

# 控制与决策

Control and Decision

## 多群多策略差分大规模多目标优化算法

葛媛媛, 陈得宝, 邹锋

引用本文:

葛媛媛, 陈得宝, 邹锋. 多群多策略差分大规模多目标优化算法[J]. *控制与决策*, 2024, 39(2): 429–439.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.1154>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### [具有重组学习和混合变异的动态多种群粒子群优化算法](#)

Dynamic multi-population particle swarm optimization algorithm with recombined learning and hybrid mutation

*控制与决策*. 2021, 36(12): 2871–2880 <https://doi.org/10.13195/j.kzyjc.2020.0898>

#### [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>

#### [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>

#### [基于树形结构无界存档的多目标粒子群算法](#)

Multi-objective particle swarm optimization algorithm based on tree-structured unbounded archive

*控制与决策*. 2020, 35(11): 2675–2686 <https://doi.org/10.13195/j.kzyjc.2019.0276>

# 多群多策略差分大规模多目标优化算法

葛媛媛<sup>1,3</sup>, 陈得宝<sup>2,3†</sup>, 邹 锋<sup>2</sup>

1. 淮北师范大学 计算机科学与技术学院, 安徽 淮北 235000;
2. 淮北师范大学 物理与电子信息学院, 安徽 淮北 235000;
3. 安徽省认知行为智能计算工程研究中心, 安徽 淮北 235000)

**摘要:** 针对差分进化算法在解决大规模多目标优化问题时, 出现优化后期多样性不足、收敛速度慢等问题, 提出一种多群多策略差分大规模多目标优化算法. 根据个体特性不同, 将种群分为 3 个等级不同的子群, 利用多群策略的优势维持种群多样性. 为减少种群陷入局部最优的概率, 在不同等级的子群中引入多个变异策略以较好地平衡子群个体的多样性和收敛性. 为保证不同子群间信息得到有效交换, 根据 3 个子群的进化状态确定重新分群时机, 既保证个体在本群内得到充分进化, 又保证个体在一定的条件下进行信息交换. 为利用更多的信息生成优秀的子代, 将更新后的子群与其父代子群合并, 选出下一代子群. 为验证所提出算法的有效性, 在一组大规模基准测试问题上评估算法的性能, 实验结果表明, 所提出算法在两个常用测试指标 IGD 和 HV 上明显优于其他对比算法.

**关键词:** 大规模多目标优化; 多目标优化; 差分进化; 多种群策略; 变异策略

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2022.1154

引用格式: 葛媛媛, 陈得宝, 邹锋. 多群多策略差分大规模多目标优化算法[J]. 控制与决策, 2024, 39(2): 429-439.

## A large-scale multi-objective optimization based on multi-population and multi-strategy differential algorithm

GE Yuan-yuan<sup>1,3</sup>, CHEN De-bao<sup>2,3†</sup>, ZOU Feng<sup>2</sup>

1. School of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China;
2. School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, China;
3. Anhui Engineering Research Center for Intelligent Computing and Application on Cognitive Behavior, Huaibei 235000, China)

**Abstract:** In the middle or late evolutionary stage of large-scale multi-objective optimization problems (LSMOPs), the differential evolution (DE) algorithm has problems such as diversity shortage and slow convergence. A large-scale multi-objective optimization based on multi-population and multi-strategy differential evolution (LMOMMDE) is proposed. According to the characteristics of individuals in the population, the population is divided into three subpopulations with different levels, and the advantages of the multi-population strategy are used to maintain the diversity of the population. To reduce the probability that the population will fall into local optimum, multiple mutation strategies are introduced for subpopulations on different levels, this operation better balances the diversity and convergence of individual in subpopulations. To ensure the effective exchange of information among different subpopulations, this paper determines the timing of regrouping according to the evolutionary status of the three subpopulations, the individual can fully evolve within the population, and the individual can effectively exchange information under certain conditions. To use more information to generate excellent offspring, the updated subpopulations and their parent subpopulations are combined and generate the next generation subpopulations. To verify the effectiveness of the LMOMMDE, the performance of this algorithm is evaluated on a set of large-scale benchmark problems. The experimental results show that LMOMMDE is significantly better than the comparison algorithms in the two commonly used test indicators IGD and HV.

**Keywords:** large-scale optimization; multi-objective optimization; difference evolution; multiple population strategy; mutation strategy

收稿日期: 2022-07-01; 录用日期: 2022-09-21.

基金项目: 国家自然科学基金项目(61976101); 安徽省学术和技术带头人后备人选科研活动经费项目(2021H264); 安徽省高校学科(专业)拔尖人才学术项目(gxbjZD2022021).

责任编辑: 巩敦卫.

†通讯作者. E-mail: chendb\_8@163.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

多目标优化问题 (multi-objective optimization problems, MOPs) 广泛存在于实际应用中<sup>[1-3]</sup>. 为利用进化算法 (evolutionary algorithms, EAs) 实现多目标问题优化, 研究人员提出了大量的多目标优化进化算法 (multi-objective optimization evolutionary algorithms, MOEAs), 如基于指标的 MOEA<sup>[4]</sup>、基于分解的 MOEA<sup>[5]</sup> 和基于帕累托的 MOEA<sup>[6]</sup> 等. 但随着决策变量数量的增加, 搜索空间呈指数增长, 用 MOEAs 来处理高维的多目标问题时, 算法性能会急剧下降. 为解决大规模多目标优化问题 (large-scale multi-objective optimization problems, LSMOPs), 近些年, 人们从不同的方向提出了一些解决 LSMOPs 的方法, 到目前为止, 这些算法大体可分为4种类型.

第1类是基于协同进化 (cooperative coevolution, CC) 方法. 该方法主要将高维决策变量划分为若干组, 并优化每组决策变量. 最典型的算法是基于广义差分的协同进化算法 (CCGDE3)<sup>[7]</sup>. 首先, 种群被分解为几个独立的子群, 每个子群中高维决策变量被随机划分为几个数量相等的变量组; 然后, 通过广义差分进化算法 (GDE3)<sup>[8]</sup> 对每组决策变量进行优化. 这种随机变量分组的方法虽然操作简单, 但具有一定的盲目性, 算法的收敛速度受到一定的影响.

第2类是基于决策变量分析方法. 该类方法主要将 LSMOP 分解为多个简单的 MOPs. 代表性算法有基于决策变量分析的多目标优化算法 (MOEA/DVA)<sup>[9]</sup> 和基于决策变量聚类的大规模多目标优化算法 (LMEA)<sup>[10]</sup>. MOEA/DVA 算法将决策变量划分为距离变量、位置变量和混合变量, 再将距离变量分成若干组进行优化, 以提高种群的收敛性; 然后, 优化所有变量以提高种群的收敛精度. LMEA 算法则使用聚类方法将决策变量分为收敛性相关变量和多样性相关变量, 并对其分别使用不同优化策略. 之后, 文献[11]提出了可扩展小种群的协方差自适应进化算法 ( $S^3$ -CMA-ES), 该算法是在 MOEA/DVA 基础上利用协方差矩阵自适应策略对距离变量再次划分, 从提高变量的划分精度上改善算法的性能, 但对于变量之间存在相关性且难以划分的问题, 会出现变量分类不够准确, 算法的运算量也会增大.

第3类是基于问题转换的方法. 这类方法将 LSMOP 转换成小规模 MOP, 缩小了决策空间的搜索范围. 文献[12]提出一种加权优化框架 (WOF), 文献[13]提出一种基于问题重构的大规模框架 (LSMOF). WOF 算法将决策变量分成若干组并为

每组分配一个权重, 将权重视为待优化的变量, 将 LSMOP 转换为小规模 MOP, 从而降低计算量, 但该方法无法对同组的不同决策变量施加不同的操作; LSMOF 则是在决策空间中获取一组参考方向向量, 并与一组权重变量相关联, 使个体沿着参考方向搜索优秀解. 此类方法利用权重变量重构决策空间, 使决策变量维度得到有效降低, 节省大量计算时间. 但问题的重构, 损失了部分决策变量信息, 算法的收敛精度难以得到保证.

第4类是基于高效搜索方法. 主要是设计高效的子代生成方法来改善算法性能. 文献[14]提出一种求解 LSMOP 的自适应后代生成算法 (DGEA). 首先, 利用非支配排序和参考向量选出两组侧重收敛性和多样性的预选解集; 然后, 利用预选解集构造方向向量来产生优秀后代. 但当决策空间信息相似时, DGEA 算法将难以逃离局部最优. 在竞争群算法 (CSO)<sup>[15]</sup> 中, 竞争机制将种群分成失败者和成功者, 且失败者会向成功者学习. 在此基础上, 文献[16]提出了基于竞争群的大规模多目标优化算法 (LMOCSO), 为加快收敛速度, LMOCSO 在位置更新策略中添加了加速度. 这类算法中, 每代仅对失败者进行更新, 成功者未得到合理的引导, 影响算法逃离局部最优的能力.

