (1.63e-2)	1.7518e-1 (1.81e-1)-	1.4343e-1 (5.36e-2)-
DTLZ7	3	6.1693e-1 (3.65e-1)	1.6375e+0 (1.06e+0)-	3.0153e+0 (1.98e+0)-
	4	1.2777e+0 (3.80e-1)	4.4298e+0 (1.63e+0)-	3.1305e+0 (1.95e+0)-
	6	1.8817e+0 (3.86e-1)	9.1146e+0 (2.09e+0)-	3.4162e+0 (2.39e+0)≈
	8	3.1865e+0 (5.03e-1)	7.5509e+0 (2.02e+0)-	2.2579e+0 (9.18e-1)+
	10	2.0424e+0 (3.26e-1)	2.1378e+0 (4.46e-1)≈	1.6052e+0 (1.37e-1)+
+/-/≈			1/27/7	3/31/1

表 2 反向迭代距离(IGD)指标结果(续)

Problem	<i>m</i>	K-RVEA	MOEA/D-EGO	ParEGO
DTLZ1	3	8.7794e+1 (2.33e+1)-	8.4134e+1 (1.25e+1)-	9.8745e+1 (2.77e+1)-
	4	7.7108e+1 (1.54e+1)-	6.5017e+1 (1.17e+1)-	7.5079e+1 (2.05e+1)-
	6	2.9461e+1 (7.93e+0)-	3.1707e+1 (8.41e+0)-	3.6348e+1 (1.20e+1)-
	8	9.8363e+0 (3.46e+0)-	1.1251e+1 (3.58e+0)-	8.5461e+0 (4.01e+0)-
	10	2.8911e-1 (4.47e-2)-	4.0170e-1 (1.16e-1)-	3.5485e-1 (1.07e-1)-
DTLZ2	3	1.1418e-1 (9.33e-3)-	3.3007e-1 (2.71e-2)-	2.8540e-1 (3.20e-2)-
	4	2.4341e-1 (1.87e-2)-	3.7937e-1 (2.53e-2)-	3.4529e-1 (2.19e-2)-
	6	3.8881e-1 (1.24e-2)≈	4.6918e-1 (2.22e-2)-	4.7584e-1 (2.92e-2)-
	8	<b>4.5671e-1 (1.12e-2)+</b>	5.4193e-1 (2.83e-2)+	5.8745e-1 (2.49e-2)≈
	10	<b>5.1013e-1 (8.62e-3)+</b>	5.2104e-1 (2.19e-2)+	6.6209e-1 (3.30e-2)≈
DTLZ3	3	2.3223e+2 (6.44e+1)-	1.9804e+2 (3.57e+1)-	2.7805e+2 (5.16e+1)-
	4	1.9275e+2 (3.89e+1)-	1.6163e+2 (1.97e+1)-	2.3177e+2 (4.31e+1)-
	6	9.6472e+1 (2.65e+1)-	9.3965e+1 (2.15e+1)-	8.9375e+1 (3.38e+1)-
	8	2.9345e+1 (1.26e+1)-	3.7821e+1 (1.20e+1)-	2.8573e+1 (1.28e+1)-
	10	1.0633e+0 (1.68e-1)-	1.2884e+0 (3.23e-1)-	1.0202e+0 (8.84e-2)-
