

# 融合张角拥挤控制策略的高维多目标优化

陈振兴<sup>1</sup> 严宣辉<sup>1</sup> 吴坤安<sup>1</sup> 白猛<sup>1</sup>

**摘要** 对于高维多目标优化问题,随着目标维数的增加,种群中非支配解的比例剧增,严重降低了种群的进化压力.为了对数量众多的非支配解进行有效的拥挤控制并提升种群的多样性,本文在提出张角概念的基础上设计了一种新的拥挤控制策略(Congestion control strategy based on open angle, CCSOA),它的时间复杂度并不会随着目标维数的增加而增大.与目前优秀的进化多目标优化(Evolutionary multiobjective optimization, EMO)算法 IBEA (Indicator-based evolutionary algorithm)、NSGAIII (Nondominated sorting genetic algorithm III) 和 GrEA (Grid-based evolutionary algorithm) 的比较结果表明,融合了 CCSOA 的高维多目标优化算法在收敛效果和解集分布的均匀性两个方面均有较大的优势.

**关键词** 高维多目标优化, 进化算法, 拥挤控制, 张角

**引用格式** 陈振兴, 严宣辉, 吴坤安, 白猛. 融合张角拥挤控制策略的高维多目标优化. 自动化学报, 2015, 41(6): 1145–1158

**DOI** 10.16383/j.aas.2015.c140555

## Many-objective Optimization Integrating Open Angle Based Congestion Control Strategy

CHEN Zhen-Xing<sup>1</sup> YAN Xuan-Hui<sup>1</sup> WU Kun-An<sup>1</sup> BAI Meng<sup>1</sup>

**Abstract** For the many-objective optimization problem, the proportion of non-dominated individuals increases dramatically with the increase of target dimension, which may seriously reduce the population evolutionary pressure. In order to efficiently control the congestion among the very large numbers of non-dominated solutions and improve its diversity, this paper firstly defines the concept of open angle, based on which a novel congestion control strategy is proposed, called CCSOA (Congestion control strategy based on open angle) here. It is time complexity will not increase with the increasing target dimension. Compared with some well-known algorithms such as IBEA (Indicator-based evolutionary algorithm), NSGAIII (Nondominated sorting genetic algorithm III) and GrEA (Grid-based evolutionary algorithm), the many-objective optimization algorithm integrated with CCSOA has better convergence and remains better diversity and uniformity.

**Key words** Many-objective optimization, evolutionary algorithm, congestion control, open angle

**Citation** Chen Zhen-Xing, Yan Xuan-Hui, Wu Kun-An, Bai Meng. Many-objective optimization integrating open angle based congestion control strategy. *Acta Automatica Sinica*, 2015, 41(6): 1145–1158

多目标优化问题<sup>[1]</sup>是指需要同时优化多个目标的最优化问题,它在工程实践和科学研究中有重要的应用价值.多目标优化问题起源于许多实际复杂系统的设计、建模和规划问题,这些系统所在的领域包括工业制造、城市运输、资本预算、能量分配和城市布局等,现实世界中的优化问题通常是多属性的,一般是对多个目标的同时优化,如一个国家的最优良性发展,涉及到经济的快速增长、社会秩序的稳定、环境的保护和改善等多个方面.在这里经济的快速增长和社会秩序稳定这两个优化目标是相辅相

成的、互相促进的,通常称为一致的,但在多数情况下,被同时优化的多个目标之间是相互冲突的,如企业的生产活动中,产品质量和生产成本是两个相互冲突的目标.现实生活中几乎每个重要的决策问题都蕴含多目标优化问题.

进化多目标优化 (Evolutionary multi-objective optimization, EMO) 作为一类启发式搜索算法被成功应用于多目标优化领域<sup>[1-2]</sup>,是当前计算智能研究热点之一.针对多目标优化问题,进化算法的优化效果已经被普遍接受,针对大部分问题,它可以有效地寻找到折中解集(通常用 Pareto 最优解集来衡量).进化多目标优化算法的核心问题<sup>[3]</sup>主要包括以下几点: 1) 为了尽量避免种群早熟并加快算法收敛速度,必须在种群规模受限的情况下尽可能保持种群的多样性; 2) 算法只能提供有限个数的最优解供决策者参考,为此算法需要尽可能地提供分布均匀的最优解集作为代表; 3) 为了保证种群朝

收稿日期 2014-07-30 录用日期 2015-02-02  
Manuscript received July 30, 2014; accepted February 2, 2015  
国家自然科学基金(61175123)资助  
Supported by National Natural Science Foundation of China (61175123)  
本文责任编辑 王红卫  
Recommended by Associate Editor WANG Hong-Wei  
1. 福建师范大学数学与计算机科学学院 福州 350007  
1. College of Mathematic and Computer Science, Fujian Normal University, Fuzhou 350007

着最优 Pareto 前沿面的方向搜索, 如何对种群进行交叉、变异以及选择使种群收敛到问题的最优前沿面. 以上问题对于任何一个多目标优化算法都是一个挑战. 如何有效求解高维多目标优化问题(一般指目标维数大于 4 的优化问题, 英文表述为 Many-objective optimization problem<sup>[4-5]</sup>)是当前多目标优化研究领域的热点和难点<sup>[6-7]</sup>, 其中一个难点在于如何找到一组有限个数的解集来表示整个 Pareto 前沿面, 要求这些解集能够尽量覆盖整个 Pareto 前沿面, 并且分布均匀, 这在本质上就是如何对求得的非被支配解进行有效的拥挤控制. 对于高维多目标优化问题, 随着目标维数的增加, 种群个体间互不支配的概率增大, 种群中非被支配个体的数量剧增, 这大大降低了种群的选择压力. 如何对这些个体进行有效的选择, 成了高维多目标优化的一大挑战. 为应对该问题, 本文通过深入研究, 提出了一种能够适用于高维多目标优化的种群拥挤控制策略 (Congestion control strategy based on open angle, CCSOA), 为验证该策略在高维多目标优化中的有效性, 本文将其为环境选择算子融合到进化多目标优化算法中, 得到算法 MOEA-CCSOA (Many-objective optimization integrating open angle based congestion control strategy).

文章结构安排如下: 第 1 节是本文的相关工作; 第 2 节重点阐述文章提出的基于张角的种群拥挤控制策略; 第 3 节重点介绍融合了新的拥挤控制策略的进化多目标优化 (Evolutionary multiobjective optimization, EMO) 算法; 第 4 节是实验仿真和结果分析部分; 第 5 节为结束语.

## 1 相关工作

### 1.1 多目标优化问题相关定义

不失一般性, 一个具有  $n$  个决策变量、 $m$  个目标函数的多目标优化问题<sup>[1]</sup> 可以表述为

$$\begin{cases} \min y = F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{s. t. } g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, 3, \dots, p \\ h_i(\mathbf{x}) = 0, \quad i = 1, 2, 3, \dots, q \end{cases} \quad (1)$$

其中, 目标函数  $F(\mathbf{x})$  定义了一个由决策空间到目标空间的映射函数;  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in X \subseteq \mathbf{R}^n$ ,  $\mathbf{x}$  为  $n$  维的决策向量,  $X$  为  $n$  维的决策空间;  $\mathbf{y} = (y_1, y_2, \dots, y_m) \in Y \subseteq \mathbf{R}^m$ ,  $\mathbf{y}$  为  $m$  维的目标向量,  $Y$  为  $m$  维的目标空间;  $g_i(\mathbf{x}) \leq 0$  ( $i = 1, 2, \dots, p$ ) 表示  $p$  个不等式约束;  $h_i(\mathbf{x}) = 0$  ( $i = 1, 2, \dots, q$ ) 表示  $q$  个等式约束. 在此基础上, 给出以下几个重要的定义:

**定义 1 (Pareto 占优).** 设  $\mathbf{x}, \mathbf{x}^* \in X$ , 称

$\mathbf{x}^*$  Pareto 占优  $\mathbf{x}$ , 当且仅当式 (2) 成立, 则记为:  $\mathbf{x}^* \succ \mathbf{x}$ .

$$\begin{aligned} & (\forall i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})) \wedge \\ & (\exists k \in \{1, 2, \dots, m\} : f_k(\mathbf{x}^*) < f_k(\mathbf{x})) \end{aligned} \quad (2)$$

**定义 2 (Pareto 最优解集).** Pareto 最优解集是所有 Pareto 最优解组成的集合, 定义如下:

$$P^* = \{\mathbf{x}^* | \neg \exists \mathbf{x} \in X : \mathbf{x}^* \succ \mathbf{x}\} \quad (3)$$

**定义 3 (Pareto 前沿面).** Pareto 最优解集  $P^*$  中所有 Pareto 最优解对应的目标矢量组成的曲面称为 Pareto 前沿面  $PF^*$ :

$$\begin{aligned} PF^* = \\ \{F(\mathbf{x}^*) = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_m(\mathbf{x}^*))^T | \mathbf{x}^* \in P^*\} \end{aligned} \quad (4)$$

**定义 4 (极端集).** 在多目标进化算法中, 单个目标当前最优的点称之为极端点, 所有目标的极端点所组成的集合称为极端集.

### 1.2 研究现状分析

种群的多样性决定了种群的进化潜能, 因此设计能保证种群多样性的个体选择策略是进化算法要处理的重要问题. 早期的多样性保持策略主要有以下几类<sup>[7]</sup>: 小生境技术、信息熵的方法、聚集密度的方法、与网格相关的方法和聚类的方法. 为了选择合适的个体推动种群的进化, Deb 等在 NSGA-II (Nondominated sorting genetic algorithm II)<sup>[8]</sup> 中通过 Pareto 排序来选择个体, 并设计了计算个体间聚集距离的方法来进行拥挤控制, 可以避免过分相似的同级 Pareto 个体被同时选择; Corne 等提出的 PESA (Pareto envelop-based selection algorithm)<sup>[9]</sup> 则是通过网格方法来维持解集的分布性和多样性; Zitzler 等在算法 SPEA2<sup>[10]</sup> 中提出了归档集的修剪策略. 这些代表性的算法能够很好地处理 2 维和 3 维的目标优化问题, 但随着目标维数的线性增加, 种群中非被支配解所占比例剧增, 算法的优化效果下降. 为此, 研究者们提出了多种改进来应对高维目标下种群多样性维护问题, 主要可以归纳为以下几类:

1) 基于权重系数的方法. 该类算法通过设计一组权重系数将高维多目标优化问题转化为单目标优化问题. 如 Zhang 将传统的数学规划方法与进化算法结合起来提出了 MOEA/D<sup>[11]</sup>, 算法设置了一组固定的均匀分布的目标权重向量, 将问题分解成一组子问题来求解, 各子问题的求解以固定的目标权重为基础, 目标权重的均匀性提升了种群分布的均匀性. 而毕晓君等提出的 HMMCA (High-dimensional multi-objective multi-directional co-

evolutionary algorithm)<sup>[12]</sup> 则是利用目标空间内的一组方向向量将多目标优化问题分解成多个方向进行寻优, 并应用于高维多目标优化中。

2) 改进支配关系的方法. 该类方法重新定义支配法则, 本质上是将个体的目标函数值按照一定的比例缩小或放大之后, 再与其他个体进行支配比较, 是对传统支配关系的弱化. 如 Deb 等提出了不同于传统 Pareto 占优的新型占优机制  $\epsilon$  占优<sup>[13]</sup> 的概念; 如 Yang 等在算法 GrEA (Grid-based evolutionary algorithm)<sup>[5]</sup> 中结合 Pareto 支配的概念和传统网格的概念后, 提出了网格支配的概念来处理高维多目标优化问题。

3) 降维的方法<sup>[14-15]</sup>. 该类方法主要是通过降低目标维度来提升传统算法对高维目标的优化能力。

4) 基于指标的方法. 这类方法引入衡量算法的性能指标来指导种群进化. 如 Zitzler 等将超体积指标引入到个体适应度函数中, 提出算法 IBEA (Indicator-based evolutionary algorithm)<sup>[16]</sup>; 如巩敦卫等<sup>[17]</sup> 提出基于集合的进化算法, 该方法以超体积、分布度以及延展度为新的目标, 将高维多目标优化问题转化为 3 目标优化问题。

5) 对目标空间进行划分的方法. 该方法在目标空间引入额外的参考点或者向量来辅助控制种群在目标空间的分布. 如 Deb 等通过设置了一组分布均匀的参考点, 在环境选择中选择与参考带最近的个体, 提出算法 NSGAIII (Nondominated sorting genetic algorithm III)<sup>[4]</sup>; 如 Jiao 等在算法 DVCMOA (Direction vectors based co-evolutionary multi-objective optimization algorithm)<sup>[18]</sup> 中通过一组均匀分布的方向向量将目标空间区域进行均匀划分, 种群中的个体被划分到各个子区域进行协同进化, 以提升种群的多样性; 而 Reynoso-Meza 等<sup>[19]</sup> 在算法 spMODE 中则设计了一种目标空间的球形裁剪策略来对种群进行拥挤控制。

6) 混合算法<sup>[20-21]</sup>. 该类算法将其他领域的技术引入进化算法中, 加强了算法的寻优能力. 如詹炜<sup>[21]</sup> 将“流形学习”算法引入到高维多目标优化算法中, 提出流形学习多目标优化算法, 利用“流形学习”算法降低高维多目标优化问题决策空间的数据维度, 挖掘决策空间中内蕴的“流形”结构, 建立准确的决策空间模型, 指导算法演化, 加速收敛。

以上这几类方法在提高种群多样性上都有其优势: 基于权重系数的方法如 MOEA/D<sup>[11]</sup> 通过一组均匀权重系数构造出不同子种群, 子种群具有不同的进化压力, 种群多样性得到提高; 基于指标的方法 IBEA<sup>[16]</sup> 将超体积指标融入到个体适应度值的计算中, 通过淘汰最差适应度个体来提升种群的分布性; NSGAIII<sup>[4]</sup> 通过设置固定的参考点, 选择与参考点最近的个体进入下一代, 参考点分布的均匀性提高

了解集的均匀性, 而包含标准化目标空间边界的参考点可以增强种群分布的广度; GrEA<sup>[5]</sup> 通过计算每个个体的网格排名 (Grid ranking)、网格拥挤距离 (Grid crowding distance) 和网格相关点距离 (Grid coordinate point distance), 依据这些值来进行锦标赛来提高种群的多样性, 算法设计的一套适应度惩罚机制则保证了边界个体具有较高的适应度, 提高了边界极端个体的进化概率, 从而提高种群分布的广度。