差分进化 (differential evolution, DE)<sup>[17]</sup> 常用于解决 LSMOPs, 例如: CCGDE3<sup>[7]</sup> 和  $S^3$ -CMA-ES<sup>[11]</sup>, 这两种算法性能的好坏主要受决策变量分组的准确率影响. 此外, 这两种方法中, 单一的变异策略导致算法收敛速度慢且跳出局部最优需要消耗大量函数评估. 本文利用 DE 算法具有多种变异算子的优点, 针对 LSMOP, 提出一种基于多种群多策略的差分大规模多目标优化算法 (LMOMMDE). 通过分群策略提高种群的多样性, 采用多种变异策略来弥补单一变异策略的不足. 最后, 在 LSMOP 测试函数上验证了 LMOMMDE 算法的有效性. 本文的主要贡献如下:

1) 提出一种等级分群策略. 通过结合快速非支配排序和转移密度估计方法, 快速评估个体的优劣程度并将种群分成3个等级的子群, 且每个子群具备不同的搜索能力.

2) 提出多变异策略. 每个子群分配多个策略, 通过构造子群之间的差分向量来实现个体更新, 同时每个子群以一定概率独自进化以维持子群之间的信息差, 降低种群陷入局部最优的概率.

3) 为实现不同子群间信息的有效交换, 同时保证每个子群充分进化, 本文根据多个子群的状态设置重

新分组的条件.

## 1 相关工作

### 1.1 大规模多目标优化问题描述

一个具有  $M$  个目标函数和  $D$  维决策变量的多目标优化问题 MOPs 的数学描述如下:

$$\begin{aligned} \min F(X) &= (f_1(X), f_2(X), \dots, f_M(X)); \\ \text{s.t. } X &= (x_1, x_2, \dots, x_D) \in R^D. \end{aligned} \quad (1)$$

其中:  $X$  是个体的位置向量,  $f_M(X)$  是个体的第  $M$  个目标值. 当  $D \geq 100, M \geq 2$  时, 则为大规模多目标优化问题 (LSMOPs).

### 1.2 差分进化算法

差分进化 (DE)<sup>[17]</sup> 要经过群体初始化、个体位置更新 (变异和交叉)、选择等过程, 具体如下.

#### 1.2.1 种群初始化

假设当前种群规模大小为  $N$ , 目标和决策变量维度分别为  $M$  和  $D$ , 则第  $i$  个个体初始化为

$$X_i = X_{\min} + \text{rand}[0, 1] \times (X_{\max} - X_{\min}), \quad (2)$$

其中  $X_{\max}$  和  $X_{\min}$  分别是决策变量的最大值和最小值.

#### 1.2.2 变异算子

常用的 DE 变异策略如下 (其他变异策略参考文献 [18-19]):

$$\text{DE/rand/1} : V_i = X_{r_1} + F \times (X_{r_2} - X_{r_3}). \quad (3)$$

其中:  $V_i$  是个体变异后的位置向量,  $X_{r_1}, X_{r_2}, X_{r_3}$  分别是个体  $r_1, r_2$  和  $r_3$  的位置向量,  $F$  是缩放因子.

#### 1.2.3 交叉算子

交叉算子主要采用二项式交叉, 具体公式如下:

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{rand}(0, 1) \leq \text{Cr or } j = j_{\text{rand}}; \\ x_{i,j}, & \text{otherwise.} \end{cases} \quad (4)$$

其中: Cr 是交叉因子;  $U_i = [u_{i,1}, u_{i,2}, \dots, u_{i,D}]$  是交叉操作后新生成个体的决策向量,  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, D$ ;  $j_{\text{rand}}$  是  $[1, D]$  的随机整数.

#### 1.2.4 选择算子

选择算子的主要作用是选出下一代个体, 具体公式如下:

$$X_i = \begin{cases} U_i, & f(U_i) < f(X_i); \\ X_i, & \text{otherwise.} \end{cases} \quad (5)$$

其中  $f(U_i)$  和  $f(X_i)$  分别为  $U_i$  和  $X_i$  的适应度值.

## 2 多群多策略差分优化算法

DE 算法虽然在解决一些低维的多目标优化算法中发挥了较好的作用, 但在求解大规模优化问

题时, DE 的结果并不理想. 本文从平衡算法的多样性和收敛性出发, 设计一种多群多策略大规模多目标差分算法 (LMOMMDE). 与问题转换算法相比, LMOMMDE 算法不损失问题原始信息, 有利于提升解的精度. 与变量相关性分析方法相比, LMOMMDE 算法直接对决策变量进行优化, 无需分析变量间的关系, 节省变量分析所需的计算量.

### 2.1 分群策略

在 LMOMMDE 算法中, 多群策略的引入旨在维持种群个体之间的差异, 提高种群的多样性, 降低差分向量趋向于 0 的概率. 通常, 子群数量太多可能会导致分群过程计算复杂度上升, 数量太少又不利于维持种群多样性. 目前, 大多数多群大规模差分改进算法中, 子群数量一般设置为 3 或 4 个. 由于本文采用快速非支配排序方法, 若根据个体特性进行分群, 则后期会因为支配关系的减弱, 可能导致大部分个体集中于优秀群, 而其他子群个体数量很少, 不利于维护子群的多样性. 不失一般性, LMOMMDE 算法采用操作简单的均等划分方法来维持每个子群的个体数量. 为此, 根据个体的优秀程度, 将种群平均划分为 3 个子群, 分别为: 优秀群、一般群、较差群.

传统采用基于拥挤度的快速非支配排序方法<sup>[20]</sup> 重点关注个体多样性; 而在 LMOMMDE 算法中, 采用转移密度估计 (shift-based density estimation, SDE) 策略<sup>[21]</sup>, 同时考虑个体的收敛性和多样性, 提出基于 SDE 策略的快速非支配排序方法, 对个体进行等级排序并分群. 分群过程如图 1 所示.

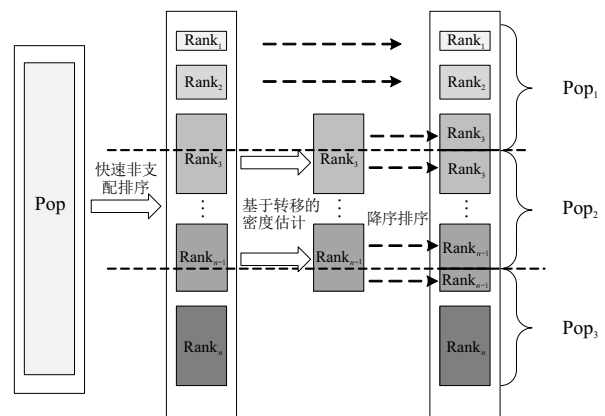


图 1 分群策略示例

首先, 对种群大小为  $N$  的种群 Pop 进行快速非支配排序, 得到种群 Pop 在目标空间中不同级非支配解集. 如图 1 所示, 假设种群 Pop 获得  $n$  级 Rank<sub>1</sub>、Rank<sub>2</sub>、...、Rank <sub>$n-1$</sub>  和 Rank <sub>$n$</sub>  的非支配解集, 其中 Rank <sub>$i$</sub>  级为第  $i$  级的非支配解集, 各级个体的优秀程度从 Rank<sub>1</sub> 到 Rank <sub>$n$</sub>  级依次递减. 假设某次分群

过程中, Rank<sub>1</sub> 和 Rank<sub>2</sub> 级个体的数量和小于  $N/3$ , 但加上 Rank<sub>3</sub> 级个体数量则大于  $N/3$ ; Rank <sub>$n$</sub>  级个体数量也小于  $N/3$ , 但加上 Rank <sub>$n-1$</sub>  级个体数量则大于  $N/3$ ; 于是对 Rank<sub>3</sub> ~ Rank <sub>$n-1$</sub>  级中的个体采用 SDE 策略来获取其优秀程度值并进行降序排序. 从 Rank<sub>3</sub> 级的个体中选择部分前排个体与 Rank<sub>1</sub> 和 Rank<sub>2</sub> 中的个体构建优秀子群 Pop<sub>1</sub>, 直至 Pop<sub>1</sub> 子群个体数量达到  $N/3$ . 对 Rank <sub>$n-1$</sub>  级的个体同样进行排序, 选择排名靠后的部分个体与 Rank <sub>$n$</sub>  中的个体构建较差群 Pop<sub>3</sub>. 剩余个体组成一般子群 Pop<sub>2</sub>. 计算个体的 SDE 适应度值公式如下:

$$\text{Fit}(p_i, \text{Pop}) = \min_{p_i \in \text{Pop}} (\text{Fit}(d(p_i, p_1), \dots, d(p_i, p_N))), \quad (6)$$

$$d(p_i, p_n) = \sqrt{\sum_{j=1}^M (\max\{0, f_j(p_i) - f_j(p_n)\})^2}. \quad (7)$$

其中:  $p_i$  和  $p_n$  分别是当代种群 Pop 中第  $i$  个和第  $n$  个个体,  $i, n \leq N$ .

