

# 控制与决策

Control and Decision

## 基于分解和聚类的昂贵高维多目标进化算法

徐三水, 李军华, 李凌, 黎明

引用本文:

徐三水, 李军华, 李凌, 黎明. 基于分解和聚类的昂贵高维多目标进化算法[J]. *控制与决策*, 2024, 39(2): 440–448.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0744>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于向量角分解的高维多目标进化算法

Many-objective evolutionary algorithm based on vector angle decomposition

控制与决策. 2021, 36(3): 761–768 <https://doi.org/10.13195/j.kzyjc.2019.0925>

#### 基于弱关联的自适应高维多目标进化算法

A weak association-based adaptive evolutionary algorithm for many-objective optimization

控制与决策. 2021, 36(8): 1804–1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

#### 一种求解约束多目标问题的协作进化算法

A collaborative evolutionary algorithm for solving constrained multi-objective problems

控制与决策. 2021, 36(11): 2656–2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

#### 基于多种群分解预测的动态多目标引力搜索算法

Dynamic multi-objective gravitational searching algorithm based on multi-population decomposition prediction

控制与决策. 2021, 36(12): 2910–2918 <https://doi.org/10.13195/j.kzyjc.2020.1002>

#### 基于R2指标和目标空间分解的高维多目标粒子群优化算法

R2 indicator and objective space partition based many-objective particle swarm optimizer

控制与决策. 2021, 36(9): 2085–2094 <https://doi.org/10.13195/j.kzyjc.2020.0113>

# 基于分解和聚类的昂贵高维多目标进化算法

徐三水, 李军华<sup>†</sup>, 李凌, 黎明

(南昌航空大学 江西省图像处理与模式识别重点实验室, 南昌 330063)

**摘要:** 使用进化算法解决昂贵高维多目标优化问题时, 因目标维数较高, 导致收敛性和多样性平衡困难, 并且消耗成本过高, 使得计算资源有限时难以收敛. 为此, 提出一种基于分解和聚类的昂贵高维多目标进化算法 (DC-EMEA), 使用克里金模型近似目标函数, 减少昂贵函数的评价次数. 在优化器对模型的最优解集搜索时, 借助参考向量分解目标空间, 有利于收敛性和多样性的平衡, 同时采取两轮选择的方式, 保证后代种群规模与父代相同, 为填充准则选择真实评价的个体时, 提供更多选择, 提升搜索效率. 同时, 提出一种自适应填充准则, 首先使用  $K$  均值算法将种群划分为  $k$  个子种群. 通过划分邻域, 将子种群自适应地分成不同类型, 根据子种群的类型选择个体, 提升计算资源的利用率. 在选择个体时, 侧重于对收敛性压力的维持, 提升收敛速度. 将选出的个体用于更新模型和档案. 实验结果表明, DC-EMEA 能够很好地平衡收敛性和多样性, 同时具有较强的收敛能力.

**关键词:** 克里金模型; 高维昂贵优化; 填充准则; 参考向量; 进化算法; 聚类算法

中图分类号: TP18 文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0744

引用格式: 徐三水, 李军华, 李凌, 等. 基于分解和聚类的昂贵高维多目标进化算法 [J]. 控制与决策, 2024, 39(2): 440-448.

## Decomposition and cluster based expensive many-objective evolutionary algorithm

XU San-shui, LI Jun-hua<sup>†</sup>, LI Ling, LI Ming

(Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China)

**Abstract:** When using evolutionary algorithms to solve expensive many-objective optimization problems, the many-objective leads to difficulties in balancing convergence and diversity and makes convergence difficult when computational resources are limited due to high consumption costs. Therefore, this paper proposes a decomposition and cluster based expensive many-objective evolutionary algorithm (DC-EMEA), which uses the Kriging model to approximate the objective function and reduces the number of evaluations of real expensive functions. When the optimizer searches for the optimal solution set of the model, the objective space is decomposed with the help of the reference vector, which is conducive to the balance of convergence and diversity. At the same time, two rounds of selection are adopted to ensure that the offspring population size is the same as that of the parents, providing more options for the selection of individuals for real evaluation by the infill criterion and improving the search efficiency. Meanwhile, an adaptive infill criterion is proposed to firstly divide the population into  $k$  subpopulations using the  $K$ -means algorithm. Then, by dividing the neighborhood, the subpopulations are adaptively divided into different types, and individuals are selected according to the types of subpopulations to improve the utilization of computational resources. In the selection of individuals, the focus is on the maintenance of convergence pressure to improve the convergence speed. Finally, the selected individuals are used to update the model and the archive. The experiments show that the DC-EMEA can balance convergence and diversity well and has a strong convergence ability.

**Keywords:** Kriging model; expensive many-objective optimization; infill criterion; reference vector; evolutionary algorithm; cluster algorithm

收稿日期: 2022-05-02; 录用日期: 2022-09-20.

基金项目: 国家自然科学基金项目 (62066031, 61866025, 61866026); 江西省研究生创新基金项目 (YC2021-S678).

责任编辑: 巩敦卫.

<sup>†</sup>通讯作者. E-mail: jhlee126@126.com.

\*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

## 0 引言

许多优化问题包含多个相互冲突的目标,这类问题称为多目标优化问题(multi-objective optimization problems, MOPs)<sup>[1]</sup>. 进化算法<sup>[2]</sup>被广泛应用于多目标优化问题,但传统的进化算法通常需要上万次的函数评价才能获得较好的解集. 然而,在一些现实优化问题中,函数评价需要昂贵的物理实验或者复杂的数值模拟,消耗大量的成本,这类优化问题称为昂贵优化问题(expensive optimization problems, EOPs)<sup>[3]</sup>. 例如空气动力学形状设计问题<sup>[4]</sup>,单次数值模拟需要耗费几分钟到几小时,如果使用传统进化算法进行一次完整的优化设计,则需要耗费数月的时间,这在实际优化问题中难以接受. 有学者提出使用机器学习<sup>[5]</sup>的方法,建立消耗资源少的代理模型,替代真实函数评价,将真实昂贵的函数评价次数降低到几百次,提高进化算法的效率.

代理模型辅助的进化算法简称为 SAEA. 在 SAEAs 中,常用的代理模型<sup>[6]</sup>有:克里金模型、径向基函数模型和神经网络模型等. 其中,克里金模型因其拟合精度高,且能够给出预测值的不确定度而被广泛应用于 SAEAs.

随着科技的发展,工业设计优化问题变得越来越复杂,需要同时优化的目标数往往超过3个,并且评价过程消耗大量成本,这类问题称为昂贵高维多目标优化问题<sup>[7]</sup>(many-objective EOPs, MaEOPs). 为了提高进化算法在处理 MaEOPs 时的性能,学者们相继提出了许多优秀算法,根据代理模型在进化算法中的用途不同,可以将这些算法大致分为3类:1) 基于分类的 SAEAs,利用代理模型比较候选解的优劣,过滤掉质量差的解,例如 CSEA (classification based surrogate-assisted evolutionary algorithm)<sup>[7]</sup>、CPS-MOEA (classification based preselection multi-objective evolutionary algorithm)<sup>[8]</sup>和 REMO (relation selection based evolutionary multi-objective optimization algorithm)<sup>[9]</sup>等;2) 基于回归的 SAEAs,将代理模型用来近似目标函数,减少真实函数计算次数,此类经典算法有 ParEGO (Pareto based efficient global optimization)<sup>[10]</sup>、MOEA/D-EGO (efficient global optimization assisted multi objective evolutionary based on decomposition)<sup>[11]</sup>、K-RVEA (Kriging-assisted reference vector-guide evolutionary algorithm)<sup>[12]</sup>、KTA 2 (Kriging-assisted two-archive evolutionary algorithm)<sup>[13]</sup>和 EDN-ARMOEA (efficient dropout neural network assisted adaptive reference points based multi-objective evolutionary algorithm)<sup>[14]</sup>等;3) 分类

与回归模型结合的 SAEAs,这类算法结合了分类模型和回归模型的优点,但模型的管理更加复杂,经典算法有 CRADE (classification-and regression-assisted differential evolutionary algorithm)<sup>[15]</sup>和 C-M-EA (hybrid algorithm for computationally expensive many-objective optimization)<sup>[16]</sup>等.

