

# 基于 SOM 聚类 and 自适应算子选择的 高维多目标进化算法

钟沛龙<sup>1,2</sup>, 黎 明<sup>1,2</sup>, 何 超<sup>3</sup>, 陈 昊<sup>1,2</sup>

(1. 南昌航空大学信息工程学院, 江西南昌 330063; 2. 南昌航空大学无损检测技术教育部重点实验室, 江西南昌 330063;  
3. 南京航空航天大学自动化学院, 江苏南京 211106)

**摘 要:** 在高维多目标进化算法中, 通常利用重组算子产生优质子代来引导种群搜索, 已有研究表明, 利用相似个体进行重组可以提高子代个体质量. 由于自组织映射(Self-Organizing Mapping, SOM)网络能够通过聚类的方式保持种群个体原有的拓扑逻辑关系并获得个体的相似信息, 因此本文提出一种基于 SOM 聚类 and 自适应算子选择的高维多目标进化算法(Many-Objective Evolutionary Algorithm based on SOM Clustering and Adaptive Operator Selection, MaOEA-SCAOS). 本文首先通过自组织映射网络进行种群分类, 提取个体数据结构信息, 并利用相似性构建邻域交配池; 然后根据类内个体支配信息进行自适应算子选择, 提高算法搜索和收敛性能; 最后, 采用环境选择策略对种群进行多样性管理以保证种群在帕累托前沿均匀分布. 仿真结果表明, 本文提出的基于 SOM 聚类 and 自适应算子选择(SOM Clustering and Adaptive Operator Selection, SCAOS)方法在处理高维多目标优化问题时具有较强的竞争力并且性能指标整体优于其他方法.

**关键词:** 高维多目标优化; 自组织映射网络; 聚类; 自适应选择; 进化算法

**中图分类号:** TP18

**文献标识码:** A

**文章编号:** 0372-2112(2022)08-1959-16

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20201438

## Many-Objective Evolutionary Algorithm Based on SOM Clustering and Adaptive Operator Selection

ZHONG Pei-long<sup>1,2</sup>, LI Ming<sup>1,2</sup>, HE Chao<sup>3</sup>, CHEN Hao<sup>1,2</sup>

(1. School of Information Engineering, Nanchang Hangkong University, Nanchang, Jiangxi 330063, China;  
2. Key Laboratory of Nondestructive Testing, Ministry of Education, Nanchang Hangkong University, Nanchang, Jiangxi 330063, China;  
3. School of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China)

**Abstract:** In the many-objective evolutionary algorithm, recombination operators are usually used to generate high-quality offspring to guide the population search. Previous studies have shown that using similar individuals to reorganize can improve the quality of individual offspring. Since the self-organizing mapping(SOM) network can maintain the original topological relationship of the population individuals and obtain the similar information of the individuals through clustering, this paper proposes a many-objective evolutionary algorithm based on SOM clustering and adaptive operator selection (MaOEA-SCAOS). First, the proposed method use self-organizing mapping network to classify the population, extract individual data structure information, and use similarity to build a neighborhood mating pool. Then the method select the adaptive operator based on the individual dominance information in the class to improve the search and convergence performance. Finally, the environmental selection strategy is adopted to manage the diversity of the population to ensure that the population is evenly distributed in the Pareto front. The experimental simulation results show that the SOM clustering and adaptive operator selection(SCAOS) method proposed in this paper has strong competitiveness while dealing with many-objective optimization problems, and the overall performance index is better than other methods.

收稿日期: 2020-12-15; 修回日期: 2021-06-04; 责任编辑: 王天慧

基金项目: 国家自然科学基金(No.61772255, No.61866025, No.61866026, No.61961030); 江西省教育厅科学技术项目(No.GJJ170608); 江西省优势科技创新团队计划项目(No.20181BCB24008); 江西省自然科学基金(No.20181BAB202025, No.20202BAB204036); 无损检测教育部重点实验室开放基金(No.EW201708505); 江西省研究生创新专项资金项目(No.YC2020-S520)

Key words: many-objective optimization; self-organizing mapping network; clustering; adaptive selection; evolutionary algorithm

## 1 引言

在实际生活中,多目标优化问题存在于各类现实场景中,往往存在多个彼此相互冲突的目标,其中当目标个数为4个及4个以上时,多目标优化问题又被称为高维多目标优化问题(Many-Objective optimization Problems, MaOPs)<sup>[1]</sup>. 为了有效地求解高维多目标优化问题,现有的文献采用了多种策略来增强高维多目标进化算法的性能<sup>[2-4]</sup>,例如引入新的指标策略<sup>[5-7]</sup>、构造新型支配关系<sup>[8-11]</sup>等.

现有方法主要将工作集中在如何去平衡算法的收敛性和多样性,并保持向 Pareto 前沿的选择压力. 如 NSGA-III<sup>[12]</sup>利用分解的思想来保持种群的多样性,并且使用 Pareto 支配加快种群的收敛速度; $\epsilon$ -MOEA<sup>[13]</sup>利用 $\epsilon$ -支配和网格框架在环境选择准则中提出了收敛性和多样性度量来保存精英个体进入存档;KnEA<sup>[14]</sup>, RVEA<sup>[15]</sup>和 MOEA/DDU<sup>[16]</sup>等也是如此. 然而上述算法并没有考虑种群个体的数据结构信息及个体间的相似性. 尽管部分基于种群个体数据结构信息的多目标进化算法被提出,但它们主要利用邻域关系发掘多目标优化问题的个体结构,此外仅采用此类结构信息,并不能最有效地产生优质个体,具有一定的局限性. 如基于多目标优化的细胞遗传算法(Cell genetic algorithm based on Multi-Objective optimization, MOCeal)<sup>[17]</sup>通过将每个个体置于一个网格结构的元胞中来构建个体之间的邻域关系,其中每个个体仅同它的邻域个体交互产生新个体,该方式对目标空间的搜索能力有限;基于规则模型的多目标分布估计算法(Regularity Model based Multi-objective Estimation of Distribution Algorithm, RM-MEDA)<sup>[18]</sup>应用主成分分析方法(Principal Component Analysis method, PCA)<sup>[19]</sup>将 Pareto 集划分为若干个不相交的类,在每个类中开展主成分分析并建立概率模型,并在模型中抽样产生新个体以更新种群,然而类的数目  $K$  值难以设置,不同的  $K$  值,对算法性能的表现影响较大.

现有文献显示,利用相似个体作为父代进行重组能够显著提高子代个体的质量,并提高算法的搜索效率<sup>[20,21]</sup>. 另外,聚类算法是一类将未知标签的数据对象集进行分组的无监督学习方法,其分类的结果能够保证同一类的数据对象间的相似性较高,不同类的数据对象间的相似性较低<sup>[22]</sup>. 并且,聚类算法仅利用数据本身的特征以及数据之间的相似性关系进行分类. 因此自组织映射网络(Self-Organizing Map, SOM)模型能够通过无监督学习的方式对来自高维输入空间的训练点

产生低维的表征,并保持输入数据间的近邻关系,从而实现聚类的目的. 综上所述,SOM模型能够通过聚类的方式保持种群个体原有的拓扑逻辑关系,并可以自动地根据个体的相似性构建邻域结构. 因此本文提出了一种基于 SOM 聚类和自适应算子选择的方法(SOM Clustering and Adaptive Operator Selection, SCAOS)并将其作用于高维多目标进化算法中. 与现有的基于数据结构的方法不同,SCAOS可以更好地利用种群个体的数据结构信息及个体之间的相似性来增强高维多目标进化算法的性能. SCAOS的主要贡献在于:(1)采用 SOM 聚类算法提取数据结构信息,获得个体间相似性,将每个个体与相应的神经元相关联,相似个体分配到相邻神经元;(2)在类内利用个体支配信息进行自适应算子选择来产生优质子代引导个体重组,增强算法性能;(3)将提出的重组方法与环境选择策略相结合,增强了种群个体的多样性和收敛性.

## 2 相关工作

自组织映射网络模型由 Kohonen<sup>[23]</sup>提出,是一种无监督学习的神经网络模型,可以将输入的  $n$  维空间数据映射到一个较低的维度输出,并能够保持数据原有的拓扑逻辑关系. SOM 这类基于自组织映射的神经网络,已经被广泛应用于数据聚类<sup>[24]</sup>、图像分割<sup>[25]</sup>、数据可视化<sup>[26]</sup>等实际应用中. 在处理多目标优化问题方面,根据 SOM 的使用方式不同,基于 SOM 的多目标进化算法被分为两类:第一类是直接应用 SOM 模型产生新解,第二类是利用 SOM 引导算法的搜索.

在第一类中,Hakimi-Asiabar 等<sup>[27]</sup>使用精英个体对 SOM 模型进行训练,然后将被训练后的 SOM 输出层神经元的权值作为新产生的解,提高了产生优质新解的效率. Zhan 等<sup>[28]</sup>利用 SOM 来学习多目标优化问题 Pareto 解集的流型结构,然后通过均匀随机扰动 SOM 的神经元权值的方式来产生新个体以提高算法在目标空间中的探索能力. 在第二类中,Norouzi 等<sup>[29]</sup>使用 NSGA-II 先进化一定代数以对目标空间进行初步勘探,然后利用 SOM 将种群划分到若干个子种群中,每个子种群独立开展遗传操作并且进化若干代,有效提高了算法的搜索性能. Gu 等<sup>[30]</sup>提出了一种基于 SOM 的权值向量设计方法,在基于分解的 MOEA 的权值向量设计中利用 SOM 网络来发现个体在目标空间的分布.

以上研究表明,SOM 网络模型不仅能够通过提高产生个体的质量来增强算法在处理多目标优化问题时的性能,而且还可以通过引导算法搜索使优质的个体

向着 Pareto 前沿移动,进而提高多目标优化算法的搜索效率.此外,利用 SOM 进行聚类的方式较为简单,无需设置初始类个数,对初始数据不太敏感.因此,基于 SOM 的多目标进化算法能够适用于各种复杂多目标问题的处理,具有较好的应用前景.图 1 所示为典型的 SOM 拓扑结构,输入层为  $n$  维空间,输出层为二维空间,每个神经元  $u$  上都有其对应的权值向量  $W^u=(w_1^u, w_2^u, \dots, w_n^u)$ ,神经元权值的维数等于目标个体的维数.

可以通过竞争与合作的方式来对 SOM 网络进行训练.

(1) 竞争:输入个体  $X$ ,根据个体  $X$  与神经元的欧氏距离判断获胜神经元,当欧氏距离最小时,此神经元即为获胜神经元,求解如下:

$$u' = \arg \min_{1 \leq u \leq N} \|X^c - W^u\|_2 \quad (1)$$

$u'$  为当前个体  $X^c$  的获胜神经元,其中,  $1 \leq c \leq N$ .

(2) 合作:在训练过程中,除了更新获胜神经元的权值,还要更新与获胜神经元邻近神经元的权值.表达式如下:

$$U = \left\{ u \mid \|Z^u - Z^{u'}\|_2 < \sigma \right\} \quad (2)$$

$$W^u = W^u + \tau \times e^{-\|Z^u - Z^{u'}\|_2} \times (X^c - W^u) \quad (3)$$

其中,  $U$  为获胜神经元  $u'$  的邻近个体集合,  $\sigma$  为邻域半径,  $\tau$  为当前训练的学习率,  $u \in U$ .

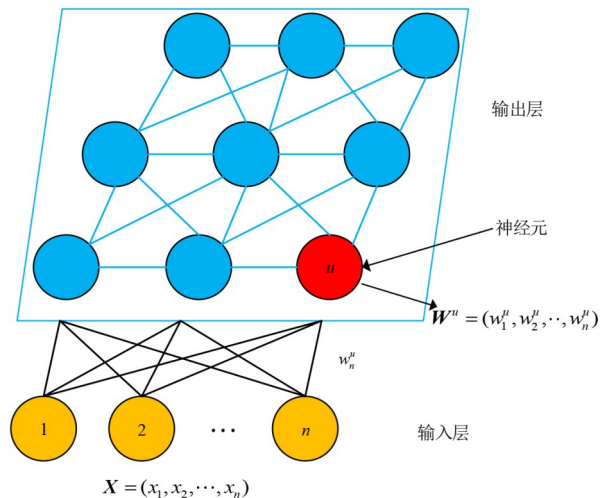


图 1 典型的 SOM 拓扑结构

### 3 基于 SOM 聚类 and 自适应算子选择的高维多目标进化算法

#### 3.1 SOM 聚类

本文利用 SOM 网络特性来有效提取个体数据信息,获得个体间相似性,并得到相似个体的集合.算法 1 为 SOM 聚类算法的具体框架.