## 2.2 变异策略

DE 算法有 6 种常用的变异策略, 不同变异策略有不同的优势. 针对不同性质的个体使用不同的变异策略, 有利于充分发挥不同变异策略的优势, 提升算法的整体性能. 本文为平衡子群的收敛性与多样性, 采用多变异策略, 具体方法如下.

优秀子群 Pop<sub>1</sub> 收敛性强于子群 Pop<sub>2</sub> 和 Pop<sub>3</sub>, 但是, 随着进化代数的增加, 该子群易因多样性缺失而出现局部最优. 因此, 对于子群 Pop<sub>1</sub> 中的一部分个体 (当  $\text{rand}(0, 1) \leq \text{Tr}$  时), 采用有利于多样性的变异策略 “DE/rand/1”; 从种群 Pop (Pop<sub>1</sub>  $\cup$  Pop<sub>2</sub>  $\cup$  Pop<sub>3</sub>) 中随机选择个体构建方差向量, 以增大子群 Pop<sub>1</sub> 的搜索范围, 进而增强子群 Pop<sub>1</sub> 的全局搜索能力. 对于剩下的个体, 采用局部勘察能力强的 “DE/current-to-best/1” 变异策略, 以提升子群的收敛精度, 变异策略公式如下:

当  $\text{rand}(0, 1) \leq \text{Tr}$  时

$$V_{i, \text{Pop}_1} = X_{i, \text{Pop}_1} + F_1 \times (X_{r_{11}, \text{Pop}} - X_{r_{12}, \text{Pop}}); \quad (8)$$

否则

$$V_{i, \text{Pop}_1} = X_{i, \text{Pop}_1} + F_1 \times (X_{\text{best}, \text{Pop}_1} - X_{i, \text{Pop}_1}) + F_1 \times (X_{r_{13}, \text{Pop}_1} - X_{r_{14}, \text{Pop}_1}). \quad (9)$$

其中:  $X_{r_{11}, \text{Pop}}$  和  $X_{r_{12}, \text{Pop}}$  表示分别从种群 Pop 中选择两个随机个体的位置向量,  $X_{i, \text{Pop}_1}$  和  $X_{\text{best}, \text{Pop}_1}$  分别表示子群 Pop<sub>1</sub> 的第  $i$  个个体的位置和当前子群最优个体的位置,  $X_{r_{13}, \text{Pop}_1}$  和  $X_{r_{14}, \text{Pop}_1}$  表示分别从

种群 Pop 中选择两个随机个体的位置,  $r_{11}$  和  $r_{12}$  是  $[0, N]$  范围内的随机整数,  $r_{13}$  和  $r_{14}$  是  $[0, N/3]$  范围内的随机整数, Tr 是概率阈值,  $F_1$  是子群 Pop<sub>1</sub> 的缩放因子.

子群 Pop<sub>2</sub> 的个体具有一定的多样性和收敛性, 为了加强子群的收敛速度, 同时维持子群的局部探测能力. 本文设计一种改进的变异策略. 当  $\text{rand}(0, 1) \leq \text{Tr}$  时, 利用 Pop<sub>1</sub> 的个体引导 Pop<sub>2</sub> 的部分个体进行变异, 提高子群的收敛速度; 子群 Pop<sub>2</sub> 的其他个体采用 “DE/current-to-best/1” 变异策略进行局部探测, 以提升子群的收敛精度. 由于按照个体优劣排序, Pop<sub>3</sub> 中的个体一般劣于 Pop<sub>2</sub> 的个体, Pop<sub>2</sub> 个体更新主要利用 Pop<sub>1</sub> 和 Pop<sub>2</sub> 的个体信息, 以保证子群 Pop<sub>2</sub> 具有一定的收敛速度. 子群 Pop<sub>2</sub> 的变异策略如下所示:

当  $\text{rand}(0, 1) \leq \text{Tr}$  时

$$V_{i, \text{Pop}_2} = X_{i, \text{Pop}_2} + F_2 \times (X_{r_{21}, \text{Pop}_1} - X_{i, \text{Pop}_2}); \quad (10)$$

否则

$$V_{i, \text{Pop}_2} = X_{i, \text{Pop}_2} + F_2 \times (X_{\text{best}, \text{Pop}_2} - X_{i, \text{Pop}_2}) + F_2 \times (X_{r_{22}, \text{Pop}_2} - X_{r_{23}, \text{Pop}_2}). \quad (11)$$

其中:  $X_{r_{21}, \text{Pop}_1}$  为从 Pop<sub>1</sub> 中随机选择一个个体的位置,  $X_{\text{best}, \text{Pop}_2}$  为子群 Pop<sub>2</sub> 的最优个体的位置,  $X_{r_{22}, \text{Pop}_2}$  和  $X_{r_{23}, \text{Pop}_2}$  表示分别从种群 Pop<sub>2</sub> 中选择两个随机个体的位置,  $r_{21}$ 、 $r_{22}$  和  $r_{23}$  是  $[0, N/3]$  范围内的随机整数,  $F_2$  是子群 Pop<sub>2</sub> 的缩放因子.

较差子群 Pop<sub>3</sub> 中个体多样性较好但收敛性较差, 为了提高种群的收敛精度且不易陷入局部最优, 当  $\text{rand}(0, 1) \leq \text{Tr}$  时, Pop<sub>3</sub> 分别从 Pop<sub>1</sub> 和 Pop<sub>2</sub> 中随机选择个体构造两个差分向量, 部分个体利用 Pop<sub>1</sub> 和 Pop<sub>2</sub> 的信息引导个体进化, 提升收敛速度. 同样, Pop<sub>3</sub> 需要维持自身多样性和提升其收敛性, Pop<sub>3</sub> 采用与 Pop<sub>1</sub> 和 Pop<sub>2</sub> 相同的 “DE/current-to-best/1” 变异策略来更新部分个体, 平衡子群的多样性与收敛性. 具体操作如下所示:

当  $\text{rand}(0, 1) \leq \text{Tr}$  时

$$V_{i, \text{Pop}_3} = X_{i, \text{Pop}_3} + F_3 \times (X_{r_{31}, \text{Pop}_1} - X_{i, \text{Pop}_3}) + F_3 \times (X_{r_{32}, \text{Pop}_2} - X_{i, \text{Pop}_3}); \quad (12)$$

否则

$$V_{i, \text{Pop}_3} = X_{i, \text{Pop}_3} + F_3 \times (X_{\text{best}, \text{Pop}_3} - X_{i, \text{Pop}_3}) + F_3 \times (X_{r_{33}, \text{Pop}_3} - X_{r_{34}, \text{Pop}_3}). \quad (13)$$

其中:  $X_{r_{31}, \text{Pop}_1}$  和  $X_{r_{32}, \text{Pop}_2}$  分别表示从子群 Pop<sub>1</sub> 和

Pop<sub>2</sub>中随机选择两个个体的位置,  $X_{best,Pop_3}$  为子群 Pop<sub>3</sub> 的最优个体的位置,  $X_{r_{33},Pop_3}$  和  $X_{r_{34},Pop_3}$  分别表示从子群 Pop<sub>3</sub> 中随机选择两个个体的位置,  $r_{31}$ 、 $r_{32}$ 、 $r_{33}$  和  $r_{34}$  是  $[0, N/3]$  范围内的随机整数,  $F_3$  是子群 Pop<sub>3</sub> 的缩放因子. 各子群的交叉操作与基本的DE方法相同, 此处不再赘述.

