

## 基于平均距离聚类的 NSGA-II

崔志华<sup>1</sup> 张茂清<sup>2</sup> 常宇<sup>1</sup> 张江江<sup>1</sup> 王晖<sup>3</sup> 张文生<sup>4</sup>

**摘要** 拥挤度距离是一种用于度量解集多样性的指标。然而,在许多情况下,该指标无法有效区分多样性较优个体。其原因为拥挤度距离主要利用每个位置的局部信息。为解决该问题,基于整个种群全局位置信息,本文设计了基于平均距离聚类的多样性度量指标,并进一步提出了基于平均距离聚类的 NSGA-II。该算法利用平均距离将种群划分为若干大致均匀分布的小种群,然后分别在各小种群内执行选择、交叉和变异等操作。实验结果表明,本文所提算法可以有效地保持种群多样性。

**关键词** 多目标优化算法, NSGA-II, 拥挤度距离, 多样性

**引用格式** 崔志华, 张茂清, 常宇, 张江江, 王晖, 张文生. 基于平均距离聚类的 NSGA-II. 自动化学报, 2021, 47(5): 1171–1182

**DOI** 10.16383/j.aas.c180540

### NSGA-II With Average Distance Clustering

CUI Zhi-Hua<sup>1</sup> ZHANG Mao-Qing<sup>2</sup> CHANG Yu<sup>1</sup> ZHANG Jiang-Jiang<sup>1</sup> WANG Hui<sup>3</sup> ZHANG Wen-Sheng<sup>4</sup>

**Abstract** Crowding distance is an index for measuring the diversity of solutions. However, in many cases, it may fail to identify individuals with better diversity. The reason is that crowding distance mainly takes advantage of the local information of each position. To tackle this issue, based on the global position information of entire population, this paper designs average-distance-clustering diversity index, and further proposes NSGA-II with average distance clustering (ADCNSGA-II). ADCNSGA-II divides the entire population into several small populations using average distance, then the selection, crossover and mutation operators are performed in each small population. Simulation results show the proposed algorithm can maintain the diversify effectively.

**Key words** Multi-objective optimization algorithms, NSGA-II, crowding distance, diversity

**Citation** Cui Zhi-Hua, Zhang Mao-Qing, Chang Yu, Zhang Jiang-Jiang, Wang Hui, Zhang Wen-Sheng. NSGA-II with average distance clustering. *Acta Automatica Sinica*, 2021, 47(5): 1171–1182

近年来,随着复杂工程优化问题的大量出现,多目标优化问题 (Multi-objective optimization problems, MOP) 逐渐受到许多研究人员的重视。不同于单目标优化问题,多目标优化问题具有高复杂性、非线性和目标函数相互冲突等特点,且不存在全局最

优解,而是一个最优解集 (Pareto optimal set)。

为了有效求解多目标优化问题,学者们引入许多解决方法,如加权和法、目标规划法和极大极小法等。在具有一定先验知识前提下,上述方法可以取得较好结果,但它们通常只能获得一个 Pareto 最优解。进化算法的出现为解决多目标优化问题开辟了新思路,利用进化算法并行性、智能性、自适应性和自组织性,多目标进化算法可以计算多个 Pareto 最优解<sup>[1–3]</sup>。

多目标问题复杂性主要体现在目标函数个数上,因此,一种较为常见的研究思路是设计高效策略以减少目标函数个数。Yang<sup>[4]</sup> 提出采用随机加权和方法把多个目标函数动态整合为一个单目标函数,通过变化不同权重得到 Pareto 最优解集。Zhang 等<sup>[5]</sup> 提出的 MOEA/D (Multi-objective evolutionary algorithm based on decomposition) 算法将多目标优化问题分解为一定数量的单目标优化子问题,然后对这些子问题同步求解。巩敦卫等<sup>[6]</sup> 将高维多目标优化问题分解为若干子优化问题,每一个子优化问题除了包含原优化问题的少数目标函数之外,还将其他目标函数相关信息融合成一个新目标函数,

收稿日期 2018-08-09 录用日期 2019-01-22  
Manuscript received August 9, 2018; accepted January 22, 2019  
国家自然科学基金 (61806138, U1636220, 61663028), 山西省自然科学基金 (201801D121127), 江西省杰出青年人才计划 (20171BCB23075), 江西省自然科学基金 (20171BAB202035) 资助

Supported by National Natural Science Foundation of China (61806138, U1636220, 61663028), Natural Science Foundation of Shanxi Province (201801D121127), Distinguished Young Talents Plan of Jiangxi Province (20171BCB23075), and Natural Science Foundation of Jiangxi Province (20171BAB202035)

本文责任编辑 张敏灵  
Recommended by Associate Editor ZHANG Min-Ling  
1. 太原科技大学复杂系统与计算智能实验室 太原 030024 2. 同济大学电子与信息工程学院 上海 201804 3. 南昌工程学院江西省水信息协同感知与智能处理重点实验室 南昌 330029 4. 中国科学院自动化研究所 北京 100190

1. Complex System and Computational Intelligence Laboratory, Taiyuan University of Science and Technology, Taiyuan 030024 2. School of Electronics and Information, Tongji University, Shanghai 201804 3. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang 330029 4. Institute of Automation, Chinese Academy of Sciences, Beijing 100190

以降低问题求解难度. Schaffer<sup>[7]</sup> 把种群按照目标数量分成若干子种群, 在每个子种群中按照个体适应度选择出每个目标上最优个体, 然后将这些个体重新组合成种群. 丁进良等<sup>[8]</sup> 提出一种基于参考点预测策略的动态多目标优化算法, 该算法能快速跟踪动态变化的 Pareto 前沿. 李文彬等<sup>[9]</sup> 提出一种基于决策空间变换的最近邻预测方法, 通过属性趋势模型引入决策空间到目标空间的映射知识, 使决策空间的最近邻更有效反映目标空间的最近邻, 从而预测多目标优化 Pareto 支配解精度<sup>[10]</sup>.

根据无免费午餐定理, 每个智能优化算法都只对某些问题有效, 因此, 利用不同算法优点设计混合策略就成了一种提高算法性能的常见方法<sup>[11-14]</sup>. Zitzler 等<sup>[15]</sup> 利用适应度分配策略、密度估计和增强的存档截断方法设计了 SPEA2. Yang 等<sup>[16]</sup> 根据种群进化过程中支配解和非支配解比例大小, 把搜索过程分割为三个阶段, 并在每个阶段设计了不同搜索策略. 徐斌等<sup>[17]</sup> 和 Wang 等<sup>[18]</sup> 针对约束多目标优化问题, 提出了一种差分进化算法和  $\alpha$  约束支配处理的混合优化算法, 该算法在初期能有效利用不可行解所携带有用信息, 增加种群多样性, 在后期则控制不可行解比例, 使得算法朝可行域方向进化. 陈志旺等<sup>[19]</sup> 将高斯过程和智能进化算法进行结合提出一种融合多属性决策的双层种群筛选策略, 并将其嵌入到遗传算法求解高斯模型参数. 针对解分布性问题, 林浒等<sup>[20]</sup> 以免疫克隆算法为框架, 引入适应度共享策略, 提出了具有良好分布性的多目标优化进化算法. Tran 等<sup>[21]</sup>、Pooja 等<sup>[22]</sup> 和 Wang 等<sup>[23]</sup> 将差分进化算法中交叉算子和人工蜂群算法结合, 提出了新的混合多目标进化算法.

针对求解问题特点设计策略, 能得到较好满意解<sup>[24-26]</sup>. 刘潭等<sup>[27]</sup> 针对单位产油量综合能耗模型输出与实际值存在较大误差的问题, 利用高斯混合模型 (Gaussian mixture model, GMM) 对单位产油量综合能耗混合模型误差特性进行描述, 实现对模型的误差补偿, 并将误差补偿后的单位产油量综合能耗引入到已建优化模型中, 使得优化结果更接近实际最优值. 付亚平等<sup>[28]</sup> 针对生产工序的合并造成一种串并联共存的生产布局, 建立了混合整数规划模型. 针对模型特点, 他们设计了一种改进的非支配排序遗传算法, 同时采用基于启发式方法的种群初始方式提高种群多样性, 并引入一种局域搜索策略以改善算法所获得非支配解及分布性<sup>[29-30]</sup>. 乔俊飞等<sup>[31]</sup> 针对污水处理过程控制能耗过大和水质超标严重等问题, 通过记忆多目标智能优化算法的动态处理信息, 建立环境变量参数与最优解之间的知识模型. 利用定向局部区域寻优以及随机全局寻优策略, 提高了算法收敛性, 获得了更高质量的

解<sup>[32-33]</sup>.

NSGA-II<sup>[34]</sup> 是 Deb 等在 2002 年提出的多目标优化算法, 其具有较低计算复杂性且在相同时间内能获得较优的求解性能. 为了得到汽车制动器中多目标参数优化问题, 张屹等<sup>[35]</sup> 提出了基于正交设计的 NSGA-II, 得到了分布更优的 Pareto 前沿面. 为了解决多目标柔性作业车间调度问题, Yuan 等<sup>[36]</sup> 把分层策略融入了 NSGA-II, 提出了混合局部搜索的 NSGA-II. 路艳雪等<sup>[37]</sup> 尝试把多输入多输出的反向传播 (Back-propagation, BP) 神经网络和 NSGA-II 融合, 提高了算法运算速度和计算精度.