DTLZ4	3	3.8752e-1 (9.90e-2)-	6.2227e-1 (5.24e-2)-	4.6181e-1 (1.11e-1)-
	4	4.7711e-1 (9.45e-2)-	6.7619e-1 (4.98e-2)-	6.1837e-1 (7.31e-2)-
	6	5.3293e-1 (6.60e-2)-	6.8987e-1 (3.35e-2)-	6.4966e-1 (5.69e-2)-
	8	5.5506e-1 (4.75e-2)-	6.4652e-1 (1.49e-2)-	7.0838e-1 (4.59e-2)-
	10	5.9644e-1 (3.34e-2)-	6.4151e-1 (1.15e-2)-	6.9226e-1 (2.36e-2)-
DTLZ5	3	7.4675e-2 (1.35e-2)-	2.5758e-1 (3.00e-2)-	1.8449e-1 (2.70e-2)-
	4	<b>7.4204e-2 (1.43e-2)+</b>	2.2970e-1 (2.20e-2)-	1.7121e-1 (2.67e-2)-
	6	<b>4.1696e-2 (9.84e-3)+</b>	1.5111e-1 (1.84e-2)-	1.3482e-1 (1.79e-2)-
	8	<b>3.3471e-2 (8.98e-3)+</b>	7.9811e-2 (8.18e-3)-	7.0680e-2 (9.54e-3)-
	10	1.3828e-2 (2.11e-3)+	2.1128e-2 (1.97e-3)-	2.1497e-2 (2.67e-3)-
DTLZ6	3	1.7591e+0 (3.60e-1)≈	2.0638e+0 (7.35e-1)-	4.3254e+0 (3.79e-1)-
	4	1.3823e+0 (2.51e-1)≈	1.8642e+0 (4.93e-1)-	4.1099e+0 (4.64e-1)-
	6	7.6180e-1 (2.33e-1)≈	1.1981e+0 (4.55e-1)-	2.9873e+0 (3.31e-1)-
	8	3.2327e-1 (8.41e-2)-	5.3158e-1 (2.36e-1)-	1.5660e+0 (3.19e-1)-
	10	<b>5.7897e-2 (1.57e-2)≈</b>	1.9007e-1 (7.23e-2)-	2.3082e-1 (1.20e-1)-
DTLZ7	3	<b>1.3785e-1 (1.32e-2)+</b>	2.0830e-1 (4.87e-2)+	3.5891e-1 (5.22e-2)≈
	4	<b>2.7038e-1 (7.64e-2)+</b>	5.1056e-1 (5.46e-2)+	5.8401e-1 (8.09e-2)+
	6	<b>5.9232e-1 (6.65e-2)+</b>	8.9901e-1 (6.69e-2)+	9.4171e-1 (6.67e-2)+
	8	<b>8.8726e-1 (4.84e-2)+</b>	1.0752e+0 (6.22e-2)+	1.2105e+0 (1.31e-1)+
	10	<b>1.1357e+0 (4.30e-2)+</b>	1.2152e+0 (1.85e-2)+	1.2925e+0 (5.38e-2)+
+/-/≈		11/19/5	7/28/0	4/28/3