### 2.3 环境选择

如图2所示, 为构建下一代子群, LMOMMDE算法采用基于SDE的快速排序方法. 首先, 每个子群

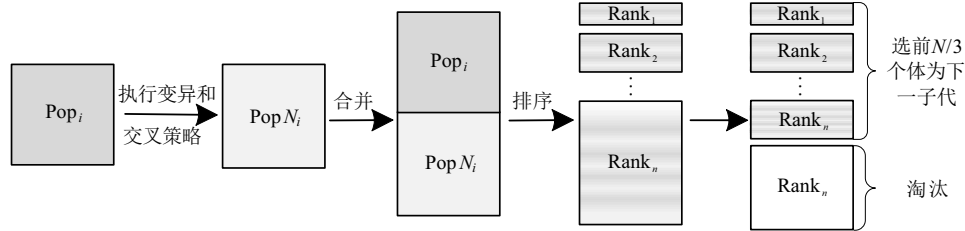


图2 选择策略示例

### 2.4 重新分群条件

通常, 在多群优化算法中, 频繁分群会导致子群的个体得不到充分进化且增加算法的运算量. 为此, 在 LMOMMDE 算法中, 重新分群的条件设置为: 若子群  $Pop_2 \cup Pop_{N_2}$  或  $Pop_3 \cup Pop_{N_3}$  中第1级非支配解集数量超过设定数量  $N_s$  时, 对种群进行重新分群,  $N_s$  的较优值利用参数敏感性分析方法得到. 设置此条件的原因是, 由于优秀群  $Pop_1 \cup Pop_{N_1}$  中的个体相对较好, 进行快速非支配排序时, 第1级的非支配解集数量一般较多, 如果将  $Pop_1 \cup Pop_{N_1}$  中非支配解集数量作为重新分群的依据, 则可能会导致种群频繁分群, 影响算法的性能. 对于子群  $Pop_2 \cup Pop_{N_2}$  或  $Pop_3 \cup Pop_{N_3}$ , 第1级的非支配解集数量超过某一给定值时, 可认为这两个子群有陷入局部最优的趋势, 此时选择重新分群, 可改善子群的多样性, 减少子群陷入局部最优的概率.

### 2.5 算法时间复杂度分析

在本文中, 假设种群大小为  $N$ , 目标数量为  $M$ , 决策变量维数为  $D$ . LMOMMDE 的初始化种群的复杂度是  $O(DN)$ . 在每一次迭代过程中, LMOMMDE 算法由交叉、变异和环境选择方法构成. 3 个子群的交叉和变异的复杂度是  $O(DN)$ , 环境选择时间复杂度是  $O((4(M+1)N^2)/3)$ . 此外, 满足重新分群条件时, 需要将 3 个子群合并并重新排序分群. 重新分群的时间复杂度是  $O((M+1)N^2)$ . 综上, LMOMMDE 算法中每次迭代的最小时间复杂度为  $O(DN + (4(M+1)N^2)/3)$ ; CCGDE3、LMEA、 $S^3$ -CMA-ES、WOF-NSGA-II 和 LMOCSSO 作为本文的对比算法, 其

通过其相应的变异和交叉策略产生新个体解集  $Pop_{N_i}(i = 1, 2, 3)$ ; 然后, 将每个子群的父代与新个体解集合并, 进而利用快速非支配排序方法对子群中的个体进行分级. 如果排在第1级的非支配解集的个体数量不满足子群个体数量的要求, 则采用 SDE 策略对其下一级的非支配解集进行排序, 选择部分排在前面的个体与上级个体共同构建满足子群个体数量的下一代子群; 剩下的个体自然淘汰.

时间复杂度分别为  $O(DN + 3MN^2)$ 、 $O(M^2N^2)$ 、 $O(MN^2 + D^2)$ 、 $O(DN + MN \log(N) + MN^2)$  和  $O((3MN)/2 + DN + (M+1)N^2)$ . 由此可见, LMOMMDE 算法与 CCGDE3、WOF-NSGA-II 和 LMOCSSO 算法在时间复杂度方面相当.

## 3 仿真实验

### 3.1 实验环境和测试函数

本文对比算法包括: CCGDE3<sup>[7]</sup>、LMEA<sup>[10]</sup>、 $S^3$ -CMA-ES<sup>[11]</sup>、WOF-NSGA-II<sup>[12]</sup> 和 LMOCSSO<sup>[16]</sup>. 这些算法各有特点, 代表目前对大规模多目标优化的不同方向. 为公平起见, 所有算法在 64 位 Windows 11 系统独立运行 20 次, 其中 CPU 为 Intel(R) Core(TM) i7-10700/2.9 GHz, 内存为 32 GB.

本文选择基准测试函数 LSMOP<sup>[22]</sup> 来测试算法的性能, 具体的测试函数名称和特点如表 1 所示. 目标空间维度  $M$  分别取 2 和 3, 决策空间维度  $D$  分别取 100、200 和 500.

表 1 LSMOP 测试函数特点

函数名称	PF 形状	形态特点	目标函数可分性
LSMOP 1	线性	单峰	完全可分
LSMOP 2	线性	混合	部分可分
LSMOP 3	线性	多峰	混合
LSMOP 4	线性	混合	混合
LSMOP 5	凹形	单峰	完全可分
LSMOP 6	凹形	混合	部分可分
LSMOP 7	凹形	多峰	混合
LSMOP 8	凹形	混合	混合
LSMOP 9	不连续	混合	完全可分

### 3.2 性能指标

本文采用反向世代距离 (IGD)<sup>[23]</sup> 和超体积 (HV)<sup>[24]</sup> 指标来衡量算法的收敛性和多样性. IGD 值越小, 表明算法获得的最终解集所组成的 Pareto front (PF) 越靠近真实 PF; HV 值越大, 表明算法的综合性能越好. 指标的计算公式如下:

$$IGD(S, Pop) = \frac{\sum_{i=1, s_i \in S}^{|S|} \min d(s_i, Pop)}{|S|}, \quad (14)$$

$$HV = \bigcup_{j=1}^N \{ \delta_{p_j \in Pop} (p_j, Z) \}. \quad (15)$$

其中:  $S$  是真实 PF 上的解集;  $Pop$  是最后一代种群;  $p_j$  是种群  $Pop$  的一个解;  $N$  是种群  $Pop$  的大小;  $\delta$  是勒贝格测度, 用来测量个体  $p_j$  与参考点  $Z$  所形成的超立方体的体积.

### 3.3 参数设置

在实验中, 当测试函数的目标数  $M = 2$  时, 种群规模  $N$  取 300; 目标数  $M = 3$  时, 种群规模  $N$  取

496;  $D$  为决策变量的数量. CCGDE3 和  $S^3$ -CMA-ES 的参数设置为文献 [7] 和文献 [11] 给出的建议值, 其他对比算法的参数值参考了文献 [16] 的参数设置. 所有算法的参数设置如表 2 所示.

表 2 6 种算法的具体参数值

算法	参数
CCGDE3	$F = 0.5, Cr = 0.5$
LMEA	$nPer = 5, nSel = 5, nCor = 6$
$S^3$ -CMA-ES	$m = 5, \Delta = 1e-6, RT = 400, \lambda = 20, F = 0.5, Cr = 0.2$
WOF-NSGA-II	$t_1 = 1\ 000, t_2 = 1\ 000, q = 2, \gamma = 3, \delta = 0.7$
LMOCSSO	$\alpha = 2$
LMOMMDE	$Ns = 70, Tr = 0.4, F \in [0.3, 0.8], Cr = 0.05$

LMEA、WOF-NSGA-II 和 LMOCSSO 使用多项式变异策略, 概率  $p_m = 1/D$ , 分布指数  $\eta_m = 20$ . LMEA 和 WOF-NSGA-II 使用二进制交叉策略, 交叉概率  $p_c = 0.1$ , 分布指数  $\eta_c = 20$ .