为了进一步提升 NSGA-II 性能, 近年来许多研究者尝试把聚类思想融入该算法中. 例如, Mukhopadhyay 等<sup>[38]</sup> 提出将模糊聚类思想应用于 NSGA-II, 并基于该方法从所求 Pareto 前沿面中获得了较优的聚类结果. 不同于以往单个标准聚类方法, Handl 等<sup>[39]</sup> 提出两标准的聚类方法, 并进一步将其与进化算法结合; 与一些单标准聚类算法相比, 该改进方法具有较优效果<sup>[40-41]</sup>. 类似地, 刘丛等<sup>[42]</sup> 利用使用欧氏距离和 Path 距离设计出新的聚类框架, 并融合多目标进化算法有效提升了算法性能.

上述聚类思想与多目标进化算法融合大多根据已有聚类方法进行改进. 与以上对 NSGA-II 改进不同, 本文受聚类思想启发, 针对算法多样性方面存在的缺陷<sup>[43-44]</sup> 设计了新的平均距离聚类的多样性指标. 基于平均距离聚类多样性指标, 整个种群可均匀划分成若干个小种群, 并可将 NSGA-II 的选择和交叉算子等操作应用在小种群中, 从而保证算法所求结果均匀分布在 Pareto 前沿面上, 称改进后算法为基于平均距离聚类的 NSGA-II 算法 (NSGA-II with average distance clustering, ADCNSGA-II).

本文组织如下: 第 1 节介绍多目标优化领域中的基本概念及 NSGA-II 基本框架; 第 2 节详细介绍了基于平均距离聚类的 NSGA-II, 包括选择和交叉算子的设计; 第 3 节利用 SCH 及 ZDT 测试集, 将所提算法与常用多目标优化算法进行对比实验, 并分析了参数  $S$  对算法性能影响; 最后给出了本文结论和未来研究方向.

## 1 基本 NSGA-II 框架

本文考虑如下多目标优化问题:

$$\min f(\mathbf{x}) = \min[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})] \quad (1)$$

其中,  $f_i(\mathbf{x})$  表示第  $i$  个目标函数,  $i = 1, 2, \dots, M$ ,  $\mathbf{x} = (x_1, x_2, \dots, x_D) \in \mathbf{R}^D$  为决策变量空间.

**定义 1.** 若变量  $\mathbf{x}$  的目标函数为  $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x}))^T$ , 变量  $\mathbf{x}'$  的目标函数为  $f(\mathbf{x}') = (f_1(\mathbf{x}'), f_2(\mathbf{x}'), \dots, f_M(\mathbf{x}'))^T$ , 当且仅当对

于  $\forall i \in \{1, 2, \dots, M\}$ ,  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$  成立, 且存在  $k \in \{1, 2, \dots, M\}$ , 使得  $f_k(\mathbf{x}) < f_k(\mathbf{x}')$  严格成立, 则称  $\mathbf{x}$  支配  $\mathbf{x}'$ , 记作:  $\mathbf{x} \prec \mathbf{x}'$ .

**定义 2.** 决策空间上所有 Pareto 最优解构成的集合称为 Pareto 最优解集 (*ParetoSet*, *PS*)

**定义 3.** Pareto 最优解集在目标空间上对应的点集称为 Pareto 前沿面 (*ParetoFront*, *PF*).

NSGA-II<sup>[34]</sup> 主要采用了快速非支配排序 (Fast non-dominated sorting) 和拥挤度距离 (Crowding distance) 的概念. 快速非支配排序主要思想如下: 首先, 对种群  $P$  中所有个体计算两个参数, 支配个体  $p$  的个体数量 ( $n_p$ ) 和个体  $p$  所支配的个体集合 ( $S_p$ ). 对于任意个体  $p$ , 若  $n_p = 0$ , 则其被放入第 1 层非支配前沿面  $F_1$ . 对  $n_p = 0$  的个体, 遍历其所支配的个体集合  $S_p$  中每一个个体 ( $q$ ) 并执行  $n_q - 1$ , 此时若  $n_q = 0$ , 则放入临时集合  $Q$  中. 遍历完  $F_1$  所有个体的支配集合之后, 此时  $Q$  所有个体组成第 2 层非支配前沿面  $F_2$ . 重复上述步骤, 直到所有个体都被识别出其所在前沿面.

拥挤度距离用于衡量种群多样性, 设  $f_k(\mathbf{x}_i)$ ,  $k = 1, 2, \dots, M$ ,  $i = 1, 2, \dots, N$  为目标函数, 基于每个目标函数值对所有个体进行升序排列, 排在第一位 (即函数值最小) 的个体, 其拥挤度距离为  $d_1 = \infty$ , 排在最后一位 (即函数值最大) 的个体, 其拥挤度距离为  $d_L = \infty$ , 其余个体的拥挤度距离按照下式计算:

$$d_i = \sum_{k=1}^M |f_k(\mathbf{x}_{i-1}) - f_k(\mathbf{x}_{i+1})| \quad (2)$$

其中,  $d_i$  表示个体  $\mathbf{x}_i$  的拥挤度距离,  $f_k(\mathbf{x}_{i+1})$  表示个体  $\mathbf{x}_{i+1}$  的第  $k$  个目标函数.

拥挤度距离计算过程可通过图 1 简要说明如下.

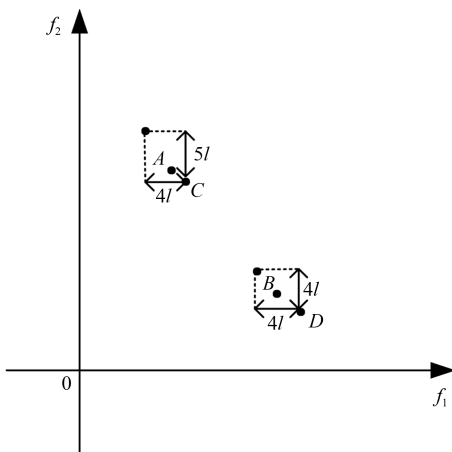


图 1 熔化率与比例、积分系数之间的关系

Fig. 1 Comparing melting rate with proportion and integral parameters of the consarc controller

假设图 1 中有  $A$  点、 $B$  点及其相邻的邻居, 且  $A$  与  $B$  互不支配. 根据式 (2), 可以得出  $B$  点的拥挤度距离为与  $B$  点相邻的两点的每个函数维度距离之和, 即  $B$  点的拥挤度为  $8l$ ; 同样, 可得  $A$  点拥挤度距离为  $9l$ . 从数值结果上看,  $B$  点比  $A$  点更加拥挤; 从图 1 可以直观看出  $A$  点比  $B$  点更加拥挤. 根据拥挤度距离的定义, 算法应该去掉  $B$  点, 保留  $A$  点. 如果交叉的父代选择了  $A$  点和  $C$  点, 由于  $A$  和  $C$  的距离过近, 会导致子代个体与父代  $A$  和  $C$  没有明显差异, 进而导致算法收敛到某一局部最优解而不能均匀地分布在整个 Pareto 前沿面上. 这显然不符合我们对拥挤度距离的预期.

## 2 基于平均距离聚类的 NSGA-II

为更清楚地介绍基于平均距离聚类的 NSGA-II, 第 2.1 节首先介绍匹配选择, 然后在第 2.2 节中详细介绍平均距离聚类的多样性指标, 之后在第 2.3 节中将其应用于 NSGA-II 的选择算子, 第 2.4 节将其应用于交叉算子, 最后总结本文所提算法的基本框架.

### 2.1 匹配选择

为加快整个种群的收敛速度和保证种群多样性, 本文引入了匹配选择操作对种群进行选择. 在匹配选择中, 本文借鉴 SPEA2 赋值方法对种群个体适应值进行重新赋值, 该赋值方法不仅考虑种群个体之间的支配关系, 也考虑种群个体间的被支配关系 (其中种群个体赋值方法可参考 SPEA2, 不再详细说明). 基于以上赋值的适应度值, 将非支配个体保存到一个初始为空的一个种群  $P_0$  中, 为防止种群  $P_0$  大小超出边界, 这里采用聚类截断方法使得整个种群大小达到要求的种群大小值  $N$ , 最后将得到种群作为匹配选择算子的子代输出. 具体匹配选择算子的伪代码如下.

#### 算法 1. 匹配选择算子

输入参数:  $P$  (种群),  $N$  (种群大小).

1.  $P \leftarrow \text{Initialize}(N)$  // 初始种群;
2. 对种群  $P$  个体进行适应度赋值;
3. **while** ( $|P_0| < N$ ) **do**
4. 基于分配的自适应度值对种群进行快速非支配排序;
5. 将非支配个体加入到一个种群  $P_0$  中;
6. **end**;
7. 判断种群  $P_0$  个体数量是否超出边界值;
8. 若超出边界, 采用聚类截断方法删除冗余个体;
9. 输出种群  $P_0$ .

### 2.2 平均距离聚类的多样性指标

平均距离聚类的多样性指标 (Average-distance-clustering diversity index) 主要采用聚类

思想,把平均距离内多个个体划分为一个小种群.

假设种群  $P = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$  有  $N$  个个体,  $f(\mathbf{x}_i) = (f_1(\mathbf{x}_i), f_2(\mathbf{x}_i), \dots, f_M(\mathbf{x}_i))$  为个体  $\mathbf{x}_i$  的目标函数,则种群  $P$  中第  $j$  个目标函数最大值  $f_{j,\max}$  和最小值  $f_{j,\min}$  分别为

$$f_{j,\max} = \max\{f_j(\mathbf{x}_1), f_j(\mathbf{x}_2), \dots, f_j(\mathbf{x}_N)\} \quad (3)$$

$$f_{j,\min} = \min\{f_j(\mathbf{x}_1), f_j(\mathbf{x}_2), \dots, f_j(\mathbf{x}_N)\} \quad (4)$$

定义群体在第  $j$  个目标函数的平均距离  $f_{j,\text{step}}$  为

$$f_{j,\text{step}} = \frac{S \times (f_{j,\max} - f_{j,\min})}{2N} \quad (5)$$

其中,  $S$  表示每个小种群所含个体数量. 由于在实际算法运行中,个体之间相对位置具有随机性,因此实际小种群中个体数量需要根据问题的特点设置.

