

# 高维多目标头脑风暴优化算法

吴亚丽<sup>1,2†</sup>, 付玉龙<sup>1,2</sup>, 李国婷<sup>1,2</sup>, 张亚崇<sup>3</sup>

(1. 西安理工大学 自动化与信息工程学院, 陕西 西安 710048;

2. 陕西省复杂系统控制与智能信息处理重点实验室, 陕西 西安 710048;

3. 西安飞行自动控制研究所, 陕西 西安 710065)

**摘要:** 多目标优化的两个核心指标是收敛性和多样性, 而对二者加以优化和权衡是多目标进化算法的关键. 头脑风暴优化算法作为一种新型的群体智能优化算法, 一经提出便引起了众多研究者的关注. 本文在对现有的多目标头脑风暴优化算法研究的基础上, 通过对决策变量进行分析, 围绕收敛性和多样性分别进行优化, 在对收敛性优化时通过分解策略增加选择压力, 而在对多样性优化时以参考点更新种群增加多样性, 最终扩展并提出了高维多目标头脑风暴优化算法. 此外, 本文提出一种以角点为聚类中心的自适应聚类方式, 明确个体的导向, 提高种群的扩展性. 与现有的几种效果较好的多目标进化算法进行比较, 大量的仿真结果表明了本文的算法具有优秀的性能.

**关键词:** 头脑风暴优化算法; 聚类; 决策变量聚类; 分解策略; 参考点

**引用格式:** 吴亚丽, 付玉龙, 李国婷, 等. 高维多目标头脑风暴优化算法. 控制理论与应用, 2020, 37(1): 193 – 204  
DOI:10.7641/CTA.2019.80898

## Many-objective brain storm optimization algorithm

WU Ya-li<sup>1,2†</sup>, FU Yu-long<sup>1,2</sup>, LI Guo-ting<sup>1,2</sup>, ZHANG Ya-chong<sup>3</sup>

(1. School of Automation and Information Engineering, Xi'an University of Technology, Xi'an Shaanxi 710048, China;

2. Shaanxi Province Key Laboratory of Complex System Control and Intelligent Information Processing, Xi'an Shaanxi 710048, China;

3. Xi'an Flight Automatic Control Research Institute, Xi'an Shaanxi 710065, China)

**Abstract:** Convergence and diversity are two core indicators of multi-objective optimization. The optimization and the balance of them are the keys of multi-objective evolutionary algorithms (MOEA). As a new kind of swarm intelligence optimization algorithm, brain storm optimization (BSO) has paid more attention of more researchers in different fields. Based on the research of the existing multi-objective BSO (MOBSO), this paper optimizes the convergence and diversity by analyzing the decision variables of problems. Decomposition strategy is carried out to increase the select pressure while the convergence optimization is performed, and the strategy of reference points is adopted to update the population to increase the diversity while the diversity optimization is performed. Finally, we extend and propose the many-objective brain storm optimization algorithm (MaOBSO). In addition, this paper proposes a new adaptive clustering strategy based on the corner points, which could clarify the orientation of individuals and improve the expansibility of population. Compared with several existing multi-objective evolutionary algorithms with better performance, a large number of simulation results show that the algorithm of this paper has excellent performance.

**Key words:** brain storm optimization (BSO); cluster; decision variable clustering; decomposition strategy; reference points

**Citation:** WU Yali, FU Yulong, LI Guoting, et al. Many-objective brain storm optimization algorithm. *Control Theory & Applications*, 2020, 37(1): 193 – 204

## 1 引言

作为一种典型的复杂优化问题, 多目标优化问题 (multi-objective optimization problem, MOP) 已广泛应用于工程领域以及现实生活中, 如地下水监测问题<sup>[1]</sup>、PID控制器设计问题<sup>[2]</sup>、网络社会物理系统<sup>[3]</sup>等

领域. 近几年, 随着时代的发展, 现实所面临的问题越来越复杂, 并且呈现出动态、多约束、大规模、目标个数增加等特性. 通常, 称目标个数多于3个的MOP为高维多目标优化问题 (many-objective optimization problem, MaOP).

收稿日期: 2018–11–19; 录用日期: 2019–04–28.

†通信作者. E-mail: yliwu@xaut.edu.cn; Tel.: +86 13119121204.

本文责任编辑: 王凌.

国家自然科学基金青年基金项目(61503299, 61502385)资助.

Supported by the National Youth Foundation of China (61503299, 61502385).

与单目标优化问题 (single-objective optimization problem, SOP) 相比, MOP 需要得到一组解来逼近问题真实的前沿, 这组解在决策空间中被称为 Pareto 最优解集 (pareto set, PS), 在目标空间中被称为 Pareto 最优前沿 (pareto front, PF). 当问题的目标个数增加时, 用来解决多目标问题的经典算法表现出了较大的局限性, 主要表现在收敛能力和多样性保持能力两个方面.

由于目标数目的增加, 种群中的非支配解所占比重随着目标个数急剧增加<sup>[4]</sup>, 选择压力减小, 支配关系难以适用, 导致算法的进化速度降低. 基于此问题, 主要研究分为以下两点: ① 开发或改进支配方式, 增加选择压力, 如  $\epsilon$ -支配<sup>[5]</sup>、 $L$ -最优<sup>[6]</sup>、模糊支配<sup>[7]</sup>、网格支配<sup>[8]</sup>、 $\theta$ -支配<sup>[9]</sup>、增强支配 (strengthened dominance relation, SDR) 等<sup>[10]</sup>, 其主要通过更加松弛的支配关系来对原本互不支配的个体进行区分, 以更加合理地进行适应度选择和环境选择; ② Pareto 支配与其他选择指标相结合, 如基于分解的多目标进化算法<sup>[11-13]</sup>、基于指标的多目标进化算法<sup>[14-15]</sup>等, 主要通过权重向量对个体进行衡量或根据个体对整体指标的贡献来对个体进行适应度值分配.

在多样性的保持能力方面, 由于目标函数的增加使得候选解稀疏地分布在高维空间里, 导致了传统多样性维持策略的低效率. 尽管拥挤距离的策略已较为成熟且表现出良好的性能<sup>[16]</sup>, 但在 MaOP 方面仍然表现出较低的效率. 为此, 文献<sup>[17]</sup>提出角度惩罚距离的概念, 并证明了基于角度的度量不仅具有可扩展性, 而且性能优于基于欧氏距离的方式. 基于参考点的选取也是当前流行的一种 Pareto 解集的维持策略, 已被应用于很多比较流行的算法中<sup>[12, 17-18]</sup>.

由上述分析可以看出, MaOP 问题的研究难点是收敛性和多样性之间的权衡. 大多数算法将其归结于根据目标值进行适应度值分配以及选择问题, 并从各方面提出了分配以及选择策略. 文献<sup>[13]</sup>和文献<sup>[19]</sup>从问题角度出发, 围绕各决策变量对目标的影响将其分为收敛性相关变量和多样性相关变量, 然后分别对其进行优化, 增强优化结构; 此外, 采用分而治之的思想对无关收敛性相关变量分别进行优化, 提高优化效率, 取得了非常好的效果.

在众多的智能优化算法中, 史玉回教授 2011 年提出的头脑风暴优化算法 (brain storm optimization, BSO) 以聚类中心为导向, 并以概率朝着其他个体进行学习, 能够较好地兼顾收敛性和多样性<sup>[20]</sup>. 在经典算法的基础上, 不同研究者围绕其聚类、变异、以及更新归档集的方式衍生出了多种多目标头脑风暴优化算法 (multi-objective brain storm optimization, MOBSO), 均取得了较好的结果<sup>[21-23]</sup>. 文献<sup>[21]</sup>将聚类应用于目标空间, 并采用精英类策略, 加快 MOBSO 的收敛速

度. 文献<sup>[22]</sup>采用 DBSCAN 聚类代替经典 BSO 中的  $k$ -means 聚类, 用差分变异代替高斯变异, 提出了改进的多目标头脑风暴优化算法 (modified multi-objective brain storm optimization, MMBSO), 效果大幅提高. 文献<sup>[23]</sup>采用了 MBSO 中的简单聚类, 并在差分变异的基础上引入了开放性概率, 有效避免陷入局部最优 (self-adaptive multi-objective brain storm optimization, SMOBSO), 吴亚丽等人<sup>[24]</sup>将其用于解决热电联供环境经济调度 (environment economic dispatch, EED) 问题, 结果优于效果较好的准对抗教学优化算法 (quasi-oppositional teaching learning based optimization, QOTLBO).

基于上述的研究基础, 本文综合 BSO 算法的进化优势与聚类思想的选择策略, 提出一种有效的高维多目标头脑风暴优化算法. 算法对 MaOP 问题的决策变量特性进行进一步特征提取, 并基于不同的特征设计不同的进化策略. 主要步骤为: 首先, 根据决策变量的特征将其分为收敛性相关变量和多样性相关变量, 并采取不同的优化策略. 对于收敛性相关变量, 对收敛性相关变量进行独立性分析, 对于相互独立的变量“分而治之”, 提高算法优化效率; 并采用分解的策略将多个目标转换为一个目标, 增加选择压力, 提高收敛速度; 对于多样性相关变量, 提出了一种新的以角点为聚类中心的自适应聚类方式, 以一定的概率基于类中心产生新个体提高算法的扩展性: 采用参考点策略更新种群, 提高算法的均匀性. 此外, 为了增强算法的导向性, 个体对应的权重向量和参考点均采用标准边界交叉方法 (normal-boundary intersection, NBI) 方式生成且相对应, 且每次迭代中进行动态分配. 在以权重向量进行目标转换和以参考点进行更新时, 个体以参考点为导向. 文中给出大量的仿真算例来验证算法的有效性.