表 3 超体积(HV)指标结果

Problem	<i>m</i>	DCDREA	CSEA	CPS-MOEA
DTLZ1	3	<b>3.8710e-1 (1.81e-1)</b>	1.2035e-1 (1.37e-1)-	1.4018e-1 (1.46e-1)-
	4	<b>6.2617e-1 (1.82e-1)</b>	4.7747e-1 (1.98e-1)≈	1.7636e-1 (1.14e-1)-
	6	<b>8.9030e-1 (6.87e-2)</b>	8.8823e-1 (6.12e-2)≈	3.5164e-1 (2.84e-1)-
	8	9.8652e-1 (1.13e-2)	<b>9.8963e-1 (9.10e-3)≈</b>	9.5185e-1 (4.90e-2)-
	10	<b>6.3704e-1 (8.46e-2)</b>	5.6739e-1 (1.95e-1)≈	2.8127e-1 (1.80e-1)-
DTLZ2	3	<b>5.2297e-1 (1.13e-2)</b>	2.4190e-1 (5.92e-2)-	1.5342e-1 (4.96e-2)-
	4	<b>6.1708e-1 (2.39e-2)</b>	3.1313e-1 (5.55e-2)-	7.7482e-2 (3.91e-2)-
	6	6.2976e-1 (4.76e-2)	4.4183e-1 (4.55e-2)-	5.5830e-2 (3.25e-2)-
	8	5.8139e-1 (4.07e-2)	5.4614e-1 (2.95e-2)-	2.1205e-1 (5.32e-2)-
	10	6.5430e-1 (3.67e-2)	6.3873e-1 (2.54e-2)≈	5.2675e-1 (2.84e-2)-
DTLZ3	3	<b>2.3488e-1 (2.21e-1)</b>	4.1531e-2 (9.11e-2)-	1.1308e-1 (2.95e-1)-
	4	<b>3.4835e-1 (1.83e-1)</b>	1.4957e-1 (1.72e-1)-	6.6512e-2 (1.04e-1)-
	6	<b>8.1672e-1 (8.87e-2)</b>	7.4672e-1 (1.44e-1)≈	1.6143e-1 (1.42e-1)-
	8	<b>9.8479e-1 (1.32e-2)</b>	9.7743e-1 (1.65e-2)≈	7.0308e-1 (2.00e-1)-
	10	2.1726e-1 (8.53e-2)	<b>2.4786e-1 (1.64e-1)≈</b>	1.6453e-2 (3.17e-2)-
DTLZ4	3	<b>3.8446e-1 (6.21e-2)</b>	2.5919e-1 (6.78e-2)-	5.8182e-2 (3.12e-2)-
	4	<b>5.2892e-1 (4.87e-2)</b>	3.8827e-1 (7.77e-2)-	4.7312e-2 (4.21e-2)-
	6	<b>7.0549e-1 (4.89e-2)</b>	5.7649e-1 (4.93e-2)-	6.9550e-2 (5.23e-2)-
	8	<b>8.1665e-1 (2.85e-2)</b>	6.8987e-1 (4.64e-2)-	3.1900e-1 (5.67e-2)-
	10	<b>8.8611e-1 (1.13e-2)</b>	8.0089e-1 (3.58e-2)-	7.1893e-1 (3.60e-2)-

表 3 超体积(HV)指标结果(续 1)

Problem	m	DCDREA	CSEA	CPS-MOEA
DTLZ5	3	<b>1.6108e-1 (1.38e-2)</b>	9.0637e-2 (3.01e-2)-	7.4370e-2 (2.29e-2)-
	4	1.1354e-1 (1.47e-2)	7.0312e-2 (2.01e-2)-	2.0278e-2 (1.20e-2)-
	6	8.3102e-2 (1.50e-2)	9.2042e-2 (1.43e-2)+	3.2575e-2 (2.19e-2)-
	8	9.3451e-2 (4.52e-3)	9.9619e-2 (2.48e-3)+	7.5687e-2 (1.55e-2)-
	10	9.7285e-2 (1.19e-3)	<b>9.9786e-2 (3.72e-4)+</b>	9.8014e-2 (6.47e-4)+
DTLZ6	3	<b>1.7854e-1 (1.62e-1)</b>	0.0000e+0 (0.00e+0)-	3.2579e-2 (9.18e-2)-
	4	<b>2.1406e-1 (1.39e-1)</b>	0.0000e+0 (0.00e+0)-	1.3671e-2 (3.08e-2)-
	6	4.5767e-1 (1.55e-1)	7.3764e-3 (2.67e-2)-	4.0017e-2 (9.79e-2)-
	8	<b>6.1020e-1 (1.08e-1)</b>	3.1738e-1 (1.95e-1)-	3.4743e-1 (1.78e-1)-
	10	7.0666e-1 (9.84e-3)	6.8139e-1 (5.62e-2)≈	6.8452e-1 (2.55e-2)-
DTLZ7	3	1.9034e-1 (3.47e-2)	3.3204e-2 (3.85e-2)-	5.4078e-3 (1.74e-2)-
	4	1.6420e-1 (2.46e-2)	2.2172e-4 (5.90e-4)-	4.6282e-4 (1.01e-3)-
	6	1.5647e-1 (1.37e-2)	0.0000e+0 (0.00e+0)-	4.9351e-5 (2.16e-4)-
	8	1.3745e-1 (1.16e-2)	5.6533e-3 (1.94e-2)-	3.2441e-4 (9.73e-4)-
	10	1.3962e-1 (2.69e-2)	3.2552e-2 (3.65e-2)-	9.1211e-4 (1.59e-3)-
+/-/≈			3/23/9	1/34/0