在算法 1 中,  $T_{\max}$  和  $t$  分别表示最大迭代次数和当前迭代次数. SOM 聚类算法先对神经元的权值向量  $W$  及神经元邻域半径  $\sigma_0$  进行初始化操作(第 1 行).然后通过阈值  $\delta$  限制学习率及邻域半径的更新策略来训练权值向量,其具体步骤为:首先学习率从  $\tau_0$  降低到  $\delta$ ,邻域的大小从  $\sigma_0$  降低到  $\sigma$ ,持续优化权值向量,使得输出层能够表示出输入个体的分布情况;然后学习率  $\tau_0$  降低到  $\delta$  后进一步减小,而邻域  $\sigma$  则保持不变,这种方式能够自动优化网络结构,可以在一定程度上保持个体分布的多样性(第 5~10 行);再通过式(1)获取当前个体最接近的神经元  $u'$ ,由式(2)获得邻近神经元集合  $U$ ,式(3)更新邻近神经元的权值向量  $W^u$ (第 11~16 行);最后将每个个体与神经元关联,相似个体将被分配到邻

#### 算法 1 SOM 聚类

输入: 训练集  $TS$ , 学习率  $\tau_0$ , 阈值  $\delta$

输出: 神经元权值向量  $W^u$ , 相似个体集合  $X^u$

1. 初始化神经元权值向量  $W = \{W^1, W^2, \dots, W^N\}$ , 初始邻域半径

$$\sigma_0 = \frac{\sqrt{\sum_{i=1}^{M-1} \frac{N_i^2}{M-1}}}{2};$$

2. WHILE  $t \leq T_{\max}$  DO

3. FOR  $c = 1$  TO  $N$  DO

4. /\*调整神经元学习率  $\tau$  及邻域半径  $\sigma^*$ \*/

5. IF  $\tau > \delta$

$$6. \quad \tau = \tau_0 \times \left( 1 - \frac{(t-1) \times N + c}{T_{\max} \times N} \right);$$

$$7. \quad \sigma = \sigma_0 \times \left( 1 - \frac{(t-1) \times N + c}{T_{\max} \times N} \right);$$

8. ELSE

$$9. \quad \tau = \tau_0 \times \left( 1 - \frac{(t-1) \times N + c}{T_{\max} \times N} \right);$$

10. END

11. /\*寻找最接近  $X^c$  的神经元  $u'$ \*/

$$12. \quad u' = \arg \min_{1 \leq u \leq N} \|X^c - W^u\|_2;$$

13. /\*获得邻近神经元集合  $U$ \*/

$$14. \quad U = \left\{ u \mid \|Z^u - Z^{u'}\|_2 < \sigma \right\};$$

15. /\*更新邻近神经元的权值向量  $W^u$ \*/

$$16. \quad W^u = W^u + \tau \times e^{-\|Z^u - Z^{u'}\|_2} \times (X^c - W^u);$$

17. END

18. /\*将每个个体与神经元相关联\*/

19. FOR  $i = 1$  TO  $N$  DO

20. /\*随机选择个体  $X$ , 获得与个体  $X$  最接近的神经元  $u$ , 将  $X$  与  $u$  相关联起来\*/

$$21. \quad X_i^u = X_i;$$

22. END

23. END

近神经元,得到相似个体集合 $X^u$ (第19~22行)。

### 3.2 自适应算子选择

在多目标优化过程中,相似个体之间重组,能够提高产生新个体的质量,增强算法的局部搜索能力<sup>[20]</sup>。本文依据此特点,设计一种自适应算子选择方式,充分利用相似个体数据信息产生高质量的子代,引导算法在目标空间中的搜索。其利用SOM网络得到的相似个体集合,建立父代交配池,根据交配池内个体支配信息选择重组算子,产生优质子代。具体步骤如算法2所示。

在算法2中,首先根据相似个体集合 $X^u$ ,获得当前个体交配池MP,大小设置为 $S$ (第2行);然后获得当前交配池内个体支配信息(第3行和第4行);最后根据个体支配信息选择算子,若个体均互不支配,从交配池内随机选择2个作为父代个体进行重组产生子代 $y$ ,否则,从相似个体集合随机选取2个不同于当前个体的个体作为父代个体进行重组产生子代 $y$ (第5~11行)。本文同时引入2种不同的重组算子,分别应用在交配池内和相似个体集合中产生子代,具体如算法3和算法4所示。

#### 算法2 自适应算子选择

输入:相似个体集合 $X^u$ ,交配池大小 $S$

输出:子代 $y$

1. FOR  $i = 1$  TO  $N$  DO
2.  $MP = \bigcup_{s=1}^S \{X^{u^s}\} \cup \{X^i\}$ ;
3.  $PN = \text{Nondominated}(MP)$ ;
4.  $PD = \text{Dominated}(MP)$ ;
5. IF  $PD \in \emptyset$
6. Parent = MP;
7.  $y = \text{SBX}(X^a, X^b)$ ;
8. ELSE
9. Parent = X;
10.  $y = \text{DE}(X^a, X^b, X^i)$ ;
11. END
12. END

算法3为在交配池内选择父代个体进行子代生成。首先在交配池内选取2个父代个体 $X^a, X^b$ ;其次采用模拟二进制交叉算子(Simulated Binary Crossover, SBX)<sup>[31]</sup>产生子代 $y'$ ;然后为了防止新产生的子代 $y'$ 超出个体的边界值,对其进行修正为 $y''$ ;最后通过多项式变异操作<sup>[32]</sup>产生最终的子代 $y$ 。

算法4为在相似个体集合选择父代个体进行子代生成。首先在相似个体集合中随机选择2个不同的个体 $X^a, X^b$ 和当前个体 $X$ 作为父代个体;其次采用差分进化交叉算子(Differential Evolution, DE)<sup>[33]</sup>产生子代 $y'$ ;然后修正 $y'$ 为 $y''$ ;最后由多项式变异生成子代 $y$ 。

值得注意的是,相较于比较所有个体,在交配池内

进行比较可以有效减少计算复杂度并提取个体之间的相互信息,更加有利于优质个体的产生。此外,为了保持最终产生的子代个体数为 $N$ ,在进行SBX交叉算子操作时只产生1个子代。

#### 算法3 模拟二进制交叉

输入: MP

输出:  $y$

1.  $[X^a, X^b] = \text{random}(MP)$ ;
2.  $y' = 0.5 \times [(1 + \alpha) \cdot X^a + (1 - \alpha) \cdot X^b]$ ;
3. 其中,  $\alpha = \begin{cases} (\text{rand} \times 2)^{\frac{1}{1+\eta}}, & \text{rand} \leq 0.5 \\ \left(\frac{1}{2 - \text{rand} \times 2}\right)^{\frac{1}{1+\eta}}, & \text{其他} \end{cases}$ ;
4. /\*修正 $y'$ \*/
5.  $y'' = \begin{cases} X_{\max}, & y' > X_{\max} \\ X_{\min}, & y' < X_{\min} \\ y', & \text{其他} \end{cases}$ ;
6. /\*多项式变异\*/
7.  $y = \begin{cases} y'' + \gamma \times (X_{\max} - X_{\min}), & \text{rand} < \rho \\ y'', & \text{其他} \end{cases}$ ;
8.  $\gamma = \begin{cases} \left[ 2r + (1 - 2r) \left( \frac{\|X_{\max}\| - \|y''\|}{\|X_{\max}\| - \|X_{\min}\|} \right)^{\xi+1} \right]^{\frac{1}{\xi+1}} - 1, & r < 0.5 \\ 1 - \left[ 2r + (1 - 2r) \left( \frac{\|X_{\max}\| - \|y''\|}{\|X_{\max}\| - \|X_{\min}\|} \right)^{\xi+1} \right]^{\frac{1}{\xi+1}}, & \text{其他} \end{cases}$ ;

#### 算法4 差分进化交叉

输入:  $X^u$

输出:  $y$

1.  $[X^a, X^b, X] = \text{random}(X) \cup X$ ;
2.  $y' = \begin{cases} X - F \times (X^a - X^b), & \text{rand} \leq \text{CR} \\ X, & \text{其他} \end{cases}$ ;
3. /\*其中,  $F$ 和 $\text{CR}$ 是DE的控制参数\*/
4. /\*修正 $y'$ \*/
5.  $y'' = \begin{cases} X_{\max}, & y' > X_{\max} \\ X_{\min}, & y' < X_{\min} \\ y', & \text{其他} \end{cases}$ ;
6. /\*多项式变异\*/
7.  $y = \begin{cases} y'' + \gamma \times (X_{\max} - X_{\min}), & \text{rand} < \rho \\ y'', & \text{其他} \end{cases}$ ;
8.  $\gamma = \begin{cases} \left[ 2r + (1 - 2r) \left( \frac{\|X_{\max}\| - \|y''\|}{\|X_{\max}\| - \|X_{\min}\|} \right)^{\xi+1} \right]^{\frac{1}{\xi+1}} - 1, & r < 0.5 \\ 1 - \left[ 2r + (1 - 2r) \left( \frac{\|X_{\max}\| - \|y''\|}{\|X_{\max}\| - \|X_{\min}\|} \right)^{\xi+1} \right]^{\frac{1}{\xi+1}}, & \text{其他} \end{cases}$ ;

### 3.3 MaOEA-SCAOS 算法描述

MaOEA-SCAOS 如算法 5 所述. 首先初始化种群  $P$ 、神经元权值向量  $W$  和训练集  $TS$ ; 其次利用 SOM 聚类算法获得相似个体集合  $X^u$ ; 然后通过集合  $X^u$ , 构建邻域交配池, 通过自适应算子选择操作重组生成优质子代; 最后通过环境选择策略选择个体更新种群. 值得注意的是本文环境选择策略框架为 NSGA-III 算法中环境选择框架. NSGA-III<sup>[12]</sup> 是最著名的进化多目标优化算法之一. 它是在 NSGA-II<sup>[34]</sup> 的基础上进一步提出的, 该算法的基本算法框架和 NSGA-II 算法相似, 但是在选择算子上进行了改进, 它通过提供和自适应更新一组预定义的参考点来维持种群个体间的多样性. 由于文章篇幅限制, NSGA-III 的环境选择策略的详细步骤见文献[12]. 另外在算法 5 的第 7 行, TP 表示暂时保存训练好的个体, 用于训练集  $TS$  的更新(第 24 行), 可以有效减少 SOM 网络的训练时间.

#### 算法 5 MaOEA-SCAOS 算法框架

输入: 种群规模  $N$ , 最大迭代次数  $T_{\max}$

输出: 最终种群  $P$

```

1. /*初始化种群  $P$ 、神经元权值向量  $W$  和训练集  $TS$ */
2.  $P = \text{Initialize}(N)$ 
3.  $TS = P, W = \{TS\}$ ;
4. WHILE  $t \leq T_{\max}$  DO
5. /*SOM 聚类得到相似个体集合  $X^u$ */
6.  $X^u = \text{SOM}(TS)$ ;
7.  $TP = X^u$ 
8. /*自适应算子选择*/
9. FOR  $i = 1$  TO  $N$  DO
10.  $MP = \bigcup_{s=1}^S \{X^{u^s}\} \cup \{X^i\}$ ;
11.  $PN = \text{Nondominated}(MP)$ ;
12.  $PD = \text{Dominated}(MP)$ ;
13. IF  $PD \in \emptyset$ 
14.   Parent =  $MP$ ;
15.    $y = \text{SBX}(X^a, X^b)$ ;
16. ELSE
17.   Parent =  $X$ ;
18.    $y = \text{DE}(X^a, X^b, X^c)$ ;
19. END
20. /*更新种群  $P^*$ */
21.  $P = \text{EnvironmentalSelection}(P \cup y)$ ;
22. END
23. /*更新训练集  $TS$ */
24.  $TS = P \setminus TP$ ;
25. END

```

## 4 实验仿真与分析

为了验证本文算法 (MaOEA-SCAOS) 的有效性, 选取 2 种利用个体数据信息的多目标进化算法 (MO-Cell<sup>[17]</sup>, RM-MEDA<sup>[18]</sup>) 和 3 种多目标进化算法 (NSGA-III<sup>[12]</sup>, MaOEA-CSS<sup>[35]</sup>, KnEA<sup>[14]</sup>) 与本文所提算法进行对比分析. 所有的算法均是在多目标进化算法开源平台 PlatEMO<sup>[36]</sup> 上运行.