### 3.4 实验结果

表 3 和表 4 给出所有算法在 2 目标和 3 目标的测试函数 LMSOP 下 IGD 的平均值和方差值. 其中: 决

表 3 6 种算法在 2 目标的基准测试函数 LSMOP 上的平均 IGD 值

测试函数	$D$	CCGDE3	LMEA	$S^3$ -CMA-ES	WOF-NSGA-II	LMOCSSO	LMOMMDE
LSMOP1	100	1.575 8e+0(1.57e-1)-	1.560 8e-2(1.64e-3)-	4.529 1e-1(1.03e-1)-	1.201 1e-1(2.98e-3)-	3.718 4e-3(1.83e-4)-	<b>3.590 1e-3(1.49e-4)</b>
	200	2.699 4e+0(1.06e-1)-	1.616 4e-2(2.12e-3)-	5.494 6e-1(1.92e-1)-	1.115 5e-1(4.41e-2)-	4.592 2e-3(1.09e-4)-	<b>4.268 1e-3(1.27e-4)</b>
	500	3.383 6e+0(1.37e-1)-	2.332 0e-2(6.46e-4)-	6.345 5e-1(1.48e-1)-	1.340 8e-1(1.84e-2)-	1.523 9e-2(1.69e-3)-	<b>5.828 9e-3(1.02e-4)</b>
LSMOP2	100	1.967 3e-1(1.44e-2)-	3.528 0e-2(8.92e-4)-	5.049 1e-2(1.44e-2)-	2.153 0e-2(1.76e-3)-	3.536 5e-2(2.22e-3)-	<b>2.1423e-2(9.16e-3)</b>
	200	1.328 3e-1(4.04e-3)-	3.637 5e-2(1.33e-3)-	1.856 1e-2(2.00e-3)+	<b>1.732 6e-2(5.38e-3)+</b>	2.749 2e-2(4.88e-3)-	2.536 1e-2(9.13e-3)
	500	6.833 9e-2(1.45e-3)-	2.345 1e-2(1.14e-4)-	<b>9.064 2e-3(1.60e-2)+</b>	9.645 7e-3(1.78e-3)+	3.163 5e-2(1.33e-3)-	1.854 0e-2(8.53e-4)
LSMOP3	100	9.968 2e+0(1.57e+0)-	9.301 5e-1(2.00e-1)+	1.422 2e+0(2.91e-2)=	<b>6.246 8e-1(1.04e-3)+</b>	7.057 9e-1(2.28e-3)+	2.169 5e+0(6.10e-1)
	200	1.232 4e+1(1.09e+0)-	8.433 3e-1(6.17e-2)+	1.997 2e+0(1.27e-1)=	9.389 8e-1(5.10e-1)+	<b>7.071 6e-1(5.49e-2)+</b>	3.281 6e+0(8.80e-1)
	500	2.055 0e+1(1.69e-1)-	1.136 0e+0(1.22e-1)+	2.101 6e+0(3.16e-1)=	1.375 7e+0(2.66e-1)=	<b>7.342 5e-1(4.69e-2)+</b>	3.364 8e+0(1.34e+0)
LSMOP4	100	2.577 8e-1(4.78e-2)-	1.084 6e-1(4.25e-3)-	3.796 7e-1(5.54e-3)-	3.598 3e-2(2.36e-2)-	2.888 8e-2(3.87e-4)-	<b>2.530 5e-2(1.54e-3)</b>
	200	1.813 0e-1(1.60e-2)-	5.945 3e-2(5.64e-3)-	2.075 4e-1(1.17e-2)-	3.523 7e-2(8.67e-3)-	1.874 4e-2(2.69e-4)-	<b>1.776 0e-2(1.96e-4)</b>
	500	1.019 4e-1(1.23e-3)-	3.194 9e-2(4.95e-4)-	1.970 1e-1(2.48e-2)-	3.666 4e-2(2.98e-3)-	1.219 4e-2(4.38e-4)-	<b>1.059 2e-2(3.91e-3)</b>
LSMOP5	100	4.557 3e+0(8.75e-1)-	5.083 7e-1(4.05e-1)-	7.421 8e-1(9.19e-6)-	4.420 5e-2(7.56e-3)-	<b>4.159 7e-3(5.73e-4)+</b>	7.973 2e-3(1.56e-3)
	200	6.388 6e+0(1.14e-1)-	4.183 9e-1(2.49e-1)-	7.421 0e-1(3.86e-7)-	4.818 4e-2(8.43e-3)-	<b>5.400 5e-3(5.89e-4)+</b>	8.820 3e-3(4.26e-4)
	500	7.643 2e+0(1.55e-1)-	4.133 4e-1(1.31e-2)-	7.401 4e-1(5.49e-5)-	1.119 7e-1(5.61e-2)-	8.957 2e-3(7.66e-4)-	<b>8.273 7e-3(8.57e-4)</b>
LSMOP6	100	1.114 3e+0(5.17e-2)-	5.463 9e-1(1.33e-1)-	7.472 4e-1(6.18e-4)-	4.448 5e-1(7.99e-2)-	7.436 3e-1(3.68e-4)-	<b>3.940 3e-1(1.05e-1)</b>
	200	9.194 3e-1(6.32e-3)-	4.963 6e-1(5.78e-2)-	7.439 7e-1(8.80e-5)-	4.346 9e-1(1.47e-1)-	7.643 2e-1(5.16e-4)-	<b>4.320 7e-1(1.86e-2)</b>
	500	8.096 8e-1(3.17e-3)-	6.032 9e-1(1.78e-2)-	7.441 5e-1(9.06e-5)-	4.167 1e-1(6.48e-2)-	7.533 7e-1(1.76e-3)-	<b>4.141 3e-1(2.12e-2)</b>
LSMOP7	100	3.361 5e+3(9.77e+2)-	1.329 9e+0(3.19e-2)=	4.318 7e+0(8.21e-2)=	<b>7.976 5e-1(4.18e-2)+</b>	1.620 7e+0(5.12e-1)=	2.871 6e+0(9.70e-1)
	200	1.010 5e+4(1.99e+2)-	1.363 0e+0(1.07e-1)=	3.911 6e+0(3.15e-1)=	<b>9.699 4e-1(1.38e-1)</b>	1.576 1e+0(2.64e-1)=	1.669 3e+0(2.15e-1)
	500	2.365 7e+4(1.18e+4)-	1.448 8e+0(2.35e-2)=	3.994 3e+0(1.64e-1)=	<b>1.409 2e+0(7.14e-2)=</b>	1.612 7e+0(3.68e-1)=	2.430 9e+0(2.34e-1)
LSMOP8	100	4.781 1e+0(1.44e+0)-	1.131 0e-1(1.21e-2)-	3.998 2e-1(3.68e-2)-	6.808 8e-2(7.46e-3)-	3.525 1e-2(4.59e-4)-	<b>2.951 1e-2(9.96e-3)</b>
	200	5.315 2e+0(8.43e-1)-	6.634 4e-2(3.41e-3)-	3.192 3e-1(7.01e-2)-	6.604 1e-2(1.48e-3)-	5.145 3e-2(5.48e-4)-	<b>5.053 6e-2(4.90e-5)</b>
	500	5.945 4e+0(6.27e-1)-	4.477 4e-2(1.59e-4)-	3.061 7e-1(4.74e-2)-	6.094 1e-2(1.01e-2)-	<b>2.513 4e-2(3.63e-4)+</b>	3.582 7e-2(8.15e-4)
LSMOP9	100	7.611 7e+0(6.35e+0)-	4.645 6e-1(2.04e-1)-	6.672 5e-1(1.12e-2)-	8.100 4e-1(9.48e-4)-	<b>2.072 3e-2(1.39e-3)+</b>	1.265 6e-1(3.69e-2)
	200	1.506 7e+1(5.46e+0)-	4.104 0e-1(1.88e-1)-	6.070 5e-1(4.38e-3)-	2.047 1e-1(2.64e-1)-	1.689 2e-1(5.19e-3)-	<b>1.072 1e-1(1.32e-2)</b>
	500	1.550 4e+1(2.29e+0)-	4.951 5e-1(9.38e-3)-	6.446 7e-1(8.65e-3)-	6.481 1e-1(2.29e-1)-	<b>1.069 8e-1(1.59e-2)+</b>	2.029 8e-1(3.65e-3)
+ / - / =		0/27/0	3/21/3	2/19/6	5/19/3	8/16/3	-
average score		5.888 9	3.222 2	4.592 6	2.777 8	2.407 4	<b>2.111 1</b>