尽管已有许多优秀算法可用来解决 MaEOPs,但仍存在许多挑战:1) 环境选择的设计. 高维目标空间收敛性和多样性平衡困难,传统 Pareto 支配方法的选择压力不足,使用参考向量对高维目标空间进行分解能够降低问题求解的难度,但传统的基于参考向量分解的算法,如 RVEA (reference vector-guide evolutionary algorithm)<sup>[17]</sup>,在其关联操作中,会产生空的参考向量,导致后代种群规模不足,降低算法的搜索效率. 2) 填充准则的设计. 填充准则用来确定哪些个体能够使用真实函数评价,选出的个体不仅可以用来训练模型,同时还作为最终解集的候选解,因此,填充准则的设计直接影响算法的性能. 同时考虑收敛性和不确定度的方法,未能根据需求状态不同选择个体,当计算资源有限时,无法取得较好解;而将收敛性和不确定度分开考虑的算法中,在选择不确定度大的个体时,忽略了对收敛性的维持,不利于算法收敛.

为了更好地解决这些问题,本文提出一种基于分解和聚类的昂贵高维多目标进化算法 (decomposition and cluster based expensive many-objective evolutionary algorithm, DC-EMEA). 主要贡献总结如下:1) 提出两轮环境选择策略,平衡种群的收敛性和多样性,提升搜索效率. 在两轮环境选择策略中,借助参考向量对目标空间分解,当选择出的后代种群规模不满足要求时,利用未关联个体的参考向量进行第2轮选择,提升后代个体数量,为填充准则提供更多的选择,提升算法的搜索效率. 2) 提出自适应填充准则,根据需求状态不同选择个体,提升计算资源的利用率,加快收敛速度. 具体的,首先使用  $K$  均值聚类算法将模型评价的个体从决策空间中划分成  $k$  个子种群,初步保证多样性;然后,通过邻域划分,将真实评价的个体划分到子种群的邻域中,根据子种群与其邻域个体的关系,将子种群自适应地分成收敛性提升种群和多样性提升种群. 根据子种群类型不同采用相应的策略选择个体.

## 1 基础知识

### 1.1 多目标优化问题

一个具有  $n$  维决策变量、 $M$  个目标函数的多目标优化问题可以表示如下:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_M(x))^T;$$

subject to  $x \in X$ . (1)

其中:  $x = (x_1, x_2, \dots, x_n)^T \in X$  为  $n$  维决策向量,  $F(x) = (f_1(x), f_2(x), \dots, f_M(x))^T$  为  $M$  维目标向量. 设  $U = (u_1, u_2, \dots, u_n)^T$  和  $Y = (y_1, y_2, \dots, y_n)^T$  是目标空间中的两个向量, 当且仅当  $u_i \leq y_i (i = 1, 2, \dots, M)$  且存在  $j = 1, 2, \dots, M$  使  $u_j < y_j$  成立, 称  $U$  支配  $Y$ . 在多目标优化问题中, 如果一个解不被其他解支配, 则称这个解为 Pareto 最优解, 由于多目标优化问题的各目标之间相互冲突, Pareto 最优解往往不止一个, 而是一组解, 这组解在决策空间中被称为 Pareto 解集 (Pareto set, PS), PS 在目标空间中的投影称为 Pareto 前沿 (Pareto front, PF).

## 1.2 克里金模型

克里金模型对非线性问题的拟合精度较高<sup>[6]</sup>, 因此本文选择克里金模型作为代理模型. 假设训练模型的个体数量为  $N_l$ , 决策变量维数为  $n$ , 克里金模型建模过程可以表示为

$$y(x) = \mu(x) + \varepsilon(x). \quad (2)$$

其中:  $\mu(x)$  为回归模型的预测值;  $\varepsilon(x)$  为一个满足均值为 0、方差为  $\sigma^2$  的随机过程. 对于任意两个变量  $x^i, x^j$ , 其随机过程的协方差表示为

$$\text{Cov}[\varepsilon(x^i), \varepsilon(x^j)] = \sigma^2 \mathbf{R}([R(x^i, x^j)]). \quad (3)$$

其中:  $\mathbf{R}$  是大小为  $N_l \times N_l$  的相关矩阵, 其表达式为

$$\mathbf{R} = \begin{bmatrix} R(x^1, x^1) & R(x^1, x^2) & \dots & R(x^1, x^{N_l}) \\ R(x^2, x^1) & R(x^2, x^2) & \dots & R(x^2, x^{N_l}) \\ \vdots & \vdots & \ddots & \vdots \\ R(x^{N_l}, x^1) & R(x^{N_l}, x^2) & \dots & R(x^{N_l}, x^{N_l}) \end{bmatrix}. \quad (4)$$

$R(x^i, x^j)$  为相关函数, 通常选用如下函数:

$$R(x^i, x^j) = \exp\left(-\sum_{k=1}^n \theta_k |x_k^i - x_k^j|^2\right). \quad (5)$$

其中:  $\theta$  为超参数,  $\theta$  由最大似然估计方法计算得出, 计算方法为

$$\psi(\theta) = -\frac{1}{2}(N_l \ln(\sigma^2) + \ln|\mathbf{R}|). \quad (6)$$

为得出预测值  $\hat{y}$ , 还需确定方差  $\sigma^2$ , 其值为

$$\sigma^2 = \frac{1}{N_l} (y - \mathbf{1}\mu(\hat{x}))^T \mathbf{R}^{-1} (y - \mathbf{1}\mu(\hat{x})), \quad (7)$$

其中  $\mathbf{1}$  表示大小为  $n \times 1$  的全 1 向量. 输入值  $\hat{x}$  与  $N_l$  个训练样本的相关函数向量记为  $r^T(\hat{x})$ , 表达式为

$$r^T(\hat{x}) = [R(\hat{x}, x^1), R(\hat{x}, x^2), \dots, R(\hat{x}, x^{N_l})]^T. \quad (8)$$

由此可以得出模型对于输入值  $\hat{x}$  的预测值  $\hat{y}$  和不确定度  $s(\hat{x})$ , 表示如下:

$$\hat{y} = \mu(\hat{x}) + r^T(\hat{x}) \mathbf{R}^{-1} (y - \mathbf{1}\mu(\hat{x})), \quad (9)$$

$$s(\hat{x}) = \sqrt{\sigma^2 \left[1 - r^T \mathbf{R}^{-1} r + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} r)^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}}\right]}. \quad (10)$$

克里金模型的不确定度指的是: 在使用克里金模型对输入值  $\hat{x}$  进行预测时, 对给出的  $\hat{y}$  值不能确定的程度. 从  $s(\hat{x})$  的计算公式可以得出, 对于输入值  $\hat{x}$ , 其距离真实训练个体越远, 克里金模型对给出的  $\hat{y}$  值不确定的程度越高.