但这些算法也存在一些不足: 基于权重系数的方法所引入的权重系数相当于引入了偏好信息, 但该信息不够准确和全面, 容易导致算法寻优的结果偏离真正的 Pareto 前沿; 而基于指标的方法严重依赖于所选择的指标, 并且不同类型指标间的平衡不容易把握; 目标空间设置参考点的方法 NSGAIII<sup>[4]</sup> 在高维空间中引入的参考点个数有限, 固定位置的选取使得该方法具有一定的局限性; GrEA<sup>[5]</sup> 通过高维空间中的网格来进行拥挤控制, 容易造成网格空洞, 较难平衡局部与全局的拥挤关系, 而且网格粒度也不容易控制; 改进支配法则中的参数难以确定, 对个体的真实目标函数值改变容易导致算法无法收敛到真实的 Pareto 前沿; 而降维的方法会造成信息丢失, 使得最终优化效果不稳定; 混合算法则是当前多目标研究的热门方向, 但它涉及多个研究领域, 算法复杂度较高, 实现难度较大. 为此, 本文通过深入研究提出了一种基于张角的种群拥挤控制策略 CCSOA, 旨在设计出更加适用于高维目标空间的环境选择策略, 提高种群的多样性以及种群在目标空间分布的均匀性和广度, 进而提高种群的收敛效果. 本文的创新点主要体现在: 1) 算法 CCSOA 采用张角值来度量个体间的相似度, 算法的时间复杂度仅与种群规模有关, 而与目标维度无关, 适用于高维多目标优化; 2) 算法 CCSOA 根据个体间的相对位置进行拥挤控制, 不需要额外引入参考点或者向量; 3) 针对存档的非被支配种群, 算法 CCSOA 能够进行有效的拥挤控制, 提高种群分布的均匀性; 而针对同时包含非被支配个体和被支配个体的混合种群, 算法能够提升种群稀疏区域被支配个体的进化概率, 提高种群多样性; 4) 融合了 CCSOA 策略的高维多目标优化算法 MOEA-CCSOA 中, 对张角阈值进行了自适应调整, 并引入了缓冲池的策略。

## 2 基于张角的种群拥挤控制策略

### 2.1 概念及定义

张角是指在目标空间中两目标向量之间的夹角, 它在一定程度上刻画了两个个体在目标空间的差异性, 比如有两个互不支配的个体, 如果它们之间的张角值越小, 说明它们越相似, 并在 Pareto 前沿面上

分布越紧凑;而偏角则被设计为目标向量与中心向量之间的夹角,表示在目标空间中该个体相对中心向量的张角偏移程度.张角和偏角的定义如下:

**定义 5 (张角).** 令种群为  $P$ , 个体  $p_1$  与  $p_2$  之间的张角 (Open angle) 定义为

$$Oa(p_1, p_2) = \arccos\left(\frac{F^{\text{norm}}(p_1) \cdot F^{\text{norm}}(p_2)}{\|F^{\text{norm}}(p_1)\|_2 \times \|F^{\text{norm}}(p_2)\|_2}\right) \times \frac{180}{\pi} \quad (5)$$

$$F^{\text{norm}}(p) = [f_1^{\text{norm}}(p), f_2^{\text{norm}}(p), \dots, f_M^{\text{norm}}(p)] \quad (6)$$

$$f_i^{\text{norm}}(p) = \frac{f_i(p) - \min_{p \in P}(f_i(p))}{\max_{p \in P}(f_i(p)) - \min_{x \in P}(f_i(p))}, \quad i = 1, 2, \dots, M \quad (7)$$

**定义 6 (偏角).** 个体  $p$  在  $M$  维目标空间中的偏角 (Drift angle) 定义为该个体与虚拟中心个体  $p_{\text{central}}$  在目标空间的张角值:

$$Da(p) = Oa(p, p_{\text{central}}) \quad (8)$$

其中, 设定虚拟中心个体  $p_{\text{central}}$  对应的所有目标值均为 1, 在目标空间不偏向任何一维坐标轴.

## 2.2 拥挤控制策略 CCSOA 描述

依据最优前沿面的非支配特性, 我们采用张角来衡量个体间的相似性, 并结合张角和偏角来控制个体的选择顺序, 设计出种群拥挤控制策略 CCSOA 来对种群个体进行筛选. 其中, RIS (Representative individual set) 表示一个具有多样性代表的个体集,  $RP(i), i = 1, 2, \dots, N$  表示由输入种群划分得到的  $N$  组相似个体集 (Resembled population).

### 算法 1 (CCSOA).

**输入.** 种群 POPin, 目标个数  $M$ , 目标函数  $F$ , 张角阈值.

**输出.** RIS,  $RP(i), i = 1, 2, \dots, N$ .

**步骤 1 (初始化操作).** 初始化 RIS 为空,  $i = 1$ ;

**步骤 2 (计算偏角).** 根据式 (5)~(8), 计算 POPin 中每个个体的偏角.

**步骤 3 (MAX 选择).** 在输入种群 POPin 中选择具有最大偏角值的个体加入集合 MaxPool (可能是一个个体, 也可能是多个个体).

**步骤 4 (MIN 选择).** 将最后一个加入 RIS 的个体记为 Plast. 如果 Plast 为空 (RIS 中尚无个体), 则从 MaxPool 中任意选取一个个体进入 RIS,

并将 Plast 更新为该个体, 更新  $i = i + 1$ ; 否则, 计算 MaxPool 中每个个体与 Plast 之间的张角, 并从中选择具有最小张角的个体进入 RIS, 并将 Plast 更新为该个体, 更新  $i = i + 1$ .

**步骤 5 (加减操作).** 1) 计算种群 POPin 内所有个体与 Plast 之间的张角; 2) 将张角值小于阈值的所有个体从 POPin 中转移至  $RP(i)$ .

**步骤 6 (停止条件).** 如果 POPin 种群大小不为 0, 则跳转到步骤 2; 否则, 输出典型代表个体集 RIS, 相似个体集  $RP(i), i = 1, 2, \dots, N$ , 算法结束.

图 1 是算法 CCSOA 在三维目标下的示意图, 输入种群在目标空间都经过标准化处理, 即每一维目标均处于区间  $[0, 1]$ , 为了说明方便, 仅在此假设所有个体在目标空间都沿着目标向量方向被投影到半径为 1 的  $1/8$  球面上 (由于计算的是向量间的夹角, 该假设对计算结果不产生影响), 并用小圆点表示. 如图 1 上半幅所示, 图 1 中圈出了通过策略 CCSOA 依次得到 10 组相似个体集  $RP(i), i = 1, 2, \dots, 10$ , 每组相似集中都有一个五角星表示通过算法得到的典型代表个体, 剩下的为种群其他个体. 图 1 下半幅从侧面展示了最终得到的典型代表集合 RIS.

## 2.3 算法分析

1) 算法步骤 3 中 MAX 操作优先选取偏角值较大的个体, 使得个体选择顺序相对中心向量由远及近, 提高了种群边缘极端个体的进化概率, 提高种群分布的广度.

2) 在阈值的控制下, 算法将输入种群划分成  $N$  组相似个体集, 标记为  $RP(i), i = 1, 2, \dots, N$ , 这  $N$  组个体集在原点的视角空间中互不重叠, 使得组间的个体间保持了一定的差异性; 阈值也同样保证了算法得到的典型代表个体间具有一定的差异性. 算法得到分布较为均匀的典型代表个体集合 RIS.

3) CCSOA 重视稀疏区域被支配个体对加强种群多样性所起的作用. 被支配个体虽然没有非支配个体优秀, 但如果它处于目标空间的个体稀疏区域, 那么它比处在高密度区域的非支配个体更能加强种群的多样性. 图 2 是二维目标情况下普通种群采用 CCSOA 算法接受环境选择的示意图, 它表明了稀疏区域中被支配个体对种群多样性的贡献: 普通种群 (Normal population, NPOP) 如图 2(a) 所示,  $p_1, p_2, p_3, p_4$  和  $p_5$  均表示被支配个体, 其余为非被支配个体; 对种群 NPOP 进行非支配过滤的结果如图 2(b) 所示; 而采用 CCSOA 策略对 NPOP 进行筛选的结果如图 2(c) 所示, 它包含了部分稀疏区域的被支配个体, 对比图 2(b) 和 (c) 中个体的分布情况可知, 当非支配个体过于集中于某些区域时, CCSOA 通过选取部分稀疏区域的被支配个体可以

提高下一代种群的多样性。

4) 设输入种群的规模为  $n$ , 目标维度为  $M$ , 角度阈值固定, 则算法 CCSOA 在最坏的情况下, 会将输入种群划分成  $n$  个相似个体集, 每个相似个体集中只有一个个体. 则此时的算法时间复杂度为  $O(n^2)$ . 策略 CCSOA 中的张角的计算只基于向量

点积, 随着目标维度的增加, 该策略的算法复杂度不会增加, 而只与种群的规模有关, 适用于高维目标.