如图 2 所示,若在目标空间中有点  $A$ 、 $B$  和  $C$ , 对应坐标轴  $f_1$  上投影分别为  $A'$ 、 $B'$  和  $C'$ , 且  $D$  为  $A'$  和  $B'$  中间点,  $E$  为  $B'$  和  $C'$  中间点. 若要在  $f_1$  上形成以  $B'$  为中心的小种群,且不能与相邻小种群有交集,则此时  $B'$  点所在小种群的选择半径只能为两个相邻点距离一半.

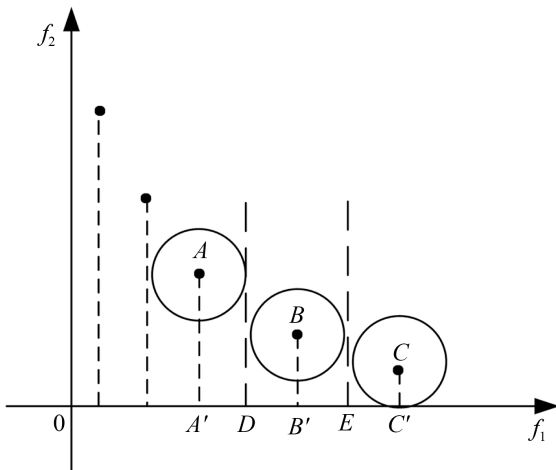


图 2 式 (5) 中参数说明

Fig. 2 Illustration of parameter in (5)

下面确定个体  $\mathbf{x}_i = (x_1^i, x_2^i, \dots, x_D^i)$  所在小种群内的其他个体. 由于平均距离为  $f_{j,\text{step}}$ , 若个体  $\mathbf{x}_j = (x_1^j, x_2^j, \dots, x_D^j)$  在同一个小种群内, 则目标函数  $f(\mathbf{x}_j)$  应满足:  $f_{\min}(\mathbf{x}_j) \leq f(\mathbf{x}_j) \leq f_{\max}(\mathbf{x}_j)$ , 其中  $f_{\max}(\mathbf{x}_i)$  为  $\mathbf{x}_i$  所在小种群所有个体目标函数上界,  $f_{\min}(\mathbf{x}_j)$  为  $\mathbf{x}_i$  所在小种群所有个体目标函数下界,  $f_{\max}$  和  $f_{\min}$  的计算式为

$$f_{\max}(\mathbf{x}_i) = ((f_1(\mathbf{x}_i) + f_{1,\text{step}}), (f_2(\mathbf{x}_i) + f_{2,\text{step}}), \dots, (f_M(\mathbf{x}_i) + f_{M,\text{step}})) \quad (6)$$

$$f_{\min}(\mathbf{x}_i) = ((f_1(\mathbf{x}_i) - f_{1,\text{step}}), (f_2(\mathbf{x}_i) - f_{2,\text{step}}), \dots, (f_M(\mathbf{x}_i) - f_{M,\text{step}})) \quad (7)$$

上述小种群划分过程可用算法 2 描述, 其中  $\mathbf{x}_i$  为随机选择个体. 在初始阶段, 首先确定目标函数平均距离 (第 1~2 行), 在此基础上进一步确定  $\mathbf{x}_i$  所在小种群, 并对小种群内未被标记个体进行逐一标记 (第 5~7 行). 需要注意的是, 在小种群划分过程中, 平均距离和小种群划分的数量由种群在运算过程中确定.

根据式 (5)~(7) 可知, 较大  $S$  取值 (例如当  $S$  取值为种群规模大小) 会直接导致  $f_{j,\text{step}}$  取值较大, 从而导致  $f_{\max}(\mathbf{x}_i) - f_{\min}(\mathbf{x}_i)$  增大. 由上述小种群划分过程可知,  $f_{\max}(\mathbf{x}_i) - f_{\min}(\mathbf{x}_i)$  较大取值不仅会导致更多个体包括在一个小种群中, 而且也会减少小种群数量, 并进一步减弱种群多样性. 因此, 在后续实验中, 本文将采用较小  $S$  值, 且将验证不同  $S$  取值对算法影响.

如图 3 所示, 通过以上方法可以将整个种群划分为若干小种群. 每个小种群范围根据种群平均距离确定, 且每个小种群相对于原始整个种群呈现大致均匀分布特点. 正是上述特点才能保证最后所求结果均匀分布在 Pareto 前沿面上, 同时小种群有限的搜索空间也能保证整个小种群内部所有个体对前沿面进行有效探索.

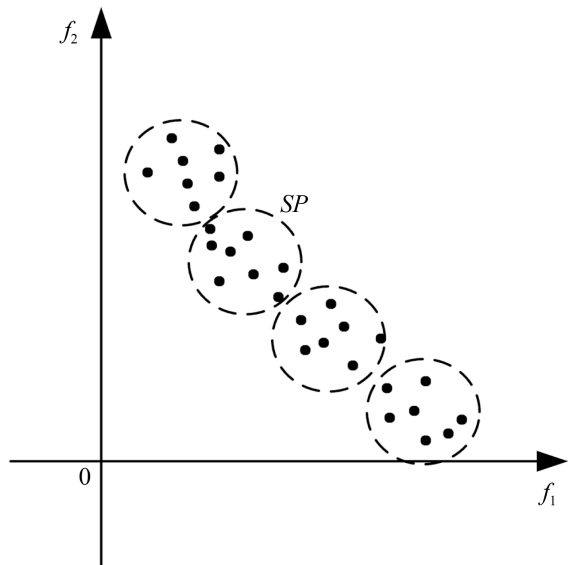


图 3 小种群划分示意图

Fig. 3 Illustration of small populations

**算法 2. 平均距离多样性指标**

输入参数:  $P$  (种群),  $N$  (种群大小).

1. 计算目标函数维度最大值 (式 (3)) 和最小值 (式 (4));
2. 计算目标函数平均距离 (式 (5));

3.  $k = 1$ ;
4. **while** (存在未标记个体) **do**
5. 随机选择未标记个体  $\mathbf{x}_i$ ;
6. 确定  $\mathbf{x}_i$  小种群范围 (式 (6) 和 (7));
7. 对  $\mathbf{x}_i$  平均距离范围内未被标记的个体标记为  $k$ ;
8.  $k = k + 1$ ;
9. **end**
10. 输出标记种群种群.

### 算法 3. 基于平均距离聚类选择算子

输入参数:  $SP$  (小种群),  $|SP|$  (种群大小).

1. 计算小种群  $SP$  的平均距离, 并计算每个个体平均距离内的个体数;
2.  $i = 1, SP_{\text{new}} = \emptyset$ ;
3. **while** ( $|SP_{\text{new}}| < |SP|$ ) **do**
4. 选择小种群  $SP$  中第  $i$  个个体  $a_i$ , 随机选择小种群  $SP$  中个体  $a_j$ ;
5. **if** ( $a_i < a_j$ ) **then**
6.  $SP_{\text{new}} = SP_{\text{new}} \cup a_i, i = i + 1$ ;
7. **else if** ( $a_j < a_i$ ) **then**
8.  $SP_{\text{new}} = SP_{\text{new}} \cup a_j, i = i + 1$ ;
9. **else if** ( $a_i$  平均距离内个体数少于  $a_{jj}$  平均距离内个体数) **then**
10.  $SP_{\text{new}} = SP_{\text{new}} \cup a_i, i = i + 1$ ;
11. **else if** ( $a_i$  平均距离内个体数大于  $a_{jj}$  平均距离内个体数) **then**
12.  $SP_{\text{new}} = SP_{\text{new}} \cup a_j, i = i + 1$ ;
13. **else if**  $\text{rand}() < 0.5$  **then**
14.  $SP_{\text{new}} = SP_{\text{new}} \cup a_i, i = i + 1$ ;
15. **else**
16.  $SP_{\text{new}} = SP_{\text{new}} \cup a_j, i = i + 1$ ;
17. **end if**
18. **end if**
19. **end if**
20. **end if**
21. **end if**
22. **end while**
23. **return**  $SP_{\text{new}}$ .

### 2.3 基于平均距离聚类选择算子

如算法 3 所示, 在对小种群  $SP$  执行选择时, 首先判断随机选择个体  $a_j$  与  $a_i$  的支配关系. 若可以确定支配关系, 则较优个体被选择 (第 5~8 行). 若  $a_j$  与  $a_i$  互不支配, 则比较平均距离内个体数量, 平均距离内个体数量少的个体被优先选择 (第 9~12 行). 如果  $a_j$  与  $a_i$  具有相同个体数量, 则随机选择两者其一 (第 13~16 行).

利用上述策略对图 1 进一步分析. 由于点  $A$  与点  $B$  互不支配 (第 5~8 行), 因此无法判断两者优劣. 若将其所在小种群平均距离内个体数量作为进一步评判指标, 则可有效区分两者优劣 (第 9~12 行). 由于  $A$  非常靠近其邻居节点, 因此在  $A$  点平均距离范围至少有一个节点. 对于  $B$  而言, 由于其与

前后两个邻居距离适中, 因此其平均距离内节点数均为零. 从上述分析中可以看出, 利用本文所提的基于平均距离聚类选择策略可以有效避免选择拥挤度大的个体.