## 2 基于分解和聚类的昂贵高维多目标进化算法

### 2.1 算法框架

算法运行过程中存在两类个体, 分别是模型评价和真实函数评价的个体. 为方便管理, 使用  $\text{Arc}_1$  保存模型评价的个体,  $\text{Arc}_2$  保存真实函数评价的个体. 算法 1 为本文的算法框架.

算法 1: 算法框架

---

输入:  $N$  (种群规模和参考向量数目)、 $\text{MaxFE}$  (真实函数最大评价次数)、 $\omega_{\max}$  (模型最大搜索代数) 和  $k$  (填充准则选择的样本个数);

输出: 最终种群.

- 1) 使用拉丁超立方取样生成  $N$  个个体, 并使用真实目标函数计算其函数值, 将个体加入到  $\text{Arc}_1$  和  $\text{Arc}_2$  中;
- 2) 生成  $N$  个均匀分布的参考向量;
- 3)  $\text{FE} = N/\text{FE}$  为当前真实函数评价次数;
- 4) 使用  $\text{Arc}_2$  中的个体为每一维目标建立初始模型;
- 5) while  $\text{FE} \leq \text{MaxFE}$ ;
- 6)  $\omega = 1$ ;
- 7) while  $\omega \leq \omega_{\max}$ ;
- 8) 对  $\text{Arc}_1$  中的个体交叉变异生成子代种群, 合并  $\text{Arc}_1$  和子代种群, 使用模型代替真实函数计算子代种群个体的函数值;
- 9) 从合并种群选择  $N$  个个体, 作为新的  $\text{Arc}_1$ ;
- 10)  $\omega = \omega + 1$ ;
- 11) end while
- 12) 从  $\text{Arc}_1$  中选择  $k$  个个体, 删除这  $k$  个个体中与  $\text{Arc}_2$  中重复的个体, 剩余  $\text{Remain}$  个个体;
- 13) 使用真实目标函数计算  $\text{Remain}$  个个体的函数值;
- 14)  $\text{FE} = \text{FE} + \text{Remain}$ ,  $\text{Arc}_2 = \text{Arc}_2 \cup \text{Remain}$ ;
- 15) 使用这  $\text{Remain}$  个个体更新模型;
- 16) 从  $\text{Arc}_2$  中选择  $N$  个个体, 作为下一轮的初始种群  $\text{Arc}_1$ /匹配池选择;
- 17) end while
- 18) 将  $\text{Arc}_2$  中的非支配解集作为最终种群输出.

---

如算法 1 所示, 第 1)~第 4) 行为初始化. 首先, 从目标空间中生成  $N$  个均匀分布的参考向量, 再使用拉丁超立方采样<sup>[18]</sup> 生成  $N$  个个体, 使用真实函数评价后, 保存到  $\text{Arc}_1$  和  $\text{Arc}_2$  中, 并用  $\text{Arc}_2$  中的个体为每一维目标建立初始克里金模型. 第 7)~第 11) 行为优化器对模型最优解集的搜索. 在这一部分中, 将  $\text{Arc}_1$  中的个体作为父代种群, 进行模拟二进制交叉<sup>[19]</sup> 和多项式变异<sup>[20]</sup>, 生成子代种群, 使用模型计算子代个体的目标值, 将父代种群与子代种群合并, 对合并种群进行选择, 作为新的  $\text{Arc}_1$ . 第 12)~第 15) 行为填充准则选择个体, 用于更新模型和  $\text{Arc}_2$ , 对模型的最优解集搜索了  $\omega_{\max}$  代之后, 从当前模型的最优解集

Arc<sub>1</sub> 中选取  $k$  个最有潜力的个体, 若这  $k$  个个体中存在与 Arc<sub>2</sub> 中重复的个体, 则删除这些重复的个体, 将剩下的个体使用真实函数评价后, 加入到 Arc<sub>2</sub> 中, 并对模型进行更新. 因为模型估值存在误差, 并且随着维数的增加, 累积误差增大, 所以在模型更新后的下一轮搜索过程中, 从 Arc<sub>2</sub> 中选择  $N$  个真实函数评价的个体作为新的初始种群. 重复上述过程, 直到满足终止条件.

### 2.2 两轮环境选择

在昂贵高维多目标优化问题中, 使用参考向量对问题进行分解, 有助于收敛性和多样性的平衡. 然而, 传统的使用参考向量分解的方法, 选择出的后代种群规模通常小于父代, 不利于算法搜索. 针对这种情况, 本文采取两轮选择的方式, 进一步满足后代种群规模. 算法2给出了两轮环境选择的伪代码.

算法2: 两轮环境选择

```

输入: Arc;
输出: 环境选择后的 Arc.
1) 对种群中的个体进行目标值转换;
2) 将 Arc 中的个体关联到离其最近的参考向量;
3) 第1轮选择, 对已关联个体的参考向量, 从其关联的个体中
   选择离原点欧氏距离最小的个体, 保存到临时种群 temp 中;
4) if |temp| < N then
5) 第2轮选择, 对于未关联个体的参考向量, 从 Arc 剩下的
   个体中选择与其 JDF 值最小的个体, 加入到 temp 中;
6) end if
7) 将 temp 作为环境选择后的 Arc 输出.
    
```

如算法2所示, 在两轮环境选择的初始阶段, 对种群中的个体进行目标值转换, 记第  $t$  代个体  $p$  的函数值为  $F_t(p) = (f_{t,1}(p), f_{t,2}(p), \dots, f_{t,M}(p))^T$ , 目标值转换公式如下:

$$F_t'(p) = F_t(p) - Z_t^{\min}. \quad (11)$$

其中:  $Z_t^{\min} = (z_{t,1}^{\min}, z_{t,2}^{\min}, \dots, z_{t,M}^{\min})^T$  为种群中每一维目标函数的最小值组成的向量, 称为理想点. 经过目标值转换的好处是: 个体全部移动到第一象限, 极点移动到坐标轴上, 个体之间的位置关系不会发生改变, 增大了参考向量的覆盖范围<sup>[17]</sup>. 对合并种群进行选择时, 首先, 将个体与参考向量关联, 关联的方法为: 计算个体与原点之间的连线到各参考向量的夹角, 选择夹角最小的参考向量关联, 夹角计算如下:

$$\theta_{p,i} = \arccos \frac{F_t'(p) \cdot V_{t,i}}{\|F_t'(p)\| \cdot \|V_{t,i}\|}. \quad (12)$$

完成关联操作后, 从参考向量关联的个体中选择离原点欧氏距离最近的个体进入下一代. 为了更直观地展示关联操作, 在图1中, 以两目标问题为例, 假设初始种群规模为5, 将合并种群中的10个个体与5

个参考向量进行关联.