5) 当输入种群全为非被支配个体时, 运用该策略输出的 RIS 是分布较为均匀且有一定广度的精英个体集; 当输入种群同时包含非被支配个体与被支配个体时, 输出的 RIS 不仅包含非支配个体也包含

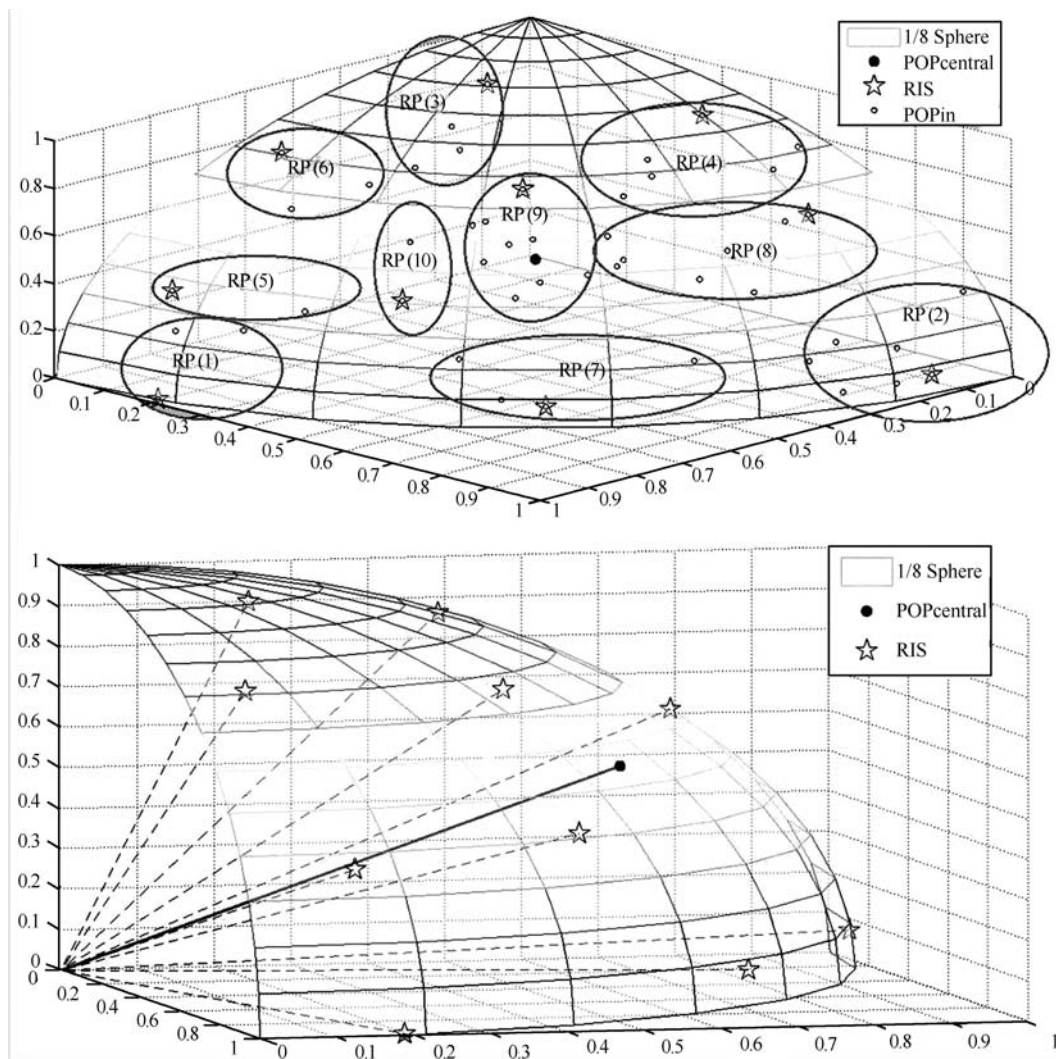


图 1 拥挤控制策略 CCSOA 实例示意图

Fig. 1 The diagram of congestion control strategy CCSOA

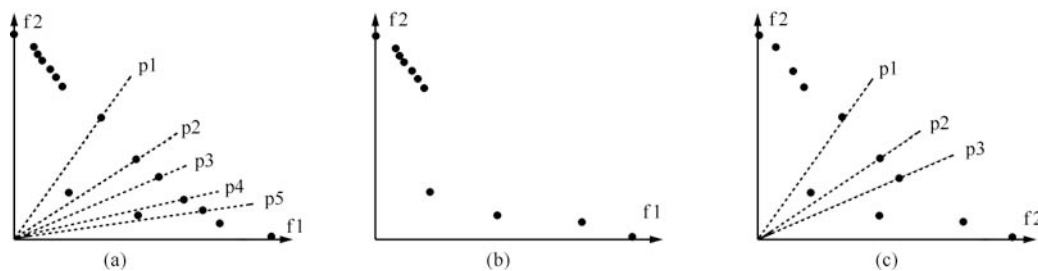


图 2 拥挤控制策略 CCSOA 作用于二维普通种群的示意图

Fig. 2 The diagram for the congestion strategy CCSOA execute on 2-dimensional normal population

部分稀疏区域的被支配个体, 稀疏区域的被支配个体对种群多样性的作用得到重视, CCSOA 策略对不同种群都有较强的适用性. 另外, 本文只认为 RIS 中的所有个体作为一个整体时是输入种群的典型代表, 而不是所在相似个体集的典型代表.

### 3 融合张角拥挤控制策略的多目标优化算法 MOEA-CCSOA

为了验证基于张角的拥挤控制策略 CCSOA 在高维多目标优化算法中的效果, 本文利用该策略设计出一种融合张角拥挤控制策略的多目标优化算法 MOEA-CCSOA 来解决高维多目标优化问题. 下面给出算法 MOEA-CCSOA 的流程, 其中, 算法设置一个普通种群 (Normal population, NPOP)、存档种群 (Achieved population, APOP) 和缓冲池 (Buffer pool). 存档种群只包含非被支配个体.

#### 算法 2 (MOEA-CCSOA).

**输入.** 目标函数  $F$  和目标总个数  $M$ , 最大目标评价次数 MaxEval, 缓冲池规模上限  $N_{\text{buff}}$ .

**输出.** 存档种群 APOP

**步骤 1 (参数初始化).** 初始化角度阈值  $\sigma_{\text{npop}}$ 、 $\sigma_{\text{apop}}$  和  $\sigma_{\text{buff}}$ ; 初始化进化种群规模  $N_{\text{npop}}$  和存档种群规模  $N_{\text{apop}}$ ; 目标评价次数  $t_{\text{eval}} < 0$ .

**步骤 2 (种群初始化).** 根据种群规模随机初始化种群 NPOP, 初始化 APOP 为空.

**步骤 3 (环境选择操作).** 采用 CCSOA 策略从当前存档种群 APOP 和缓冲池中的非支配解中, 筛选出典型代表个体集更新为下一代存档种群; 采用 CCSOA 策略从缓冲池中筛选出典型代表个体集更新为下一代普通种群 NPOP; 更新阈值和目标评价次数  $t_{\text{eval}}$ . 调整角度阈值, 更新目标评价次数.