### 2.4 基于平均距离聚类交叉算子

与常规交叉算子不同, 本文将小种群决策变量的平均距离加入到交叉算子计算过程中, 即引入了扰动操作, 从而可以进一步增加种群个体多样性.

一般交叉过程如图 4 所示. 在决策空间中, 假设点  $A$  (小圆中心位置) 和点  $B$  (小圆中心位置) 为随机选择的交叉父代. 按照标准交叉原则,  $A$  与  $B$  交叉子代一定落在小矩形框内 (这里仅讨论交叉算子, 不涉及变异, 因此后代产生范围可以确定). 如果引入基于平均距离聚类, 则可以确定  $A$  点所在的小种群范围 (对应  $A$  点所在小圆圈区域), 同时确定  $B$  点所在小种群范围 (对应  $B$  点所在小圆圈区域). 通过随机选择  $A$  点所在区域的个体  $A'$  和  $B$  点所在区域的个体  $B'$ ,  $A$  与  $B$  交叉产生的子代范围 (实际是  $A'$  和  $B'$  进行交叉运算) 将比标准交叉算子产生的子代范围更大 (大矩形框所在区域). 根据以上分析, 可以看出采用基于平均距离聚类之后的子代区域搜索范围扩大了许多.

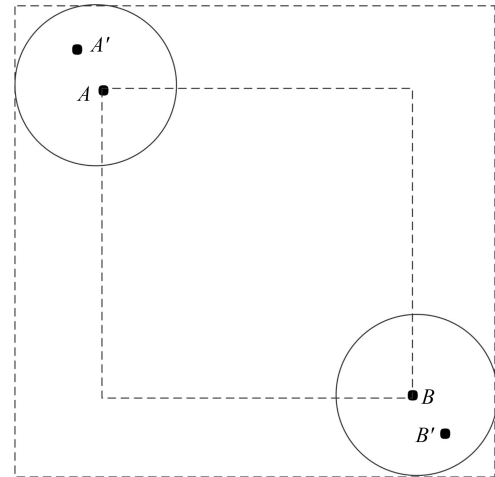


图 4 交叉算子示意图

Fig. 4 Illustration of crossover operator

定义  $\otimes$  为交叉运算,  $\mathbf{x}^A = (x_1^A, x_2^A, \dots, x_D^A)$  和  $\mathbf{x}^B = (x_1^B, x_2^B, \dots, x_D^B)$  为进行交叉运算的父代, 上述交叉过程可用式 (8) 描述为

$$\mathbf{x}^{\text{new}} = (\mathbf{x}^A + (r_1 \times \mathbf{x}_{\text{step}}^A)) \otimes (\mathbf{x}^B + (r_2 \times \mathbf{x}_{\text{step}}^B)) \quad (8)$$

其中,  $\mathbf{x}^{\text{new}}$  表示产生的子代,  $r_1$  和  $r_2$  分别是  $[-1, 1]$  内的随机数.  $\mathbf{x}_{\text{step}}^{SPA}$  和  $\mathbf{x}_{\text{step}}^{SPB}$  分别为  $A$  点和  $B$  点所在小种群每维决策变量的平均距离.

设  $A$  点所在小种群为  $SP_A$ , 其个体数量为  $|SP_A|$ ,  $x_j^i$  表示  $A$  点所在小种群  $SP_A$  第  $i$  个个体第  $j$  维决策变量, 则小种群  $SP_A$  中第  $j$  维决策变量最大值  $x_{j,\max}^{SP_A}$  与最小值  $x_{j,\min}^{SP_A}$  分别为

$$x_{j,\max}^{SP_A} = \max(x_j^1, x_j^2, \dots, x_j^{|SP_A|}) \quad (9)$$

$$x_{j,\min}^{SP_A} = \min(x_j^1, x_j^2, \dots, x_j^{|SP_A|}) \quad (10)$$

进而, 可得  $A$  点所在小种群第  $j$  维决策变量的平均距离为

$$x_{j,\text{step}}^{SP_A} = \frac{x_{j,\max}^{SP_A} - x_{j,\min}^{SP_A}}{2|SP_A|} \quad (11)$$

在每一维决策变量上计算决策变量的平均距离, 可得到  $A$  点所在小种群  $SP_A$  在每维决策变量的平均距离,  $x_{\text{step}}^{SP_A} = (x_{1,\text{step}}^{SP_A}, x_{2,\text{step}}^{SP_A}, \dots, x_{D,\text{step}}^{SP_A})$ , 同理可以得到  $x_{\text{step}}^{SP_B} = (x_{1,\text{step}}^{SP_B}, x_{2,\text{step}}^{SP_B}, \dots, x_{D,\text{step}}^{SP_B})$ .

## 2.5 基于平均距离聚类的 NSGA-II 基本框架

基于平均距离聚类的 NSGA-II 本质上是把平均距离聚类策略融入到 NSGA-II 的选择和交叉算子中. 它由 5 个操作组成: 平均距离聚类多样性指标、基于平均距离聚类的选择算子、基于平均距离聚类的交叉算子、变异算子和环境选择.

基于平均距离聚类的 NSGA-II 主要过程如算法 4 所示. 在初始化种群和相关参数之后, 首先对所有个体进行快速非支配排序, 然后通过平均距离聚类的多样性指标把整个种群按照平均距离聚类划分为若干小种群, 并根据小种群范围进一步确定每个个体所属小种群; 利用基于平均距离聚类的选择算子在每个小种群内执行选择算子. 根据个体支配等级和其所在小种群个体数量确定较优个体, 并基于平均距离聚类交叉算子对小种群内部较优个体执行交叉运算. 变异算子表示对所有个体进行变异操作, 然后利用环境选择对整个种群个体更新. 重复上述步骤, 直到满足终止条件. 需要说明的是, 此处变异算子不涉及平均距离聚类, 因此本文不再详述. 环境选择为从父代种群和子代种群中选择较优个体, 其主要涉及的选择算子与上面所述相同, 因此也不再赘述. ADCNSGA-II 详细流程图如图 5 所示.

### 算法 4. 基于平均距离聚类的 NSGA-II (ADCNSGA-II)

输入参数:  $P$  (种群),  $N$  (种群大小),  $P_c$  (交叉概率),  $P_m$  (变异概率).

1.  $P \leftarrow \text{Initialize}(N)$  //初始化种群;
2. **while** (是否满足结束条件) **do**
3.  $P_0 \leftarrow \text{mating\_selection}(P)$  //匹配选择;
4.  $P_0 \leftarrow \text{fast-nondominated-sorting}(P_0)$  //快速非支配排序;

5.  $P_1 \leftarrow \text{average-distance-clustering diversity index}(P_0)$  //计算平均距离聚类的多样性指标;
6.  $P_2 \leftarrow \text{selection operator with average-distance-clustering}(P')$  //基于平均距离聚类选择算子;
7.  $P_3 \leftarrow \text{crossover operator with average-distance-clustering}$  //基于平均距离聚类交叉算子;
8.  $P_4 \leftarrow \text{mutation}(P_3, P_c)$  //变异操作;
9.  $P \leftarrow \text{environmental selection}(P \cup P_4)$  //环境选择, 更新种群;
10. **end while**.

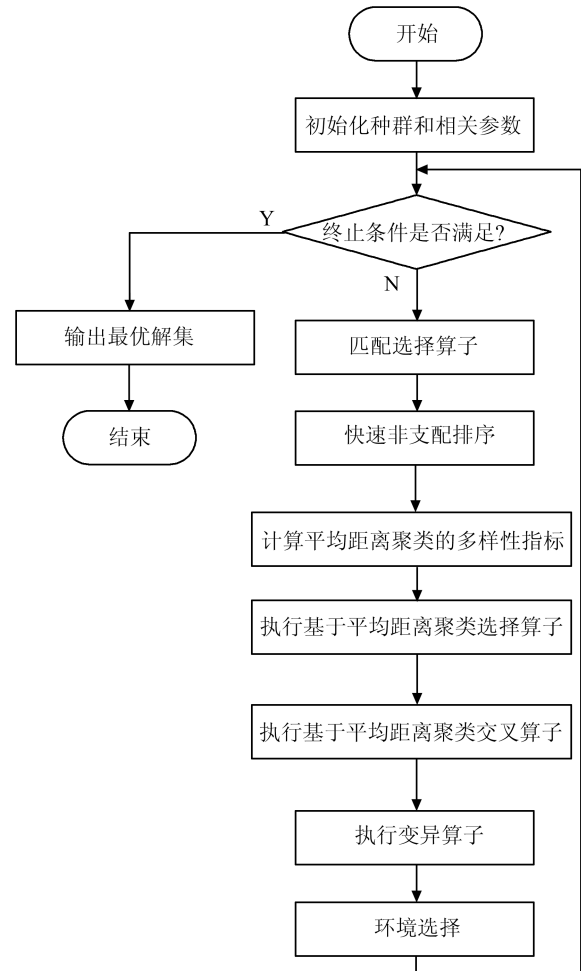


图 5 ADCNSGA-II 算法流程图

Fig. 5 The flowchart of ADCNSGA-II

## 3 实验及结果分析

本实验分为两部分: 1) 分析 ADCNSGA-II 中参数  $S$  取不同值时对算法性能的影响; 2) 验证 ADCNSGA-II 算法性能. 为了综合测试算法性能, 本文采用 ZDT 测试函数集和 SCH 测试函数. ZDT 测试函数集<sup>[45]</sup> 由 Zitzler 等在 2000 年提出, 有 6 个测试函数, 本文采用其中 5 个测试函数, SCH<sup>[46]</sup> 由 Schaffer 提出. 这些测试函数具有凸、凹、连续、非

连续和具有多重局部最优点等特点.