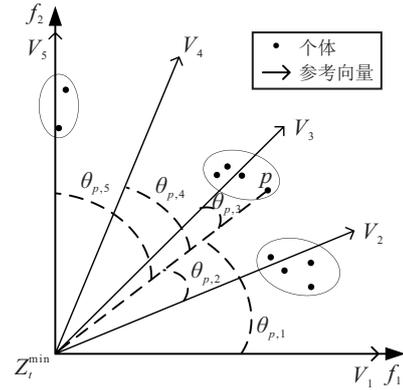


图1 个体与参考向量的关联操作

对于图1中的个体  $p$ , 其与原点之间的连线到参考向量  $V_3$  的夹角最小, 因此将  $p$  关联到  $V_3$  上. 图1中仅有参考向量  $V_2$ 、 $V_3$  和  $V_5$  关联了个体, 分别从其关联的个体中选择与原点欧氏距离最小的个体, 此时选出的后代只有3个个体, 小于初始种群规模, 需要进行第2轮选择. 在第2轮选择中, 利用未关联个体的参考向量选择剩下的个体.

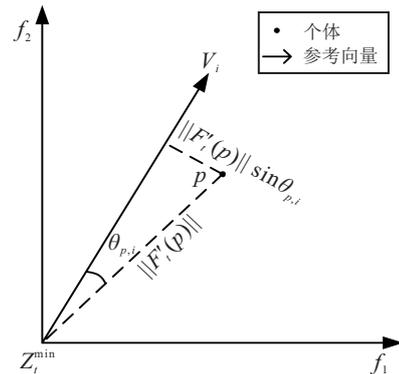


图2 联合距离函数计算

如图2所示, 通过计算个体到参考向量的距离, 结合个体到原点的欧氏距离, 作为联合距离函数 (joint distance function, JDF). JDF 计算公式如下:

$$JDF_{p,i} = \|F_t'(p)\| + \|F_t'(p)\| \cdot \sin \theta_{p,i}. \quad (13)$$

在 JDF 中, 通过个体与参考向量之间的距离提供多样性压力、个体与原点之间的距离提供收敛压力, 能够有效地平衡收敛性和多样性, JDF 值越小的个体质量越好. 对于图1中的参考向量  $V_1$  和  $V_4$ , 分别选择与剩下个体中 JDF 值最小的个体进入下一代, 使得后代种群满足规模要求, 提升搜索效率.

### 2.3 自适应填充准则

昂贵高维多目标优化问题的真实函数评价次数较少, 填充准则选择个体时需要考虑收敛性与多样性的平衡, 在计算资源有限的情况下, 根据需求的不同选择个体, 能够高效地利用计算资源. 本文提出一种

自适应填充准则,算法3给出伪代码.

算法3: 自适应填充准则

```

输入: Arc1, Arc2;
输出: k个个体.
1) 将 Arc1 和 Arc2 中所有个体在每个目标上的最小值作为理想点,使用
   K 均值聚类算法,将 Arc1 分成 k 个子种群,并得到聚类中心;
2) 将 Arc2 中的个体分配到离其最近的聚类中心,与子种群具有相同聚类
   中心的个体作为子种群的邻域个体;
3) for i = 1 : k
4)   计算第 i 个子种群及其邻域个体到理想点的距离;
5)   if 子种群中存在个体到理想点的距离小于任意邻域个体到理想点的
      距离 then
6)     第 i 个子种群为收敛性提升种群;
7)   else
8)     第 i 个子种群为多样性提升种群;
9)   end if
10)  if 第 i 个子种群为收敛性提升种群 then
11)   选择子种群中收敛度最小的个体,加入到临时档案 temp 中;
12) else
13)   计算子种群中个体的不确定度和收敛度,按照不确定度降序排列,
      收敛度升序排列,分别编号 1, 2, ..., |Gi|, 作为两个独立目标;
14)   将个体按两目标进行非支配排序,从非支配个体中,选择收敛度序
      号最小的个体,加入到临时档案 temp 中;
15) end if
16) end for
17) 输出 temp 中的 k 个个体.

```

算法3主要分为3个部分.

第1部分为初始化(第1)行和第2)行). 因为填充准则选择的个体数仅占模型评价个体的一部分,为使选出的个体在模型评价的个体中分布均匀,使用K均值聚类的方法,将 Arc<sub>1</sub> 中的个体从决策空间中分成 k 个子种群  $G_1, G_2, \dots, G_k$ . 同时,得到每个子种群的聚类中心  $Cen_1, Cen_2, \dots, Cen_k$ , 并将 Arc<sub>1</sub> 和 Arc<sub>2</sub> 中所有个体在每一维目标上的最小值作为理想点  $z^*$ .

第2部分为自适应判断子种群类型(第5)~第9)行). 判断子种群类型的方法为: 首先,划分邻域. 划分邻域的方法为: 计算 Arc<sub>2</sub> 中真实评价的个体到聚类中心  $Cen_1, Cen_2, \dots, Cen_k$  的距离,将 Arc<sub>2</sub> 中的个体分配到最近的聚类中心,被分配到  $Cen_i$  的真实评价个体作为第 i 个子种群的邻域个体. 然后,计算子种群及其邻域个体到理想点的距离,若子种群中存在个体到理想点的距离,比其任意邻域个体到理想点的距离都小,则该子种群为收敛性提升种群,否则为多样性提升种群.

第3部分为根据子种群类型选择相应的个体(第11)~第15)行). 虽然克里金模型提供了个体的不确定度量方法,然而普适性不高,不适用于其他模型,如径向基模型和多项式回归模型等. 为此提出一种通用的不确定度量方法,将个体与训练模型的样本集在决策空间中的欧氏距离最小值作为该个体的不确定度. 个体的不确定度越大,表明该个体附近的训练样本越稀疏,对多样性的提升越明显. 个体  $x$  的不确定度计算公式如下:

$$\text{Uncertainty}(x) = \min_{y \in \text{Arc}_2} \|x - y\|. \quad (14)$$

另外,将目标空间中个体与理想点之间的距离作为个体的收敛度. 收敛度越小的个体收敛性越强. 个体  $x$  的收敛度计算公式如下:

$$\text{Convergence}(x) = \|F(x) - z^*\|. \quad (15)$$

若子种群为收敛性提升种群,则从子种群中选择收敛度最小的个体进入下一代,增强种群的收敛性. 若子种群为多样性提升种群,则在选择不确定度大的个体时,为了维持选择压力,同时结合收敛度来选择个体. 因为不确定度是从决策空间中计算的,而收敛度是从目标空间中计算的,决策空间与目标空间的维度不同,所以将不确定度和收敛度作为两个单独的目标进行优化. 由于不确定度越大的个体,多样性越强,而收敛度越小的个体,收敛性越强,不确定度大和收敛度小的个体被优先选择. 为使两个目标均为最小化问题,首先,将个体的不确定度值从大到小排序,编号为  $1, 2, \dots, |G_i|$ , 作为第1个目标; 然后,将个体的收敛度值从小到大排序,编号为  $1, 2, \dots, |G_i|$ , 作为第2个目标. 编号完成后,按照个体编号进行非支配排序,得到的非支配个体为收敛性和多样性最佳的个体,为维持收敛性压力,从这些非支配个体中选择收敛度最小的个体用于真实评价.