## 2 头脑风暴优化算法的基本原理

### 2.1 头脑风暴优化算法

头脑风暴优化算法 (BSO) 是受人类头脑风暴会议所启发, 将产生的每一个想法看做是一个潜在的可行解, 产生想法的过程遵循“庭外判决”、“各抒己见”、“取长补短”、“追求数量”的原则, 即产生尽可能多的解, 量变产生质变. 与人类头脑风暴会议过程类似, 头脑风暴优化算法以聚类中心为导向产生新个体对应于以一个较好的想法为参考生成新的想法. 在得到导向之后, 新个体通过在其基础上进行变异来产生, 每一次迭代完后进行比较, 保留较好的个体进入下一次迭代. 头脑风暴优化算法过程如下所示<sup>[20]</sup>:

**Step 1** 种群初始化;

**Step 2** 评价个体, 聚类;

**Step 3** 产生新个体;

**Step 4** 更新种群和聚类中心;

**Step 5** 若达到最大迭代次数则输出最优个体, 否则转Step 2.

传统的BSO采用 $k$ -means进行聚类, 并将每个类中最优的个体作为聚类中心. 在产生新个体时, 选择个体以一定的概率分别为一个聚类中心、一个类中个体、两个聚类中心或者类中个体的加权和. 在此基础上, 通过加变异实现对当前解的探索过程, 常用的高斯变异定义如下:

$$X_{\text{new}}^d = X_{\text{old}}^d + \xi * n(\mu, \sigma), \quad (1)$$

其中:  $X_{\text{old}}^d$  表示选择个体的第 $d$ 维,  $X_{\text{new}}^d$  是产生的新个体的第 $d$ 维,  $n(\mu, \sigma)$  是以 $\mu$ 为均值、 $\sigma$ 为方差的高斯随机函数,  $\xi$  是一个衡量高斯随机值权重的系数, 表示如下:

$$\xi = \text{logsig}((T/2 - t)/K) * \text{rand}, \quad (2)$$

其中:  $\text{logsig}$  是一个S型对数传递函数;  $T$  为最大迭代次数;  $t$  为当前迭代次数;  $K$  用来改变 $\text{logsig}$ 的斜率, 可调节算法由全局搜索到局部搜索的速度;  $\text{rand}$  产生(0, 1)之间的随机值.

张军等人<sup>[25]</sup>提出以差分变异代替高斯变异, 并取得了更好的效果且已经成功用于多种版本的BSO中<sup>[23-24, 26]</sup>. 差分变异如式(3)所示:

$$X_{\text{new}}^d = \begin{cases} \text{rand}(L_d, H_d), & \text{rand} < P_r, \\ X_{\text{old}}^d + \text{rand} * (X_1^d - X_2^d), & \text{其他}, \end{cases} \quad (3)$$

其中:  $L_d, H_d$  为第 $d$ 维的上下界,  $X_1^d, X_2^d$  为在当代全局中选择的两个不同的个体的第 $d$ 维. 为了保持种群的多样性, 防止陷入局部最优, 设置开放性概率 $P_r$ .

从式(3)可以看出, 与高斯变异相比, 差分变异运算量减小, 运行速度提高. 而且变异量的产生是基于当代种群内其他个体而来, 可根据种群内个体的离散程度来自适应调节, 能够更好地与种群内其他个体共享信息, 其搜索效率显著提高. 因此, 差分变异能够在搜索过程中更好地平衡局部搜索和全局搜索, 有效地提高算法性能.

## 2.2 多目标头脑风暴优化算法

与BSO相比, MOBSO引入了归档集的策略, 用来保存Pareto最优解. 在每次迭代中对种群进行非支配排序, 取非支配解计入归档集, 然后进行更新, 最终得到一组足够靠近最优前沿且均匀的解. MOBSO采用循环拥挤距离策略更新归档集. 首先令两个端点的拥挤距离为无穷大, 然后分别计算所有个体相邻点在目标空间的曼哈顿距离, 逐次删除拥挤距离最小的个体, 以保持归档集内的个体具有均匀性. MOBSO能够很好地解决两个目标的优化问题并应用于实际, 由于其对新个体的选择机制使得难以用于解决MaOP, 且原始的基于Pareto支配策略导致使得算法收敛速度过慢.

基于此, 本文将归档集的更新策略和多样性选择机制进行重新设计, 提出了一种求解高维多目标优化问题的头脑风暴优化算法.

## 3 高维多目标头脑风暴优化算法

### 3.1 主要框架

多目标优化算法的目的在于得到一组优秀且均匀的Pareto最优解, 由于不同特性的决策变量对优化结果有着不同的影响<sup>[19]</sup>, 因此在每一次迭代中先后进行收敛性优化和多样性优化, 即先使得个体向好的方向移动, 然后保持种群的多样性. 在进行优化之前, 首先需产生参考点以及对决策变量进行聚类. 算法的主要框架见算法1.

**算法 1** MaOBBSO.

**输入:**  $M$  (目标个数),  $E_{\text{max}}$  (最大评价次数),  $P_o$  (选择一个个体产生新个体的概率),  $P_c$  (选择聚类中心产生新个体的概率);

**输出:**  $P$  (最终种群);

1)  $P \leftarrow \text{Initialize}(M)$ ; //初始化

2) 用NBI的方式生成权重向量 $w$  (参考点);

3)  $[w, \text{cluster}] \leftarrow \text{VectorAllocation}(P, w, M)$ ; //参考点分配以及聚类\*/

4)  $[DV, CV] \leftarrow \text{VariableCluster}(P)$ ; //决策变量聚类

5)  $\text{sub CVs} \leftarrow \text{InteractionAnalysis}(P, CV)$ ; //独立性分析

6) 更新 $Z_{\text{min}}$ 和 $Z^{\text{nad}}$ ;

7) **while** 没有达到终止条件 **do**

8) **if** 收敛指标未达到 **then**

9)  $P \leftarrow \text{ConverOptimi}(P, \text{sub CVs}, Z_{\text{min}})$ ;

//收敛优化

10) 更新 $Z_{\text{min}}$ ;

11) **end if**

12)  $P \leftarrow \text{DiverOptimi}(P, P_o, P_c, DV, w, \text{cluster}, Z_{\text{min}}, Z^{\text{nad}})$ ; //多样性优化\*/

13) 更新 $Z_{\text{min}}$ 和 $Z^{\text{nad}}$ ;

14) **end while**

MaOBBSO采用NBI<sup>[18]</sup>的方式生成权重向量, 因此直接将种群视为归档集. 在算法1中, 首先进行初始化, 包括生成新的种群以和参考点, 并以角点为聚类中心对参考点进行聚类. 然后将个体与参考点进行匹配, 并将参考点的聚类结果对应到种群中. 接着将决策变量分为收敛性相关变量和多样相关变量, 并对收敛性相关变量进行独立性分析. 最后进行变量优化.

**算法 2**  $\text{VectorAllocation}(P, w, M)$ .

**输入:**  $P$  (种群),  $w$  (参考向量),  $M$  (目标个数);

**输出:**  $w$  (分配后的向量),  $\text{cluster}$  (聚类结果);

1)  $\text{cluster} \leftarrow$  以角点为聚类中心对 $w$ 进行 $k$ -

means聚类;

2)  $\text{Angle} \leftarrow$  计算  $P$  中所有个体与  $w$  的角度; /\*  $\text{Angle}$  为一个  $N \times N$  的矩阵,  $\text{Angle}[i][j]$  表示第  $i$  个体与第  $j$  个向量的夹角,  $N$  为种群规模\*/

3) **for**  $k = 1$  to  $N$  **do**

4)  $\text{sortindex}[i] \leftarrow j$ ; //  $[i, j]$  是  $\text{Angle}$  最大值的下标

5)  $Q1 \leftarrow$  当前  $\text{Angle}$  的第  $i$  行;

6)  $\text{Angle} \leftarrow \text{Angle}/Q1$ ;

7)  $Q2 \leftarrow$  当前  $\text{Angle}$  的第  $j$  列;

8)  $\text{Angle} \leftarrow \text{Angle}/Q2$ ;

9)  $w'[k] \leftarrow w[\text{sortindex}[i]]$ ;

10)  $\text{cluster}'[k] \leftarrow \text{cluster}[\text{sortindex}[i]]$ ;

11) **end for**

12)  $w \leftarrow w'$ ;

13)  $\text{cluster} \leftarrow \text{cluster}'$ .

算法2为将参考点和个体相对应以及聚类的过程. 本文采用参考向量的策略, 首先对参考向量根据其角点(极 endpoint)为中心进行聚类, 确定每个向量所属类别. 然后计算每个向量与个体在目标空间的夹角, 根据角度进行分配, 并以分配关系将参考点的角力聚类结果对应到种群. 以3目标DTLZ1为例, 在进化途中种群在目标空间中的聚类结果如图1所示.