在多目标优化领域有很多评价指标, 本文采用  $GD$  值和  $SP$  值作为评价标准, 其中,  $GD^{[47]}$  用于评价最终解集逼近真实前沿面程度,  $SP^{[47]}$  用于评价种群分布多样性. 通过对两个指标分别测试, 可清楚地反映算法的整体性能.

### 3.1 参数 $S$ 对算法的影响

本小节研究参数对算法 ADCNSGA-II 性能的影响. 实验所用计算机为 Inter Core i 5-2400 3.10 GHz CPU, 6.00 GB 内存, Windows7 操作系统, 运行环境为 MATLAB7.9. 每个算法独立运行 30 次, 算法最大迭代 100 次, 种群个体为 50, 分别取值为 1, 2, 3, 4, 5 和 6.

不同  $S$  取值对算法 ADCNSGA-II 性能的影响

如表 1 所示. 其中,  $\text{mean}(GD)$  和  $\text{mean}(SP)$  分别表示算法运行 30 次后  $GD$  值和  $SP$  值平均值. 同样,  $\text{std}(GD)$  和  $\text{std}(SP)$  分别表示算法运行 30 次后  $GD$  和  $SP$  方差. 表中最优结果用粗体标注. 从表 1 可以看出, 在 ZDT4 测试函数上, 不同  $S$  取值对算法分布多样性和收敛效率有较大影响. 但就整个测试函数而言,  $S = 1$  比  $S$  取其他值时在算法性能上具有较大优势. 就测试结果的数值变化范围而言, 不同  $S$  取值对算法性能并无明显差异. 上述现象表明, 较小  $S$  取值 (即每个小种群中个体数量较少), 可以有效限制小种群内部个体变化范围; 同时, 也使多个小种群之间保持相对均匀的分布, 进而保证整个种群的多样性.

为分析不同  $S$  取值对算法的影响, 本文利用 Friedman 检验和 Wilcoxon 检验进一步分析上述结

表 1 参数  $S$  对算法 ADCNSGA-II 性能影响  
Table 1 Influence of parameter  $S$  on ADCNSGA-II

测试函数	指标	$S = 1$	$S = 2$	$S = 3$	$S = 4$	$S = 5$	$S = 6$
SCH	$\text{mean}(GD)$	<b><math>5.87 \times 10^{-2}</math></b>	$6.47 \times 10^{-2}$	$6.68 \times 10^{-2}$	$6.77 \times 10^{-2}$	$6.67 \times 10^{-2}$	$6.69 \times 10^{-2}$
	$\text{std}(GD)$	$8.14 \times 10^{-3}$	<b><math>3.84 \times 10^{-3}</math></b>	$5.03 \times 10^{-3}$	$5.11 \times 10^{-3}$	$5.28 \times 10^{-3}$	$4.15 \times 10^{-3}$
	$\text{mean}(SP)$	<b><math>9.87 \times 10^{-3}</math></b>	$1.18 \times 10^{-2}$	$1.77 \times 10^{-2}$	$1.89 \times 10^{-2}$	$2.28 \times 10^{-2}$	$1.77 \times 10^{-2}$
	$\text{std}(SP)$	$6.23 \times 10^{-3}$	<b><math>2.62 \times 10^{-3}</math></b>	$1.12 \times 10^{-2}$	$1.07 \times 10^{-2}$	$2.16 \times 10^{-2}$	$1.01 \times 10^{-2}$
ZDT1	$\text{mean}(GD)$	<b><math>2.05 \times 10^{-6}</math></b>	$4.07 \times 10^{-6}$	$4.71 \times 10^{-5}$	$3.28 \times 10^{-5}$	$7.25 \times 10^{-5}$	$7.71 \times 10^{-5}$
	$\text{std}(GD)$	<b><math>1.11 \times 10^{-5}</math></b>	$2.10 \times 10^{-5}$	$8.90 \times 10^{-5}$	$6.66 \times 10^{-5}$	$1.32 \times 10^{-4}$	$1.28 \times 10^{-4}$
	$\text{mean}(SP)$	<b><math>4.48 \times 10^{-3}</math></b>	$5.34 \times 10^{-3}$	$5.94 \times 10^{-3}$	$6.48 \times 10^{-3}$	$5.87 \times 10^{-3}$	$5.42 \times 10^{-3}$
	$\text{std}(SP)$	$1.59 \times 10^{-3}$	$1.31 \times 10^{-3}$	<b><math>1.20 \times 10^{-3}</math></b>	$1.92 \times 10^{-3}$	$2.57 \times 10^{-3}$	$1.78 \times 10^{-3}$
ZDT2	$\text{mean}(GD)$	<b><math>1.32 \times 10^{-5}</math></b>	$3.80 \times 10^{-5}$	$5.17 \times 10^{-5}$	$5.51 \times 10^{-5}$	$8.31 \times 10^{-5}$	$8.55 \times 10^{-5}$
	$\text{std}(GD)$	<b><math>6.55 \times 10^{-5}</math></b>	$9.60 \times 10^{-5}$	$9.26 \times 10^{-5}$	$7.18 \times 10^{-5}$	$9.89 \times 10^{-5}$	$1.21 \times 10^{-4}$
	$\text{mean}(SP)$	<b><math>4.47 \times 10^{-3}</math></b>	$5.40 \times 10^{-3}$	$6.32 \times 10^{-3}$	$6.91 \times 10^{-3}$	$6.78 \times 10^{-3}$	$6.79 \times 10^{-3}$
	$\text{std}(SP)$	$1.79 \times 10^{-3}$	<b><math>1.02 \times 10^{-3}</math></b>	$1.44 \times 10^{-3}$	$2.60 \times 10^{-3}$	$3.32 \times 10^{-3}$	$2.46 \times 10^{-3}$
ZDT3	$\text{mean}(GD)$	<b><math>1.09 \times 10^{-6}</math></b>	$6.07 \times 10^{-6}$	$2.48 \times 10^{-5}$	$3.57 \times 10^{-5}$	$1.38 \times 10^{-5}$	$5.13 \times 10^{-5}$
	$\text{std}(GD)$	<b><math>5.91 \times 10^{-6}</math></b>	$2.61 \times 10^{-5}$	$5.56 \times 10^{-5}$	$5.54 \times 10^{-5}$	$2.20 \times 10^{-5}$	$5.81 \times 10^{-5}$
	$\text{mean}(SP)$	<b><math>4.93 \times 10^{-3}</math></b>	$6.20 \times 10^{-3}$	$6.01 \times 10^{-3}$	$7.84 \times 10^{-3}$	$6.85 \times 10^{-3}$	$8.04 \times 10^{-3}$
	$\text{std}(SP)$	$1.89 \times 10^{-3}$	<b><math>1.21 \times 10^{-3}</math></b>	$2.12 \times 10^{-3}$	$3.61 \times 10^{-3}$	$2.22 \times 10^{-3}$	$2.91 \times 10^{-3}$
ZDT4	$\text{mean}(GD)$	$1.51 \times 10^{-5}$	$8.46 \times 10^{-6}$	$2.03 \times 10^{-5}$	<b><math>4.54 \times 10^{-6}</math></b>	$2.67 \times 10^{-5}$	$7.20 \times 10^{-6}$
	$\text{std}(GD)$	$4.49 \times 10^{-5}$	$2.13 \times 10^{-5}$	$4.46 \times 10^{-5}$	<b><math>1.25 \times 10^{-5}</math></b>	$6.31 \times 10^{-5}$	$1.76 \times 10^{-5}$
	$\text{mean}(SP)$	<b><math>3.29 \times 10^{-3}</math></b>	$5.21 \times 10^{-3}$	$6.01 \times 10^{-3}$	$5.98 \times 10^{-3}$	$6.43 \times 10^{-3}$	$6.43 \times 10^{-3}$
	$\text{std}(SP)$	$1.64 \times 10^{-3}$	<b><math>1.20 \times 10^{-3}</math></b>	$1.75 \times 10^{-3}$	$1.65 \times 10^{-3}$	$2.01 \times 10^{-3}$	$3.32 \times 10^{-3}$
ZDT6	$\text{mean}(GD)$	<b><math>1.11 \times 10^{-2}</math></b>	$2.07 \times 10^{-2}$	$2.07 \times 10^{-2}$	$3.27 \times 10^{-2}$	$9.92 \times 10^{-3}$	$2.29 \times 10^{-2}$
	$\text{std}(GD)$	<b><math>2.34 \times 10^{-2}</math></b>	$3.86 \times 10^{-2}$	$4.77 \times 10^{-2}$	$5.42 \times 10^{-2}$	$3.61 \times 10^{-2}$	$4.08 \times 10^{-2}$
	$\text{mean}(SP)$	<b><math>1.77 \times 10^{-2}</math></b>	$6.14 \times 10^{-2}$	$5.52 \times 10^{-2}$	$7.81 \times 10^{-2}$	$1.09 \times 10^{-2}$	$4.55 \times 10^{-2}$
	$\text{std}(SP)$	$5.80 \times 10^{-2}$	$1.64 \times 10^{-1}$	$1.26 \times 10^{-1}$	$1.18 \times 10^{-1}$	<b><math>3.05 \times 10^{-2}</math></b>	$1.14 \times 10^{-1}$

果. Friedman 检验用于分析多个样本之间的差异, 根据秩均值大小排名, 秩均值越小说明算法性能越好. Wilcoxon 检验用于分析两样本是否具有显著性差异. 若 p-value 小于 0.05, 则认为两个算法具有显著性差异. 表 2 和表 3 中所有数据根据算法运行 30 次平均  $GD$  值求出. 从表 2 中可以看出,  $S = 1$  相对于  $S$  取其他值时在 ADCNSGA-II 上具有一定优势; 从表 3 中可进一步得出, 不同的  $S$  取值并没有导致

表 2 Friedman 测试结果  
Table 2 Comparison results of Friedman test

参数	秩均值
$S = 1$	<b>1.92</b>
$S = 2$	2.40
$S = 3$	3.67
$S = 4$	4.38
$S = 5$	4.23
$S = 6$	4.42

表 3 Wilcoxon 检测测试结果  
Table 3 Comparison results of Wilcoxon test

$S = 1$ 对比	p 值
$S = 2$	0.0765
$S = 3$	0.0018
$S = 4$	<b>0.0003</b>
$S = 5$	0.0258
$S = 6$	<b>0.0003</b>

算法具有明显的差异. 从上述分析中可以得出, ADCNSGA-II 算法对于参数  $S$  变化不敏感, 即 ADCNSGA-II 对参数  $S$  具有较强的鲁棒性.