## 2.4 时间复杂度分析

在 DC-EMEA 中,算法的时间成本主要来源于代理模型的建立、模型最优解集的搜索和填充准则选择个体. 将 DC-EMEA 的种群规模记为  $N$ , 决策向量维数记为  $D$ , 目标维数记为  $M$ , 填充准则选择个体数记为  $k$ . 因为在 DC-EMEA 中对每一维目标都建立了克里金模型,所以建模的时间复杂度为  $O(MN^3)$ . 在模型最优解集搜索过程中,将个体关联到参考向量上的时间复杂度为  $O(N^2)$ . 对合并种群进行第1轮选择的时间复杂度为  $O(N)$ . 第2轮选择的最坏情况是第1轮仅选出一个个体,此时的时间复杂度为  $O(N^2)$ . 在填充准则中,聚类的时间复杂度为  $O(kDN)$ , 判断子种群类型的时间复杂度为  $O(N^2)$ . 从子种群中选择个体时,最坏情况是子种群都为多样性提升种群,选择个体的时间复杂度为  $O(N^2)$ . 在 DC-EMEA 中,  $k, D, M$  的值远远小于  $N$ , 因此算法的时间复杂度为  $O(MN^3)$ .

## 3 实验与分析

在本节中,设计了两组实验,第1组实验将两轮环境选择与只进行单轮环境选择的算法进行对比,用于验证两轮环境选择的有效性. 第2组实验将 DC-EMEA 与 CSEA<sup>[7]</sup>、MOEA/D-EGO<sup>[11]</sup>、K-RVEA<sup>[12]</sup>、

KTA 2<sup>[13]</sup> 以及 EDN-ARMOEA<sup>[14]</sup> 进行对比. 其中: CSEA 使用单层前馈神经网络作为分类代理模型; MOEA/EGO、K-RVEA 和 KTA 2 使用克里金模型作为回归模型; EDN-ARMOEA 使用 Dropout 神经网络作为回归模型. 本文所有算法均在 PlatEMO<sup>[21]</sup> 平台中运行.

### 3.1 实验设置

实验选用 DTLZ 测试集<sup>[22]</sup> 作为基准测试问题, 分别测试算法在目标数为 3、4、6、8 和 10, 决策变量维数为 10 维情况下的性能. 使用反世代距离 (inverted generational distance, IGD) 指标<sup>[23]</sup> 比较算法的性能.

1) 种群规模: 由于 DC-EMEA 和 K-RVEA 的种群规模取决于参考向量数量, 而参考向量由单纯型网格设计<sup>[24]</sup> 生成, 为了保证公平性, 所有算法种群规模设置相同, 分别设置为 91、84、112、128 和 110.

2) 终止条件: 本文使用函数的最大评价次数作为终止条件. 第 1 组实验不需要使用真实目标函数计算, 最大评价次数设置为 10 000; 第 2 组实验需要使用真实目标函数计算, 消耗大量时间成本, 因此最大评价次数设置为 300.

3) 交叉和变异算子设置: 采用模拟二进制交叉和多项式变异生成子代种群, 交叉算子和变异算子均设置为 20, 交叉概率设置为 1, 变异概率设置为 1/*D*, 其中 *D* 为决策变量的维数.

4) 其他参数设置: 本文模型最大使用代数  $\omega_{max}$  和填充准则选取的个体数 *k* 的设置参考 K-RVEA, 分别为 20 和 5, 其他算法参数设置与其原文相同.

### 3.2 结果分析

1) 第 1 组实验: 将本文两轮环境选择算法 (two

round environment selection, TRES), 在经过第 1 轮环境选择后, 不再进行第 2 轮环境选择的算法, 作为单轮环境选择算法 (single round environment selection, SRES). 将 TRES 与 SRES 进行对比, 用于验证两轮环境选择的有效性. 图 3 为两算法在 3 维 DTLZ 5 上获得的非支配解集. 表 1 为两算法在 DTLZ 测试问题上获得的 IGD 值统计结果, 最好的结果显示为黑体, 其中 ‘+’, ‘-’ 和 ‘=’ 分别表示 SRES 与 TRES 相比更好、更差和相似.

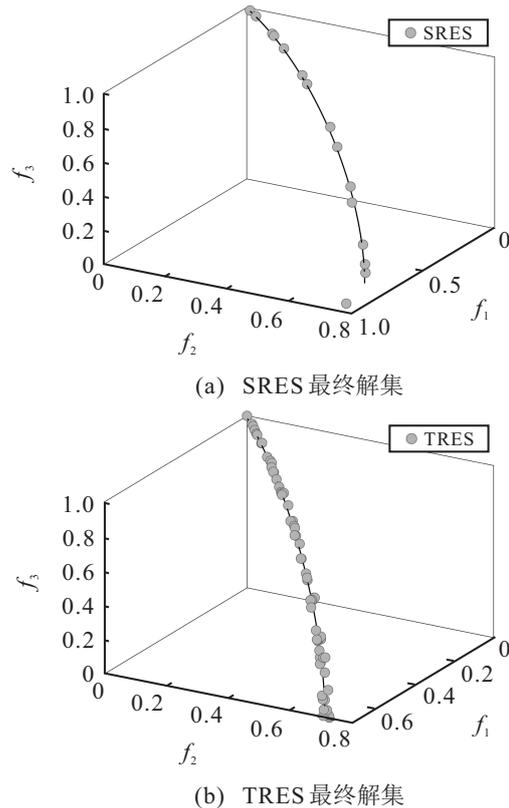


图 3 SRES 和 TRES 在 3 维 DTLZ 5 上的非支配解集

表 1 SRES 和 TRES 在 DTLZ 测试问题上获得的 IGD 值统计结果 (均值)

| problem       | <i>M</i> | SRES                | TRES              | problem       | <i>M</i> | SRES                | TRES              |
|---------------|----------|---------------------|-------------------|---------------|----------|---------------------|-------------------|
| DTLZ 1        | 3        | 1.833 0e+0 =        | <b>1.794 4e+0</b> | DTLZ 5        | 3        | 5.890 2e-2 -        | <b>3.908 1e-2</b> |
|               | 4        | 9.229 7e-1 =        | <b>8.289 7e-1</b> |               | 4        | 4.469 1e-1 -        | <b>2.070 2e-1</b> |
|               | 6        | 3.256 3e-1 -        | <b>2.450 2e-1</b> |               | 6        | 2.013 5e-1 -        | <b>1.922 7e-1</b> |
|               | 8        | <b>1.394 2e-1</b> + | 1.447 4e-1        |               | 8        | 6.817 7e-1 -        | <b>1.664 9e-1</b> |
|               | 10       | 1.679 3e-1 -        | <b>1.620 1e-1</b> |               | 10       | 5.664 6e-1 -        | <b>1.922 1e-1</b> |
| DTLZ 2        | 3        | 6.040 4e-2 =        | <b>6.018 9e-2</b> | DTLZ 6        | 3        | 5.969 9e-2 -        | <b>4.126 6e-2</b> |
|               | 4        | 1.436 3e-1 =        | <b>1.407 4e-1</b> |               | 4        | 1.455 0e-1 =        | <b>1.283 7e-1</b> |
|               | 6        | 2.810 5e-1 =        | <b>2.736 9e-1</b> |               | 6        | <b>1.495 6e-1</b> = | 1.878 2e-1        |
|               | 8        | 4.126 3e-1 -        | <b>3.730 8e-1</b> |               | 8        | <b>1.750 4e-1</b> = | 1.881 5e-1        |
|               | 10       | 5.787 0e-1 -        | <b>4.680 1e-1</b> |               | 10       | 2.578 1e-1 -        | <b>1.816 8e-1</b> |
| DTLZ 3        | 3        | 6.520 1e+0 =        | <b>5.608 6e+0</b> | DTLZ 7        | 3        | 1.140 8e-1 =        | <b>1.098 3e-1</b> |
|               | 4        | 3.253 0e+0 =        | <b>3.132 9e+0</b> |               | 4        | <b>2.955 3e-1</b> + | 3.405 3e-1        |
|               | 6        | <b>1.144 0e+0</b> = | 1.162 7e+0        |               | 6        | 6.843 0e-1 =        | <b>6.658 1e-1</b> |
|               | 8        | 4.856 5e-1 -        | <b>4.114 8e-1</b> |               | 8        | <b>1.154 6e+0</b> + | 1.457 2e+0        |
|               | 10       | 6.236 6e-1 -        | <b>4.825 3e-1</b> |               | 10       | <b>2.228 4e+0</b> + | 2.791 7e+0        |
| DTLZ 4        | 3        | 6.063 6e-2 =        | <b>6.062 7e-2</b> |               |          |                     |                   |
|               | 4        | 1.651 9e-1 =        | <b>1.443 5e-1</b> |               |          |                     |                   |
|               | 6        | 2.814 3e-1 =        | <b>2.710 1e-1</b> |               |          |                     |                   |
|               | 8        | 3.723 9e-1 =        | <b>3.629 9e-1</b> |               |          |                     |                   |
|               | 10       | 4.916 2e-1 -        | <b>4.755 3e-1</b> |               |          |                     |                   |
| +/-/= 4/13/18 |          |                     |                   | +/-/= 4/13/18 |          |                     |                   |