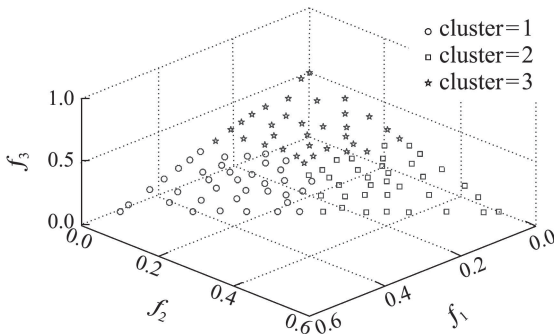


图1 3目标DTLZ1进化中聚类结果

Fig. 1 The cluster result of DTLZ1 of 3 objectives while evolutionary

### 3.2 决策变量聚类

现实中的问题是多种多样的. 对于有的问题来说, 其对维度变化非常敏感, 当某些维度有较小偏离时, 其适应度值便极度远离真实的Pareto前沿, 或者对整体均匀性造成很大的影响. 以以下多目标问题为例:

$$\begin{cases} \min f_1 = x_1 + x_2, \\ \min f_2 = 1 - x_1 + x_2, \\ \text{s.t. } x_i \in [0, 1], i = 1, 2. \end{cases} \quad (4)$$

分别对式(4)中  $x_1$  和  $x_2$  保持不变, 让另一个变量在定义域内均匀变化, 观察其在目标空间中的变化, 如图2所示.

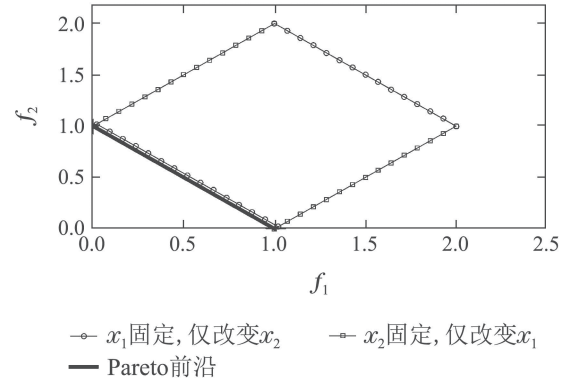


图2 分别改变  $x_1$  和  $x_2$  得到的适应度值分布

Fig. 2 The fitness distribution obtained by perturbing  $x_1$  and  $x_2$  respectively

由图2可见, 当分别改变  $x_1$  和  $x_2$  而固定其他维度时在目标空间中有着不同的影响, 显然,  $x_1$  影响MOP的收敛性, 而  $x_2$  影响多样性, 分别称为收敛性相关变量 (convergence variables, CV) 和多样性相关变量 (diversity variables, DV). 对于 CV, 需要找到一个最优值, 来支配其他所有值下对应的解; 而对于 DV, 需要使其合理分布, 促使对应的解在目标空间中均匀分布<sup>[13, 18]</sup>.

#### 算法3 VariableCluster( $P$ ).

输入:  $P$ (种群);

输出: DV(多样性相关变量); CV(收敛性相关变量);

1)  $\text{sample} \leftarrow$  随机从  $P$  中选择一个个体;

2)  $D \leftarrow$  决策变量维度;

3) **for**  $i = 1$  to  $D$  **do**

4)  $\text{sample} \leftarrow$  干扰  $\text{sample}$  第  $i$  维  $n$  次得到一个种群; /\*  $n$  为自定义变量(如20),  $\text{sample}$  是一个  $n \times D$  矩阵, 其仅第  $i$  列不同\*/

5)  $\text{FNo}[i] \leftarrow$   $\text{sample}$  进行非支配排序得到非支配前沿; //  $\text{FNo}[i]$  为对  $\text{sample}$  非支配排序后的结果;

6) **end for**

7)  $[\text{CV}, \text{DV}] \leftarrow$  对  $\text{FNo}$  进行  $k$ -means 聚类, 将决策变量聚为两类;

8) **if**  $\text{mean}(\text{FNo}[\text{CV}]) > \text{mean}(\text{FNo}[\text{DV}])$  **then** //  $\text{FNo}[\text{CV}]$  和  $\text{FNo}[\text{DV}]$  分别表示 CV 中决策变量和 DV 中决策变量所对应的前沿.

9)  $\text{CV} \leftarrow \text{DV}$ ;

10)  $\text{DV} \leftarrow \{j = 1, \dots, D | j \notin \text{CV}\}$ ;

11) **end if**

算法3为决策变量聚类的过程. 本文从定义出发, 对于每一个决策变量, 分别在其他决策变量不变的情况下对其进行扰动, 即由一个个体得到种群, 然后对

种群进行非支配排序, 则非支配序列便可反映决策变量的特征.

所有决策变量得到的非支配序列构成的矩阵反映了所有的特征. 严格来说, 收敛性相关变量得到种群的非支配序列个数为1, 而多样性相关变量的序列个数等于对应个体数. 因此, 本文采用 $k$ -means聚类将序列得到的矩阵分为两类, 较小的类别为收敛性相关变量. 较大的则为多样性相关变量.

### 3.3 变量独立性分析

决策变量规模增加时优化难度也随之增加, 但现实中往往需要解决较大规模的问题. 一直以来, 学者们致力于找到一种更好的方法来找到全局最优解, 另一种策略便是对问题进行简化, 将一个大规模优化问题分解为多个简单的小规模优化问题. 对于一个SOP, 称满足

$$\begin{aligned} \arg \min_{(x_1, \dots, x_n)} f(x_1, \dots, x_n) = \\ \left[ \arg \min_{x_1} f(x_1, \dots, x_n) \dots \right. \\ \left. \arg \min_{x_i} f(x_1, \dots, x_n) \dots \right. \\ \left. \arg \min_{x_n} f(x_1, \dots, x_n) \right] \end{aligned} \quad (5)$$

的函数为可分离函数, 否则称为不可分离函数.

对优化问题“分而治之”能够很大程度上提高优化效率, 其表示变量之间互不影响, 因此能够逐个对变量进行优化. 但问题的变量之间是否具有依赖性难以提前所知, 判断的方式为: 对于两个决策变量 $x_i$ 和 $x_j$ , 若存在 $a_1, a_2, b_1, b_2$ 满足

$$\begin{cases} f(x)|_{x_i=a_2, x_j=b_1} < f(x)|_{x_i=a_1, x_j=b_1} \wedge \\ f(x)|_{x_i=a_2, x_j=b_2} > f(x)|_{x_i=a_1, x_j=b_2}, \\ f(x)|_{x_i=a_2, x_j=b_1} = \\ f(x_1, \dots, x_{i-1}, a_2, \dots, x_{j-1}, b_1, \dots, x_n), \end{cases} \quad (6)$$

则称 $x_i$ 和 $x_j$ 为互不独立变量. 对于收敛性相关变量独立性分析算法见算法4.

**算法4** InteractionAnalysis( $P, CV$ ).

输入:  $P$ (种群);  $CV$ (收敛性相关变量);

输出: sub CVs(非独立的收敛性相关变量集合);

- 1) sub CVs  $\leftarrow \emptyset$ ;
- 2) **forall the**  $v \in CV$  **do**
- 3) CorSet  $\leftarrow \emptyset$ ;
- 4) **forall the** Group  $\in$  sub CVs **do**
- 5) **forall the**  $u \in$  Group **do**;
- 6) flag  $\leftarrow$  **false**;
- 7) **for**  $i = 1$  to  $nCor$  **do** //  $nCor$ 为自定义变量(如5)
- 8) **if**  $v$ 和 $u$ 非独立 **then**
- 9) flag  $\leftarrow$  **true**;

- 10) CorSet  $\leftarrow$  CorSet  $\cup$  {Group};
- 11) **break**;
- 12) **end if**
- 13) **end for**
- 14) **if** flag **then**
- 15) **break**;
- 16) **end if**
- 17) **end for**
- 18) **end for**
- 19) **if** CorSet ==  $\emptyset$  **then**
- 20) sub CVs  $\leftarrow$  sub CVs  $\cup$  { $v$ };
- 21) **else**
- 22) sub CVs  $\leftarrow$  sub CVs/CorSet
- 23) Group  $\leftarrow$  CorSet和 $v$ 的所有变量;
- 24) sub CVs  $\leftarrow$  sub CVs  $\cup$  {Group};
- 25) **end if**

### 3.4 变量优化

本文将决策变量分为收敛性相关变量和多样性相关变量, 对其进行不同的优化策略对算法的进化结构有着显著的提升. 算法5为收敛性优化过程, 为了增加选择压力, 使种群更快地趋近于Pareto前沿, 本文采用加权边界相交 (penalty-based boundary intersection, PBI)的策略来评价一个解的优劣<sup>[12]</sup>. PBI策略能将个体的多个目标值转换为一个目标值, 其计算公式为

$$\begin{cases} \min g^{pbi}(x|w, z^{\min}) = d_1 + \theta d_2, \\ d_1 = \frac{\|(F(x) - z^{\min})^T w\|}{\|w\|}, \\ d_2 = \|F(x) - (z^{\min} + d_1 \frac{w}{\|w\|})\|, \end{cases} \quad (7)$$

其中:  $\theta$ 为自定义的惩罚参数, 用来权衡收敛性与多样性. 可见,  $d_1$ 为所有目标构成的点 $f$ 距离所有目标最小值构成的点的距离, 能够衡量 $x$ 的收敛程度,  $d_2$ 表示点 $f$ 与权重 $w$ 的间距, 能够衡量 $x$ 的分布特性, 因此PBI能够更加全面地用一个值来衡量一个解的优劣程度.