### 4.1 测试问题与性能指标

在实验中, 采用 DTLZ 测试问题集和 WFG 测试问题集. DTLZ 测试问题集都有  $n = M + k - 1$  个决策变量, 其中  $M$  为目标维数. 根据文献[37]中的决策变量设置建议, DTLZ1 中设置  $k = 5$ , DTLZ2-DTLZ6 中设置  $k = 10$ , DTLZ7 中设置  $k = 20$ . WFG 测试问题集中所有决策变量个数  $n = k + l$ , 其中设置  $k = M - 1, l = 10$ .

为了验证 MaOEA-SCAOS 获得的近似 Pareto 前沿的收敛性和多样性, 在实验研究中采用反世代距离 (Inverted Generational Distance, IGD) 和超体积指标 (Hyper-Volume, HV). HV 指标表示的是被种群中所有个体支配并且以参考点  $R$  为边界的区域, 其表达的是非支配个体覆盖的目标空间区域大小. 给定参考点  $R$ , 获得的种群个体  $P^*$ , 那么关于参考点  $R$  的超体积定义为

$$HV(P^*, R) = L \left( \bigcup_{i=1}^{|S|} v_i \right) \quad (4)$$

其中,  $L$  为 Lebesgue 测度,  $|S|$  表示种群个体  $P^*$  支配  $R$  的数目,  $v_i$  表示参考点与种群中第  $i$  个个体构成的超体积. 从 HV 定义可看出, HV 指标的值越大, 算法求得的种群质量越高, 算法性能优化效果越佳.

IGD 指标定义为每个参考点与其最近个体距离之和的平均值. 其公式定义如下:

$$IGD(P^*, P) = \frac{\sum_{x \in P^*} \min_{y \in P} \text{dis}(x, y)}{|P|} \quad (5)$$

其中,  $\text{dis}(x, y)$  表示个体与参考点间的欧氏距离, 由算法获得的种群个体记为  $P^*$ ,  $P$  为真实的 Pareto 前沿面均匀采样得到的一组参考点,  $|P|$  为分布在真实 Pareto 前沿面上种群  $P$  的规模. IGD 的值越小表示所得种群个体越是靠近真实 PF, 个体分布越均匀. IGD 指标既可以评价收敛性, 也可以评价多样性.

### 4.2 实验参数设置

(1) 种群大小  $N$ : 本文采用文献[12]中设置的方法. 并且为了公平对比, 其他算法的种群规模设置相同. 所有算法种群大小设置如表 1 所示.

(2) 终止条件: 算法运行一次的终止条件是最大评价次数. 最大评价次数等于种群规模乘以最大迭代次数. 本文在 3 目标和 5 目标的最大迭代次数设置为 500, 在 8 目标和 10 目标上设置为 1 000.

表 1 种群大小设置

目标维数 $M$	$(H_1, H_2)$	种群大小 $N$
3	(13,0)	105
5	(5,0)	126
8	(3,2)	156
10	(3,2)	275

(3)统计方法:所有算法在每个测试问题上均独立运行 20 次,对 MaOEA-SCAOS 算法和其他对比算法获得的结果采用 Wilcoxon 秩和检验方法进行比较,其中均值分析的显著性水平设置为 5%. 根据 Wilcoxon 秩和检验方法,其中“+”表示对比算法所得结果要优于 MaOEA-SCAOS,“-”表示对比算法所得结果要劣于 MaOEA-SCAOS,“=”表示对比算法和 MaOEA-SCAOS 获得的结果没有明显的差异.

(4)对比算法主要参数设置:对于 MOCeII 和 NSGA-III 中 DE 运算符的控制参数  $F$  和  $CR$  分别设置为 0.3 和 0.2;在 RM-MEDA 中,本地 PCA 算法中的簇数  $K$  设置为 5;在 MaOEA-CSS 中,环境选择的阈值  $t$  大小设置为 0;KnEA 的参数设置参照文献 [14];本文算法 MaOEA-SCAOS 中初始 SOM 学习率  $\tau_0$ 、交配池大小  $MP$  和学习率下降阈值  $\delta$  分别设置为 0.9, 5 和 0.1.

(5)交叉和变异算子设置:在 MaOEA-SCAOS 算法中 SBX 分布因子  $\eta$  设置为 20, DE 控制参数  $F$  和  $CR$  分别设置为 0.5 和 1. 多项式变异的变异分布因子  $\zeta$  均设置为 20, 变异概率  $\rho$  均为  $1/n$  ( $n$  代表决策变量数目).

### 4.3 参数分析

以 DTLZ1~DTLZ7 测试问题作为目标函数进行分析,使用不同参数的 MaOEA-SCAOS 对每种测试问题运用 IGD 性能指标值随性能评价次数的变化进行分析,其中详细分析了交配池大小  $MP$  和阈值  $\delta$ .

#### 4.3.1 交配池大小 $MP$ 分析

为了客观评价交配池大小对算法的影响,本文选取 3, 5, 10, 15 和 20 五种不同大小的交配池,并设置相同的其他参数来验证交配池对算法的影响. 图 2 显示了 MaOEA-SCAOS 在 DTLZ1, DTLZ3, DTLZ6 和 DTLZ7 测试问题上独立运行 20 次后的平均 IGD 指标值随着评价次数的变化曲线. 从图 2 可以看出,不同的  $MP$  在不同测试问题上有着不同的性能,其中较大的  $MP$  所得算法性能较低,较小的  $MP$  所得算法性能较优,故较小的交配池适合 MaOEA-SCAOS 算法. 可以看出,当  $MP=5$  时, MaOEA-SCAOS 所呈现的收敛性能较好,故本文所设置交配池大小为 5.

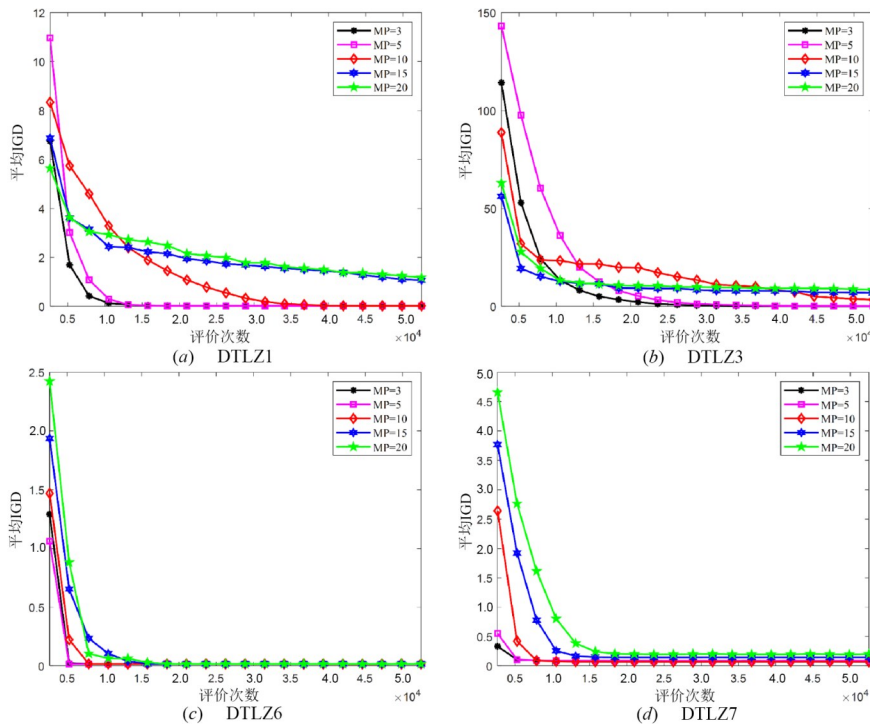


图 2 不同交配池大小在 DTLZ 测试问题独立运行 20 次平均 IGD 指标值

#### 4.3.2 阈值 $\delta$ 分析

不同阈值  $\delta$ , 决定了邻域的多样性分布情况, 最终会影响算法的收敛性. 为了获取最佳阈值参数, 选取阈值大小为 0.1, 0.3, 0.5, 0.7 来进行验证, 其他参数设置相同.

图 3 显示了 MaOEA-SCAOS 在 DTLZ1, DTLZ3, DTLZ6 和 DTLZ7 测试问题上独立运行 20 次后的平均 IGD 指标值随着评价次数的变化曲线. 由图 3 可以看出, 较小的阈值得到算法的收敛性较优, 故本文设置阈值  $\delta$  大小为 0.1.

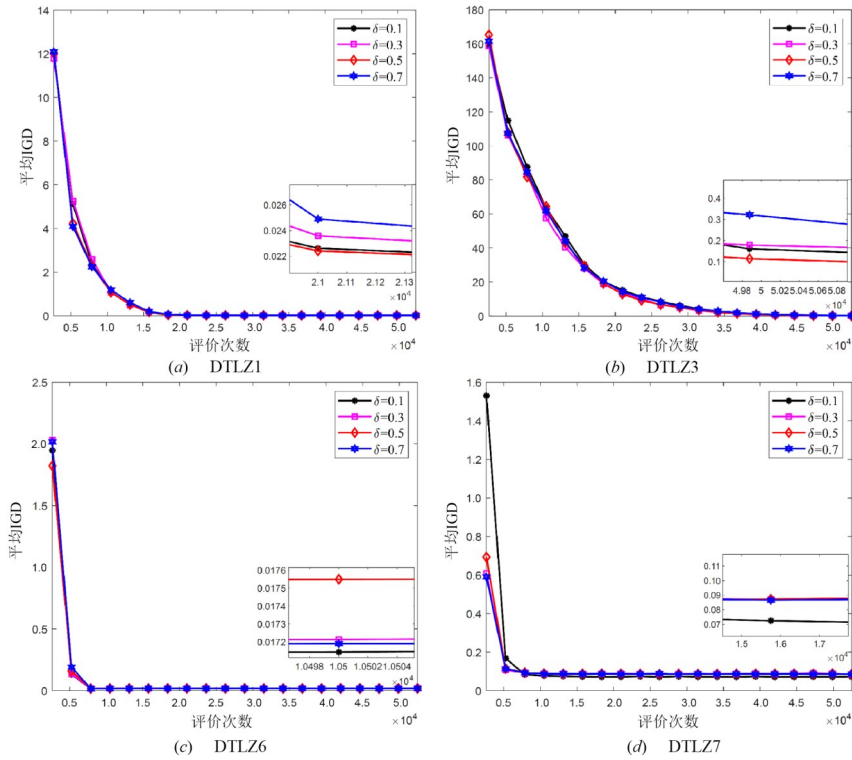


图3 不同阈值大小在DTLZ测试问题独立运行20次平均IGD指标值

4.4 计算复杂度分析

本小节分析 MaOEA-SCAOS 算法在一代内的最坏情况总体计算复杂度. 在 SOM 聚类阶段, 首先训练阶段通过个体寻找最近神经元对其权值向量进行更新; 其次聚类操作将每一个个体与最近神经元相匹配, 将相似个体分配到邻近神经元, 因此该过程的计算复杂度为  $O(N^2)$ , 其中  $N$  为种群个体数量. 在自适应算子选择阶段, 首先考虑交配选择, 对于一个个体规模为  $N$  和  $M$  目标的优化问题, 构建交配池大小  $MP$  需要  $O(MN \cdot MP)$ ; 其次交叉变异对父代个体的每个决策变量, 产生  $N$  个后代需要  $O(DN)$  的运行时间, 其中,  $D$  为决策变量

的个数. 在环境选择阶段, 在最坏的情况下需要  $O(MN^2)$  的运行时间. 其他操作的复杂度较小. 因此, MaOEA-SCAOS 在一代内的最坏情况总体计算复杂度为  $O(MN^2)$ . MaOEA-SCAOS 算法在计算上非常高效.