表4 6种算法在3目标的基准测试函数LSMOP上的平均IGD值

测试函数	<i>D</i>	CCGDE3	LMEA	S <sup>3</sup> -CMA-ES	WOF-NSGA-II	LMOCSO	LMOMMDE
LSMOP1	100	3.038 8e+0(6.68e-1)-	4.284 5e-2(2.73e-3)-	1.902 0e-1(2.64e-2)-	1.256 0e-1(5.25e-2)-	8.216 8e-2(2.55e-2)-	<b>2.943 1e-2(3.07e-4)</b>
	200	3.577 4e+0(1.70e+0)-	4.161 2e-2(1.82e-3)-	2.354 5e-1(1.03e-2)-	5.095 8e-2(1.46e-3)-	1.062 1e-1(3.21e-3)-	<b>3.018 8e-2(2.86e-4)</b>
	500	6.830 5e+0(3.00e-1)-	<b>4.516 3e-2(5.63e-3)+</b>	3.730 7e-1(9.52e-2)-	1.564 2e-1(5.52e-2)-	1.271 0e-1(3.27e-3)-	1.228 7e-1(5.04e-3)
LSMOP2	100	1.983 1e-1(3.24e-5)-	<b>4.675 7e-2(1.85e-3)+</b>	6.218 8e-2(9.79e-3)+	1.003 4e-1(1.02e-3)-	7.721 2e-2(1.94e-3)-	6.585 0e-2(1.01e-2)
	200	1.226 8e-1(5.02e-4)-	4.273 6e-2(3.74e-4)+	<b>3.826 1e-2(8.12e-3)+</b>	4.978 3e-2(1.39e-3)-	5.721 2e-2(9.97e-4)+	6.8314e-2(1.74e-3)
	500	5.988 6e-2(6.49e-4)-	<b>3.075 3e-2(4.98e-5)+</b>	3.128 1e-2(8.96e-2)+	3.559 5e-2(2.02e-3)+	3.167 5e-2(3.00e-4)+	4.655 0e-2(8.63e-4)
LSMOP3	100	9.779 9e+0(7.62e-1)-	8.444 6e-1(9.89e-2)+	1.498 9e+0(3.46e-1)=	8.596 9e-1(1.41e-3)+	<b>7.736 1e-1(8.09e-2)+</b>	1.476 7e+0(3.66e-1)
	200	1.337 2e+1(1.17e+0)-	9.398 6e-1(1.03e-1)+	3.391 0e+0(2.95e-1)=	<b>7.722 8e-1(1.25e-1)+</b>	8.564 9e-1(7.32e-3)+	1.293 9e+0(2.92e-1)
	500	1.523 9e+1(1.70e+0)-	8.928 1e-1(3.70e-3)+	6.991 7e+0(2.16e-1)-	<b>8.606 6e-1(2.94e-5)+</b>	8.607 2e-1(1.36e-16)+	1.648 5e+0(1.39e-1)
LSMOP4	100	5.313 2e-1(1.57e-2)-	1.322 1e-1(3.05e-3)-	3.590 2e-1(4.13e-2)-	1.244 3e-1(6.13e-3)-	1.3990e-1(6.29e-3)-	<b>6.984 2e-2(5.72e-3)</b>
	200	3.632 9e-1(1.96e-2)-	8.627 9e-2(4.26e-3)-	2.481 7e-1(9.75e-3)-	1.073 9e-1(1.11e-2)-	1.126 6e-1(2.37e-3)-	<b>7.044 8e-2(1.86e-3)</b>
	500	1.969 7e-1(4.80e-3)-	<b>4.680 5e-2(4.17e-4)+</b>	1.680 1e-1(8.29e-3)-	9.866 5e-2(3.74e-3)=	8.623 3e-2(1.65e-3)+	9.8629e-2(1.12e-2)
LSMOP5	100	3.622 8e+0(1.20e+0)-	2.676 3e+0(2.07e+0)-	9.459 3e-1(5.01e-7)-	3.510 7e-1(8.48e-3)-	2.057 5e-1(9.58e-2)-	<b>4.926 4e-2(2.51e-3)</b>
	200	4.856 2e+0(1.73e+0)-	6.057 8e+0(5.29e-1)-	8.960 0e-1(8.65e-2)-	3.572 7e-1(1.14e-4)-	2.717 3e-1(2.09e-2)-	<b>5.130 5e-2(3.18e-3)</b>
	500	7.195 3e+0(3.74e-2)-	8.663 6e+0(1.68e-1)-	7.401 6e-1(7.24e-5)-	<b>3.596 1e-1(3.25e-3)+</b>	5.0746e-1(2.22e-2)-	4.256 1e-1(2.52e-4)
LSMOP6	100	7.911 8e+2(7.47e+0)-	1.136 0e+0(7.20e-2)-	2.612 1e+0(9.15e-1)-	8.762 6e-1(3.76e-2)-	9.6748e-1(8.02e-2)-	<b>6.346 0e-1(1.36e-1)</b>
	200	2.354 9e+3(4.89e+2)-	2.602 5e+2(4.48e+2)-	1.377 7e+0(1.77e-2)=	1.025 3e+0(7.54e-2)=	1.236 1e+0(9.66e-2)=	<b>1.020 8e+0(7.86e-2)</b>
	500	5.020 8e+3(7.45e+2)-	6.547 6e+1(9.03e+1)-	<b>1.147 1e+0(1.274-2)=</b>	1.244 2e+0(2.36e-2)=	1.262 2e+0(1.19e-1)=	1.200 2e+0(3.21e-2)
LSMOP7	100	1.997 1e+0(1.91e-1)-	2.273 8e+0(8.65e-1)-	9.535 6e-1(5.81e-4)-	<b>7.400 2e-1(1.06e-3)+</b>	9.459 3e-1(7.65e-6)-	8.140 6e-1(2.74e-1)
	200	1.716 8e+0(1.13e-1)-	1.712 7e+0(2.41e-2)-	9.486 3e-1(2.14e-4)-	7.829 5e-1(1.24e-3)-	9.4012e-1(1.00e-2)-	<b>7.521 2e-1(3.35e-2)</b>
	500	1.259 6e+0(4.55e-2)-	9.728 9e-1(9.83e-2)-	9.249 7e-1(4.73e-4)-	8.146 0e-1(1.30e-3)-	9.459 3e-1(3.10e-6)-	<b>7.615 9e-1(6.72e-3)</b>
LSMOP8	100	9.642 9e-1(5.49e-2)-	1.619 7e-1(6.96e-2)-	9.500 3e-1(9.06e-4)-	3.629 0e-1(5.75e-3)-	1.600 6e-1(1.79e-2)-	<b>9.020 5e-2(1.18e-2)</b>
	200	9.099 5e-1(1.21e-2)-	1.510 3e-1(1.30e-2)-	9.468 0e-1(1.35e-4)-	1.333 6e-1(9.23e-3)-	1.052 8e-1(3.44e-3)=	<b>1.014 8e-1(4.12e-3)</b>
	500	7.005 0e-1(6.13e-2)-	1.096 6e-1(1.05e-2)-	9.390 6e-1(2.68e-4)-	7.295 6e-2(1.11e-3)+	<b>6.979 8e-2(1.29e-3)+</b>	8.578 0e-2(1.52e-3)
LSMOP9	100	3.081 6e+1(9.21e+0)-	5.399 5e-1(2.39e-2)-	7.836 9e-1(6.81e-2)-	1.145 0e+0(1.72e-5)-	<b>2.174 6e-1(3.32e-2)+</b>	3.764 8e-1(3.32e-2)
	200	3.459 3e+1(5.59e-1)-	4.779 7e-1(2.97e-2)=	8.185 3e-1(2.92e-2)-	1.1451e+0(9.40e-5)-	<b>3.743 6e-1(2.40e-2)+</b>	4.962 0e-1(4.52e-2)
	500	4.435 0e+1(1.19e+1)-	4.165 6e-1(2.51e-3)+	7.914 0e-1(5.61e-2)-	1.145 1e+0(1.75e-5)-	<b>4.005 0e-1(3.11e-2)+</b>	6.921 4e-1(1.40e-2)
+ / - / =		0/27/0	9/17/1	3/20/4	7/17/3	10/14/3	-
average score		5.814 8	3.259 3	4.148 1	2.963 0	2.629 6	<b>2.185 2</b>

策变量维度 *D* 分别是 100、200 和 500, 最优结果加粗表示. 为了直观地展示 LMOMMDE 算法的性能, 实验采用显著性水平为 0.05 的 Wilcoxon 秩和检验方法并对统计结果进行分析, “+ / - / =” 分别表示对比算法的性能显著“优于/劣于/类似于” LMOMMDE 性能的问题数量. “average score” 为各算法在所有问题上的 IGD 结果等级排序的平均值. 表 5 给出不同算法在 HV 指标上的 Wilcoxon 秩和检验的统计结果.

表5 6种算法在基准测试函数LSMOP上关于HV指标的Wilcoxon秩和检验的统计结果

<i>M</i>	<i>D</i>	CCGDE3	LMEA	S <sup>3</sup> -CMA-ES	WOF	LMOCSO
2	100	0/8/1	1/7/1	1/7/1	2/7/0	3/6/0
	200	0/7/2	0/8/1	1/7/1	2/6/1	1/6/2
	500	0/7/2	0/7/2	0/8/1	1/6/2	2/5/2
3	100	0/7/2	2/6/1	2/6/1	3/6/0	2/6/1
	200	0/7/2	2/6/1	2/6/1	4/5/0	3/4/2
	500	1/6/2	3/5/1	2/5/2	4/5/0	4/5/0

此外, 在图3和图4中给出每个算法在20次独立运行中的最优IGD值的最终PF. 其中: 三角形点构成

LSMOP测试问题的真实PF, 灰色点则表示各个算法的最优解集(从线性、凹形、不连续类型中分别选取3种测试函数, 且决策变量维数*D*取100).