**步骤 4 (种群基因操作).** 对种群 NPOP 和 APOP 分别进行交叉和变异操作, 并将 APOP 的子代参与 NPOP 的环境选择.

**步骤 5 (停机判断).** 如果  $t_{\text{eval}} < \text{MaxEval}$ , 则算法跳转到步骤 3; 否则, 输出当前存档种群 APOP, 算法结束.

#### 3.1 环境选择操作

为了提升种群的选择压力, 算法仅从普通种群的非支配个体中进行环境选择, 但是由于种群中的非支配个体的数量可能由于进化而不稳定, 算法 CCSOA 的优势也无法得到有效利用. 为此, 算法设置了一个容量比普通种群大的缓冲池来辅助环境选择操作, 缓冲池中存储了历代普通种群得到的非支配个体, 并对其进行了维护. 环境选择是一个适者生存、优胜劣汰的过程. 这里的环境选择机制具体包含 4 个步骤:

**步骤 1 (非支配过滤).** 将 NPOP 中的非被支配

个体加入缓冲池.

**步骤 2 (更新存档种群).** 在当前存档种群 APOP 和缓冲池的非支配个体中, 采用 CCSOA 策略筛选出一定规模的典型代表个体集 RIS, 并将其更新为下一代存档种群.

**步骤 3 (缓冲池维护).** 如果缓冲池的种群规模不足  $N_{\text{buff}}$ , 则随机生成一定量的个体使它的规模达到  $N_{\text{buff}}$ ; 相反, 如果缓冲池中种群规模大于  $N_{\text{buff}}$ , 采用选择策略 CCSOA 进行划分得到  $N$  组相似个体集  $\text{RP}(i), i = 1, 2, \dots, N$ , 每次均从规模最大的相似个体集中删除一个在标准化目标空间中模长较长的个体直至整个缓冲池的规模刚好为  $N_{\text{buff}}$ .

**步骤 4 (更新普通种群).** 采用 CCSOA 策略从缓冲池中选出典型代表个体 RIS, 并将 NPOP 更新为 RIS, 如果 NPOP 的种群规模不足  $N_{\text{npop}}$ , 随机生成个体进行补充使之达到  $N_{\text{npop}}$  的规模.

**步骤 5.** 根据式 (9) 更新角度阈值  $\sigma_{\text{npop}}$ 、 $\sigma_{\text{apop}}$  和  $\sigma_{\text{buff}}$ ; 更新目标评价次数  $t_{\text{eval}}$ .

#### 3.2 基因操作

由于缓冲池中既包含非被支配也包含被支配个体, 对 NPOP 的环境选择虽然可以提高种群的多样性, 但可能会降低种群的占优比例. 为了进一步加强种群的进化压力, 算法充分利用筛选出的存档种群进行内部交叉变异操作. 算法采用如下操作算子: 1) 存档种群只包含非支配个体, 采用模拟二元交叉操作 SBX<sup>[22]</sup> 以使不同精英强强结合, 可充分发挥精英的作用, 而高斯变异<sup>[23]</sup> 操作也为精英个体的进化提供了动力; 将存档种群产生的子代加入普通种群参与环境选择. 2) 普通种群中的个体比较混杂, 具有较好的多样性, 采用差分进化操作策略<sup>[24]</sup>, 以引导那些不够优秀但对种群多样性做出重要贡献的个体加快进化, 也为其中的优秀个体提供了更多的进化方向. 基因操作方法的多样性也有利于增强进化过程中种群的多样性.

#### 3.3 讨论分析

1) 参数设置策略 CCSOA 的有效执行严重依赖于角度阈值参数  $\sigma$  ( $\sigma_{\text{npop}}$ 、 $\sigma_{\text{apop}}$  和  $\sigma_{\text{buff}}$ ), 参数值越小, 输出的典型代表个体数越多, 而划分的相似个体集的区域数越多; 相反, 阈值越大, 输出的典型代表个体数越少, 而划分得到的相似个体集的组数越少, 但每组相似个体集内个体数越多. 典型代表个体数过多或者过少都不利于种群的进化, 因此需要对阈值  $\sigma$  进行自适应动态调整. 在标准化的目标空间中, 张角范围内的空间区域大小与张角值有近似正比例关系. 而算法 CCSOA 中为得到 RIS, 会将张角范围内的相似个体移除, 可知张角值与典型代表个体数量有近似反比例关系, 即  $\frac{\sigma_{\text{desire}}}{\sigma_{\text{old}}} \approx \frac{N_{\text{old}}}{N_{\text{desire}}}$ .

为了使算法得到的 RIS 规模接近我们预设的规模  $N_{\text{desire}}$ , 算法采用取中间值的方法对阈值进行调整 ( $\sigma_{\text{new}} = \frac{1}{2}(\sigma_{\text{desire}} + \sigma_{\text{old}})$ ), 得到的调整策略为

$$\sigma_{\text{new}} = \frac{1}{2} \left( \frac{N_{\text{old}}}{N_{\text{desire}}} \sigma_{\text{old}} + \sigma_{\text{old}} \right) \quad (9)$$

其中,  $\sigma_{\text{old}}$  和  $N_{\text{old}}$  为上一次环境选择使用的阈值和得到的典型代表个体数量,  $N_{\text{desire}}$  和  $\sigma_{\text{new}}$  为期望得到的典型代表个体的数量和调整后的阈值, 在普通种群和存档种群中,  $N_{\text{desire}}$  即为种群规模  $N_{\text{npop}}$  和  $N_{\text{apop}}$ ; 针对缓冲池, 设置  $N_{\text{desire}} = 1/4 \times N_{\text{buff}}$ . 另外, 算法随机初始化为  $(0, 25]$  区域内的一个值, 随着进化进行调整.

2) 环境选择策略在标准化的目标空间, 如果  $\|f(\mathbf{x})\|_2 > \|f(\mathbf{y})\|_2$ , 那么绝对不会出现  $\mathbf{x}$  占优  $\mathbf{y}$  的情况. 算法在同一组的相似个体集中选择保留模长较短的个体 (或者淘汰模长较长的个体), 是对传统 Pareto 支配的一种弱化, 选择模长较短的个体本质上是对非被支配个体的选择.

## 4 仿真实验

为了验证 MOEA-CCSOA 对多目标优化问题的求解性能, 将其与相关算法 IBEA、NSGAIII、GrEA 和 MOEA-CCSOA 在 DTLZ 系列标准测试函数集<sup>[25]</sup> 上进行对比实验. 所有实验均在硬件配置为 Intel Core i5-2450M CPU 2.50 GHz、内存 4GB 的 PC 上进行, 编程工具为 Matlab, 版本号为 R2012a.

### 4.1 算法性能评价指标

在对比实验中, 为了衡量出算法在不同方面的性能, 选用了 3 个相关指标. 其中指标 Generational distance (GD)<sup>[26]</sup> 衡量了算法的收敛效果; 指标 Spacing<sup>[27]</sup> 主要对算法结果集分布的均匀性做出评价; Delta<sup>[28]</sup> 指标主要衡量算法最终获得的非被支配解集的扩展程度及分布均匀性, Spacing 和 Delta 共同衡量了种群的多样性. GD<sup>[26]</sup> 用来评价 PF 与 PF\* 的远近程度:

$$\text{GD} = \frac{1}{n} \left( \sum_{i=1}^n d_i^2 \right)^{\frac{1}{2}} \quad (10)$$

针对某个具体的优化问题, PF\* 是均匀分布的理论 Pareto 最优前沿面, PF 表示通过算法求得的 Pareto 最优前沿面 (下同),  $n$  是 PF 中个体的个数. 表示 PF 中第  $i$  个个体与 PF\* 中的个体在目标空间的最小欧氏距离. GD 的值越小, 得到的前沿面 PF 越逼近理论上真实的 Pareto 最优前沿面.