4.5 算法在 WFG 问题上对比分析

表 2 和表 3 分别给出了 6 个算法在 WFG1-WFG9 测试问题上进行 20 次独立计算 HV 和 IGD 指标的实验结果, 其中表 2 汇总了对比算法所获得 HV 值的均值和标准差 (括号内为标准差), 表 3 汇总了对比算法所获得 IGD 值的均值和标准差 (括号内为标准差), 其中用黑色粗体突出最好的结果.

表 2 6 种算法在 WFG1-WFG9 测试问题上获得的 HV 指标值的实验结果

问题	M	MaOEA-SCAOS	MOCeII	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
WFG1	3	8.850 9e-1 (2.54e-2)	6.233 9e-1 (3.99e-2) -	4.837 1e-2 (9.38e-3) -	<b>9.239 5e-1</b> (6.92e-3) +	8.955 7e-1 (1.38e-2) =	9.001 2e-1 (2.13e-2) +
	5	8.206 1e-1 (3.39e-2)	4.404 2e-1 (3.23e-2) -	1.750 7e-1 (1.00e-2) -	<b>9.341 4e-1</b> (2.66e-2) +	9.187 6e-1 (2.86e-2) +	8.595 6e-1 (4.73e-2) +
	8	9.946 4e-1 (1.69e-2)	4.361 5e-1 (4.08e-2) -	2.232 1e-1 (3.16e-3) -	9.954 0e-1 (1.29e-3) +	9.949 8e-1 (1.02e-3) +	<b>9.990 3e-1</b> (2.73e-4) +
	10	9.989 7e-1 (4.24e-4)	3.649 3e-1 (3.15e-2) -	2.164 5e-1 (1.69e-3) -	9.974 8e-1 (9.71e-4) -	9.967 4e-1 (6.15e-4) -	<b>9.992 9e-1</b> (2.98e-4) +
WFG2	3	<b>9.322 9e-1</b> (9.15e-4)	9.132 1e-1 (2.07e-3) -	8.828 5e-1 (4.04e-3) -	9.301 6e-1 (2.23e-3) -	8.926 9e-1 (1.66e-2) -	9.322 5e-1 (6.85e-4) =
	5	<b>9.931 6e-1</b> (1.14e-3)	9.679 0e-1 (3.51e-3) -	7.956 4e-1 (1.47e-2) -	9.887 9e-1 (1.52e-3) -	9.407 3e-1 (1.16e-2) -	9.926 6e-1 (1.02e-3) =
	8	9.969 3e-1 (1.86e-3)	9.854 4e-1 (3.51e-3) -	6.869 3e-1 (2.84e-2) -	9.937 4e-1 (1.03e-3) -	9.689 4e-1 (7.11e-3) -	<b>9.974 4e-1</b> (8.93e-4) =

续表

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
	10	9.975 9e-1 (1.31e-3)	9.838 3e-1 (3.57e-3) -	6.377 8e-1 (2.43e-2) -	9.950 2e-1 (7.25e-4) -	9.814 8e-1 (5.32e-3) -	<b>9.976 1e-1</b> <b>(1.05e-3) =</b>
WFG3	3	3.885 9e-1 (5.09e-3)	3.885 2e-1 (2.79e-3) =	3.072 0e-1 (7.73e-3) -	3.782 2e-1 (7.57e-3) -	2.209 1e-1 (3.36e-2) -	<b>3.893 7e-1</b> <b>(2.92e-3) =</b>
	5	1.551 0e-1 (1.89e-2)	<b>1.749 2e-1</b> <b>(1.77e-2) +</b>	1.890 3e-2 (1.06e-2) -	6.319 9e-2 (2.39e-2) -	0.000 0e+0 (0.00e+0) -	1.474 3e-1 (1.55e-2) =
	8	<b>5.004 1e-2</b> <b>(2.05e-2)</b>	3.658 6e-2 (1.70e-2) -	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	4.985 0e-2 (2.03e-2) =
	10	1.889 5e-3 (6.17e-3)	0.000 0e+0 (0.00e+0) =	0.000 0e+0 (0.00e+0) =	0.000 0e+0 (0.00e+0) =	0.000 0e+0 (0.00e+0) =	<b>6.262 2e-3</b> <b>(1.28e-2) =</b>
WFG4	3	<b>5.592 2e-1</b> <b>(6.52e-4)</b>	5.011 3e-1 (4.60e-3) -	4.585 9e-1 (4.60e-3) -	5.410 7e-1 (2.59e-3) -	4.992 9e-1 (1.30e-2) -	5.588 3e-1 (8.39e-4) =
	5	<b>7.844 1e-1</b> <b>(1.76e-3)</b>	6.289 6e-1 (1.12e-2) -	4.765 2e-1 (1.07e-2) -	7.653 9e-1 (2.89e-3) -	5.340 7e-1 (3.20e-2) -	7.826 5e-1 (2.16e-3) -
	8	<b>9.167 3e-1</b> <b>(1.19e-2)</b>	6.362 9e-1 (2.47e-2) -	4.978 7e-1 (1.07e-2) -	9.079 7e-1 (3.59e-3) -	5.433 8e-1 (2.31e-2) -	9.147 0e-1 (1.00e-2) -
	10	<b>9.652 8e-1</b> <b>(1.10e-3)</b>	6.287 9e-1 (1.84e-3) -	5.290 0e-1 (9.13e-3) -	9.621 6e-1 (8.11e-4) -	5.936 2e-1 (3.51e-2) -	9.581 4e-1 (7.71e-3) -
WFG5	3	<b>5.216 6e-1</b> <b>(8.52e-5)</b>	4.812 8e-1 (5.66e-3) -	4.517 4e-1 (8.00e-3) -	5.046 6e-1 (3.90e-3) -	4.800 5e-1 (8.31e-3) -	5.216 3e-1 (1.18e-4) =
	5	<b>7.429 2e-1</b> <b>(5.39e-4)</b>	5.810 9e-1 (1.44e-2) -	4.892 2e-1 (1.23e-2) -	7.247 8e-1 (3.03e-3) -	5.599 0e-1 (1.56e-2) -	7.424 7e-1 (8.36e-4) =
	8	<b>8.631 7e-1</b> <b>(2.43e-4)</b>	5.533 0e-1 (2.78e-2) -	5.445 8e-1 (8.27e-3) -	8.410 3e-1 (5.56e-3) -	5.891 5e-1 (2.68e-2) -	8.627 9e-1 (3.62e-4) -
	10	<b>9.039 5e-1</b> <b>(1.86e-4)</b>	5.758 0e-1 (2.46e-2) -	6.253 3e-1 (8.52e-3) -	8.978 5e-1 (8.76e-4) -	6.397 9e-1 (2.30e-2) -	9.034 6e-1 (3.26e-4) -
WFG6	3	5.098 9e-1 (1.04e-2)	4.526 0e-1 (1.36e-2) -	4.481 2e-1 (1.49e-2) -	4.846 9e-1 (9.70e-3) -	4.491 6e-1 (1.82e-2) -	<b>5.117 1e-1</b> <b>(1.31e-2) =</b>
	5	<b>7.310 8e-1</b> <b>(1.45e-2)</b>	5.478 0e-1 (2.81e-2) -	4.363 7e-1 (1.37e-2) -	6.951 0e-1 (1.82e-2) -	4.260 9e-1 (4.71e-2) -	7.185 1e-1 (1.69e-2) -
	8	8.450 3e-1 (1.55e-2)	5.665 0e-1 (3.85e-2) -	5.027 1e-1 (1.32e-2) -	7.996 5e-1 (2.02e-2) -	3.752 0e-1 (3.62e-2) -	<b>8.492 6e-1</b> <b>(1.56e-2) =</b>
	10	<b>8.843 2e-1</b> <b>(1.04e-2)</b>	6.120 6e-1 (3.36e-2) -	5.759 6e-1 (1.31e-2) -	8.735 1e-1 (1.84e-2) -	4.085 6e-1 (3.04e-2) -	8.767 2e-1 (1.62e-2) =
WFG7	3	<b>5.599 5e-1</b> <b>(4.22e-4)</b>	5.086 9e-1 (3.87e-3) -	4.562 3e-1 (6.39e-3) -	5.469 2e-1 (2.73e-3) -	4.979 6e-1 (1.45e-2) -	5.594 6e-1 (4.26e-4) -
	5	<b>7.881 6e-1</b> <b>(8.79e-4)</b>	6.192 6e-1 (2.02e-2) -	4.061 0e-1 (1.48e-2) -	7.730 3e-1 (3.13e-3) -	4.723 5e-1 (3.30e-2) -	7.859 4e-1 (1.65e-3) -
	8	<b>9.203 1e-1</b> <b>(5.66e-4)</b>	5.960 6e-1 (2.99e-2) -	4.273 5e-1 (1.39e-2) -	8.931 1e-1 (6.83e-3) -	3.914 4e-1 (5.96e-2) -	9.189 6e-1 (8.64e-4) -
	10	<b>9.658 9e-1</b> <b>(4.26e-3)</b>	6.274 9e-1 (2.54e-2) -	4.562 0e-1 (1.11e-2) -	9.581 1e-1 (5.19e-3) -	4.733 6e-1 (2.52e-2) -	9.596 2e-1 (1.43e-2) -
WFG8	3	<b>4.786 7e-1</b> <b>(2.51e-3)</b>	4.297 2e-1 (3.45e-3) -	3.785 9e-1 (8.33e-3) -	4.505 2e-1 (4.87e-3) -	3.989 0e-1 (1.81e-2) -	4.742 7e-1 (3.21e-3) -
	5	<b>6.835 7e-1</b> <b>(7.80e-3)</b>	5.033 5e-1 (1.25e-2) -	3.418 6e-1 (1.37e-2) -	6.321 8e-1 (5.46e-3) -	3.550 9e-1 (5.73e-2) -	6.735 9e-1 (3.61e-3) -
	8	<b>8.110 6e-1</b> <b>(3.72e-2)</b>	5.608 5e-1 (1.43e-2) -	3.741 3e-1 (1.09e-2) -	7.614 4e-1 (3.24e-2) -	2.195 5e-1 (4.90e-2) -	7.939 3e-1 (2.18e-2) =
	10	<b>8.855 0e-1</b> <b>(2.73e-2)</b>	5.912 5e-1 (1.26e-2) -	4.075 0e-1 (1.17e-2) -	8.334 5e-1 (5.79e-2) -	3.541 4e-1 (5.37e-2) -	8.758 3e-1 (1.47e-2) =
WFG9	3	5.187 3e-1 (4.20e-2)	4.933 2e-1 (7.48e-3) -	4.755 9e-1 (1.26e-2) -	5.169 0e-1 (3.83e-2) -	4.867 0e-1 (1.05e-2) -	<b>5.357 9e-1</b> <b>(4.20e-3) =</b>
	5	7.321 4e-1 (3.06e-2)	5.434 2e-1 (4.40e-2) -	3.846 6e-1 (1.92e-2) -	<b>7.416 0e-1</b> <b>(3.06e-3) =</b>	5.749 8e-1 (2.21e-2) -	7.158 4e-1 (3.95e-2) -
	8	8.430 6e-1 (4.63e-2)	5.159 2e-1 (5.19e-2) -	4.066 5e-1 (1.31e-2) -	<b>8.639 5e-1</b> <b>(3.94e-3) =</b>	6.442 7e-1 (2.24e-2) -	8.289 9e-1 (5.46e-2) -
	10	8.850 6e-1 (5.12e-2)	5.942 2e-1 (1.54e-2) -	4.721 1e-1 (8.66e-3) -	<b>9.199 3e-1</b> <b>(2.48e-3) +</b>	6.894 0e-1 (3.50e-2) -	8.842 8e-1 (3.67e-2) -
+/-/=			1/34/1	0/35/1	4/29/3	2/32/2	4/15/17