从图3可以看出, TRES能得到更多的非支配解且收敛性更强. 这些实验验证了两轮环境选择的有效性.

从表1中IGD值对比情况来看, TRES在大多数测试实例上性能更优, 仅在8目标的DTLZ1与4、8和10目标的DTLZ7测试实例上性能差于SRES.

2) 第2组实验: 将DC-EMEA与K-RVEA、CESA、EDN-ARMOEA、KTA2和MOEA/D-EGO进行对比, 表2为6个算法在DLTZ测试问题上IGD值的统计结果, 最好的结果标为黑体, 其中‘+’ ‘-’和‘=’分别表示其他算法获得的结果与DC-EMEA相比更好、更差和相似.

表2 6个算法在DLTZ测试问题上获得的IGD值统计结果(均值)

| problem | M  | K-RVEA              | CSEA                | EDN-ARMOEA          | KTA 2               | MOEA/D-EGO   | DC-EMEA         |
|---------|----|---------------------|---------------------|---------------------|---------------------|--------------|-----------------|
| DTLZ 1  | 3  | 8.715 2e+1 -        | 6.030 4e+1 -        | 9.750 0e+1 -        | 4.965 2e+1 -        | 8.512 0e+1 - | <b>3.24e+01</b> |
|         | 4  | 7.088 5e+1 -        | 3.842 5e+1 -        | 7.344 8e+1 -        | 3.541 7e+1 -        | 5.907 6e+1 - | <b>2.28e+01</b> |
|         | 6  | 2.910 4e+1 -        | 1.771 1e+1 -        | 3.572 7e+1 -        | 1.783 7e+1 -        | 3.255 7e+1 - | <b>1.37e+01</b> |
|         | 8  | 1.066 2e+1 -        | 5.595 1e+0 =        | 1.083 7e+1 -        | <b>5.090 2e+0 =</b> | 1.200 1e+1 - | 5.25e+00        |
|         | 10 | 4.718 4e-1 -        | 3.220 0e-1 +        | 4.336 2e-1 =        | <b>2.983 9e-1 +</b> | 4.268 2e-1 - | 3.72e-01        |
| DTLZ 2  | 3  | 1.863 3e-1 -        | 2.124 4e-1 -        | 3.084 4e-1 -        | <b>6.081 4e-2 +</b> | 3.225 7e-1 - | 6.94e-02        |
|         | 4  | 2.8835 e-1 -        | 2.898 0e-1 -        | 3.739 5e-1 -        | <b>1.393 8e-1 +</b> | 3.742 1e-1 - | 1.86e-01        |
|         | 6  | <b>3.088 4e-1 +</b> | 4.262 9e-1 -        | 4.260 5e-1 -        | 3.280 6e-1 +        | 4.724 8e-1 - | 4.08e-01        |
|         | 8  | 5.008 3e-1 -        | 5.743 3e-1 -        | <b>4.450 4e-1 +</b> | 4.643 6e-1 =        | 5.350 7e-1 - | 4.65e-01        |
|         | 10 | 5.438 8e-1 -        | 6.676 5e-1 -        | <b>4.546 0e-1 +</b> | 4.571 4e-1 +        | 5.232 8e-1 - | 4.99e-01        |
| DTLZ 3  | 3  | 2.495 2e+2 -        | 1.561 6e+2 -        | 3.143 1e+2 -        | 1.342 6e+2 -        | 1.853 8e+2 - | <b>9.27e+01</b> |
|         | 4  | 1.797 3e+2 -        | 1.237 3e+2 -        | 2.527 2e+2 -        | 1.001 5e+2 =        | 1.626 8e+2 - | <b>8.31e+01</b> |
|         | 6  | 1.033 3e+2 -        | 4.925 1e+1 =        | 1.283 4e+2 -        | 5.803 1e+1 =        | 9.296 4e+1 - | <b>4.87e+01</b> |
|         | 8  | 3.366 5e+1 -        | 1.974 3e+1 =        | 3.920 2e+1 -        | <b>1.630 3e+1 =</b> | 3.429 0e+1 - | 1.79e+01        |
|         | 10 | 1.383 2e+0 -        | <b>1.026 3e+0 +</b> | 1.339 7e+0 =        | 1.089 0e+0 +        | 1.338 5e+0 = | 1.25e+00        |
| DTLZ 4  | 3  | 4.194 0e-1 -        | 4.978 5e-1 -        | <b>3.195 7e-1 +</b> | 3.420 2e-1 =        | 6.128 5e-1 - | 3.50e-01        |
|         | 4  | 5.049 3e-1 -        | 4.690 1e-1 =        | <b>3.821 8e-1 +</b> | 4.571 1e-1 =        | 6.743 4e-1 - | 4.40e-01        |
|         | 6  | 5.265 1e-1 +        | 4.898 7e-1 +        | <b>4.839 1e-1 +</b> | 6.005 0e-1 =        | 6.869 8e-1 - | 5.97e-01        |
|         | 8  | 5.611 7e-1 +        | 5.806 1e-1 =        | <b>5.071 6e-1 +</b> | 6.330 7e-1 -        | 6.458 3e-1 - | 5.98e-01        |
|         | 10 | 5.948 7e-1 +        | 6.328 6e-1 =        | <b>5.131 6e-1 +</b> | 6.234 2e-1 =        | 6.457 7e-1 = | 6.39e-01        |
| DTLZ 5  | 3  | 1.098 9e-1 -        | 1.049 8e-1 -        | 1.645 3e-1 -        | 1.194 5e-2 -        | 2.558 8e-1 - | <b>9.77e-03</b> |
|         | 4  | 1.045 6e-1 -        | 1.232 2e-1 -        | 1.643 3e-1 -        | 5.028 1e-2 -        | 2.358 8e-1 - | <b>9.97e-03</b> |
|         | 6  | 5.254 6e-2 -        | 8.443 6e-2 -        | 1.203 0e-1 -        | 9.368 2e-2 -        | 1.520 6e-1 - | <b>1.65e-02</b> |
|         | 8  | 2.433 8e-2 -        | 3.610 5e-2 -        | 5.659 7e-2 -        | 6.422 5e-2 -        | 7.641 2e-2 - | <b>2.17e-02</b> |
|         | 10 | 9.952 4e-3 =        | 1.127 7e-2 -        | 1.517 3e-2 -        | 2.046 5e-2 -        | 2.125 6e-2 - | <b>9.48e-03</b> |
| DTLZ 6  | 3  | 3.154 7e+0 -        | 4.614 6e+0 -        | 5.274 9e+0 -        | 1.806 1e+0 =        | 2.126 6e+0 - | <b>1.64e+00</b> |
|         | 4  | 2.315 6e+0 -        | 4.658 9e+0 -        | 5.008 4e+0 -        | 1.845 2e+0 -        | 1.704 7e+0 - | <b>1.35e+00</b> |
|         | 6  | 1.423 3e+0 -        | 3.226 3e+0 -        | 3.430 6e+0 -        | 1.319 9e+0 -        | 1.132 0e+0 - | <b>8.40e-01</b> |
|         | 8  | 4.843 4e-1 =        | 1.399 9e+0 -        | 1.853 9e+0 -        | 8.860 9e-1 -        | 4.859 6e-1 = | <b>4.21e-01</b> |
|         | 10 | <b>4.0728e-2 +</b>  | 1.7890e-1 =         | 2.6826e-1 -         | 1.0662e-1 =         | 1.540 9e-1 - | 1.07e-01        |
| DTLZ 7  | 3  | <b>1.246 4e-1 +</b> | 1.465 5e+0 -        | 1.164 7e+0 -        | 2.591 9e-1 =        | 2.509 1e-1 - | 1.41e-01        |
|         | 4  | 2.904 0e-1 -        | 2.626 6e+0 -        | 1.226 0e+0 -        | 4.269 8e-1 -        | 4.921 7e-1 - | <b>2.31e-01</b> |
|         | 6  | 5.750 2e-1 -        | 4.502 3e+0 -        | 1.559 4e+0 -        | 6.895 4e-1 -        | 8.665 3e-1 - | <b>5.09e-01</b> |
|         | 8  | 1.026 4e+0 -        | 5.101 8e+0 -        | 1.930 3e+0 -        | 1.061 3e+0 -        | 1.100 2e+0 - | <b>8.65e-01</b> |
|         | 10 | <b>1.098 8e+0 +</b> | 2.141 6e+0 -        | 1.381 3e+0 -        | 1.248 1e+0 =        | 1.227 0e+0 = | 1.22e+00        |
| +/-/=   |    | 7/26/2              | 3/25/7              | 7/26/2              | 6/16/13             | 0/31/4       |                 |