**算法5** ConverOptimi( $P, \text{sub CVs}, Z_{\min}$ ).

输入:  $P$ (种群), sub CVs(非独立的收敛性相关变量集合),  $Z_{\min}$ (当前种群中每个目标的最小值);

输出:  $P$ (新种群);

- 1) **for**  $i = 1$  to  $N$  **do** //  $N$ 为种群规模;
- 2) **forall the**  $j \in$  sub CVs **do**
- 3) [ $p_1, p_2$ ]  $\leftarrow$  根据PBI值以轮盘赌的方式选择父代;
- 4)  $p' \leftarrow$  SBX( $p_1[j], p_2[j]$ ); //产生子代
- 5) offspring  $\leftarrow$   $s[j] = p'$ ; //  $s$ 为 $P$ 的第 $i$ 个个体;

6)  $P \leftarrow$  通过比较  $s$  和 offspring 的 PBI 值, 保留较好的一个;  
 7) **end for**  
 8) **end for**

算法5以个体的PBI值为标准以轮盘赌的方式选择父代, 通过模拟二进制交叉(simulated binary crossover, SBX)来产生新个体, 在种群更新方面, 通过比较父代与新个体的PBI值保留较好的进入新的种群.

算法6为多样性优化的过程. 与原始BSO的机制类似, 算法以聚类中心为导向, 并以概率选择其他个体作为当前个体, 通过变异实现对个体周围的搜索. 在得到新的种群后, 将原种群与新种群进行合并, 通过计算与参考点的夹角来对种群进行更新.

此外, 为了避免不必要的搜索, 本文设置了收敛优化指标. 当种群的收敛性相关变量的方差的平均值小于某一阈值时停止收敛性优化. 值得注意的是, 在收敛性优化和多样性优化时, 分别只改变收敛性相关变量和多样性相关变量.

**算法 6** DiverOptimi( $P, P_o, P_c, DV, w, cluster, Z_{min}, Z^{nad}$ ).

**输入:**  $P$ (当前种群),  $P_o$ (用一个个体产生新个体的概率),  $P_c$ (用聚类中心产生新个体的概率),  $DV$ (多样性相关变量),  $w$ (参考点),  $cluster$ (聚类结果),  $Z_{min}$ (当前种群中每个目标上的最小值),  $Z^{nad}$ (Pareto前沿在每个目标上的截距);

**输出:**  $P$ (新种群);

1) **for**  $i = 1$  to  $N$  **do** //  $N$  为种群规模;  
 2) **if** random  $< P_o$  **then**  
 3) **if** random  $< P_c$  **then**  
 4)  $p_{select} \leftarrow s$ 所在类的角点; //  $s$  是  $P$  的第  $i$  个个体  
 5) **else**  
 6)  $p_{select} \leftarrow$  随机从  $s$  所在类选择一个个

体;  
 7) **end if**  
 8) **else**  
 9)  $p_{select} \leftarrow$  随机从  $P$  中选择两个个体并加权求和;  
 10) **end if**  
 11) offspring[ $i$ ]  $\leftarrow$  对  $p_{select}$  进行变异 // 仅对 DV 变异  
 12) **end for**  
 13) POP  $\leftarrow P \cup$  offspring; // POP 为将原种群和子种群进行合并后的种群子集;  
 14)  $P \leftarrow$  根据参考点  $w$  对 POP 进行选择, 更新种群  $P$ .

### 4 仿真结果与分析

为了测试本文算法的性能, 本文分别选择ZDT和DTLZ测试函数集来评价算法解决MOP和MaOP的性能, 以反演世代距离(inverted generational distance, IGD)<sup>[27]</sup>为指标来综合比较本文算法与几种经典算法的性能.

#### 4.1 多目标优化问题的性能

本文采用ZDT测试函数集来评价算法解决MOP的性能, 并与算法MOBSO, MMBSO, SMOBSO以及经典算法NSGA-II进行综合比较. 其中, 选用30维的ZDT1-ZDT4和ZDT6; 归档集大小和总评价次数统一为51和20000; 细节方面, MOBSO, MMBSO和SMOBSO分别设  $p_1, p_2, p_3, p_4$  为0.8, 0.8, 0.2, 0.2, 聚类个数为4, NSGA-II采用模拟二进制交叉和多项式变异, MaOBSO分别设  $P_o$  和  $P_c$  为0.8, 0.3. 拥挤距离策略在处理双目标优化问题时非常高效且操作简单, 因此在测试算法解决MOP能力时, 采用拥挤距离策略来进行环境选择. 算法运行30次, 记录平均值以及标准如表1所示, 其中, 加粗以及加黑的结果表示最优的性能.

表 1 MaOBSO与经典算法解决MOP时IGD比较

Table 1 The comparison of IGD between MaOBSO and classic algorithms while solving MOP

| 测试函数 | MOBSO <sup>[21]</sup> | MMBSO <sup>[22]</sup> | SMOBSO <sup>[23]</sup> | NSGA-II <sup>[16]</sup> | MaOBSO           |
|------|-----------------------|-----------------------|------------------------|-------------------------|------------------|
| ZDT1 | 1.7758e-2             | 9.4890e-3             | 1.2479e-2              | 9.5487e-3               | <b>7.6230e-3</b> |
|      | (7.99e-6)             | (1.34e-4)             | (2.57e-6)              | (1.99e-2)               | <b>(1.18e-4)</b> |
| ZDT2 | 2.4656e-2             | 3.1281e-1             | 1.1160e-2              | 1.9453e-2               | <b>7.7004e-3</b> |
|      | (2.05e-5)             | (8.60e-2)             | (4.45e-2)              | (2.43e-2)               | <b>(1.18e-4)</b> |
| ZDT3 | 1.7336e-2             | 1.5343e-2             | 1.2957e-2              | 6.6734e-2               | <b>9.2050e-3</b> |
|      | (8.53e-6)             | (8.53e-6)             | (6.12e-6)              | (7.77e-2)               | <b>(2.24e-4)</b> |
| ZDT4 | 7.4364e-2             | 6.1547e-1             | 1.2509e-1              | 1.3578e-2               | <b>8.9753e-3</b> |
|      | (2.22e-2)             | (7.10e-2)             | (5.63e-1)              | (1.52e-2)               | <b>(1.10e-3)</b> |
| ZDT5 | 2.1277e-1             | 7.8331e-2             | 1.4387e-2              | 4.9961e-3               | <b>4.5485e-3</b> |
|      | (3.79e-5)             | (2.33e-6)             | (9.59e-6)              | (1.02e-3)               | <b>(3.67e-4)</b> |



由表1可见,与其他算法相比, MaOBSO表现出非常优异的性能. 对于较为常规的ZDT1和ZDT2, MaOBSO较大幅度领先于经典版本的多目标BSO, 与经典算法NSGA-II相比也有所领先; 对于有着不连续前沿的ZDT3, 算法在收敛性能和多样性能优化方面均难度较大, MaOBSO性能提升更加明显; 对于具有

大量局部极值点的ZDT4和非均匀函数ZDT6, 其他算法综合性能变差, 而MaOBSO仍能够保持非常优异的性能. 为了更加直接地观察算法得到的Pareto最优解, 以ZDT1, ZDT3, ZDT4为例, 取30次中最接近IGD平均值的一次结果如图3所示.