表 3 超体积(HV)指标结果(续 2)

Problem	m	K-RVEA	MOEA/D-EGO	ParEGO
DTLZ1	3	2.0949e-2 (4.21e-2)-	9.0370e-3 (2.18e-2)-	2.2044e-2 (8.74e-2)-
	4	2.3815e-2 (3.33e-2)-	5.5542e-2 (7.95e-2)-	5.1560e-2 (9.14e-2)-
	6	6.0647e-1 (1.08e-1)-	6.1504e-1 (1.23e-1)-	3.9553e-1 (1.58e-1)-
	8	9.5801e-1 (2.19e-2)-	9.6629e-1 (2.91e-2)-	9.3981e-1 (3.22e-2)-
	10	5.6661e-1 (1.43e-1)≈	2.6987e-1 (1.76e-1)-	3.7093e-1 (1.90e-1)-
DTLZ2	3	4.8070e-1 (1.74e-2)-	1.5295e-1 (3.88e-2)-	1.6993e-1 (4.70e-2)-
	4	5.7239e-1 (3.86e-2)-	2.1608e-1 (4.60e-2)-	2.6122e-1 (4.17e-2)-
	6	<b>7.3046e-1 (2.34e-2)+</b>	3.6329e-1 (4.98e-2)-	3.8260e-1 (4.08e-2)-
	8	<b>8.2842e-1 (2.58e-2)+</b>	5.6131e-1 (3.62e-2)≈	5.4222e-1 (3.51e-2)-
	10	<b>8.9448e-1 (1.14e-2)+</b>	8.1666e-1 (2.18e-2)+	6.6286e-1 (2.94e-2)≈
DTLZ3	3	9.9678e-3 (4.46e-2)-	0.0000e+0 (0.00e+0)-	0.0000e+0 (0.00e+0)-
	4	7.6846e-3 (3.44e-2)-	4.2296e-3 (1.89e-2)-	0.0000e+0 (0.00e+0)-
	6	2.3205e-1 (1.42e-1)-	2.8923e-1 (1.62e-1)-	2.8853e-1 (1.94e-1)-
	8	9.3182e-1 (4.24e-2)-	9.1209e-1 (4.60e-2)-	8.4333e-1 (8.25e-2)-
	10	1.3818e-1 (9.58e-2)-	6.4332e-2 (6.66e-2)-	1.4099e-1 (6.24e-2)-
DTLZ4	3	1.0118e-1 (1.03e-1)-	1.4506e-2 (1.98e-2)-	1.3416e-1 (7.35e-2)-
	4	1.3822e-1 (9.84e-2)-	2.8731e-2 (2.39e-2)-	1.3448e-1 (5.41e-2)-
	6	3.8572e-1 (1.28e-1)-	1.0969e-1 (4.48e-2)-	3.2167e-1 (6.90e-2)-
	8	5.9983e-1 (1.34e-1)-	3.5826e-1 (6.46e-2)-	4.2339e-1 (7.34e-2)-
	10	8.2775e-1 (4.93e-2)-	7.5020e-1 (2.29e-2)-	6.7813e-1 (4.06e-2)-
DTLZ5	3	1.4157e-1 (1.05e-2)-	2.8418e-2 (2.71e-2)-	5.6991e-2 (2.21e-2)-
	4	<b>1.2538e-1 (8.77e-3)+</b>	2.9408e-2 (2.35e-2)-	3.0916e-2 (1.95e-2)-
	6	<b>1.0397e-1 (3.34e-3)+</b>	5.7465e-2 (2.03e-2)-	5.2187e-2 (2.01e-2)-
	8	<b>9.9846e-2 (1.41e-3)+</b>	7.9708e-2 (8.51e-3)-	7.5478e-2 (1.46e-2)-
	10	9.9639e-2 (4.61e-4)+	9.6606e-2 (9.97e-4)-	9.6110e-2 (9.25e-4)-
DTLZ6	3	1.2750e-1 (1.32e-1)≈	1.3563e-1 (1.96e-1)≈	0.0000e+0 (0.00e+0)-
	4	2.1212e-1 (1.23e-1)≈	1.0450e-1 (1.44e-1)-	0.0000e+0 (0.00e+0)-
	6	<b>4.6690e-1 (1.13e-1)≈</b>	2.8227e-1 (2.32e-1)-	3.6003e-3 (1.56e-2)-
	8	6.0807e-1 (6.83e-2)≈	5.6810e-1 (1.63e-1)≈	1.7467e-1 (9.38e-2)-
	10	<b>7.1615e-1 (5.97e-3)+</b>	7.0005e-1 (6.85e-3)-	6.9554e-1 (3.57e-2)≈
DTLZ7	3	<b>2.6052e-1 (3.81e-3)+</b>	2.1355e-1 (1.73e-2)≈	1.4082e-1 (2.61e-2)-
	4	<b>2.3877e-1 (7.00e-3)+</b>	1.0728e-1 (4.78e-2)-	8.1969e-2 (3.48e-2)-
	6	<b>2.0845e-1 (3.31e-3)+</b>	1.2158e-2 (1.70e-2)-	2.0588e-2 (2.07e-2)-
	8	<b>1.8387e-1 (5.26e-3)+</b>	8.3091e-3 (1.45e-2)-	5.6940e-3 (9.34e-3)-
	10	<b>1.6830e-1 (3.18e-3)+</b>	1.5842e-2 (1.60e-2)-	2.4331e-2 (2.12e-2)-
+/-/≈		13/17/5	1/30/4	0/33/2