对表2中数据进行分析, 相比于MOEA/D-EGO, DC-EMEA在大部分测试实例上取得更好或者相似的性能, 这是因为MOEA/D-EGO采用最大期望提升准则, 同时考虑收敛性和多样性来选择个体, 在计算资源有限的情况下, 难以取得较好的效果. DC-EMEA在10目标的DTLZ1和DTLZ3上性能差于CSEA和KTA2, 这是因为在CSEA中, 使用代理模型估计个体与参考解之间的支配关系, 选出来的个体收敛性较强, 而KTA2在填充准则的收敛性需求阶

段, 选择收敛性最强的个体, 能够提升种群的收敛性. DC-EMEA在DTLZ6总体性能较好, 仅在10目标情况下差于K-RVEA; DTLZ2和DTLZ4的真实PF面为凹型, 用于测试算法保持种群多样性的能力, 由于在DC-EMEA的填充准则中, 在选择多样性良好的解时, 同时考虑了对收敛性的维持, 导致DC-EMEA没有在这两个问题上取得最好的结果, 但总体性能良好; DTLZ5为退化问题, DC-EMEA取得最佳结果; DTLZ7为Pareto最优区域不连通的问题, 用于测

试算法在不同子区域上保持种群分布的能力,DC-EMEA 能够取得较好的结果. 这些实验验证了 DC-EMEA 的有效性.

图4为6个算法在 DTLZ 1、 DTLZ 3和 DTLZ 6 测试问题上的收敛曲线. 可以看出,DC-EMEA 在

DTLZ 1 和 DTLZ 3 上的收敛速度和最终结果获得了最佳性能,在 DTLZ 6 上取得与 KTA 2 和 MOEA/D-EGO 相似的性能,但比其他 3 个算法效果更好,这在一定程度上表明自适应填充准则能够有效地提升收敛速度.

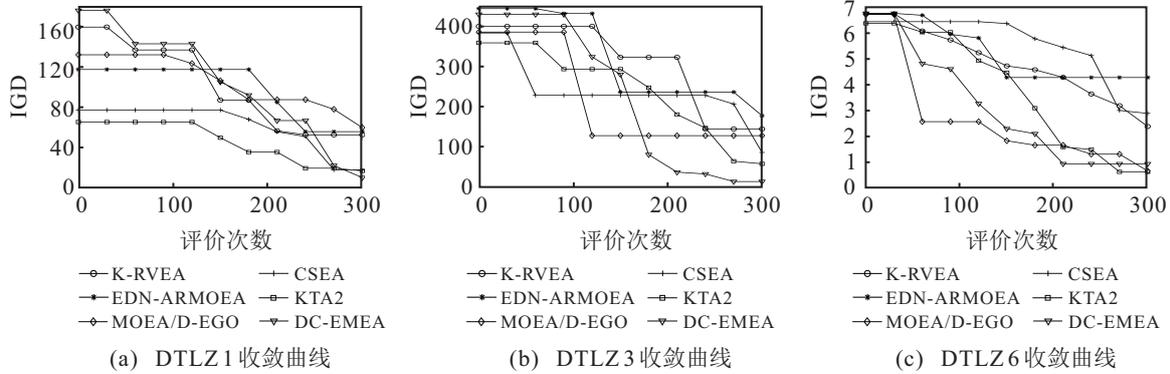


图 4 6个算法在3维DTLZ 1、DELZ 3和DTLZ 6测试问题上取得最佳IGD值时的收敛曲线

图5为6个算法在 DTLZ 2 测试问题上取得最佳 IGD 值时的非支配解集. 可以看出,DC-EMEA 非支配解集中的个体离 PF 面较近,而其他算法均存在

非支配解离 PF 面较远的情况,从解集的分布情况来看,DC-EMEA 的多样性仅次于 KTA 2,说明 DC-EMEA 在增强收敛性的同时能够维持较好的多样性.