注: +, -和=分别表示获得的结果与MaOEA-SCAOS相比更好,更差和相似

表 3 6种算法在WFG1~WFG9测试问题上获得的IGD指标值的实验结果

问题	M	MaOEA-SCAOS	MOCeII	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
WFG1	3	2.170 8e-1 (3.79e-2)	6.962 4e-1 (8.67e-2) -	2.032 0e+0 (3.09e-2) -	2.119 5e-1 (2.71e-2) =	2.970 0e-1 (3.23e-2) -	1.983 3e-1 (3.07e-2) =
	5	6.312 3e-1 (3.99e-2)	1.709 9e+0 (9.63e-2) -	2.470 4e+0 (3.72e-2) -	5.095 7e-1 (2.04e-2) +	6.655 5e-1 (4.10e-2) -	5.736 5e-1 (6.24e-2) +
	8	8.934 7e-1 (3.37e-2)	2.176 2e+0 (1.01e-1) -	3.029 0e+0 (3.62e-2) -	9.425 6e-1 (2.18e-2) -	1.296 3e+0 (7.21e-2) -	8.854 6e-1 (2.19e-2) =
	10	9.757 4e-1 (2.01e-2)	2.475 2e+0 (1.36e-1) -	3.344 2e+0 (2.93e-2) -	1.018 3e+0 (1.51e-2) -	1.409 1e+0 (6.05e-2) -	9.581 9e-1 (1.56e-2) +
WFG2	3	1.503 3e-1 (1.37e-3)	2.115 7e-1 (8.97e-3) -	2.475 6e-1 (1.11e-2) -	1.845 5e-1 (7.77e-3) -	2.242 6e-1 (2.41e-2) -	1.500 9e-1 (1.14e-3) =
	5	4.678 2e-1 (2.37e-3)	8.048 0e-1 (5.82e-2) -	1.068 9e+0 (1.01e-1) -	5.334 9e-1 (2.58e-2) -	6.836 0e-1 (4.20e-2) -	4.681 1e-1 (2.29e-3) =
	8	1.024 6e+0 (1.38e-1)	1.466 8e+0 (7.29e-2) -	2.068 9e+0 (1.06e-1) -	1.063 1e+0 (2.96e-2) =	1.571 9e+0 (8.43e-2) -	9.880 8e-1 (1.54e-1) =
	10	1.163 3e+0 (1.23e-1)	1.532 7e+0 (7.02e-2) -	2.400 1e+0 (8.09e-2) -	1.184 3e+0 (3.52e-2) =	1.680 4e+0 (8.61e-2) -	1.183 4e+0 (1.42e-1) =
WFG3	3	1.002 3e-1 (1.01e-2)	1.146 0e-1 (1.12e-2) -	2.798 5e-1 (2.26e-2) -	1.064 2e-1 (1.14e-2) =	6.106 2e-1 (1.69e-1) -	9.752 5e-2 (6.74e-3) =
	5	5.372 8e-1 (6.52e-2)	5.272 2e-1 (8.51e-2) =	1.083 7e+0 (7.96e-2) -	5.426 1e-1 (7.19e-2) =	2.171 1e+0 (3.23e-1) -	5.766 4e-1 (5.62e-2) -
	8	1.527 8e+0 (3.03e-1)	1.237 9e+0 (2.01e-1) +	1.765 5e+0 (1.35e-1) -	1.419 6e+0 (2.55e-1) =	5.612 5e+0 (4.85e-1) -	1.686 3e+0 (2.20e-1) -
	10	1.812 7e+0 (2.34e-1)	1.411 2e+0 (2.16e-1) +	2.035 3e+0 (1.93e-1) -	1.830 1e+0 (2.39e-1) =	7.141 3e+0 (4.67e-1) -	1.968 2e+0 (3.01e-1) =
WFG4	3	2.042 6e-1 (1.43e-4)	2.849 3e-1 (1.03e-2) -	3.383 9e-1 (1.51e-2) -	2.489 6e-1 (8.80e-3) -	3.021 6e-1 (3.60e-2) -	2.044 7e-1 (1.83e-4) -
	5	1.174 8e+0 (1.43e-3)	1.229 6e+0 (2.49e-2) -	1.490 0e+0 (2.24e-2) -	1.232 3e+0 (1.98e-2) -	1.793 6e+0 (1.95e-1) -	1.174 6e+0 (1.83e-3) =
	8	2.970 1e+0 (3.88e-2)	3.369 4e+0 (5.47e-2) -	3.463 7e+0 (2.29e-2) -	3.450 1e+0 (5.66e-2) -	4.525 1e+0 (1.70e-1) -	2.978 0e+0 (7.20e-2) =
	10	4.547 4e+0 (4.68e-3)	4.489 2e+0 (3.76e-2) +	4.505 3e+0 (2.58e-2) +	4.578 4e+0 (3.80e-2) -	6.216 5e+0 (2.23e-1) -	4.547 6e+0 (2.88e-2) -
WFG5	3	2.145 1e-1 (1.09e-4)	2.799 8e-1 (9.11e-3) -	3.122 2e-1 (1.14e-2) -	2.606 0e-1 (1.10e-2) -	2.810 3e-1 (1.42e-2) -	2.145 4e-1 (1.03e-4) =
	5	1.162 9e+0 (5.70e-4)	1.209 8e+0 (1.80e-2) -	1.399 1e+0 (3.04e-2) -	1.211 1e+0 (1.88e-2) -	1.547 2e+0 (7.30e-2) -	1.162 7e+0 (1.37e-3) =
	8	2.941 0e+0 (1.31e-3)	3.423 5e+0 (6.63e-2) -	3.623 6e+0 (4.71e-2) -	3.338 1e+0 (2.98e-2) -	4.286 7e+0 (2.09e-1) -	2.941 3e+0 (1.32e-3) =
	10	4.531 1e+0 (3.32e-3)	4.599 6e+0 (5.91e-2) -	4.764 3e+0 (3.95e-2) -	4.565 3e+0 (3.36e-2) -	5.903 8e+0 (1.79e-1) -	4.530 0e+0 (3.41e-3) =
WFG6	3	2.239 9e-1 (7.93e-3)	3.326 4e-1 (1.70e-2) -	3.455 4e-1 (1.49e-2) -	2.893 7e-1 (1.31e-2) -	3.865 9e-1 (7.01e-2) -	2.230 2e-1 (1.08e-2) =
	5	1.164 7e+0 (2.80e-3)	1.316 0e+0 (3.06e-2) -	1.597 0e+0 (3.88e-2) -	1.246 2e+0 (2.03e-2) -	2.180 0e+0 (2.28e-1) -	1.162 7e+0 (1.80e-3) +
	8	2.947 7e+0 (5.27e-3)	3.506 5e+0 (4.39e-2) -	3.582 9e+0 (2.48e-2) -	3.490 8e+0 (6.50e-2) -	5.515 6e+0 (2.01e-1) -	2.953 0e+0 (4.60e-3) -
	10	4.544 7e+0 (6.62e-3)	4.617 3e+0 (4.44e-2) -	4.555 9e+0 (2.39e-2) -	4.669 0e+0 (4.48e-2) -	7.288 3e+0 (2.25e-1) -	4.543 9e+0 (8.18e-3) =
WFG7	3	2.047 4e-1 (2.06e-4)	2.826 6e-1 (1.04e-2) -	3.363 6e-1 (1.00e-2) -	2.399 1e-1 (8.46e-3) -	3.223 3e-1 (6.99e-2) -	2.047 8e-1 (2.62e-4) =
	5	1.176 3e+0 (8.69e-4)	1.255 4e+0 (3.07e-2) -	1.583 9e+0 (2.76e-2) -	1.230 3e+0 (1.81e-2) -	2.314 8e+0 (1.73e-1) -	1.175 6e+0 (1.12e-3) +
	8	2.967 7e+0 (7.16e-3)	3.499 1e+0 (9.59e-2) -	3.614 3e+0 (3.45e-2) -	3.324 5e+0 (4.54e-2) -	4.919 2e+0 (2.46e-1) -	2.968 8e+0 (6.33e-3) =
	10	4.547 8e+0 (2.02e-2)	4.584 2e+0 (4.81e-2) -	4.630 3e+0 (3.39e-2) -	4.431 9e+0 (3.86e-2) +	6.175 7e+0 (2.19e-1) -	4.644 6e+0 (2.51e-1) =
WFG8	3	2.671 6e-1 (3.98e-3)	3.834 3e-1 (5.79e-3) -	4.457 8e-1 (1.56e-2) -	3.322 4e-1 (1.14e-2) -	4.431 3e-1 (8.88e-2) -	2.751 7e-1 (5.22e-3) -
	5	1.148 0e+0 (1.28e-3)	1.446 1e+0 (2.49e-2) -	1.773 6e+0 (2.73e-2) -	1.282 7e+0 (2.25e-2) -	1.960 5e+0 (2.06e-1) -	1.154 7e+0 (8.40e-3) -

续表

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
	8	3.198 2e+0 (2.62e-1)	3.695 7e+0 (4.49e-2)-	3.769 1e+0 (2.87e-2)-	3.533 6e+0 (6.48e-2)-	5.381 5e+0 (3.58e-1)-	3.236 4e+0 (2.01e-1)-
	10	4.490 8e+0 (2.05e-1)	4.841 5e+0 (3.77e-2)-	4.766 5e+0 (2.84e-2)-	4.703 7e+0 (5.82e-2)-	7.363 7e+0 (2.50e-1)-	4.486 6e+0 (2.20e-1)=
WFG9	3	2.239 3e-1 (4.25e-2)	2.682 4e-1 (1.08e-2)-	2.794 4e-1 (1.74e-2)-	2.363 5e-1 (3.60e-2)-	2.774 2e-1 (1.65e-2)-	2.072 8e-1 (2.21e-3)=
	5	1.126 1e+0 (3.98e-3)	1.276 2e+0 (4.07e-2)-	1.678 9e+0 (4.22e-2)-	1.165 3e+0 (1.73e-2)-	1.495 6e+0 (9.89e-2)-	1.121 7e+0 (5.43e-3)+
	8	2.931 5e+0 (7.68e-3)	3.766 6e+0 (7.03e-2)-	4.105 7e+0 (7.16e-2)-	3.295 2e+0 (3.07e-2)-	4.016 2e+0 (1.54e-1)-	2.930 5e+0 (5.78e-3)=
	10	4.441 3e+0 (3.80e-2)	4.869 4e+0 (5.63e-2)-	5.380 5e+0 (8.20e-2)-	4.487 4e+0 (5.71e-2)-	5.439 9e+0 (1.51e-1)-	4.438 3e+0 (4.17e-2)=
+/-/=			3/32/1	1/35/0	2/27/7	0/36/0	5/8/23

注: +, -和=分别表示获得的结果与MaOEA-SCAOS相比更好, 更差和相似

综合 HV 和 IGD 指标的统计分析结果, 对算法在 WFG 问题上的性能进行对比分析. WFG1 具有复杂的 Pareto 前沿, MaOEA-SCAOS 在 3 和 8 目标问题上的 IGD 指标相较其他对比方法具有较好的竞争力. WFG2 具有凸面不连续且多模态的 Pareto 前沿, MaOEA-SCAOS 的 HV 指标值在 3 和 5 目标上获得最优性能, 具有较好的鲁棒性. WFG3 具有线性退化的 Pareto 前沿, MaOEA-SCAOS 算法的 HV 指标结果在 8 目标取得最优, 并且在 IGD 指标上也具有一定优势. WFG4~WFG9 在决策变量空间中的设计存在不同的困难, 其中 WFG4 具有凹型和多模态的 Pareto 前沿, WFG5 加入欺骗性特征, WFG6 具有不可分离且缩放的 Pareto 前沿, WFG7 具有参数依赖

性的可分离单峰, WFG8 在 WFG7 基础上增加了不可分特性, WFG9 具有以上测试问题特征, 因此 WFG4~WFG9 测试问题在多目标进化算法极具挑战性. 对于 WFG4 和 WFG5 测试问题, MaOEA-SCAOS 在 HV 指标上均获得最优性能, 这表明对于具有凹面的多目标优化问题, 本文方法具有较好的可靠性. 图 4 展示了各个算法获得的非支配解在 WFG4 上 3 目标实例中的分布情况, 其中 MaOEA-SCAOS 算法获得的非支配解集较好地平衡了收敛性和多样性. 对于 WFG6 测试问题, MaOEA-SCAOS 大体上优于其他算法, 仅在 5 目标问题上性能略差于 NSGA-III 算法, 由图 5 可以看到, 各个算法获得的非支配解在 WFG6 上 8 目标实例中的分布情况, 可以看出