(1) 收敛性和多样性

从表 2、表 3 可知, CSEA、CPS-MOEA、K-RVEA、MOEA/D-EGO 和 ParEGO 在绝大多数测试用例中的

表现都不如 DCDREA. 其中, 仅有 K-RVEA 的表现较为突出, 有 11 个 IGD 结果和 13 个 HV 结果要优于 DCDREA, 但仍有 19 个 IGD 结果和 17 个 HV 结果要差于 DCDREA. 总的来看, DCDREA 的表现是最好的.

根据表 2、表 3 和收敛曲线, DCDREA 在目标数较少的情况下表现更好. 随着目标数的增加, 分类和回归模型的拟合难度大幅度增加, 这就限制了代理模型预测的准确度, 从而导致表现变差.

相比使用回归模型的算法, DCDREA 在大多数时候表现更好, 仅在 DTLZ5–DTLZ7 问题上表现不佳. DTLZ5 和 DTLZ6 的  $PF$  是降维的超平面,  $PF$  相对集中, 这直接减少了有效的参考向量的数量, 从而间接导致最小相关解的数量的减少. 对于 DCDREA, 最小相关解是全局代理模型的训练样本中的正样本. 较少的正样本导致进化压力不够, 从而生成了质量较差的候选解. 另外, 分类模型选出的候选解倾向于正样本附近, 而正样本较少且收敛性较好, 不易被新产生的解更新掉, 最终导致结果不理想. DTLZ7 的  $PS$  由多个断开的区域组成, 分类模型很难较好地学习多区域的  $PS$ , 尤其是直线区域. CPS-MOEA 和 CSEA 的表现也恰好说明了这一点. 虽然 DCDREA 在多区域  $PS$  或者降维  $PF$  的问题上表现不佳, 但其对应的收敛曲线仍是下行的, 这表明, 其在优化方面仍是有效的.

相比使用分类模型的算法, DCDREA 在绝大多数时候明显表现更好. 这说明, 引入回归模型作为局部代理模型对选解有一定的帮助.

总的来说, DCDREA 在收敛性和多样性方面表现较好.