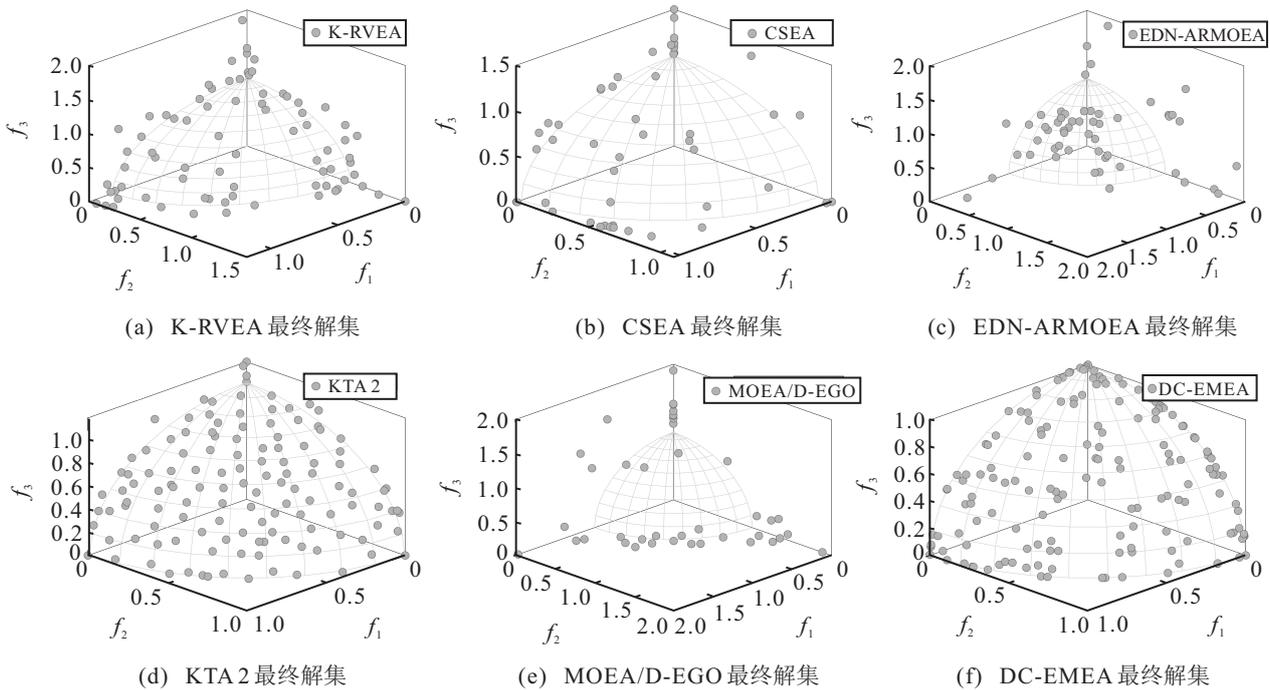


图 5 6个算法在3维DTLZ 2测试问题上取得最佳IGD值时的非支配解集

### 4 结论

本文提出了一种基于分解和聚类的昂贵高维多目标进化算法(DC-EMEA). 使用两轮环境选择策略对模型的最优解集进行搜索,能够很好地平衡收敛性和多样性,提升搜索效率;同时,使用自适应填充准则选择真实函数评价的个体,提升了收敛速度. 通过与 5 个先进的算法对比,验证了 DC-EMEA 的有效性. 后

续工作将研究使用 DC-EMEA 解决实际生活中的昂贵高维多目标优化问题.

### 参考文献(References)

[1] Jin Y C, Sendhoff B. A systems approach to evolutionary multiobjective structural optimization and beyond[J]. IEEE Computational Intelligence Magazine, 2009, 4(3): 62-76.

- [2] Zhou A M, Qu B Y, Li H, et al. Multiobjective evolutionary algorithms: A survey of the state of the art[J]. *Swarm and Evolutionary Computation*, 2011, 1(1): 32-49.
- [3] Regis R G. Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(3): 326-347.
- [4] Giannakoglou K C. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence[J]. *Progress in Aerospace Sciences*, 2002, 38(1): 43-76.
- [5] Jin Y, Olhofer M, Sendhoff B. A framework for evolutionary optimization with approximate fitness functions[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(5): 481-494.
- [6] 刘建昌, 赵阳杰, 李飞, 等. 基于R2指标的昂贵多目标进化算法[J]. *控制与决策*, 2020, 35(4): 823-832. (Liu J C, Zhao Y J, Li F, et al. Expensive multi-objective optimization algorithm based on R2 indicator[J]. *Control and Decision*, 2020, 35(4): 823-832.)
- [7] Pan L Q, He C, Tian Y, et al. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(1): 74-88.
- [8] Zhang J Y, Zhou A M, Zhang G X. A classification and Pareto domination based multiobjective evolutionary algorithm[C]. 2015 IEEE Congress on Evolutionary Computation. Sendai, 2015: 2883-2890.
- [9] Hao H, Zhou A M, Qian H, et al. Expensive multiobjective optimization by relation learning and prediction[J]. *IEEE Transactions on Evolutionary Computation*, 2022, 26(5): 1157-1170.
- [10] Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(1): 50-66.
- [11] Zhang Q F, Liu W D, Tsang E, et al. Expensive multiobjective optimization by MOEA/D with Gaussian process model[J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(3): 456-474.
- [12] Chugh T, Jin Y C, Miettinen K, et al. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 129-142.
- [13] Song Z S, Wang H D, He C, et al. A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 25(6): 1013-1027.
- [14] Guo D, Wang X L, Gao K L, et al. Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022, 52(4): 2084-2097.
- [15] Lu X F, Tang K. Classification-and regression-assisted differential evolution for computationally expensive problems[J]. *Journal of Computer Science and Technology*, 2012, 27(5): 1024-1034.
- [16] Wan K Z, He C, Camacho A, et al. A hybrid surrogate-assisted evolutionary algorithm for computationally expensive many-objective optimization[C]. 2019 IEEE Congress on Evolutionary Computation. Wellington, 2019: 2018-2025.
- [17] Cheng R, Jin Y C, Olhofer M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 773-791.
- [18] McKay M D, Beckman R J, Conover W J. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code[J]. *Technometrics*, 2000, 42(1): 55-61.
- [19] Deb K, Agrawal R. Simulated binary crossover for continuous search space[J]. *Complex Systems*, 1994, 9(3): 115-148.
- [20] Deb K, Goyal M. A combined genetic adaptive search (GeneAS) for engineering design[J]. *Computer Science and Informatics*, 1996, 26: 30-45.
- [21] Tian Y, Cheng R, Zhang X Y, et al. PlatEMO: A Matlab platform for evolutionary multi-objective optimization[educational forum[J]. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73-87.
- [22] Deb K, Thiele L, Laumanns M, et al. Scalable test problems for evolutionary multiobjective optimization evolutionary multiobjective optimization[C]. *Proceedings of the 2005 Evolutionary Multiobjective Optimization*. London, 2005: 105-145.
- [23] Coello Coello C A, Cortes N C. Solving multiobjective optimization problems using an artificial immune system[J]. *Genetic Programming and Evolvable Machines*, 2005, 6(2): 163-190.
- [24] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach—Part I: Solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.

### 作者简介

徐三水(1998—), 男, 硕士生, 从事进化算法的研究, E-mail: 2542962464@qq.com;

李军华(1974—), 男, 教授, 博士, 从事进化算法、智能控制等研究, E-mail: jhlee126@126.com;

李凌(1980—), 女, 博士, 从事医学图像处理、模式识别等研究, E-mail: ling.li@atizs.ac.cn;

黎明(1965—), 男, 教授, 博士生导师, 从事进化算法、模式识别等研究, E-mail: liming@nchu.edu.cn.