Spacing (SP)<sup>[27]</sup> 主要对算法结果集分布的均匀

性做出评价, 计算公式如下:

$$\begin{cases} \text{Spacing} = \frac{\left[ \frac{1}{n} \sum_{i=1}^n (d'_i - \bar{d})^2 \right]^{\frac{1}{2}}}{\bar{d}} \\ \bar{d} = \frac{1}{n} \sum_{i=1}^n d'_i \end{cases} \quad (11)$$

其中,  $d'_i$  示 PF 中第  $i$  个个体与其最近个体之间的欧氏距离,  $n$  是 PF 中个体的数量.

Delta<sup>[28]</sup> 指标主要衡量算法最终获得的非被支配解集 PF 中个体分布的扩展程度及均匀性:

$$\text{Delta} = \frac{\sum_{i=1}^m d(e_i, \text{PF}) + \sum_{X \in \text{PF}^*} |d(X, \text{PF}^*) - \bar{d}|}{\sum_{i=1}^M d(e_i, \text{PF}) + |\text{PF}^*| \bar{d}} \quad (12)$$

其中,  $d(X, \text{PF}^*) = \min_{Y \in \text{PF}^*, Y \neq X} \|f(X) - f(Y)\|^2$ ,  $\bar{d} = \frac{1}{|\text{PF}^*|} \sum_{X \in \text{PF}^*} d(X, \text{PF}^*)$ , 而  $(e_1, e_2, \dots, e_m)$  是 PF\* 中的  $m$  个极端集, Delta 指标值越小说明算法群体多样性越好, 其获得非被支配解分布越均匀, 且扩展性越好, 越能完整覆盖 Pareto 前沿.

### 4.2 算法参数设置

仿真对比的各个算法的参数值, 尽量参考算法文献中的取值. 交叉和变异相关参数设置: 交叉概率  $Pc = 1$ , 变异概率  $Pm = 1/n$ , 其中  $n$  表示决策变量的个数; 二元交叉分布指数、多项式变异分布指数和高斯变异分布指数均设置为 20; 差分变异伸缩因子  $F = 0.5$ . NSGAIII 中引入参考点的个数分别为 91、165、210、126、156 和 275, 分别对应目标维数 3、4、5、6、8 和 10, 而种群规模与参考点个数一致. 算法 GrEA 中网格参数 div 的设置均依照文献. IBEA 和 GrEA 的种群规模均设置为 100. MOEA-CCSOA 种群规模的大小设置为: 普通种群  $N_{\text{npop}} = 100$ , 存档种群  $N_{\text{apop}} = 100$ , 缓冲池上限  $N_{\text{buff}} = 400$ . 停机条件参照 GrEA 中的设置: 对于测试函数集 DTLZ1、DTLZ3 和 DTLZ6, 所有对比算法的最大目标评价次数为 100 000 次; 对于测试函数集 DTLZ2、DTLZ4、DTLZ5 和 DTLZ7, 所有对比算法的最大目标评价次数为 30 000 次. 最后, 所有对比算法要在每一个标准测试函数上独立运行 30 次, 以此结果来对各指标进行统计分析.

### 4.3 仿真结果及分析

采用 GD<sup>[26]</sup>、Spacing<sup>[27]</sup> 和 Delta<sup>[28]</sup> 指标来衡量算法的性能, 对算法 IBEA、NSGAIII、GrEA 和 MOEA-CCSOA 在不同目标维度的 DTLZ 系列测试函数上各进行 30 次独立实验, 来验证算法



的性能,并在图 3、图 4 和图 5 中给出了以上 3 种指标的箱型图. 为了方便各算法的比较和图示表示,图中用字母 A、B、C 和 D 分别指代算法 IBEA、NSGIII、GrEA 和 MOEA-CCSOA,且在图中用黑色圆角矩形框标出了 MOEA-CCSOA 算法表现最好时对应的指标箱型图.

### 4.3.1 解集分布的均匀性

Spacing 指标表征的是解集分布的均匀性,它仅与算法求得解集的分布性相关(与理论 Pareto 前沿无关),指标值越小,说明得到的解集分布越均匀,越有利于给决策者提供决策支持,这也是算法 CC-SOA 的直接目标,即要提高最终输出解集的均匀性.

图 3 给出了实验得到的 Spacing 指标统计箱型图,从中可以看出:在所有测试维度的 DTLZ2、DTLZ4 和 DTLZ7 系列测试函数上,算法 MOEA-CCSOA 都有最好的均匀性表现;在 DTLZ1 系列测试函数中,NSGIII 的表现最好,但 MOEA-CCSOA 在 DTLZ3 系列测试函数中,MOEA-CCSOA 维度为 3、4 和 5 的测试中种群均匀性维持能力与 NSGIII 相当;在目标维数为 3、4 和 5 时表现中等偏上,从四分位差可以看稳定性表现较差,而算法 GrEA 在各个维度上都有相对最佳的效果;在 DTLZ5 系列测试函数中,MOEA-CCSOA 在维度为 4 和 5 时具有较大的优势,而在维度为 6、8 和 10 的时候,4 个对比算法性能相当,都有较好的表现. 综上,显式地