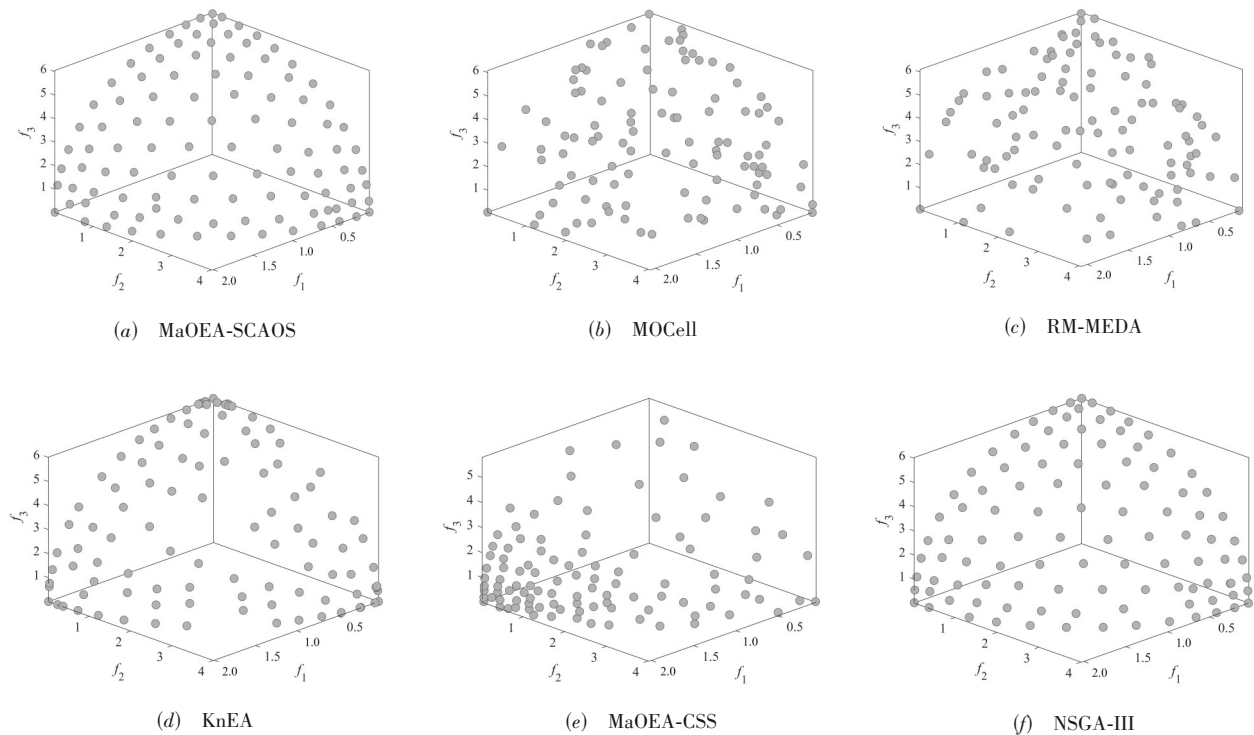


图4 WFG4问题3目标上获得的非支配解

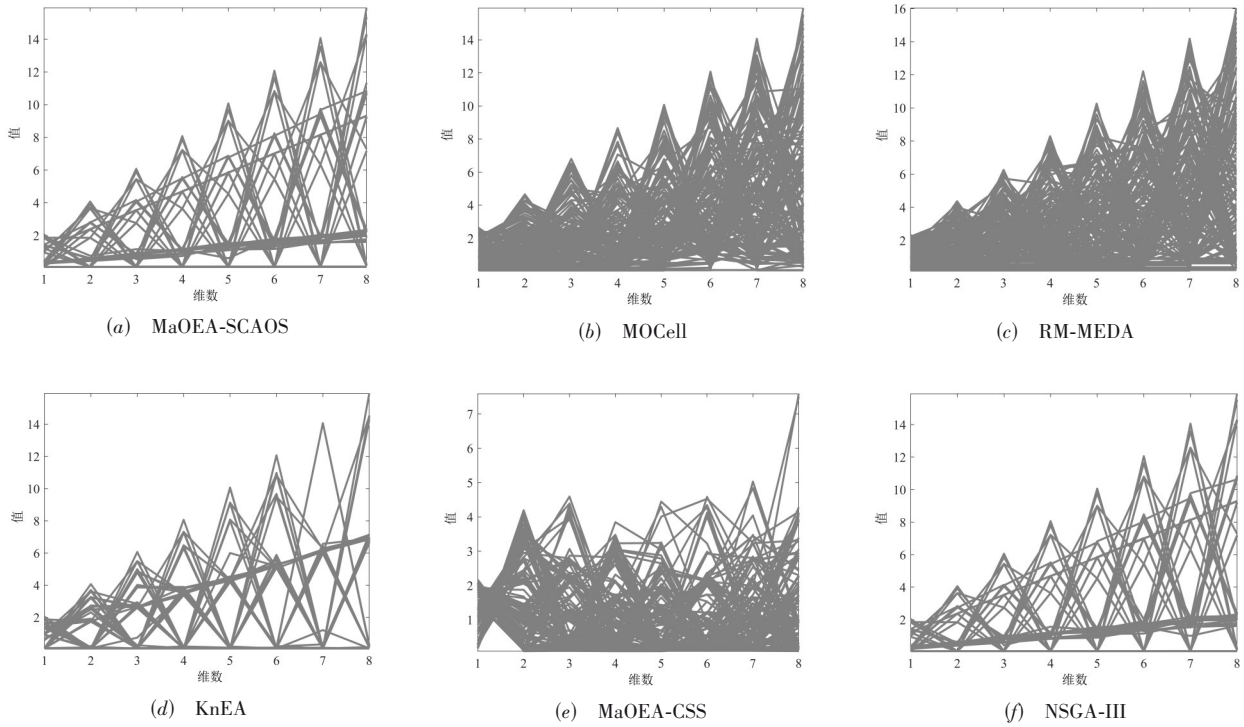


图5 WFG6问题8目标上获得的非支配解

MaOEA-SCAOS 获得的非支配解优于其他对比算法. 对于 WFG7 和 WFG8 测试问题, MaOEA-SCAOS 在处理 3, 5, 8 和 10 目标时性能表现突出, 说明在 WFG7 和 WFG8 测试问题上处理不同目标数目的普适性较好. 此外, 在 WFG9 问题的测试中, 本文算法也表现出一定的竞争力. 综上所述, 可以看出 MaOEA-SCAOS 在 WFG 测试问题集上性能显著且在处理不同目标优化问题时普适性较高.

#### 4.6 算法在 DTLZ 问题上对比分析

表 4 和表 5 分别给出了 6 个算法在 DTLZ1~DTLZ7 测试问题上进行 20 次独立计算 HV 和 IGD 指标的实验结果, 其中, 表 4 汇总了对比算法所获得 HV 值的均值和标准差 (括号内为标准差), 表 5 汇总了对比算法所获得 IGD 值的均值和标准差 (括号内为标准差), 其中用黑色粗体突出最好的结果.

表 4 6 种算法在 DTLZ1~DTLZ7 测试问题上获得的 HV 指标值的实验结果

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
DTLZ1	3	<b>8.442 1e-1</b> (2.45e-4)	8.241 4e-1 (3.87e-3)	0.000 0e+0 (0.00e+0)	7.176 4e-1 (8.20e-2)	8.078 8e-1 (1.10e-2)	8.436 0e-1 (1.01e-3)
	5	<b>9.745 0e-1</b> (3.96e-4)	2.319 1e-1 (3.24e-1)	0.000 0e+0 (0.00e+0)	5.602 7e-1 (1.61e-1)	9.073 1e-1 (1.02e-2)	9.744 6e-1 (3.50e-4)
	8	9.626 6e-1 (1.08e-1)	0.000 0e+0 (0.00e+0)	0.000 0e+0 (0.00e+0)	6.661 1e-1 (3.18e-1)	9.455 0e-1 (7.01e-3)	<b>9.972 5e-1</b> (1.46e-3)
	10	<b>9.901 7e-1</b> (2.91e-2)	0.000 0e+0 (0.00e+0)	0.000 0e+0 (0.00e+0)	8.793 8e-4 (3.93e-3)	9.803 1e-1 (2.66e-3)	9.807 0e-1 (5.89e-2)
DTLZ2	3	<b>5.630 0e-1</b> (1.16e-5)	5.251 6e-1 (3.33e-3)	5.213 2e-1 (4.32e-3)	5.428 2e-1 (4.55e-3)	5.538 6e-1 (3.14e-3)	5.629 9e-1 (1.98e-5)
	5	<b>7.947 3e-1</b> (5.06e-4)	5.278 2e-1 (3.84e-2)	5.517 1e-2 (2.08e-2)	7.679 4e-1 (5.15e-3)	7.557 5e-1 (8.62e-3)	7.946 1e-1 (3.92e-4)
	8	<b>9.145 0e-1</b> (2.31e-2)	2.298 2e-5 (1.03e-4)	5.360 4e-3 (1.09e-2)	8.827 2e-1 (4.24e-3)	8.810 1e-1 (7.35e-3)	9.125 9e-1 (3.01e-2)
DTLZ2	10	<b>9.625 1e-1</b> (1.81e-2)	1.667 5e-3 (4.99e-3)	1.339 2e-4 (3.29e-4)	9.587 1e-1 (1.60e-3)	9.349 5e-1 (4.71e-3)	9.621 1e-1 (1.94e-2)
DTLZ3	3	<b>5.527 4e-1</b> (6.27e-3)	5.131 6e-1 (1.33e-2)	0.000 0e+0 (0.00e+0)	4.790 1e-1 (5.20e-2)	5.496 0e-1 (4.09e-3)	5.401 1e-1 (1.18e-2)

续表

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
	5	5.980 4e-1 (2.92e-1)	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	3.993 6e-1 (1.89e-1) -	<b>7.464 4e-1</b> (1.22e-2) =	5.835 5e-1 (3.08e-1) =
	8	<b>8.908 0e-1</b> (7.45e-2)	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	8.555 0e-1 (1.92e-2) -	8.515 2e-1 (2.08e-1) -
	10	8.364 6e-1 (3.00e-1)	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	0.000 0e+0 (0.00e+0) -	9.232 0e-1 (7.64e-3) +	<b>9.298 0e-1</b> (5.81e-2) =
DTLZ4	3	<b>5.628 7e-1</b> (2.01e-4)	5.295 6e-1 (3.19e-3) -	5.167 0e-1 (7.45e-3) -	5.228 7e-1 (1.02e-1) -	5.585 2e-1 (1.69e-3) -	4.950 7e-1 (1.31e-1) =
	5	<b>7.946 0e-1</b> (5.00e-4)	5.993 8e-1 (3.19e-2) -	2.438 9e-1 (7.33e-2) -	7.744 1e-1 (5.25e-3) -	7.797 2e-1 (3.98e-3) -	7.567 6e-1 (6.02e-2) -
	8	9.071 9e-1 (3.34e-2)	0.000 0e+0 (0.00e+0) -	7.206 2e-3 (1.31e-2) -	9.019 9e-1 (4.47e-3) -	9.070 8e-1 (3.48e-3) -	<b>9.117 5e-1</b> (3.02e-2) =
	10	<b>9.696 2e-1</b> (1.40e-4)	0.000 0e+0 (0.00e+0) -	1.727 4e-3 (3.86e-3) -	9.555 7e-1 (5.48e-3) -	9.543 4e-1 (2.40e-3) -	9.676 9e-1 (8.56e-3) =
DTLZ5	3	1.946 2e-1 (8.08e-4)	1.983 8e-1 (2.55e-4) +	<b>1.991 7e-1</b> (1.72e-4) +	1.940 7e-1 (2.10e-3) =	1.853 6e-1 (3.13e-3) -	1.948 6e-1 (8.93e-4) =
	5	6.439 3e-2 (4.97e-2)	<b>1.017 7e-1</b> (8.56e-3) +	2.007 9e-2 (1.59e-2) =	5.697 8e-2 (3.16e-2) =	9.805 2e-2 (1.92e-2) +	3.765 8e-2 (4.51e-2) =
	8	<b>9.427 7e-2</b> (2.59e-3)	5.347 5e-2 (3.13e-2) -	7.120 6e-4 (2.16e-3) -	6.680 9e-2 (1.92e-2) -	5.538 6e-2 (2.73e-2) -	9.418 0e-2 (2.39e-3) =
	10	<b>9.133 6e-2</b> (1.92e-3)	3.539 7e-2 (2.62e-2) -	2.841 8e-10 (1.27e-9) -	5.344 1e-2 (1.99e-2) -	4.673 7e-2 (3.13e-2) -	9.105 2e-2 (1.82e-3) =
DTLZ6	3	1.920 1e-1 (1.24e-3)	<b>1.998 4e-1</b> (1.05e-4) +	4.921 1e-3 (2.20e-2) -	1.902 9e-1 (8.85e-3) =	1.855 2e-1 (1.58e-3) -	1.918 3e-1 (9.85e-4) =
	5	6.864 7e-2 (3.84e-2)	0.000 0e+0 (0.00e+0) -	4.773 4e-2 (4.96e-2) =	6.607 7e-2 (3.92e-2) =	<b>8.530 1e-2</b> (2.83e-2) =	6.287 1e-2 (4.21e-2) =
	8	8.183 2e-2 (2.80e-2)	0.000 0e+0 (0.00e+0) -	<b>8.766 4e-2</b> (2.07e-2) +	3.929 9e-2 (4.16e-2) -	4.301 1e-2 (4.10e-2) -	8.190 0e-2 (2.80e-2) =
	10	7.721 1e-2 (3.33e-2)	0.000 0e+0 (0.00e+0) -	<b>9.211 3e-2</b> (8.47e-4) +	6.357 8e-11 (1.55e-10) -	2.527 4e-2 (2.91e-2) -	8.653 3e-2 (2.04e-2) =
DTLZ7	3	2.697 4e-1 (1.55e-3)	2.578 8e-1 (1.10e-2) -	2.183 9e-1 (1.36e-2) -	<b>2.760 2e-1</b> (8.07e-3) +	2.521 8e-1 (1.13e-2) -	2.712 7e-1 (1.44e-3) +
	5	2.260 4e-1 (8.01e-3)	1.493 5e-1 (9.49e-3) -	2.898 3e-5 (4.21e-5) -	<b>2.514 7e-1</b> (7.86e-3) +	1.340 8e-1 (1.51e-2) -	2.216 4e-1 (1.01e-2) =
	8	1.950 7e-1 (4.04e-3)	5.276 9e-4 (8.36e-4) -	3.964 5e-7 (1.38e-6) -	1.034 8e-1 (9.82e-3) -	1.134 5e-2 (1.93e-3) -	<b>2.038 6e-1</b> (2.27e-3) +
	10	1.808 2e-1 (6.46e-3)	9.885 1e-6 (2.38e-5) -	1.246 0e-8 (4.35e-8) -	9.423 6e-2 (2.58e-2) -	2.379 9e-4 (1.68e-4) -	<b>1.830 2e-1</b> (8.59e-3) =
+/-/=			3/25/0	3/24/1	2/22/4	1/25/2	2/5/21