### 3.4.1 2目标测试函数的实验结果分析

在表3中, LMOMMDE 平均排名得分为 2.111 1, 在所有比较算法中, 排名第 1. 其中: LMOMMDE 在 14 个问题上获得的结果最优; LMOCSO、WOF-NSGA-II 和 S<sup>3</sup>-CMA-ES 分别在 7 个、5 个和 1 个测试问题上优于其他算法. LMOMMDE 在 LSMOP1、LSMOP4 和 LSMOP6 问题上优化效果最好. 在 200 和 500 维决策变量的 LSMOP2 问题上, WOF-NSGA-II 和 S<sup>3</sup>-CMA-ES 表现最优; 在 LSMOP3 问题上, LMEA 和 LMOCSO 算法的优化效果优于 LMOMMDE; 在 100 和 500 维决策变量的 LSMOP5、500 维决策变量的 LSMOP8、100 和 500 维决策变量的 LSMOP9 问题上, LMOMMDE 的性能略低于 LMOCSO; 在 LSMOP7 问题上, 除了 CCGDE3 算法, LMOMMDE 与其他对比算法的表现相似.

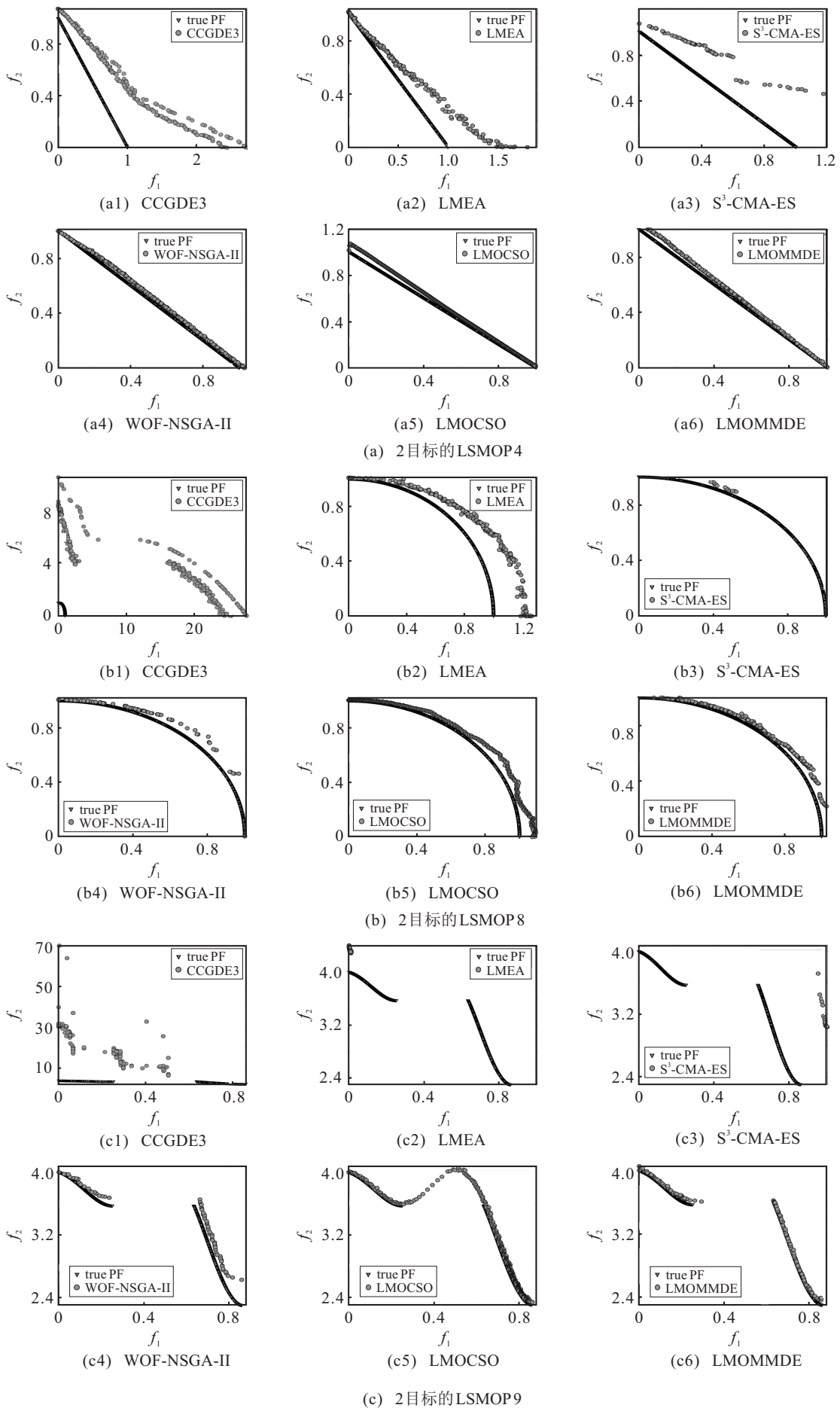


图3 6种算法在2目标的LSMOP4、LSMOP8、LSMOP9测试函数上的Pareto前沿

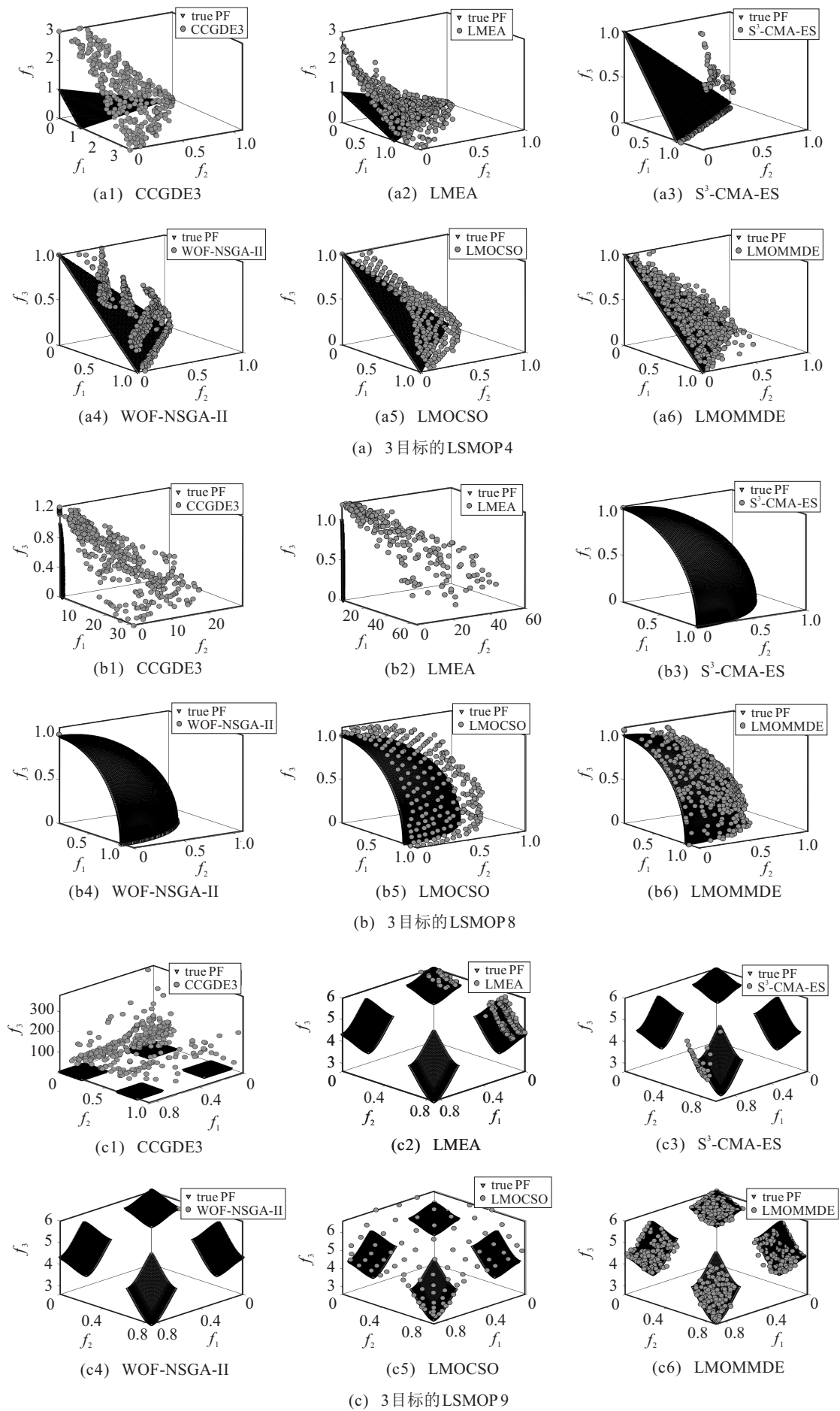


图4 6种算法在3目标的LSMOP4、LSMOP8、LSMOP9测试函数上的Pareto前沿

图3中, LMOMMDE在LSMOP4、LSMOP8和LSMOP9测试函数上的收敛性和分布性优于其他几种算法;在不连续测试函数LSMOP9中, CCGDE3、LMEA和S<sup>3</sup>-CMA-ES都陷入局部最优,得到的解集的多样性远不如WOF-NSGA-II、LMOCSO和LMOMMDE,其中LMOMMDE表现最优。