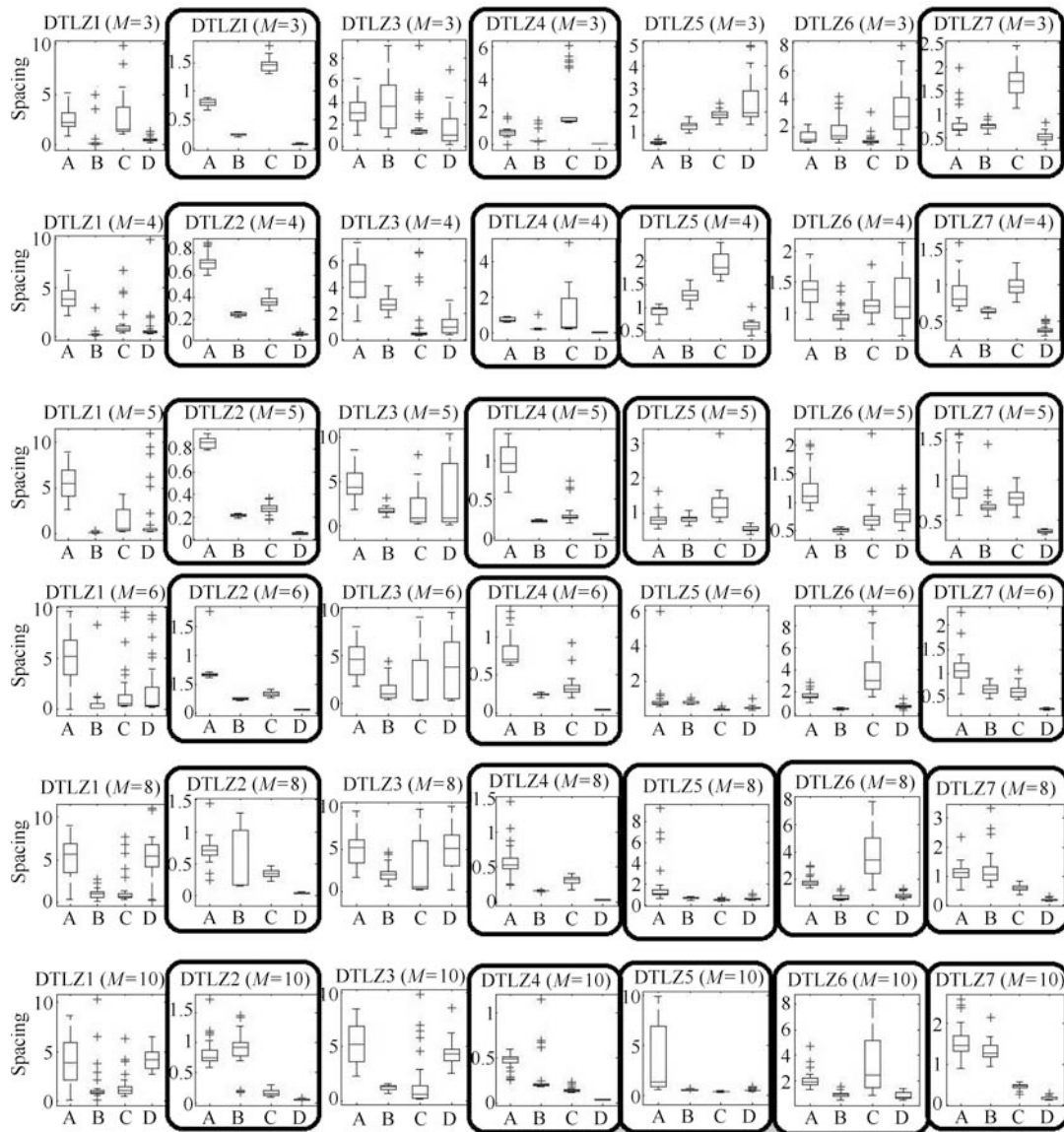


图 3 4 种算法在 42 个测试函数上独立运行 30 次得到结果的 Spacing 指标箱型图

Fig. 3 Box plots of the four algorithms for 42 benchmark problems on metric Spacing (30 independent runs)



说明采用 CCSOA 对存档种群进行维护, 可以有效提高种群分布的均匀性; 另外, 拥挤控制策略对提高种群的多样性也做出了贡献。

#### 4.3.2 解集分布的宽广性

Delta 是解集分布的宽广性度量指标, 该指标的计算必须提供已知的理论 Pareto 前沿面的数据, 它主要表征的是解集对理论 Pareto 前沿面上边界的覆盖率. Delta 指标值越小, 说明算法得到的 Pareto 前沿面的边界与理论 Pareto 前沿面边界的越逼近, 具有很好的宽广性, 且其均匀性较好。

图 4 给出了实验得到的 Delta 指标统计箱型图, 从中可以看出: 在绝大多数的

DTLZ4、DTLZ4、DTLZ6 和 DTLZ7 系列测试函数上, 算法 MOEA-CCSOA 都有最好的宽广性表现; 在 DTLZ3 系列测试函数中, MOEA-CCSOA 与 GrEA 性能相当, 均为最好; 仅在 DTLZ1 系列测试函数中, MOEA-CCSOA 的表现较差, 排在第三. 综上, CCSOA 中对个体从外到内的选择策略 (优先选取离中心向量张角值最大的个体) 对种群分布的广度起到了效果, 达到了设计的目的。

#### 4.3.3 解集的收敛效果

GD 指标衡量了算法得到解集对理论 Pareto 前沿面的逼近程度, 即收敛效果; GD 的值越小, 说明

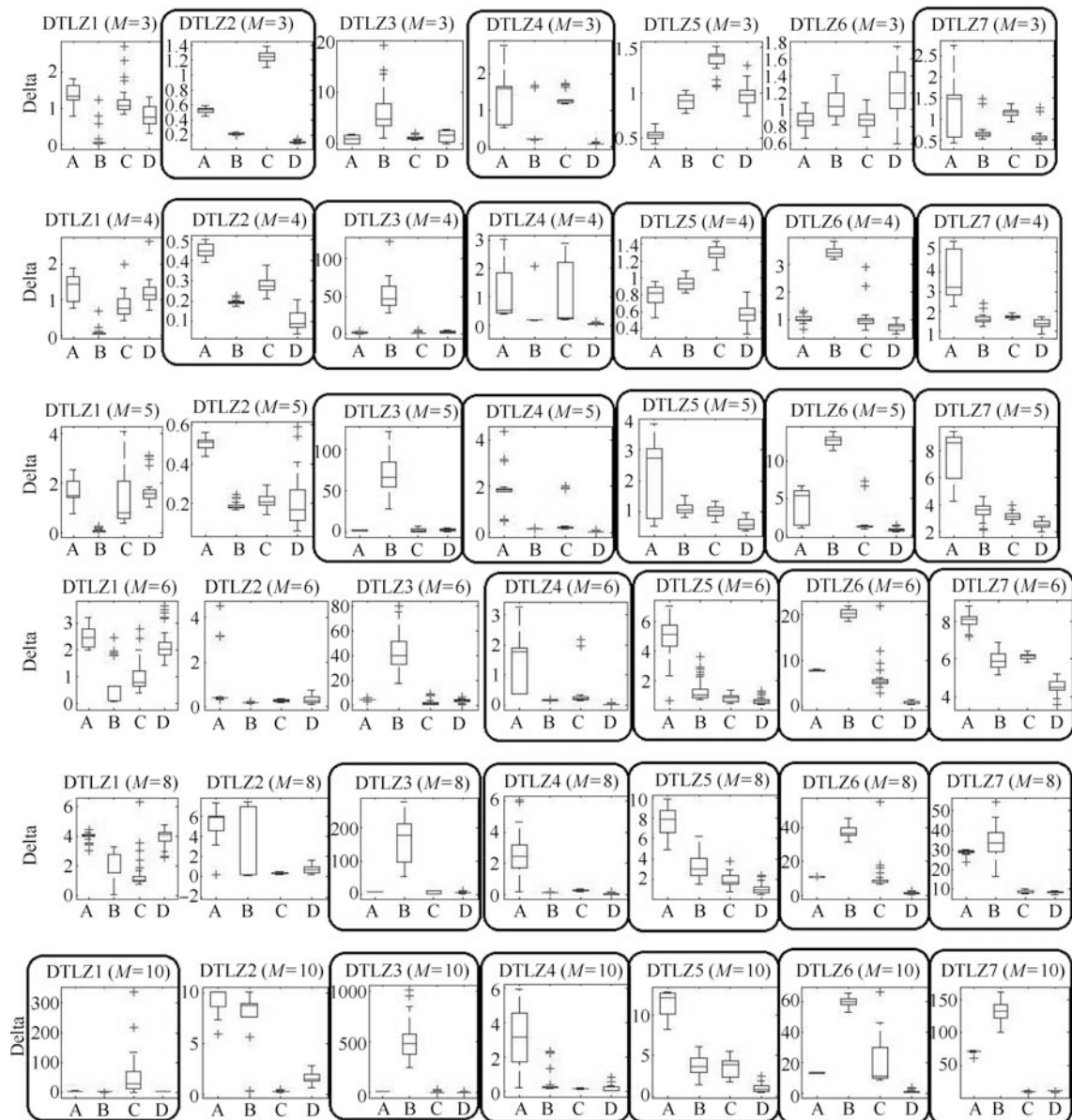


图 4 4 种算法在 42 个测试函数上独立运行 30 次得到结果的 Delta 指标箱型图

Fig. 4 Box plots of the four algorithms for 42 benchmark problems on metric Delta (30 independent runs)