## (2) 运行时间

图 8 为各算法在不同目标数下的平均运行时间, 其中, 各算法的运行时间是取算法在同一目标数的 DTLZ1–DTLZ7 问题上运行时间的平均值. 在图中可以明显地看到: DCDREA 的运行时间要大大低于使用回归模型的算法, 甚至在 10 目标时, 它比 MOEA/D-EGO 要快 50 倍以上. 而 DCDREA 的运行时间与使用分类模型的算法相比相差非常小. 总的来说, DCDREA 在运行时间方面表现得较为出色.

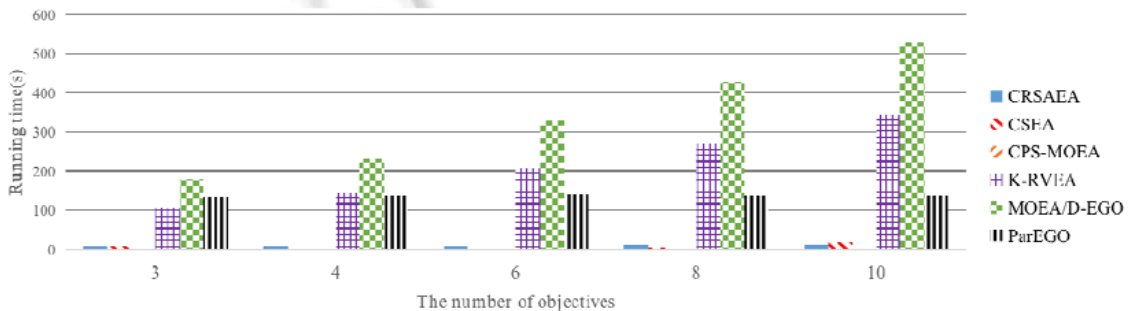


图 8 各算法在不同目标数下的平均运行时间

## (3) DCDREA 分析

由此可见, DCDREA 可以有效地解决选用的超多目标测试问题, 主要分析有以下几点.

- 首先, 构建精度较好的全局回归模型需要大量的样本, 而对于昂贵问题来说, 一般真实评估次数十分有限, 因此, 构建的全局回归模型往往是不准确的或者只是局部准确的. 而分类模型是学习样本中包含的分类标准, 即根据正负样本分布来分类选解. 对于 DCDREA, 训练样本中的正样本每次迭代都会更靠近  $PF$ , 其分类标准也会更加靠近  $PF$ , 对应的进化压力也会更大. 相比回归模型, 分类模型更容易选出合适的候选解, 同时也更加符合进化算法的思想.
- 其次, DCDREA 在选择算子中引入了局部回归模型, 从而进一步从有希望的候选解中选出进行真实评估的解. 由于全局分类模型产生的候选解分布在靠近正样本的局部空间, 而在局部空间构建的回归模型对候选解来说往往具有较好的精度. 另外, 局部回归模型没有直接拟合目标函数, 而是拟合解与理想点的距离, 从而避开了目标之间的耦合关系. 这也是同样使用 Kriging 但 DCDREA 运行速度更快的原因.



## 4 总 结

本文提出了一个基于多样性分类和距离回归的进化算法来解决昂贵超多目标优化问题。一开始, 算法先对训练样本进行分类, 并训练全局分类模型, 使其学习到样本中包含的分类标准。在候选解生成算子中, 算法用全局分类模型对候选解进行分类, 以保留有希望的候选解。在选择算子中, 算法先使用 *K-means* 方法对候选解聚类, 划分为 $\mu$ 个簇, 再对每个簇使用局部回归模型选出一个进行真实评估的解。另外, DCDREA 的局部回归模型只选用候选解附近的样本, 大大缩短了建模时间。实验部分选用大规模 3、4、6、8、10 目标的 DTLZ 测试问题, 与目前流行的 CSEA、CPS-MOEA、K-RVEA、MOEA/D-EGO 和 ParEGO 算法进行对比实验。结果表明, DCDREA 在大多数情况下表现得更好。因此, 本文提出的方法是有效可行的。