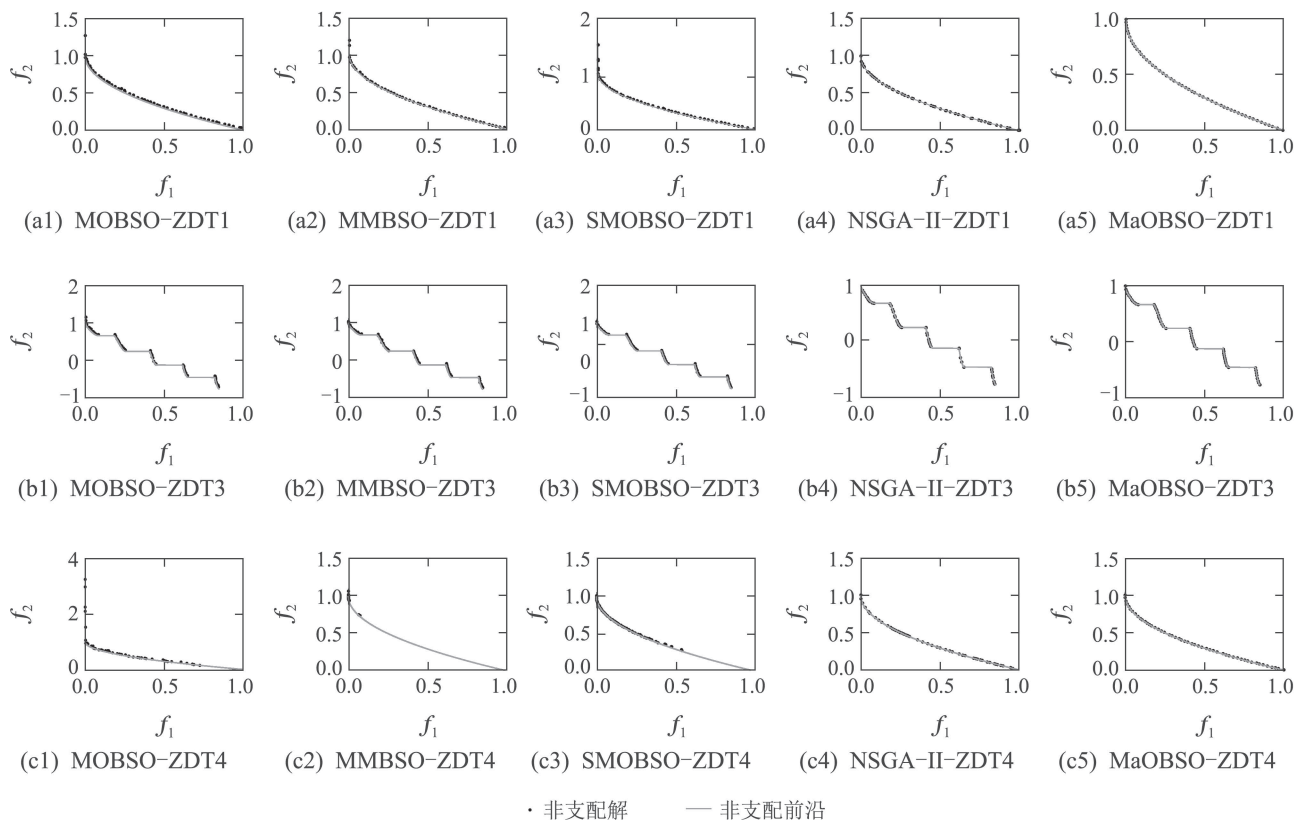


图3 MaOBSO与经典算法解决MOP时的Pareto最优解集

Fig. 3 The Pareto set of MaOBSO and classic algorithms while solving MOP

由图3可见,对于几个测试函数而言, MaOBSO从综合性能上有着最优异的表现. 从收敛性角度看, MaOBSO在ZDT1, ZDT3上略微领先于其他算法, 在ZDT1以及SMOBSO处理ZDT3时更加明显. 对于ZDT4, MaOBSO与相比其他算法有着较为明显的提升, 对比几种版本的头脑风暴优化算法可见, MMBSO中差分变异相比MOBSO的高斯变异能够更加积极地促进算法的收敛性, 而SMOBSO引入的开放性概率对于算法收敛起到了进一步的促进作用. 从多样性角度来看, MaOBSO在ZDT1上略微领先于其他算法, 对于ZDT3而言, MMBSO, SMOBSO和MaOBSO性能优于MOBSO和NSGA-II, 表明了循环拥挤距离在维持多样性方面领先于拥挤距离. 而对于ZDT4而言, 即使MMBSO和SMOBSO采用了循环拥挤距离, 但依然有着较差的多样性能, 可见算法的收敛性能较差时对多样性能有着一定程度的影响.

综上分析, MaOBSO在处理MOP时从收敛性能和多样性能方面均有较大幅度的提升, 当优化难度增加

时更为明显, 表明了采用决策变量聚类策略在改进算法进化结构方面的合理性与高效性, 且具有更大的潜力. 此外, 从稳定性角度来说, MaOBSO处理MOP时IGD指标标准差较小, 表明了算法具有较好的稳定性.

#### 4.2 高维多目标优化问题的性能

本文采用经典的DTLZ测试函数集来评价算法解决MaOP的性能, 并与当前较为流行的算法MOEA/DD, NSGA-III, LMEA, MOEA/DVA以及RVEA进行综合比较. 对于DTLZ测试函数集而言, 决策维度 $D = K + M - 1$ , 其中:  $M$ 为目标个数,  $K$ 为难度因子. 本文对于DTLZ1设 $K = 5$ , DTLZ2-DTLZ6设 $K = 10$ , DTLZ7设 $K = 20$ ; 分别测试 $M = 3, 5, 8, 10$ 下的性能, 归档集大小分别设置为105, 126, 72, 65, 最大迭代次数统一设为100000; 细节方面, MOEA/DD中设 $\delta = 0.9$ , RVEA中设 $\alpha = 2$ ,  $f_r = 0.1$ ; 算法运行30次, 记录平均值以及标准差如表2所示, 其中, 加粗以及加黑的结果表示最优的性能. 除了本文算法外, 其他算法在高维多目标优化算法platEMO<sup>[28]</sup>上运行.

表2 MaOBOS与几种算法IGD的比较  
Table 2 The comparison of IGD between MaOBOS and several algorithms

| 测试函数  | 目标个数 | MOEA/DD <sup>[12]</sup>               | NSGA-III <sup>[18]</sup>        | LMEA <sup>[19]</sup>                  |
|-------|------|---------------------------------------|---------------------------------|---------------------------------------|
| DTLZ1 | 3    | 2.0559e-2(2.87e-6) <sub>2</sub>       | 2.0566e-2(9.53e-6) <sub>4</sub> | 2.0943e-2(2.28e-4) <sub>6</sub>       |
|       | 5    | 6.8087e-2(6.91e-5) <sub>4</sub>       | 6.8113e-2(7.78e-5) <sub>5</sub> | 6.6469e-2(6.63e-4) <sub>2</sub>       |
|       | 8    | <b>1.0850e-1(1.96e-4)<sub>1</sub></b> | 1.0878e-1(1.71e-3) <sub>2</sub> | 1.1506e-1(2.56e-3) <sub>4</sub>       |
|       | 10   | 1.5336e-1(6.44e-4) <sub>4</sub>       | 1.8203e-1(4.49e-2) <sub>5</sub> | <b>1.3449e-1(6.11e-3)<sub>1</sub></b> |
| DTLZ2 | 3    | 5.4464e-2(1.99e-7) <sub>3</sub>       | 5.4465e-2(1.65e-5) <sub>4</sub> | 5.3739e-2(4.69e-4) <sub>2</sub>       |
|       | 5    | 2.1221e-1(3.43e-6) <sub>3</sub>       | 2.1221e-1(5.98e-6) <sub>3</sub> | 2.0430e-1(1.62e-3) <sub>2</sub>       |
|       | 8    | 3.8693e-1(1.13e-5) <sub>2</sub>       | 4.1907e-1(6.62e-2) <sub>4</sub> | <b>3.8290e-1(2.51e-3)<sub>1</sub></b> |
|       | 10   | 5.0027e-1(1.07e-4) <sub>3</sub>       | 5.6664e-1(7.69e-2) <sub>5</sub> | <b>4.7109e-1(1.76e-3)<sub>1</sub></b> |
| DTLZ3 | 3    | 5.4607e-2(1.64e-4) <sub>3</sub>       | 5.4784e-2(3.36e-4) <sub>5</sub> | 5.4159e-2(7.59e-4) <sub>2</sub>       |
|       | 5    | 2.1261e-1(3.00e-4) <sub>3</sub>       | 2.1334e-1(9.13e-4) <sub>5</sub> | 2.0529e-1(1.61e-3) <sub>2</sub>       |
|       | 8    | 3.8943e-1(1.65e-3) <sub>3</sub>       | 1.1330e+0(1.21e+0) <sub>6</sub> | <b>3.8309e-1(2.33e-3)<sub>1</sub></b> |
|       | 10   | 5.0210e-1(1.44e-3) <sub>3</sub>       | 3.4232e+0(2.43e+0) <sub>6</sub> | 4.7692e-1(3.13e-2) <sub>2</sub>       |
| DTLZ4 | 3    | 7.0779e-2(8.94e-2) <sub>3</sub>       | 1.3569e-1(1.85e-1) <sub>5</sub> | 7.2436e-2(5.75e-2) <sub>4</sub>       |
|       | 5    | 2.2704e-1(5.64e-2) <sub>2</sub>       | 2.7401e-1(1.04e-1) <sub>4</sub> | 2.7677e-1(1.32e-1) <sub>5</sub>       |
|       | 8    | 4.1241e-1(4.32e-2) <sub>2</sub>       | 5.1010e-1(1.13e-1) <sub>6</sub> | 5.0814e-1(1.31e-1) <sub>5</sub>       |
|       | 10   | 5.3488e-1(5.48e-2) <sub>2</sub>       | 6.2561e-1(9.46e-2) <sub>5</sub> | 5.3721e-1(9.28e-2) <sub>3</sub>       |
| DTLZ5 | 3    | 3.1367e-2(7.90e-4) <sub>5</sub>       | 1.2129e-2(1.87e-3) <sub>3</sub> | 4.6623e-3(1.70e-4) <sub>2</sub>       |
|       | 5    | 8.7960e-2(1.01e-2) <sub>3</sub>       | 1.1260e-1(5.21e-2) <sub>5</sub> | <b>5.5556e-3(3.47e-4)<sub>1</sub></b> |
|       | 8    | 1.9294e-1(3.15e-2) <sub>4</sub>       | 3.3088e-1(1.02e-1) <sub>5</sub> | <b>6.8352e-3(5.58e-4)<sub>1</sub></b> |
|       | 10   | 1.5320e-1(1.60e-3) <sub>3</sub>       | 3.1655e-1(1.07e-1) <sub>5</sub> | <b>7.5308e-3(5.34e-4)<sub>1</sub></b> |
| DTLZ6 | 3    | 3.3817e-2(5.48e-4) <sub>4</sub>       | 1.9818e-2(2.51e-3) <sub>3</sub> | <b>4.5812e-3(1.31e-4)<sub>1</sub></b> |
|       | 5    | 9.1180e-2(1.38e-2) <sub>5</sub>       | 2.6499e-1(6.12e-2) <sub>4</sub> | <b>5.0916e-3(3.94e-4)<sub>1</sub></b> |
|       | 8    | 1.8747e-1(3.05e-2) <sub>4</sub>       | 1.0875e+0(7.60e-1) <sub>6</sub> | <b>5.7316e-3(8.31e-4)<sub>1</sub></b> |
|       | 10   | 1.5384e-1(2.84e-4) <sub>3</sub>       | 3.0677e+0(1.30e+0) <sub>6</sub> | <b>6.3166e-3(1.38e-3)<sub>1</sub></b> |
| DTLZ7 | 3    | 5.2702e-1(2.50e-1) <sub>6</sub>       | 7.6846e-2(3.46e-3) <sub>2</sub> | <b>5.8590e-2(1.11e-3)<sub>1</sub></b> |
|       | 5    | 2.7686e+0(5.87e-1) <sub>6</sub>       | 3.7384e-1(1.70e-2) <sub>2</sub> | <b>3.4907e-1(6.68e-3)<sub>1</sub></b> |
|       | 8    | 1.9071e+0(5.04e-1) <sub>5</sub>       | 9.4230e-1(5.48e-2) <sub>3</sub> | <b>7.5623e-1(2.17e-2)<sub>1</sub></b> |
|       | 10   | 2.4667e+0(3.56e-1) <sub>4</sub>       | 1.6996e+0(2.14e-1) <sub>3</sub> | <b>1.0680e+0(2.98e-2)<sub>1</sub></b> |
| 排名    | —    | 95(3)                                 | 122(5)                          | 56(1)                                 |
| 测试函数  | 目标个数 | MOEA/DVA <sup>[13]</sup>              | RVEA <sup>[17]</sup>            | MaOBOS                                |
| DTLZ1 | 3    | 2.0699e-2(7.18e-5) <sub>5</sub>       | 2.0561e-2(8.00e-6) <sub>3</sub> | <b>1.8988e-2(7.93e-6)<sub>1</sub></b> |
|       | 5    | 7.9811e-2(2.41e-3) <sub>6</sub>       | 6.8081e-2(2.76e-5) <sub>3</sub> | <b>6.3105e-2(6.50e-5)<sub>1</sub></b> |
|       | 8    | 1.4851e-1(1.29e-2) <sub>6</sub>       | 1.0947e-1(2.56e-3) <sub>3</sub> | 1.1931e-1(6.58e-4) <sub>5</sub>       |
|       | 10   | 1.9988e-1(2.05e-2) <sub>6</sub>       | 1.5033e-1(3.41e-3) <sub>3</sub> | 1.4306e-1(7.89e-4) <sub>2</sub>       |
| DTLZ2 | 3    | 5.6964e-2(1.96e-8) <sub>6</sub>       | 5.4464e-2(4.15e-7) <sub>3</sub> | <b>5.0469e-2(7.16e-5)<sub>1</sub></b> |
|       | 5    | 2.3852e-1(3.49e-3) <sub>6</sub>       | 2.1222e-1(8.60e-6) <sub>5</sub> | <b>1.9425e-1(4.29e-4)<sub>1</sub></b> |
|       | 8    | 5.1762e-1(3.64e-2) <sub>6</sub>       | 3.8708e-1(1.13e-4) <sub>3</sub> | 4.4655e-1(3.18e-2) <sub>5</sub>       |
|       | 10   | 7.1274e-1(2.25e-2) <sub>6</sub>       | 5.0630e-1(1.47e-2) <sub>4</sub> | 4.7322e-1(1.42e-3) <sub>2</sub>       |
| DTLZ3 | 3    | 3.1342e-1(6.15e-2) <sub>6</sub>       | 5.4728e-2(6.07e-4) <sub>4</sub> | <b>5.0465e-2(7.56e-5)<sub>1</sub></b> |
|       | 5    | 6.7678e-1(2.86e-1) <sub>6</sub>       | 2.1284e-1(7.59e-4) <sub>4</sub> | <b>1.9422e-1(3.30e-4)<sub>1</sub></b> |
|       | 8    | 6.6129e-1(7.48e-2) <sub>5</sub>       | 3.8887e-1(1.71e-3) <sub>2</sub> | 4.1955e-1(9.87e-4) <sub>4</sub>       |
|       | 10   | 9.0747e-1(5.46e-2) <sub>5</sub>       | 5.1413e-1(2.66e-2) <sub>4</sub> | <b>4.7353e-1(1.16e-3)<sub>1</sub></b> |