### 3.4.2 3目标测试函数的实验结果分析

表4结果显示: LMOMMDE在12个问题上性能最好; CCGDE3、LMEA、S<sup>3</sup>-CMA-ES和LMOCSO分别在0、4、2、4、5个测试问题上取得最好结果。其中LMOMMDE在LSMOP1和LSMOP4~LSMOP8问题上的性能较好。从图4可以看出: LMOCSO和LMOMMDE最优解集的分布性优于其他算法,但是LMOCSO的收敛精度不如LMOMMDE; CCGDE3和LMEA算法的收敛性表现较差;而S<sup>3</sup>-CMA-ES和WOF-NSGA-II在LSMOP8和LSMOP9问题上得到最优解的分布性较差。

综上所述, LMOMMDE在不损失群体的原始信息的前提下,采用多子群和与其特性对应的变异方法,使得子群的多样性与收敛性得到较好的平衡,在LSMOP上得到的平均效果优于对比算法。其中CCGDE3、LMEA和S<sup>3</sup>-CMA-ES的性能因易受决策变量分类准确率的影响,未充分考虑不同类个体间的引导,使得算法收敛性不如LMOMMDE。WOF-NSGA-II因为问题转换,种群部分原始决策信息丢失,所以收敛精度受到影响。从图3和图4看出, LMOCSO算法在不连续的LSMOP9问题上,因为使用参考向量选择下一代种群,所得到的个体的分布性较好,但其对离散型PF问题结果不优。

## 4 结论

本文针对大规模多目标问题,提出了一种基于多群多策略的大规模多目标差分进化算法,设计了一种基于SDE的快速非支配排序的分群方法,将种群均等划分成3个优势不同的子群,根据每个子群中个体不同的特性,为每个子群设计了不同的变异策略,较好平衡了每个子群的收敛性与多样性。在大规模测试函数集上对所设计的方法进行了仿真实验,并将结果与其他几种典型的大规模多目标算法进行比较,验证了本文所设计的算法在平均性能上优于其他算法。LMOMMDE算法不损失个体信息,有利于保证优化精度,与变量相关性分类大规模多目标优化算法相比,节省了变量相关性分析所需的计算量。但本文算法仅在2目标和3目标典型LSMOP测试函数集上进行了测试,未在高维大规模多目标问题和实际应用上

测试。下一步开展的工作是拓展算法在高维多目标问题及实际应用领域的应用。

### 参考文献(References)

- [1] Pasha J, Nwodu A L, Fathollahi-Fard A M, et al. Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings[J]. *Advanced Engineering Informatics*, 2022, 52: 101623.
- [2] 吴涛, 商慧丽, 张煜葵, 等. 基于黑洞多目标进化算法的永磁直线同步电机优化设计[J]. *控制与决策*, 2022, 37(6): 1567-1572.  
(Wu T, Shang H L, Zhang Y K, et al. Optimization of a tubular coreless linear PM synchronous machine based on multi-objective blackhole evolutionary algorithm[J]. *Control and Decision*, 2022, 37(6): 1567-1572.)
- [3] 齐东润, 陈刚. 无人驾驶机器人多目标模糊操纵策略[J]. *上海交通大学学报*, 2021, 55(10): 1310-1319.  
(Qi D R, Chen G. Multi-object fuzzy control strategy of unmanned driving robot[J]. *Journal of Shanghai Jiao Tong University*, 2021, 55(10): 1310-1319.)
- [4] Tian Y, Zhang X Y, Cheng R, et al. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric[C]. *2016 IEEE Congress on Evolutionary Computation*. Vancouver, 2016: 5222-5229.
- [5] Li H, Zhang Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284-302.
- [6] Lu C, Liu Q, Zhang B, et al. A Pareto-based hybrid iterated greedy algorithm for energy-efficient scheduling of distributed hybrid flowshop[J]. *Expert Systems with Applications*, 2022, 204: 117555.
- [7] Antonio L M, Coello C A C. Use of cooperative coevolution for solving large scale multiobjective optimization problems[C]. *2013 IEEE Congress on Evolutionary Computation*. Cancun, 2013: 2758-2765.
- [8] Kukkonen S, Lampinen J. GDE3: The third evolution step of generalized differential evolution[C]. *2005 IEEE Congress on Evolutionary Computation*. Edinburgh, 2005: 443-450.
- [9] Ma X L, Liu F, Qi Y T, et al. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 275-298.
- [10] Zhang X Y, Tian Y, Cheng R, et al. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization[J]. *IEEE Transactions on*

- Evolutionary Computation, 2018, 22(1): 97-112.
- [11] Chen H K, Cheng R, Wen J M, et al. Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations[J]. Information Sciences, 2020, 509: 457-469.
- [12] Zille H, Ishibuchi H, Mostaghim S, et al. A framework for large-scale multiobjective optimization based on problem transformation[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(2): 260-275.
- [13] He C, Li L H, Tian Y, et al. Accelerating large-scale multiobjective optimization via problem reformulation[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(6): 949-961.
- [14] He C, Cheng R, Yazdani D. Adaptive offspring generation for evolutionary large-scale multiobjective optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2): 786-798.
- [15] Cheng R, Jin Y C. A competitive swarm optimizer for large scale optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(2): 191-204.
- [16] Tian Y, Zheng X T, Zhang X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. IEEE Transactions on Cybernetics, 2020, 50(8): 3696-3708.
- [17] Storn R, Price K. Differential evolution — A simple and efficient adaptive scheme for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 23(1): 341-359.
- [18] 李香平. 多策略差分进化算法研究及在多星协同任务规划上的应用[D]. 武汉: 中国地质大学, 2019.  
(Li X P. A research on multi-strategy differential evolution algorithm and its application in the multi-satellite cooperative task planning[D]. Wuhan: China University of Geosciences, 2019.)
- [19] 邓武, 蔡幸, 周永权, 等. 一种基于多策略差分进化的分解多目标进化算法[J]. 控制与决策, 2022, 37(2): 387-392.  
(Deng W, Cai X, Zhou Y Q, et al. A novel decomposition multi-objective evolutionary algorithm based on differential evolution model with multi-strategy[J]. Control and Decision, 2022, 37(2): 387-392.)
- [20] 郑强. 带精英策略的非支配排序遗传算法的研究与应用[D]. 杭州: 浙江大学, 2006.  
(Zheng Q. Elitist nondominated sorting genetic algorithm and its application[D]. Hangzhou: Zhejiang University, 2006.)
- [21] Li M Q, Yang S X, Liu X H. Shift-based density estimation for Pareto-based algorithms in many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(3): 348-365.
- [22] Cheng R, Jin Y C, Olhofer M, et al. Test problems for large-scale multiobjective and many-objective optimization[J]. IEEE Transactions on Cybernetics, 2017, 47(12): 4108-4121.
- [23] Coello Coello C A, Cortés N C. Solving multiobjective optimization problems using an artificial immune system[J]. Genetic Programming and Evolvable Machines, 2005, 6(2): 163-190.
- [24] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271.

### 作者简介

葛媛媛(1999—),女,硕士生,从事智能多目标优化及应用的研究, E-mail: 1509327890@qq.com;

陈得宝(1975—),男,教授,博士,从事智能多目标优化及应用、动态多目标优化等研究, E-mail: chendb\_8@163.com;

邹锋(1978—),男,副教授,博士,从事机器学习、进化计算与群智能优化等研究, E-mail: zfemail@163.com.