随着目标数量的增大, 目标空间呈指数扩大, 而基于分解的算法使用大数量的参考向量是不现实的, 所以, 如何使用有限的参考向量来均匀地代表目标空间是一项挑战。此外, 在目标数大于 3 以后, 对于算法结果的可视化以及对于多区域 *PS* 和降维 *PF* 的问题, 使用分类模型的算法如何改进其选择算子也是重要方向。

### References:

- [1] Jin Y, Wang H, Chugh T, *et al.* Data-driven evolutionary optimization: An overview and case studies. *IEEE Trans. on Evolutionary Computation*, 2019, 23(3): 442–458.
- [2] Ponsich A, Jaimes AL, Coello CAC. A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Trans. on Evolutionary Computation*, 2013, 17(3): 321–344. [doi: 10.1109/TEVC.2012.2196800]
- [3] Wang H, Jin Y. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE Trans. on Cybernetics*, 2018, 50(2): 536–549. [doi: 10.1109/TCYB.2018.2869674]
- [4] Ye M, Wang YP. A new malicious nodes attack-resistant security location method in wireless sensor network. *Chinese Journal of Computers*, 2013, 36(3): 532–545 (in Chinese with English abstract).
- [5] Gong MG, Jiao LC, Yang DD, Ma WP. Research on evolutionary multi-objective optimization algorithms. *Ruan Jian Xue Bao/ Journal of Software*, 2009, 20(2): 271–289 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]
- [6] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. *TIK-Report*, 2001, 103: 1–21. [doi: 10.3929/ethz-a-004284029]
- [7] Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 2002, 6(2): 0–197.
- [8] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2008, 11(6): 712–731.
- [9] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Yao X, *et al.*, eds. *Proc. of the Parallel Problem Solving from Nature—PPSN VIII (PPSN 2004)*. LNCS 3242, Berlin, Heidelberg: Springer, 2004. 832–842.
- [10] Beume N, Naujoks B, Emmerich M. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 2007, 181(3): 1653–1669.
- [11] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach. Part I: Solving problems with box constraints. *IEEE Trans. on Evolutionary Computation*, 2014, 18(4): 577–601.
- [12] Zhang X, Tian Y, Jin Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2015, 19(6): 761–775.
- [13] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, 19(1): 45–76.
- [14] Chugh T, Jin Y, Olhofer M, *et al.* A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016, 20(5): 773–791.
- [15] Jin Y. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 2011, 1(2): 61–70.