### 3.2 与其他算法实验结果对比

为了进一步分析 ADCNSGA-II (即参数  $S = 1$ ) 的性能, 本文将其与 NSGA-II<sup>[34]</sup>、PNIA<sup>[48]</sup>、SPEA2<sup>[15]</sup> 和 g-NSGA-II<sup>[49]</sup> 进行比较, 其中 g-NSGA-II 是 Molinac 等对 NSGA-II 的进一步改进算法. 从表 4 可以看出, 在 ZDT1、ZDT3 和 ZDT4

表 4 实验性能均值和方差对比  
Table 4 Means and variances of the performance metrics

测试函数	指标	PNIA	SPEA2	NSGA-II	g-NSGA-II	ADCNSGA-II
SCH	mean( $GD$ )	<b><math>3.81 \times 10^{-2}</math></b>	$4.41 \times 10^{-2}$	$6.17 \times 10^{-2}$	$4.43 \times 10^{-2}$	$5.87 \times 10^{-2}$
	std( $GD$ )	<b><math>1.93 \times 10^{-3}</math></b>	$1.14 \times 10^{-3}$	$2.96 \times 10^{-3}$	$1.61 \times 10^{-2}$	$8.14 \times 10^{-3}$
	mean( $SP$ )	<b><math>4.96 \times 10^{-3}</math></b>	$7.01 \times 10^{-3}$	$1.75 \times 10^{-2}$	$3.25 \times 10^{-2}$	$9.87 \times 10^{-3}$
	std( $SP$ )	<b><math>3.66 \times 10^{-3}</math></b>	$1.61 \times 10^{-3}$	$1.02 \times 10^{-2}$	$1.45 \times 10^{-1}$	$6.32 \times 10^{-3}$
ZDT1	mean( $GD$ )	$6.56 \times 10^{-4}$	$2.80 \times 10^{-5}$	$2.58 \times 10^{-4}$	$1.73 \times 10^{-3}$	<b><math>2.05 \times 10^{-6}</math></b>
	std( $GD$ )	$1.64 \times 10^{-4}$	$1.34 \times 10^{-5}$	$1.17 \times 10^{-4}$	$5.72 \times 10^{-3}$	<b><math>1.11 \times 10^{-5}</math></b>
	mean( $SP$ )	$1.04 \times 10^{-3}$	$1.94 \times 10^{-3}$	$5.09 \times 10^{-3}$	$1.01 \times 10^{-3}$	<b><math>4.48 \times 10^{-3}</math></b>
	std( $SP$ )	$1.06 \times 10^{-3}$	$3.38 \times 10^{-4}$	$3.92 \times 10^{-3}$	$7.45 \times 10^{-3}$	<b><math>1.59 \times 10^{-3}</math></b>
ZDT2	mean( $GD$ )	$8.77 \times 10^{-4}$	<b><math>1.17 \times 10^{-5}</math></b>	$8.42 \times 10^{-5}$	$5.98 \times 10^{-4}$	$1.32 \times 10^{-5}$
	std( $GD$ )	$1.18 \times 10^{-3}$	<b><math>1.17 \times 10^{-5}</math></b>	$1.32 \times 10^{-4}$	$6.03 \times 10^{-4}$	$6.55 \times 10^{-5}$
	mean( $SP$ )	$2.37 \times 10^{-3}$	<b><math>2.26 \times 10^{-3}</math></b>	$2.81 \times 10^{-3}$	$2.40 \times 10^{-3}$	$4.47 \times 10^{-3}$
	std( $SP$ )	$3.17 \times 10^{-3}$	<b><math>3.59 \times 10^{-3}</math></b>	$4.36 \times 10^{-3}$	$2.70 \times 10^{-3}$	$1.79 \times 10^{-3}$
ZDT3	mean( $GD$ )	$1.89 \times 10^{-4}$	$4.43 \times 10^{-6}$	$1.40 \times 10^{-4}$	$6.22 \times 10^{-5}$	<b><math>1.09 \times 10^{-6}</math></b>
	std( $GD$ )	$1.01 \times 10^{-4}$	$1.71 \times 10^{-6}$	$9.98 \times 10^{-5}$	$4.98 \times 10^{-5}$	<b><math>5.91 \times 10^{-6}</math></b>
	mean( $SP$ )	$4.09 \times 10^{-4}$	$1.89 \times 10^{-3}$	$5.56 \times 10^{-3}$	$1.10 \times 10^{-3}$	<b><math>4.93 \times 10^{-3}</math></b>
	std( $SP$ )	$5.47 \times 10^{-4}$	$5.99 \times 10^{-4}$	$3.81 \times 10^{-3}$	$8.23 \times 10^{-4}$	<b><math>1.89 \times 10^{-3}</math></b>
ZDT4	mean( $GD$ )	$1.25 \times 10^{-2}$	$6.95 \times 10^{-2}$	$2.68 \times 10^{-1}$	$1.62 \times 10^{-2}$	<b><math>1.51 \times 10^{-5}</math></b>
	std( $GD$ )	$1.03 \times 10^{-2}$	$4.19 \times 10^{-2}$	$1.30 \times 10^{-1}$	$6.48 \times 10^{-1}$	<b><math>4.49 \times 10^{-5}</math></b>
	mean( $SP$ )	<b><math>1.43 \times 10^{-3}</math></b>	$4.37 \times 10^{-3}$	$5.09 \times 10^{-3}$	$1.70 \times 10^{-2}$	$3.29 \times 10^{-3}$
	std( $SP$ )	$3.21 \times 10^{-3}$	$4.64 \times 10^{-3}$	$1.11 \times 10^{-2}$	$2.24 \times 10^{-2}$	<b><math>1.64 \times 10^{-3}</math></b>
ZDT6	mean( $GD$ )	$1.78 \times 10^{-3}$	<b><math>1.33 \times 10^{-3}</math></b>	$5.49 \times 10^{-3}$	$1.47 \times 10^{-2}$	$1.11 \times 10^{-2}$
	std( $GD$ )	$2.31 \times 10^{-4}$	<b><math>1.84 \times 10^{-4}</math></b>	$1.27 \times 10^{-3}$	$4.00 \times 10^{-3}$	$2.34 \times 10^{-2}$
	mean( $SP$ )	$1.35 \times 10^{-3}$	$1.45 \times 10^{-3}$	$4.43 \times 10^{-3}$	<b><math>8.74 \times 10^{-4}</math></b>	$1.77 \times 10^{-2}$
	std( $SP$ )	$9.87 \times 10^{-4}$	<b><math>5.74 \times 10^{-4}</math></b>	$2.05 \times 10^{-3}$	$7.14 \times 10^{-4}$	$5.80 \times 10^{-2}$



对比结果中, ADCNSGA-II 的性能明显优于其他四个算法; 且在 ZDT2 函数上, ADCNSGA-II 和 SPEA2 有相似的收敛性和分布性. 对于 SCH 测试函数, 本文提出算法也具有相对较好的性能. 在 ZDT1~ZDT4 函数上, ADCNSGA-II 算法在分布多样性和逼近真实前沿面方面优于 NSGA-II, 在 ZDT6 中, 改进算法的收敛性明显优于 NSGA-II. 对比 NSGA-II (此时相当于  $S$  取值大小为种群数) 和表 1 中  $S = 1, 2, 3, 4, 5, 6$  的实验数据, 可以进一步验证前述结论. 当  $S$  取较大值时 (例如  $S$  为种群

数量大小或者接近种群数量), 由于每个个体在整个种群所在空间或者接近整个种群空间中变化, 会导致整个种群多样性受到减弱; 较小的  $S$  值则可以有效限制小种群内部个体变换范围, 且使多个小种群保持相对均匀分布, 从而进一步增强种群多样性. 从上述分析结果可以看出, ADCNSGA-II 有效地提高了算法多样性. 同时, 解集多样性又进一步提高了算法的收敛效率.