注: +, -和=分别表示获得的结果与MaOEA-SCAOS相比更好,更差和相似

表5 6种算法在DTLZ1~DTLZ7测试问题上获得的IGD指标值的实验结果

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
DTLZ1	3	<b>1.898 8e-2</b> (1.72e-5)	2.558 3e-2 (8.91e-4) -	1.616 9e+1 (2.13e+0) -	6.464 5e-2 (4.02e-2) -	2.287 3e-2 (9.44e-4) -	1.904 4e-2 (1.19e-4) -
	5	<b>6.352 9e-2</b> (2.56e-4)	5.903 2e-1 (6.19e-1) -	2.291 8e+1 (2.68e+0) -	2.222 5e-1 (7.45e-2) -	6.881 5e-2 (2.31e-3) -	6.362 0e-2 (3.00e-4) =
	8	1.248 6e-1 (6.29e-2)	1.358 3e+1 (4.95e+0) -	2.288 8e+1 (2.43e+0) -	3.005 5e-1 (1.05e-1) -	1.177 1e-1 (2.30e-3) +	<b>9.761 0e-2</b> (2.17e-3) =
	10	1.216 8e-1 (3.33e-2)	2.019 9e+1 (8.38e+0) -	2.258 6e+1 (2.18e+0) -	4.941 6e+0 (6.16e+0) -	<b>1.153 0e-1</b> (1.16e-3) +	1.277 9e-1 (5.52e-2) =
DTLZ2	3	<b>5.030 3e-2</b> (1.80e-6)	6.906 6e-2 (2.08e-3) -	6.775 9e-2 (2.26e-3) -	6.786 7e-2 (4.56e-3) -	5.235 5e-2 (5.80e-4) -	5.030 5e-2 (5.07e-6) -
	5	1.949 0e-1 (3.23e-5)	2.741 3e-1 (1.52e-2) -	6.591 8e-1 (4.02e-2) -	2.163 1e-1 (5.70e-3) -	<b>1.883 0e-1</b> (1.79e-3) +	1.949 0e-1 (1.53e-5) =
	8	<b>3.348 8e-1</b> (4.82e-2)	1.907 0e+0 (3.08e-1) -	1.251 6e+0 (1.14e-1) -	3.828 3e-1 (4.75e-3) -	3.480 3e-1 (2.17e-3) -	3.421 0e-1 (6.93e-2) =

续表

问题	M	MaOEA-SCAOS	MOCcell	RM-MEDA	KnEA	MaOEA-CSS	NSGA-III
	10	4.384 0e-1 (4.34e-2)	1.408 8e+0 (1.56e-1)-	1.463 7e+0 (8.52e-2)-	4.313 6e-1 (2.42e-3)+	<b>3.949 0e-1</b> (1.53e-3)+	4.384 5e-1 (4.23e-2)=
DTLZ3	3	<b>5.369 3e-2</b> (4.37e-3)	7.404 3e-2 (6.00e-3)-	1.804 6e+2 (9.51e+0)-	1.241 0e-1 (6.88e-2)-	5.665 6e-2 (1.22e-3)-	5.689 8e-2 (9.79e-3)-
	5	3.821 1e-1 (3.68e-1)	1.684 8e+1 (1.24e+1)-	1.838 0e+2 (1.48e+1)-	5.693 2e-1 (2.57e-1)-	<b>1.973 8e-1</b> (3.52e-3)+	5.549 8e-1 (8.36e-1)=
DTLZ3	8	3.614 9e-1 (8.87e-2)	8.341 6e+2 (2.95e+2)-	2.494 8e+2 (3.44e+1)-	6.097 4e+1 (2.89e+1)-	<b>3.575 7e-1</b> (4.43e-3)+	4.571 9e-1 (4.76e-1)=
	10	6.049 4e-1 (4.70e-1)	8.070 4e+2 (2.43e+2)-	2.914 8e+2 (4.04e+1)-	2.898 8e+2 (9.54e+1)-	<b>4.009 9e-1</b> (3.07e-3)+	4.781 8e-1 (7.83e-2)=
DTLZ4	3	<b>5.032 0e-2</b> (3.16e-5)	6.867 5e-2 (2.50e-3)-	7.734 7e-2 (5.29e-3)-	1.091 8e-1 (1.97e-1)-	5.312 1e-2 (6.82e-4)-	1.933 0e-1 (2.67e-1)-
	5	1.949 4e-1 (5.05e-5)	2.551 9e-1 (1.11e-2)-	5.180 0e-1 (3.40e-2)-	2.159 5e-1 (6.93e-3)-	<b>1.926 9e-1</b> (1.75e-3)+	2.666 9e-1 (1.13e-1)=
	8	3.550 4e-1 (7.29e-2)	1.986 8e+0 (2.19e-1)-	1.023 4e+0 (5.03e-2)-	3.752 6e-1 (4.26e-3)-	3.524 1e-1 (1.51e-3)+	<b>3.441 9e-1</b> (7.18e-2)=
	10	4.215 5e-1 (4.84e-4)	1.952 7e+0 (2.43e-1)-	1.183 1e+0 (4.93e-2)-	4.449 0e-1 (2.04e-2)-	<b>4.009 5e-1</b> (1.08e-3)+	4.269 3e-1 (2.86e-2)-
DTLZ5	3	1.247 0e-2 (1.63e-3)	6.037 4e-3 (3.21e-4)+	<b>5.005 8e-3</b> (2.59e-4)+	9.000 8e-3 (1.74e-3)+	2.128 4e-2 (2.06e-3)-	1.179 4e-2 (1.61e-3)=
	5	3.475 4e-1 (2.29e-1)	1.114 2e-1 (3.53e-2)+	2.875 8e-1 (7.26e-2)=	2.901 1e-1 (1.09e-1)=	<b>4.876 8e-2</b> (7.45e-3)+	3.580 7e-1 (2.25e-1)=
	8	3.043 8e-1 (9.17e-2)	3.275 9e-1 (1.94e-1)=	5.066 9e-1 (1.12e-1)-	2.764 0e-1 (5.07e-2)=	<b>7.282 7e-2</b> (9.40e-3)+	2.481 4e-1 (9.17e-2)=
	10	3.537 4e-1 (7.28e-2)	3.692 9e-1 (1.55e-1)=	6.323 1e-1 (8.11e-2)-	3.147 2e-1 (6.40e-2)=	<b>8.768 5e-2</b> (2.53e-2)+	3.361 6e-1 (6.68e-2)=
DTLZ6	3	1.747 9e-2 (2.32e-3)	<b>5.203 2e-3</b> (1.86e-4)+	1.624 4e+0 (5.69e-1)-	1.492 7e-2 (1.03e-2)=	2.735 0e-2 (1.90e-3)-	1.718 6e-2 (1.62e-3)=
	5	3.584 4e-1 (1.17e-1)	5.779 2e+0 (1.02e+0)-	8.456 0e-1 (4.97e-1)-	3.727 5e-1 (1.70e-1)=	<b>7.821 4e-2</b> (2.68e-2)+	3.801 4e-1 (2.62e-1)=
	8	4.492 3e-1 (1.89e-1)	7.077 2e+0 (7.94e-1)-	4.219 7e-1 (2.26e-1)=	5.159 2e-1 (9.26e-2)=	<b>1.602 4e-1</b> (6.88e-2)+	5.202 7e-1 (3.35e-1)=
	10	5.141 3e-1 (3.32e-1)	7.735 9e+0 (6.48e-1)-	2.918 5e-1 (9.12e-2)+	1.141 5e+0 (5.23e-1)-	<b>1.446 5e-1</b> (3.90e-2)+	3.718 4e-1 (1.37e-1)=
DTLZ7	3	7.374 7e-2 (3.23e-3)	1.316 5e-1 (1.12e-1)-	1.550 3e-1 (3.16e-2)-	8.140 6e-2 (6.36e-2)-	9.806 6e-2 (1.18e-2)-	<b>7.021 2e-2</b> (2.81e-3)+
	5	3.690 6e-1 (4.61e-2)	4.223 0e-1 (1.47e-2)-	1.724 0e+0 (5.51e-1)-	<b>2.956 4e-1</b> (1.35e-2)+	3.953 6e-1 (1.98e-2)-	3.642 0e-1 (5.04e-2)=
	8	8.402 7e-1 (5.15e-2)	1.352 6e+0 (5.77e-2)-	1.842 5e+0 (2.58e-1)-	<b>6.689 2e-1</b> (2.46e-2)+	1.212 8e+0 (1.40e-1)-	7.792 1e-1 (2.88e-2)+
	10	1.033 3e+0 (9.12e-2)	2.230 2e+0 (3.15e-1)-	2.195 8e+0 (2.55e-1)-	<b>8.508 1e-1</b> (7.46e-3)+	2.309 0e+0 (4.14e-1)-	9.696 0e-1 (6.91e-2)+
+/-/=			3/23/2	2/24/2	5/17/6	17/11/0	3/5/20

注: +, -和=分别表示获得的结果与 MaOEA-SCAOS 相比更好, 更差和相似

综合 HV 和 IGD 指标的统计数据对算法在 DTLZ 问题上的实验结果进行分析. 由表 4 可以看出, 在具有规则形状的 Pareto 前沿的 DTLZ1~DTLZ4 测试问题上, MaOEA-SCAOS 在 HV 性能指标均获得较优的性能, 这表明算法对具有规则的 Pareto 前沿面的优化问题具有较好的可靠性. 图 6 展示了各个算法获得的非支配解在 DTLZ4 上 10 目标实例中的分布情况, MaOEA-SCAOS 算法的搜索能力并没有受到目标个数增加的影响, 依然表现出较强的性能. 值得注意的是, 在具有凹且退化特征