- [16] Zhou Z, Ong YS, Nguyen MH, *et al.* A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In: Proc. of the 2005 IEEE Congress on Evolutionary Computation (CEC). Edinburgh, 2005. 2832–2839. [doi: 10.1109/CEC.2005.1555050]
- [17] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998, 13(4): 455–492.
- [18] Jin Y, Olhofer M, Sendhoff B. A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans. on Evolutionary Computation*, 2002, 6(5): 481–494. [doi: 10.1109/TEVC.2002.800884]
- [19] Yi J, Shen Y, Shoemaker CA. A multi-fidelity RBF surrogate-based optimization framework for computationally expensive multimodal problems with application to capacity planning of manufacturing systems. *Structural and Multidisciplinary Optimization*, 2020. [doi: 10.1007/s00158-020-02575-7]
- [20] Cortes C, Vapnik VN. Support-Vector networks. *Machine Learning*, 1995, 20(3): 273–297.
- [21] Breiman L. Bagging predictors. *Machine Learning*, 1996, 26(2): 123–140.
- [22] Zhang J, Xing L. A survey of multiobjective evolutionary algorithms. In: Proc. of the 2017 IEEE Int'l Conf. on Computational Science and Engineering (CSE) and IEEE Int'l Conf. on Embedded and Ubiquitous Computing (EUC). Guangzhou: IEEE, 2017. 93–100. [doi: 10.1109/CSE-EUC.2017.27]
- [23] Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans. on Evolutionary Computation*, 2006, 10(1): 50–66.
- [24] Ponweiser W, Wagner T, Biermann D, *et al.* Multiobjective optimization on a limited budget of evaluations using model-assisted s-metric selection. In: Rudolph G, Jansen T, Beume N, Lucas S, Poloni C, eds. Proc. of the Parallel Problem Solving from Nature—PPSN X (PPSN 2008). LNCS 5199, Berlin, Heidelberg: Springer, 2008. 784–794.
- [25] Zhang Q, Liu W, Tsang E, *et al.* Expensive multiobjective optimization by MOEA/D with Gaussian process model. *IEEE Trans. on Evolutionary Computation*, 2010, 14(3): 456–474.
- [26] Chugh T, Jin Y, Miettinen K, *et al.* A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016, 22(1): 129–142.
- [27] Luo J, Gupta A, Ong YS, *et al.* Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates. *IEEE Trans. on Cybernetics*, 2019, 49(5): 1708–1721. [doi: 10.1109/TCYB.2018.2811761]
- [28] Zhang J, Zhou A, Zhang G. A classification and Pareto domination based multiobjective evolutionary algorithm. In: Proc. of the 2015 IEEE Congress on Evolutionary Computation (CEC). Sendai, 2015. 2883–2890. [doi: 10.1109/CEC.2015.7257247]
- [29] Pan L, He C, Tian Y, *et al.* A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2019, 23(1): 74–88. [doi: 10.1109/TEVC.2018.2802784]
- [30] Zhang Q, Zhou A, Jin Y. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans. on Evolutionary Computation*, 2008, 12(1): 41–63.
- [31] Bandaru S, Ng AHC, Deb K. On the performance of classification algorithms for learning Pareto-dominance relations. In: Proc. of the 2014 IEEE Congress on Evolutionary Computation (CEC). Beijing, 2014. 1139–1146. [doi: 10.1109/CEC.2014.6900641]
- [32] Lu X, Tang K. Classification- and regression-assisted differential evolution for computationally expensive problems. *Journal of Computer Science & Technology*, 2012, 27(5): 1024–1034.
- [33] Breiman L. Random forests. *Machine Learning*, 2001, 45(1): 5–32.
- [34] Rasmussen CE. Gaussian processes in machine learning. In: Bousquet O, von Luxburg U, Rätsch G, eds. Proc. of the Advanced Lectures on Machine Learning (ML 2003). LNCS 3176, Berlin, Heidelberg: Springer, 2004. 64–71.
- [35] Das I, Dennis JE. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 1998, 8(3): 631–657.
- [36] Morris, Max D. The design and analysis of computer experiments. *Journal of the American Statistical Association*, 2004, 99(468): 1203–1204.
- [37] Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 2000, 42(1): 55–61.
- [38] Deb K, Agrawal RB. Simulated binary crossover for continuous search space. *Complex Systems*, 2000, 9(3): 115–148.



- [39] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design. *Computer Science and Informatics*, 1996, 26(4): 30–45.
- [40] Liu B, Zhang Q, Gielen GGE. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans. on Evolutionary Computation*, 2014, 18(2): 180–192.
- [41] Deb K, Thiele L, Laumanns M, *et al.* Scalable multi-objective optimization test problems. In: *Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002)*. Honolulu: IEEE, 2002. 825–830. [doi: 10.1109/CEC.2002.1007032]
- [42] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation*, 1999, 3(4): 0–271.
- [43] Cheng R, Jin Y, Olhofer M, *et al.* A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016, 20(5): 773–791.

#### 附中文参考文献:

- [4] 叶苗, 王宇平. 一种新的容忍恶意节点攻击的无线传感器网络安全定位方法. *计算机学报*, 2013, 36(3): 532–545.
- [5] 公茂果, 焦李成, 杨咚咚, 马文萍. 进化多目标优化算法研究. *软件学报*, 2009, 20(2): 271–289. <http://www.jos.org.cn/1000-9825/3483.htm> [doi: 10.3724/SP.J.1001.2009.03483]



孙哲人(1992—), 男, 硕士生, 主要研究领域为智能计算.



陈志远(1996—), 男, 硕士生, 主要研究领域为智能计算, RNA 二级机构预测.



黄玉划(1975—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为智能计算, 信息安全.