为了进一步分析 ADCNSGA-II 与 NSGA-II 差异, 本文给出了两者之间的直观比较. 从图 6 可以看

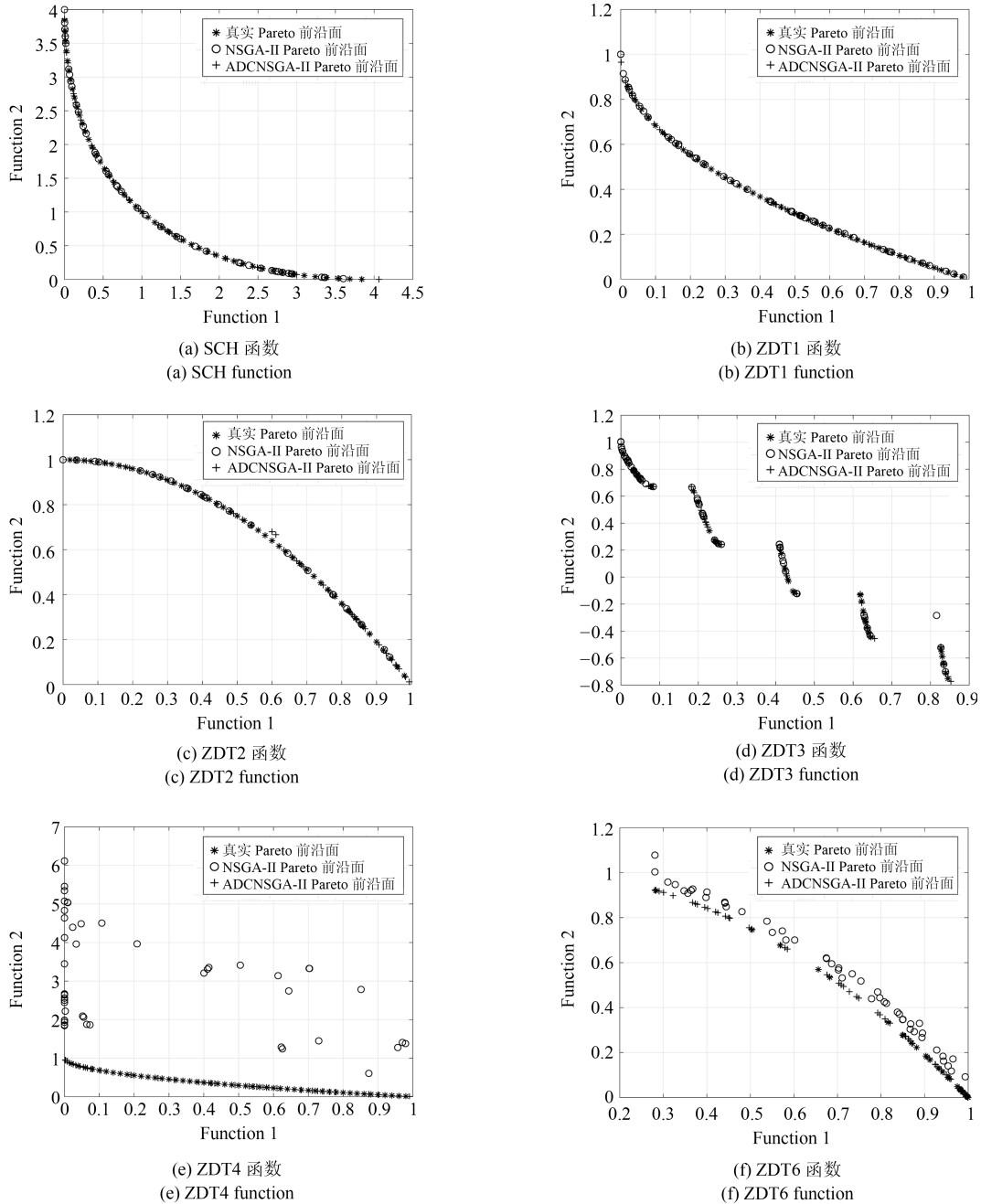


图 6 测试结果对比

Fig. 6 Comparisons of obtained solutions

出,就解集分布多样性和逼近真实前沿面的能力而言,NSGA-II 和 ADCNSGA-II 算法在函数 SCH、ZDT1 和 ZDT2 上具有相似求解结果;但在 ZDT3 函数上,ADCNSGA-II 在算法逼近最优前沿面方面明显优于 NSGA-II,说明本文采用的基于平均距离聚类的多样性指标能有效提高算法收敛效率.ZDT4 函数是几个测试函数中较为复杂的测试函数,它具有多个局部极值点.从 ZDT4 及 ZDT6 的对比效果来看,ADCNSGA-II 收敛效率明显优于 NSGA-II.从上述分析中可以看出,ADCNSGA-II 不仅能提升解集分布性,而且可以间接提高算法收敛效率;出现上述现象的原因是由于种群多样性方面的提高使得个体间差异性变大,进而改善了全局搜索的效果.

#### 4 结论与展望

NSGA-II 具有较优收敛效率,但是其采用的拥挤度距离在某些情况下不能有效保持选择多样性.为了解决该问题,本文借鉴聚类思想,提出了基于平均距离聚类的多样性度量指标,并将其与 NSGA-II 融合,提出了 ADCNSGA-II 算法.该方法利用平均距离将种群划分为若干个小种群,根据个体支配等级和平均距离内的个体数量选择较优个体,并在小种群内进行个体交叉和变异操作.实验结果表明,ADCNSGA-II 不仅有利于保持种群分布多样性,还可以间接提高种群收敛效率.在后续工作中,将会进一步完善平均距离聚类的多样性度量指标,引入其他聚类方法,使之适用于更多优化问题.

#### References

- Cai X J, Wang H, Cui Z H, Cai J H, Xue Y, Wang L. Bat algorithm with triangle-flipping strategy for numerical optimization. *International Journal of Machine Learning and Cybernetics*, 2018, **9**(2): 199–215
- Zhang M Q, Wang H, Cui Z H, Chen J J. Hybrid multi-objective cuckoo search with dynamical local search. *Memetic Computing*, 2018, **10**(2): 199–208
- Cui Z H, Xue F, Cai X J, Cao Y, Wang G G, Chen J J. Detection of malicious code variants based on deep learning. *IEEE Transactions on Industrial Informatics*, 2018, **14**(7): 3187–3196
- Yang X S. Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 2011, **3**(5): 267–274
- Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
- Gong Dun-Wei, Liu Yi-Ping, Sun Xiao-Yan, Han Yu-Yan. Parallel many-objective evolutionary optimization using objectives decomposition. *Acta Automatica Sinica*, 2015, **41**(8): 1438–1451  
(巩敦卫, 刘益萍, 孙晓燕, 韩玉艳. 基于目标分解的高维多目标并行进化优化方法. *自动化学报*, 2015, **41**(8): 1438–1451)
- Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st International Conference on Genetic Algorithms*. Pittsburgh, USA: ACM, 1985. 93–100
- Ding Jin-Liang, Yang Cui-E, Chen Li-Peng, Chai Tian-You. Dynamic multi-objective optimization algorithm based on reference point prediction. *Acta Automatica Sinica*, 2017, **43**(2): 313–320  
(丁进良, 杨翠娥, 陈立鹏, 柴天佑. 基于参考点预测的动态多目标优化算法. *自动化学报*, 2017, **43**(2): 313–320)
- Li Wen-Bin, He Jian-Jun, Feng Cai-Ying, Guo Guan-Qi. Prediction of Pareto dominance using nearest neighbor method based on decision space transformation. *Acta Automatica Sinica*, 2017, **43**(2): 294–301  
(李文彬, 贺建军, 冯彩英, 郭观七. 基于决策空间变换最近邻方法的 Pareto 支配性预测. *自动化学报*, 2017, **43**(2): 294–301)
- Sharma A, Jalal A S. Distance-based facility location problem for fuzzy demand with simultaneous opening of two facilities. *International Journal of Computing Science and Mathematics*, 2018, **9**(6): 590–601
- Cui Z H, Li F X, Zhang W S. Bat algorithm with principal component analysis. *International Journal of Machine Learning and Cybernetics*, 2019, **10**(3): 603–622
- Heraguemi K E, Kamel N, Drias H. Multi-objective bat algorithm for mining numerical association rules. *International Journal of Bio-Inspired Computation*, 2018, **11**(4): 239–248
- Cui Z H, Cao Y, Cai X J, Cai J H, Chen J J. Optimal LEACH protocol with modified bat algorithm for big data sensing systems in internet of things. *Journal of Parallel and Distributed Computing*, 2019, **132**: 217–229
- Bougherara M, Nedjah N, De Macedo Mourelle L, Rahmoun R, Sadok A, Bennouar D. IP assignment for efficient NoC-based system design using multi-objective particle swarm optimisation. *International Journal of Bio-Inspired Computation*, 2018, **12**(4): 203–213
- Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. In: *Proceedings of the 5th Conference on Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems*. Berlin, Germany: Springer-Verlag, 2001. 95–100
- Yang D D, Jiao L C, Gong M G. Adaptive multi-objective optimization based on nondominated solutions. *Computational Intelligence*, 2009, **25**(2): 84–108
- Xu Bin, Qi Rong-Bin, Qian Feng. Constrained multi-objective optimization with hybrid differential evolution and alpha constrained domination technique. *Control Theory and Applications*, 2012, **29**(3): 353–360  
(徐斌, 祁荣宾, 钱锋. 基于混合差分进化和 alpha 约束支配处理的多目标优化算法. *控制理论与应用*, 2012, **29**(3): 353–360)