(转下页)



(接上页)

|       |    |                                       |                                 |                                       |
|-------|----|---------------------------------------|---------------------------------|---------------------------------------|
| DTLZ4 | 3  | 4.7026e-1(5.04e-7) <sub>6</sub>       | 5.4464e-2(6.53e-7) <sub>2</sub> | <b>5.0776e-2(1.86e-4)<sub>1</sub></b> |
|       | 5  | 3.1383e-1(3.18e-2) <sub>6</sub>       | 2.4085e-1(7.42e-2) <sub>3</sub> | <b>2.2464e-1(1.18e-2)<sub>1</sub></b> |
|       | 8  | 4.8370e-1(1.33e-2) <sub>4</sub>       | 4.6015e-1(9.46e-2) <sub>3</sub> | <b>4.0265e-1(1.04e-3)<sub>1</sub></b> |
|       | 10 | 9.0392e-1(6.29e-2) <sub>6</sub>       | 5.4121e-1(4.86e-2) <sub>4</sub> | <b>4.9896e-1(1.89e-3)<sub>1</sub></b> |
| DTLZ5 | 3  | <b>4.3628e-3(3.47e-8)<sub>1</sub></b> | 6.2711e-2(2.71e-3) <sub>6</sub> | 3.1202e-2(5.21e-5) <sub>4</sub>       |
|       | 5  | 9.0785e-2(1.50e-2) <sub>4</sub>       | 2.3237e-1(4.29e-2) <sub>6</sub> | 3.5070e-2(1.05e-1) <sub>2</sub>       |
|       | 8  | 1.8651e-1(1.49e-2) <sub>3</sub>       | 6.0644e-1(1.65e-1) <sub>6</sub> | 6.7266e-2(1.82e-1) <sub>2</sub>       |
|       | 10 | 2.3895e-1(6.33e-2) <sub>4</sub>       | 3.9476e-1(1.74e-1) <sub>6</sub> | 6.0805e-2(2.32e-1) <sub>2</sub>       |
| DTLZ6 | 3  | 1.5501e-2(2.11e-3) <sub>2</sub>       | 9.1442e-2(2.21e-2) <sub>6</sub> | 8.3430e-2(5.55e-2) <sub>5</sub>       |
|       | 5  | 7.2658e-2(8.00e-3) <sub>2</sub>       | 2.2645e-1(5.17e-2) <sub>3</sub> | 4.3081e-1(1.56e-1) <sub>6</sub>       |
|       | 8  | 1.7722e-1(2.25e-2) <sub>3</sub>       | 3.0836e-1(1.11e-1) <sub>5</sub> | 8.1539e-2(2.72e-1) <sub>2</sub>       |
|       | 10 | 1.7452e+0(1.46e+0) <sub>5</sub>       | 1.9113e-1(3.51e-2) <sub>4</sub> | 1.0914e-1(2.77e-1) <sub>2</sub>       |
| DTLZ7 | 3  | 1.5038e-1(5.12e-3) <sub>4</sub>       | 1.0535e-1(1.43e-3) <sub>3</sub> | 3.3447e-1(1.98e-3) <sub>5</sub>       |
|       | 5  | 6.8603e-1(4.61e-2) <sub>5</sub>       | 5.0684e-1(4.41e-3) <sub>3</sub> | 6.3434e-1(1.16e-2) <sub>4</sub>       |
|       | 8  | 8.6512e-1(3.24e-2) <sub>2</sub>       | 1.7343e+0(2.41e-1) <sub>4</sub> | 3.4529e+0(1.19e-2) <sub>6</sub>       |
|       | 10 | 1.1473e+0(3.15e-2) <sub>2</sub>       | 3.1222e+0(5.37e-1) <sub>5</sub> | 4.5369e+0(1.87e-1) <sub>6</sub>       |
| 排名    | —— | 128(6)                                | 110(4)                          | 75(2)                                 |

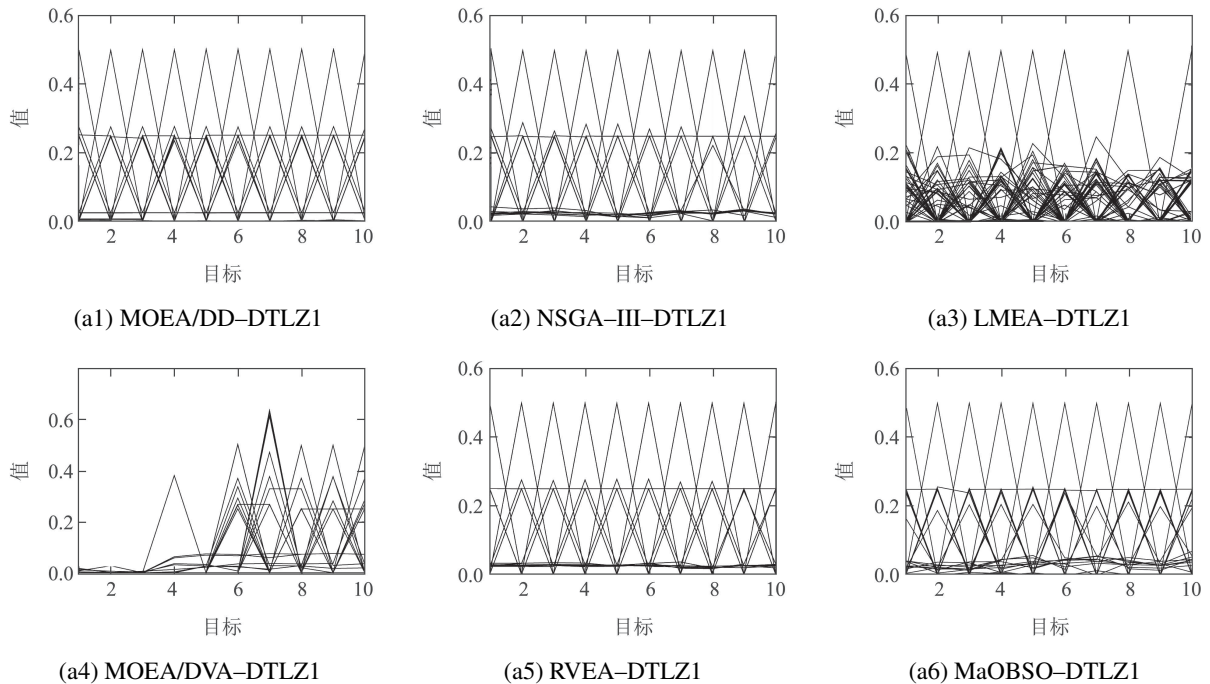
由表2可见, MaOBSO与其他算法相比整体表现出较为理想的性能. 与现有的几种算法相比, 其处于同等数量级且表现略微优异. 具体分析如下:

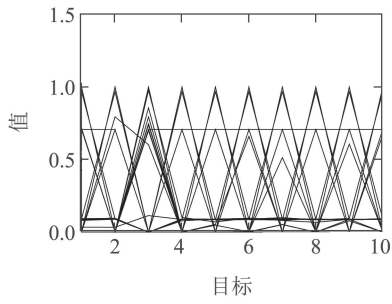
对于同一测试函数而言, 算法的IGD指标随着目标个数增加而增加, 验证了目标个数的增加对优化带来的挑战性. 可见当目标个数的增加导致种群更加离散地分布在在高维目标空间中, 导致了种群的多样性维护难度增加.

对于不同测试函数而言, 大多数算法在处理 DTLZ1-DTLZ4有着更优秀的性能, 因为他们有着更加规范的 Pareto 前沿, 算法进化过程中更容易得到 Pareto 最优解, 且种群的多样性维护更为容易. 而对于 DTLZ5-DTLZ7, 算法整体性能较差, 且受目标个数的影响更大.

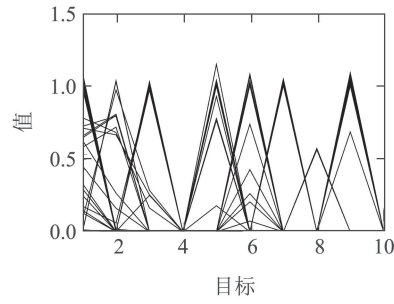
对于不同算法而言, 在处理具有较为规范的 Pareto 前沿的 DTLZ1-DTLZ4时, MaOBSO与几种经典算法性能处于同一数量级, 但由于本文算法在处理均匀性优化时有着更高的效率, 最终性能更加优秀. 特别是对于 DTLZ4, MaOBSO性能提升更为明显. 而在处理 DTLZ5-DTLZ7方面, LMEA与其他算法以及本文算法有着不同的保留机制, 具有更强的适应性, 最终得到更加均匀的前沿, 算法性能最好且更加稳定. 但相比于其他算法, MaOBSO性能更优.

从算法稳定性来讲, MaOBSO整体标准差较小, 表明算法有着较好的稳定性. 为了更加直接地观察算法得到的 Pareto 最优解, 以 10 目标 DTLZ1, DTLZ4, DTLZ5及DTLZ7为例, 取30次中最接近IGD平均值的结果对应的Pareto前沿的平行坐标如图4所示.