的 DTLZ5 和 DTLZ6 上, MaOEA-SCAOS 能够获得与 NSGA-III 算法相似的性能, 并且其在 DTLZ5 测试问题 8 目标和 10 目标上的指标优于 NSGA-III. 这说明 MaOEA-SCAOS 相较 NSGA-III 更适合处理具有退化 Pareto 前沿的高维问题. 对于具有混合特征 Pareto 前沿的 DTLZ7 测试问题, 算法 MaOEA-SCAOS 较难处理的主要原因在于没有采用自适应特征点的环境选择策略. 综上所述, MaOEA-SCAOS 在 DTLZ 测试问题集中具有规则形状 PF 的目标问题上具有良好的可靠性, 能够保持种群的收敛性

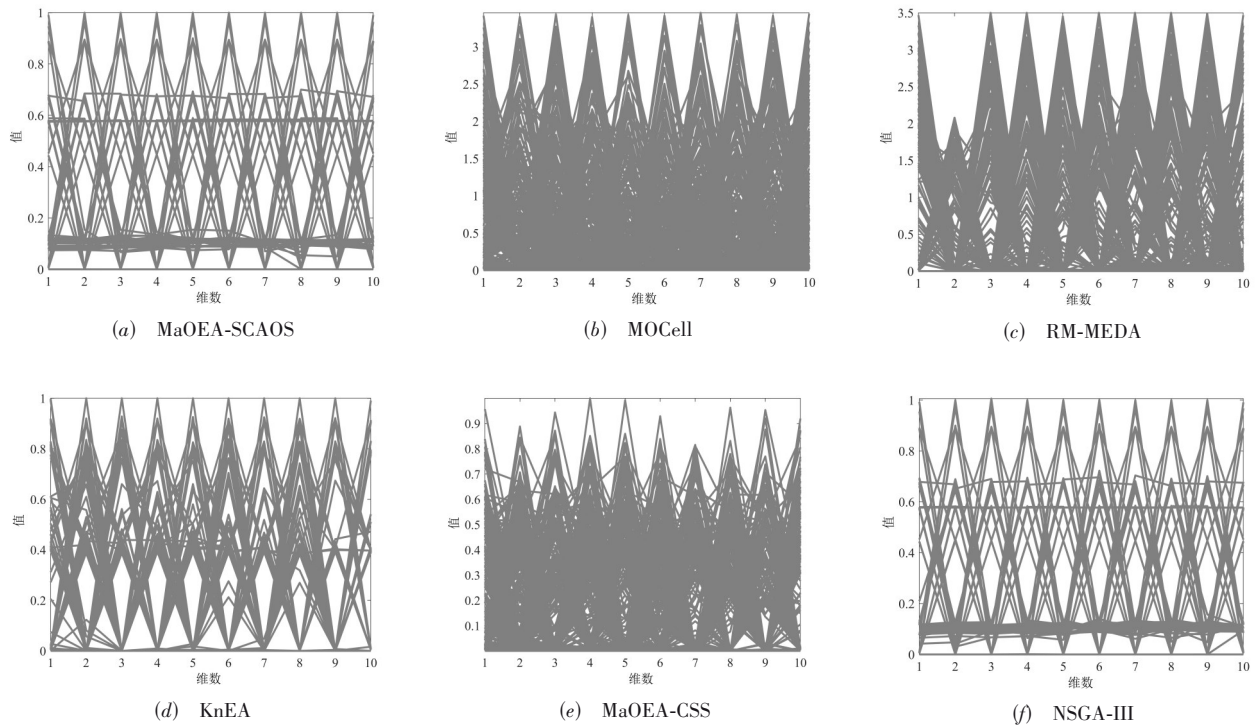


图6 DTLZ4问题10目标上获得的非支配解

和多样性;在不规则的PF上不同目标维数中仍然能够保持算法的鲁棒性,能够较好地处理不同形状Pareto前沿的问题.

## 5 总结

针对在多目标优化问题中高维多目标优化算法的搜索效率及收敛效率降低,无法产生高质量的子代个体去引导种群搜索的问题,本文提出一种基于SOM聚类和自适应算子选择的方法作用于高维多目标进化算法.该方法首先利用SOM网络特性,训练提取种群个体结构信息,获得相似个体集合,构建邻域交配池;然后通过交配池内支配信息进行自适应算子选择,重组产生优质子代引导种群搜索;最后,引入NSGA-III算法中环境策略,更新种群,以提高种群的多样性和收敛性.与现有的高维多目标优化算法进行实验对比,结果表明,本文所提出的个体重组方式的有效性优于同样利用个体结构信息的其他多目标优化算法.此外,与现有的其他算法相比,本文算法在所有基准测试函数中均具有较强的竞争力.

## 参考文献

[1] CHEUNG Y M, GU F Q, LIU H L. Objective extraction for many-objective optimization problems: Algorithm and test problems[J]. IEEE Transactions on Evolutionary Com-

putation, 2016, 20(5): 755-772.

- [2] 巩敦卫, 王更星, 孙晓燕. 高维多目标优化问题融入决策者偏好的集合进化优化方法[J]. 电子学报, 2014, 42(5): 933-939.  
GONG D W, WANG G X, SUN X Y. Set-based evolutionary optimization algorithms integrating decision-maker's preferences for many-objective optimization problems[J]. Acta Electronica Sinica, 2014, 42(5): 933-939. (in Chinese)
- [3] TIAN Y, CHENG R, ZHANG X Y, et al. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(2): 331-345.
- [4] ELARBI M, BECHIKH S, GUPTA A, et al. A new decomposition-based NSGA-II for many-objective optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 48(7): 1191-1210.
- [5] ZHANG H, ZHOU A M, SONG S M, et al. A self-organizing multiobjective evolutionary algorithm[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(5): 792-806.
- [6] 郭广颂, 陈良骥, 文振华, 等. 求解高维混合指标优化问题的交互式进化计算[J]. 电子学报, 2020, 48(7): 1361-1368.  
GUO G S, CHEN L J, WEN Z H, et al. Solving multidimensional optimization problems with hybrid indices by interactive evolutionary computation[J]. Acta Electronica Si-

- nica, 2020, 48(7): 1361-1368. (in Chinese)
- [7] GÓMEZ R H, COELLO C A. Improved metaheuristic based on the R2 indicator for many-objective optimization [C]//GECCO' 15: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. Madrid Spain: ACM, 2015: 679-686.
- [8] AGUIRRE H, TANAKA K. Adaptive  $\epsilon$ -Ranking on many-objective problems[J]. *Evolutionary Intelligence*, 2009, 2(4): 183-206.
- [9] ZOU X F, CHEN Y, LIU M Z, et al. A new evolutionary algorithm for solving many-objective optimization problems[J]. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, 2008, 38(5): 1402-1412.
- [10] 毕晓君, 张永建, 陈春雨. 基于模糊支配的高维多目标进化算法 MFEA[J]. *电子学报*, 2014, 42(8): 1653-1659.  
BI X J, ZHANG Y J, CHEN C Y. A many-objective evolutionary algorithm based on fuzzy dominance: MFEA[J]. *Acta Electronica Sinica*, 2014, 42(8): 1653-1659. (in Chinese)
- [11] 孙文静, 李军华, 黎明. 基于自适应支配准则的高维多目标进化算法[J]. *电子学报*, 2020, 48(8): 1596-1604.  
SUN W J, LI J H, LI M. Adaptive dominance criterion based evolutionary algorithm for many-objective optimization[J]. *Acta Electronica Sinica*, 2020, 48(8): 1596-1604. (in Chinese)
- [12] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [13] DEB K, MOHAN M, MISHRA S. Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions[J]. *Evolutionary Computation*, 2005, 13(4): 501-525.
- [14] ZHANG X Y, TIAN Y, JIN Y C. A knee point-driven evolutionary algorithm for many-objective optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2015, 19(6): 761-776.
- [15] CHENG R, JIN Y C, OLHOFER M, et al. A reference vector guided evolutionary algorithm for many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(5): 773-791.
- [16] YUAN Y, XU H, WANG B, et al. Balancing convergence and diversity in decomposition-based many-objective optimizers[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(2): 180-198.
- [17] NEBRO A J, DURILLO J J, LUNA F, et al. MOCcell: A cellular genetic algorithm for multiobjective optimization [J]. *International Journal of Intelligent Systems*, 2009, 24(7): 726-746.
- [18] ZHANG Q F, ZHOU A M, JIN Y C. RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 41-63.
- [19] KAMBHATLA N, LEEN T K. Dimension reduction by local principal component analysis[J]. *Neural Computation*, 1997, 9(7): 1493-1516.
- [20] ISHIBUCHI H, NARUKAWA K, TSUKAMOTO N, et al. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization[J]. *European Journal of Operational Research*, 2008, 188(1): 57-75.
- [21] HE G X, GAO J Q, HU L K. An improved immune genetic algorithm for multiobjective optimization[C]//International Conference in Swarm Intelligence. Berlin, Heidelberg: Springer, 2010: 643-650.
- [22] XU R, WUNSCH D. *Clustering*[M]. New Jersey: Wiley-IEEE Press, 2008.
- [23] KOHONEN T. The self-organizing map[J]. *Neurocomputing*, 1998, 21(1/2/3): 1-6.
- [24] PURBASARI I Y, PUSPANINGRUM E Y, PUTRA A S. Using self-organizing map(SOM) for clustering and visualization of new students based on grades[J]. *Journal of Physics: Conference Series*, 2020, 1569(2): 022037.
- [25] SRIDEVI M, MALA C. Self-organizing neural networks for image segmentation based on multiphase active contour[J]. *Neural Computing and Applications*, 2019, 31(2): 865-876.
- [26] HU R J, RATNER K, RATNER E, et al. ELM-SOM+: A continuous mapping for visualization[J]. *Neurocomputing*, 2019, 365: 147-156.
- [27] HAKIMI-ASIABAR M, GHODSYPOUR S H, KERACHIAN R. Deriving operating policies for multi-objective reservoir systems: Application of self-learning genetic algorithm[J]. *Applied Soft Computing*, 2010, 10(4): 1151-1163.
- [28] ZHAN W, LIU H, DAI G. Low earth orbit regional satellite constellation design via self organization feature maps [J]. *International Journal of Advancements in Computing Technology*, 2012, 4(13): 250-260.
- [29] NOROUZI K, RAKHSHANDEHROO G. A self organizing map based hybrid multi-objective optimization of wa-

ter distribution networks[J]. IJST, Transactions of Civil and Environmental Engineering, 2011, 35(C1): 105-119.

- [30] GU F Q, CHEUNG Y M. Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(2): 211-225.
- [31] DEB K, AGRAWAL R. Simulated binary crossover for continuous search space[J]. Complex Systems, 1994, 9(3): 115-148.
- [32] DEB K. Multi-objective Optimization Using Evolutionary Algorithms[M]. New York: John Wiley & Sons, 2001
- [33] PRICE K, STORN R M. Differential Evolution: A Practical Approach to Global Optimization(Natural Computing Series)[M]. Berlin, Heidelberg: Springer-Verlag, 2005.
- [34] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- [35] HE Z N, YEN G G. Many-objective evolutionary algorithms based on coordinated selection strategy[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(2): 220-233.
- [36] TIAN Y, CHENG R, ZHANG X Y, et al. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization educational forum[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.
- [37] DEB K, THIELE L, LAUMANN S M, et al. Scalable multi-objective optimization test problems[C]//Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600). Honolulu, HI, USA: IEEE, 2002: 825-830.



何超男, 1992年生, 浙江诸暨人. 南京航空航天大学博士研究生. 主要研究方向为进化算法与图像处理、模式识别等.

E-mail: hechao92918@163.com



陈昊男, 1982年生, 山东平度人. 南昌航空大学信息工程学院教授. 主要研究方向为进化算法理论与应用、图像处理与模式识别等.

E-mail: chenhaoshl@163.com

#### 作者简介



钟沛龙男, 1997年生, 江西赣州人. 南昌航空大学硕士研究生, 主要研究方向为进化算法.

E-mail: zplong33@163.com



黎明(通讯作者)男, 1965年生, 江西樟树人. 南昌航空大学信息工程学院教授, 南京航空航天大学博士生导师. 主要研究方向为图像处理与模式识别、智能计算等.

E-mail: liming@nchu.edu.cn