- 18 Wang R, Zhang F X, Zhang T, Peter J F. Cooperative co-evolution with improved differential grouping method for large-scale global optimisation. *International Journal of Bio-Inspired Computation*, 2018, **12**(4): 214–225
- 19 Chen Zhi-Wang, Bai Xin, Yang Qi, Huang Xing-Wang, Li Guo-Qiang. Gaussian surrogate models for expensive interval multi-objective optimization problem. *Control Theory and Applications*, 2016, **33**(10): 1389–1398  
(陈志旺, 白铨, 杨七, 黄兴旺, 李国强. 求解昂贵区间多目标优化问题的高斯代理模型. *控制理论与应用*, 2016, **33**(10): 1389–1398)
- 20 Lin Hu, Peng Yong. Immune clonal algorithm with fitness sharing for multi-objective optimization. *Control Theory and Applications*, 2011, **28**(2): 206–214  
(林浒, 彭勇. 面向多目标优化的适应度共享免疫克隆算法. *控制理论与应用*, 2011, **28**(2): 206–214)
- 21 Tran D H, Cheng M Y, Cao M T. Hybrid multiple objective artificial bee colony with differential evolution for the time-cost-quality tradeoff problem. *Knowledge-Based Systems*, 2015, **74**: 176–186
- 22 Pooja, Chaturvedi P, Kumar P, Tomar A. A novel differential evolution approach for constraint optimisation. *International Journal of Bio-Inspired Computation*, 2018, **12**(4): 254–265
- 23 Wang W J, Wang H. A new differential evolution based on Gaussian sampling for forecasting urban water resources demand. *International Journal of Computing Science and Mathematics*, 2018, **9**(2): 155–162
- 24 Liu Ding, Zhang Xin-Yu, Chen Ya-Jun. Monocrystalline silicon diameter detection image threshold segmentation method using multi-objective artificial fish swarm algorithm. *Acta Automatica Sinica*, 2016, **42**(3): 431–442  
(刘丁, 张新雨, 陈亚军. 基于多目标人工鱼群算法的硅单晶直径检测图像阈值分割方法. *自动化学报*, 2016, **42**(3): 431–442)
- 25 Su Zhao-Pin, Zhang Guo-Fu, Jiang Jian-Guo, Yue Feng, Zhang Ting. Multi-objective approach to emergency resource allocation using none-dominated sorting based differential evolution. *Acta Automatica Sinica*, 2017, **43**(2): 195–214  
(苏兆品, 张国富, 蒋建国, 岳峰, 张婷. 基于非支配排序差异演化的应急资源多目标分配算法. *自动化学报*, 2017, **43**(2): 195–214)
- 26 Cui Z H, Zhang J J, Wang Y C, Cao Y, Cai X J, Zhang W S, et al. A pigeon-inspired optimization algorithm for many-objective optimization problems. *Science China Information Sciences*, 2019, **62**(7): Article No. 70212
- 27 Liu Tan, Gao Xian-Wen, Wang Li-Na. Multi-objective optimization for oil production process with compensating model error. *Control Theory and Applications*, 2015, **32**(5): 615–622  
(刘潭, 高宪文, 王丽娜. 补偿模型误差的采油过程多目标优化. *控制理论与应用*, 2015, **32**(5): 615–622)
- 28 Fu Ya-Ping, Huang Min, Wang Hong-Feng, Wang Xing-Wei. Multi-objective optimization model and algorithm for hybrid parallel machine scheduling problem. *Control Theory and Applications*, 2014, **31**(11): 1510–1516  
(付亚平, 黄敏, 王洪峰, 王兴伟. 混合并行机调度问题的多目标优化模型及算法. *控制理论与应用*, 2014, **31**(11): 1510–1516)
- 29 Ma C X. Network optimisation design of Hazmat based on multi-objective genetic algorithm under the uncertain environment. *International Journal of Bio-Inspired Computation*, 2018, **12**(4): 236–244
- 30 Niu Y, Tian Z Y, Zhang M Q, Cai X J, Li J W. Adaptive two-SVM multi-objective cuckoo search algorithm for software defect prediction. *International Journal of Computing Science and Mathematics*, 2018, **9**(6): 547–554
- 31 Qiao Jun-Fei, Han Gai-Tang, Zhou Hong-Biao. Knowledge-based intelligent optimal control for wastewater biochemical treatment process. *Acta Automatica Sinica*, 2017, **43**(6): 1038–1046  
(乔俊飞, 韩改堂, 周红标. 基于知识的污水生化处理过程智能优化方法. *自动化学报*, 2017, **43**(6): 1038–1046)
- 32 Parouha R P, Das K N. Economic load dispatch using memory based differential evolution. *International Journal of Bio-Inspired Computation*, 2018 **11**(3): 159–170
- 33 Raj E D, Babu L D D. A firefly inspired game dissemination and QoS-based priority pricing strategy for online social network games. *International Journal of Bio-Inspired Computation*, 2018, **11**(3): 202–217
- 34 Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
- 35 Zhang Yi, Chen Ping, Wan Xing-Yu. Multi-objective optimization of automotive brake based on orthogonal design of NSGA-II. *Application Research of Computers*, 2016, **33**(3): 720–725  
(张屹, 陈平, 万兴余. 基于正交设计 NSGA-II 算法的制动器多目标优化. *计算机应用研究*, 2016, **33**(3): 720–725)
- 36 Yuan Y, Xu H. Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Transactions on Automation Science and Engineering*, 2015, **12**(1): 336–353
- 37 Lu Yan-Xue, Zhao Chao-Fan, Wu Xiao-Feng, Han Xiao-Xia. Multi-objective optimization method research based on improved NSGA-II. *Application Research of Computers*, 2018, **35**(6): 1733–1737  
(路艳雪, 赵超凡, 吴晓锋, 韩晓霞. 基于改进的 NSGA-II 多目标优化方法研究. *计算机应用研究*, 2018, **35**(6): 1733–1737)
- 38 Mukhopadhyay A, Maulik U, Bandyopadhyay S. Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. *IEEE Transactions on Evolutionary Computation*, 2009, **13**(5): 991–1005
- 39 Handl J, Knowles J. An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(1): 56–76
- 40 Biswas S K, Chakraborty M, Purkayastha B. A rule generation algorithm from neural network using classified and misclassified data. *International Journal of Bio-Inspired Computation*, 2018, **11**(1): 60–70
- 41 Zhao B P, Xue Y, Xu B, Ma T H, Liu J F. Multi-objective classification based on NSGA-II. *International Journal of Computing Science and Mathematics*, 2018, **9**(6): 539–546

- 42 Liu Cong, Wan Xiu-Hua, Peng Dun-Lu, Wu Chun-Xue. Research on multiple distance clustering based on multi-objective evolutionary algorithm. *Application Research of Computers*, 2019, **36**(1): 94–98  
(刘丛, 万秀华, 彭敦陆, 邬春学. 基于多目标进化算法的多距离聚类研究. *计算机应用研究*, 2019, **36**(1): 94–98)
- 43 Huang Chao, Hu De-Min, Yu Xing. An improved NSGA-II algorithm based on vector space model. *Journal of Chinese Computer Systems*, 2015, **36**(2): 391–396  
(黄超, 胡德敏, 余星. 一种基于向量空间模型的 NSGA-II 改进算法. *小型微型计算机系统*, 2015, **36**(2): 391–396)
- 44 Xie Jiong-Liang, Zheng Jin-Hua. Research on cause for overlapping solutions and on their influence in NSGA-II algorithm. *Computer Engineering and Applications*, 2008, **44**(29): 69–72, 145  
(谢炯亮, 郑金华. NSGA-II 中重复个体产生原因分析及影响研究. *计算机工程与应用*, 2008, **44**(29): 69–72, 145)
- 45 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000, **8**(2): 173–195
- 46 Cui Z H, Zhang M Q, Wang H, Cai X J, Zhang W S, Chen J J. Hybrid many-objective cuckoo search algorithm with Lévy and exponential distributions, *Memetic Computing*, 2020, **12**(3): 251–265
- 47 Schott J R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization [Master thesis], Massachusetts Institute of Technology, USA, 1995.
- 48 Gong M G, Jiao L C, Du H F, Bo L F. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 2008, **16**(2): 225–255
- 49 Molina J, Santana L V, Hernández-Díaz A G, Coello C A C, Caballero R. g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 2009, **197**(2): 685–692



崔志华 太原科技大学教授. 主要研究方向为智能计算, 随机算法和组合优化. 本文通信作者.

E-mail: zhihua.cui@hotmail.com

(CUI Zhi-Hua Professor at Taiyuan University of Science and Technology. His research interest covers intelligent computing, stochastic algorithms, and

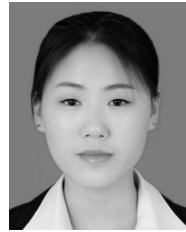
combinatorial optimization. Corresponding author of this paper.)



张茂清 同济大学博士研究生. 主要研究方向为多目标优化, 高维多目标优化, 深度学习以及应用.

E-mail: maoqing\_zhang@163.com

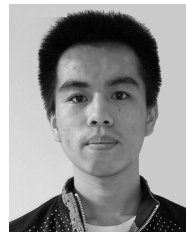
(ZHANG Mao-Qing Ph.D. candidate at Tongji University. His research interest covers multi-objective optimization, many-objective optimization, and deep learning and their applications.)



常宇 太原科技大学硕士研究生. 主要研究方向为动态高维多目标优化.

E-mail: yuchang78dd@163.com

(CHANG Yu Master student at Taiyuan University of Science and Technology. Her main research interest is dynamic many-objective optimization.)



张江江 太原科技大学硕士研究生. 主要研究方向为高维多目标优化.

E-mail: jiangofyouth@163.com

(ZHANG Jiang-Jiang Master student at Taiyuan University of Science and Technology. His main research interest is many-objective optimization.)



王晖 南昌工程学院副教授. 主要研究方向为进化计算, 群智能优化和大规模优化. E-mail: huiwang@nit.edu.cn

(WANG Hui Associate professor at Nanchang Institute of Technology. His research interest covers evolutionary computing, swarm intelligence optimization, and large-scale optimization.)



张文生 中国科学院自动化研究所研究员. 主要研究方向为大数据知识挖掘, 人工智能, 机器学习, 嵌入式视频图像处理. E-mail: wensheng.zhang@ia.ac.cn

(ZHANG Wen-Sheng Professor at the Institute of Automation, Chinese Academy of Sciences. His research interest covers large data knowledge mining, artificial intelligence, machine learning, and embedded

video image processing.)