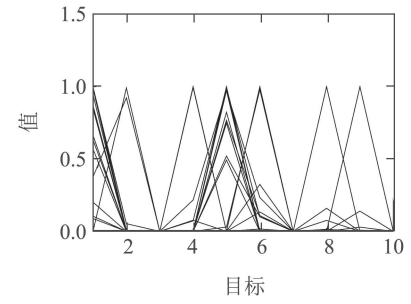




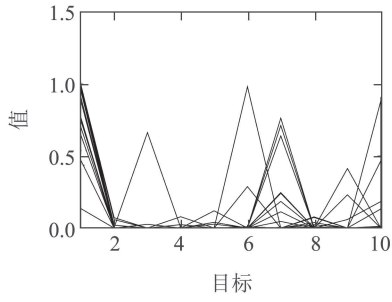
(b1) MOEA/DD-DTLZ4



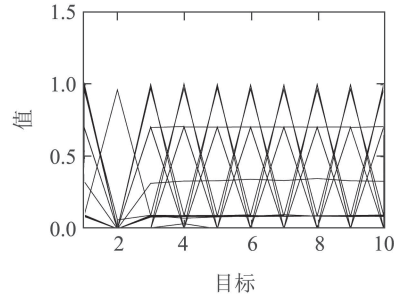
(b2) NSGA-III-DTLZ4



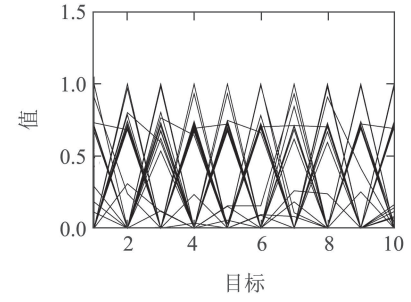
(b3) LMEA-DTLZ4



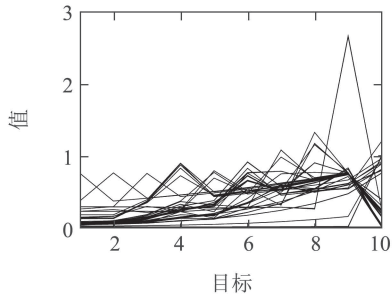
(b4) MOEA/DVA-DTLZ4



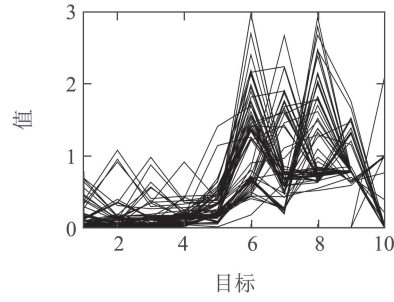
(b5) RVEA-DTLZ4



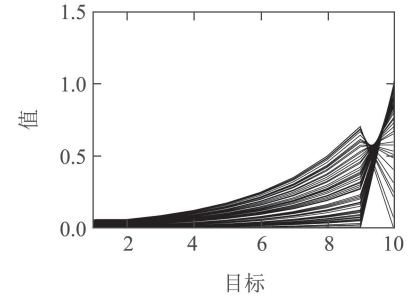
(b6) MaOBOS-DTLZ4



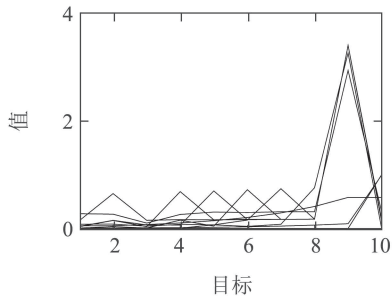
(c1) MOEA/DD-DTLZ5



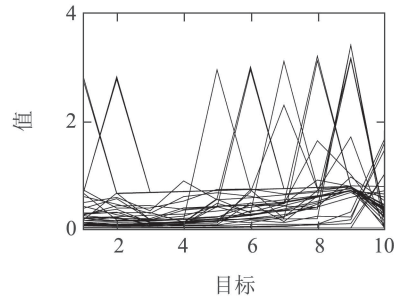
(c2) NSGA-III-DTLZ5



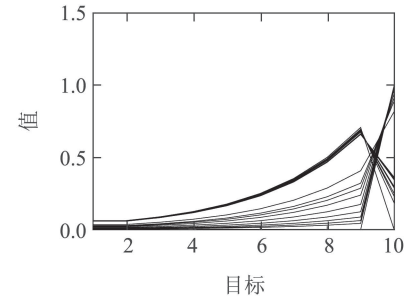
(c3) LMEA-DTLZ5



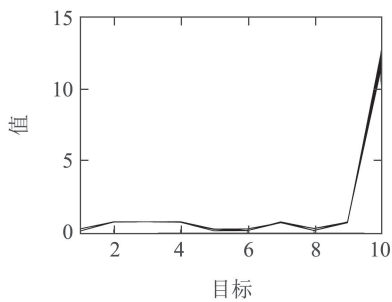
(c4) MOEA/DVA-DTLZ5



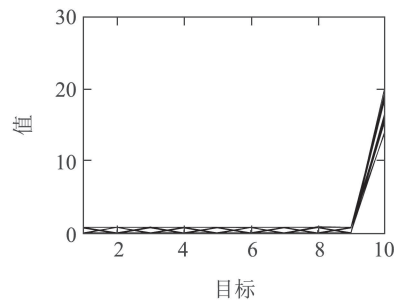
(c5) RVEA-DTLZ5



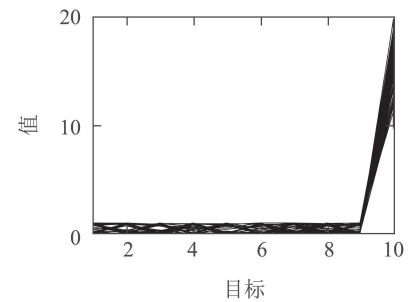
(c6) MaOBOS-DTLZ5



(d1) MOEA/DD-DTLZ7



(d2) NSGA-III-DTLZ7



(d3) LMEA-DTLZ7

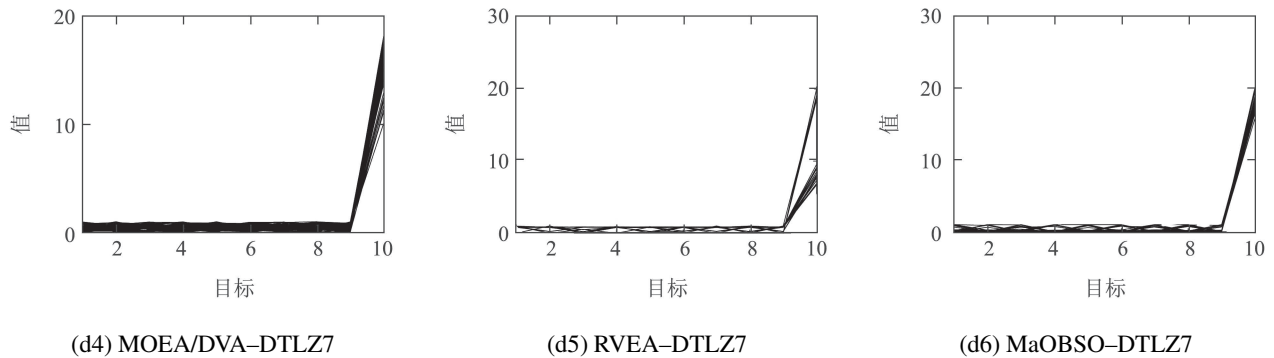


图4 MaOBSO与经典算法解决MaOP时的Pareto最优解集

Fig. 4 The Pareto set of MaOBSO and classic algorithms while solving MaOP

由图4可以看出, MaOBSO在4种测试函数上与其他几种算法性能各有胜负. 对于DTLZ1, 所有算法都表现出良好的性能, 尽管几种算法的IGD指标较为接近, 但从图4(a1)–(a6)可明显看到, LMEA均匀性稍差, MOEA/DVA收敛性稍差, 多样性较差; 对于均匀性优化难度较大的DTLZ4, 从收敛性角度看, 几种算法均有着良好的收敛性能, NSGA-III稍差; 从多样性角度来看, NSGA-III, LMEA和MOEA/DVA性能较差, 最优解集在目标空间中分布的广度以及均匀程度均不理想, 算法得到的Pareto最优解大多数分布于边界上, 且在某些目标上存在“缺失”现象, 难以使得最优解集广泛地分布于整个非支配前沿上. 而MaOBSO由于能够不受约束地产生新个体则表现出最优的性能, 与其相比, 而MOEA/DD和RVEA性能稍差; 对于具有退化前沿的DTLZ5和不连续前沿的DTLZ7, LMEA表现出最优的性能, 尽管MaOBSO难以达到理想的水平, 但与其他算法相比仍表现出良好的竞争力, 其中对于DTLZ5在收敛性能和多样性能方面均较大程度领先于其他算法, 而在DTLZ7上与其他算法相仿.

综上所述, MaOBSO能够较好地处理各种MaOP, 与当前流行的几种算法相比表现出良好的竞争力. 对具有规则前沿的(高维)多目标优化问题, MaOBSO与其他算法同样表现出良好的性能且更加优异, 而对于具有不规则前沿的问题, 算法在多样性方面稍逊于LMEA算法, 但整体性能略微领先于其他算法.

## 5 总结与展望

本文针对高维多目标优化算法的收敛性和多样性问题, 结合头脑风暴优化算法的思想, 提出了一种高维多目标头脑风暴优化算法. 算法采用决策变量聚类策略, 对收敛性相关变量和多样性相关变量分别做出优化, 并根据变量特征采用不同的个体产生策略和更新策略, 融入了头脑风暴优化算法产生新个体的机制, 仿真结果表明了算法具有良好的性能. 此外, 本文根据角点的特性采用了基于角点聚类的思想, 对于多

目标头脑风暴优化算法提出了一种新的自适应聚类策略, 为之后的研究拓宽了方向.

在算法仿真研究时, 作者发现本文基于参考点更新的方式对不规则Pareto前沿问题的性能还有待进一步的提升. 因此, 如何设计不规则前沿的多样性和收敛性更新策略, 提高算法的普适性, 并将其应用于实际的高维多目标优化问题将是作者今后进一步研究的方向.

## 参考文献:

- [1] KOLLAT J B, REED P M, MAXWELL R M. Many-objective groundwater monitoring network design using bias-aware ensemble Kalman filtering, evolutionary optimization, and visual analytics. *Water Resources Research*, 2010, 47(2): 155 – 170.
- [2] FREIRE H, OLIVEIRA P B M, PIRES E J S. From single to many-objective PID controller design using particle swarm optimization. *International Journal of Control Automation & Systems*, 2017, 15(2): 1 – 15.
- [3] WANG G G, HAO G S, CHENG S, et al. An improved brain storm optimization algorithm based on graph theory. *IEEE Congress on Evolutionary Computation*. Donostia, San Sebastián, Spain: IEEE, 2017: 509 – 515.
- [4] ISHIBUCHI H, TSUKAMOTO N, NOJIMA Y. Evolutionary many-objective optimization: a short review. *IEEE Congress on Evolutionary Computation*. Hong Kong, China: IEEE, 2008: 2419 – 2426.
- [5] LAUMANN S M, THIELE L, DEB K, et al. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 2002, 10(3): 263 – 282.
- [6] ZOU X F, CHEN Y, LIU M Z, et al. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Transactions on Systems Man & Cybernetics Part B*, 2008, 38(5): 1402 – 1412.
- [7] WANG G P, JIANG H W. Fuzzy-dominance and its application in evolutionary many objective optimization. *International Conference on Computational Intelligence and Security Workshops*. Harbin, China: IEEE, 2007: 195 – 198.
- [8] YANG S X, LI M Q, LIU X H, et al. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2013, 17(5): 721 – 736.
- [9] YUAN Y, XU H, WANG B, et al. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, 20(1): 16 – 37.

- [10] TIAN Y, CHENG R, ZHANG X Y, et al. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2019, 23(2): 331 – 345.
- [11] ZHANG Q F, LI H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712 – 731.
- [12] LI K, DEB K, ZHANG Q F, et al. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 2015, 19(5): 694 – 716.
- [13] 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. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 275 – 298.
- [14] ZITZLER E, KÜNZLI S. Indicator-based selection in multiobjective search. *International Conference on Parallel Problem Solving from Nature*. Berlin, Germany: Springer, 2004: 832 – 842.
- [15] BADER J, ZITZLER E. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 2011, 19(1): 1 – 25.
- [16] DEB K, PRATAP A, AGAWALI S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [17] CHENG R, JIN Y C, OLHOFFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 773 – 791.
- [18] 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 Transactions on Evolutionary Computation*, 2014, 18(4): 577 – 601.
- [19] ZHANG X Y, TIAN Y, CHENG R, et al. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 97 – 112.
- [20] SHI Y H. Brain storm optimization algorithm. *Advances in Swarm Intelligence*. Berlin, Heidelberg: Springer, 2011: 303 – 309.
- [21] XUE J Q, WU Y L, SHI Y H, et al. Brain storm optimization algorithm for multi-objective optimization problems. *Lecture Notes in Computer Science*, 2012, 7331(4): 513 – 519.
- [22] XIE L X, WU Y L. A modified multi-objective optimization based on brain storm optimization algorithm. *International Conference in Swarm Intelligence*. Hefei, China: Springer, 2014: 328 – 339.
- [23] GUO X P, WU Y L, XIE L X, et al. An adaptive brain storm optimization algorithm for multiobjective optimization problems. *Lecture Notes in Computer Science*, 2015, 9140(4): 365 – 372.
- [24] WU Yali, JIAO Shangbin. *Brain Storm Optimization Algorithm Theory and Application*. Beijing: Science Press, 2017. (吴亚丽, 焦尚斌. 头脑风暴优化算法理论及应用. 北京: 科学出版社, 2017.)
- [25] ZHAN Z H, ZHANG J, SHI Y H, et al. A modified brain storm optimization. *IEEE Congress on Evolutionary Computation*. Brisbane, Australia: IEEE, 2012: 1 – 8.
- [26] WU Yali, FU Yulong, WANG Xinrui, et al. Difference brain storm optimization algorithm based on clustering in objective space. *Control Theory & Applications*, 2017, 34(12): 1583 – 1593. (吴亚丽, 付玉龙, 王鑫睿, 等. 目标空间聚类的差分头脑风暴优化算法. 控制理论与应用, 2017, 34(12): 1583 – 1593.)
- [27] BOSMAN P A N, THIERENS D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 174 – 188.
- [28] TIAN Y, CHENG R, ZHANG X Y, et al. PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. *IEEE Computational Intelligence Magazine*, 2017, 12(4): 73 – 87.

#### 作者简介:

**吴亚丽** 副教授, 主要研究方向为智能优化算法理论及应用、复杂系统建模与优化, E-mail: yiliwu@xaut.edu.cn;

**付玉龙** 硕士研究生, 主要研究方向为多目标群智能优化算法及其应用, E-mail: yloongf@foxmail.com;

**李国婷** 硕士研究生, 主要研究方向为深度学习优化算法, E-mail: guotingl@foxmail.com;

**张亚崇** 高级工程师, 主要研究方向为导航信息融合和分布式导航系统技术, E-mail: whhzy802@163.com.