解集越逼近理论 Pareto 前沿面, 越接近理论最优值, GD 指标值的计算必须预先求得理论 Pareto 前沿面. 收敛效果是决策者首要考虑的因素, 也是算法的首要评价指标. Spacing 和 Delta 指标综合衡量了种群的多样性, 多样性不是决策者首要考虑的决策因素, 但却是提高算法收敛效果的重要潜在因素, 也是算法设计的出发点, 通过提高种群的多样性来进而提高算法的收敛效果. 图 5 给出了实验得到的 GD 指标统计箱型图, 从图 5 中可以看出: 在绝大多数测试的 DTLZ2 和 DTLZ4 系列函数上,

MOEA-CCSOA 均有最好的收敛效果, 在剩下 8 维 DTLZ2、8 维 DTLZ4 和 10 维 DTLZ4 上, NSGAIII 明显占优, MOEA-CCSOA 排在第二, GrEA 次之, IBEA 则较差些; 在 DTLZ1 上, 目标维数为 3、4 和 6 时, 所有对比算法的性能接近, MOEA-CCSOA 和 GrEA 异常点较多, 在目标维度为 5、8 和 10 时, IBEA 的收敛效果最好且稳定; 在 DTLZ3 上, 目标维数为 3 和 4 时, IBEA、GrEA 和 MOEA-CCSOA 收敛性能相当, 均为最好, NSGAIII 则收敛性能稍差, 在剩下的测试目标维度上, IBEA 和 GrEA 收敛

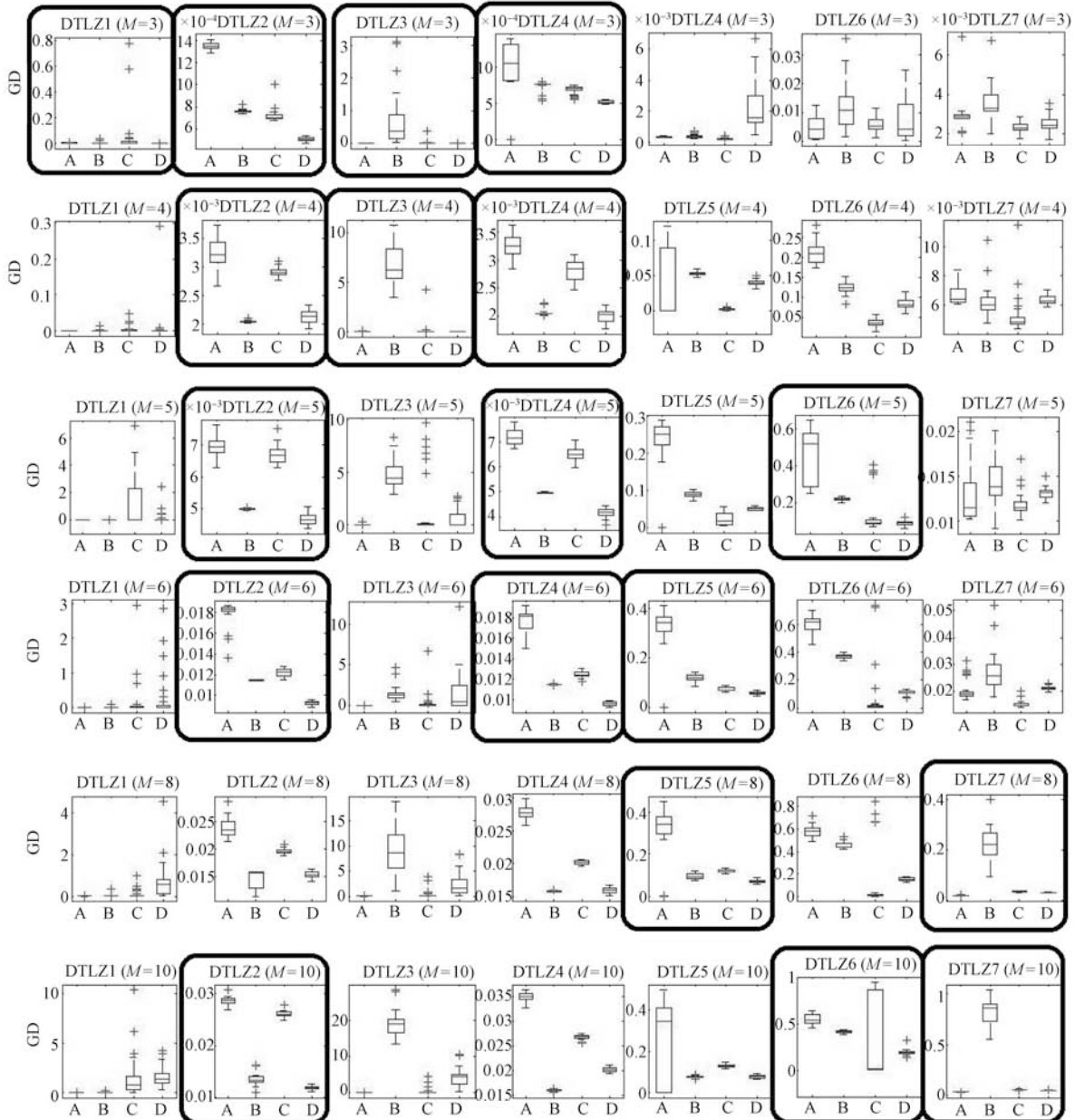


图 5 4 种算法在 42 个测试函数上独立运行 30 次得到结果的 GD 指标箱型图

Fig. 5 Box plots of the four algorithms for 42 benchmark problems on metric GD (30 independent runs)

性能仍然相当, 均为最好, MOEA-CCSOA 次之; 在 DTLZ5 上, MOEA-CCSOA 在 6、8 和 10 维目标上测试出最好的收敛性, GrEA 在 4 和 5 维目标是上占优, 而在 3 维目标上 IBEA、NSGAIII 和 GrEA 收敛性能相当, 均优于 MOEA-CCSOA; 在 DTLZ6 上, MOEA-CCSOA 则仅在 5 维目标上明显优于其他对比算法, 在 3 维目标上所有对比算法性能接近, 在剩下的 4、6、8 和 10 维目标上, GrEA 的收敛性明显占优, MOEA-CCSOA 次之, IBEA 和 NSGAIII 则较差些; 在 DTLZ7 上, 目标维数为 8 和 10 时, IBEA、GrEA 和 MOEA-CCSOA 收敛性能相当, 均优于 NSGAIII, 在剩下的测试维度上, 算法 GrEA 的收敛特性占优, MOEA-CCSOA 次之. 但总的来说, MOEA-CCSOA 在大多数测试函数上都有较好的收敛效果. 对比 Spacing 和 Delta 指标箱型图可知, 这两个指标较好时, 算法的收敛效果也较好. 说明算法 CCSOA 通过提升稀疏区域被支配个体和边缘个体的进化概率, 以及通过张角

阈值控制提升剩余个体分布的均匀性, 同时降低相似个体进化的概率, 这些都帮助提高了种群的多样性, 进而提高算法的收敛效果, 达到了算法设计的目的.

#### 4.3.4 参数调整的有效性

如果不对阈值进行自适应调整, 则会导致得到的 RIS 种群规模偏大或者偏小, 这两种情况都不利于种群的进化. 图 6 给出了在张角阈值自适应调整下, 历代种群环境选择得到的典型代表集 RIS 的规模随阈值的实际变化情况, 图 6 中的直线表示期望得到的 RIS 规模, 实心点表示实际得到的 RIS 规模, 并给出了相应的统计箱型图; 从图 6 中可以看出, 在阈值自适应调整下, RIS 的实际规模均会往期望规模方向进行调整, 在期望值 100 附近来回摆动, 从统计箱型图可看出, 这些值的中位值刚好为期望规模, 四分位差维持在 20 以内, 说明阈值自适应调整策略达到了设计的目的.

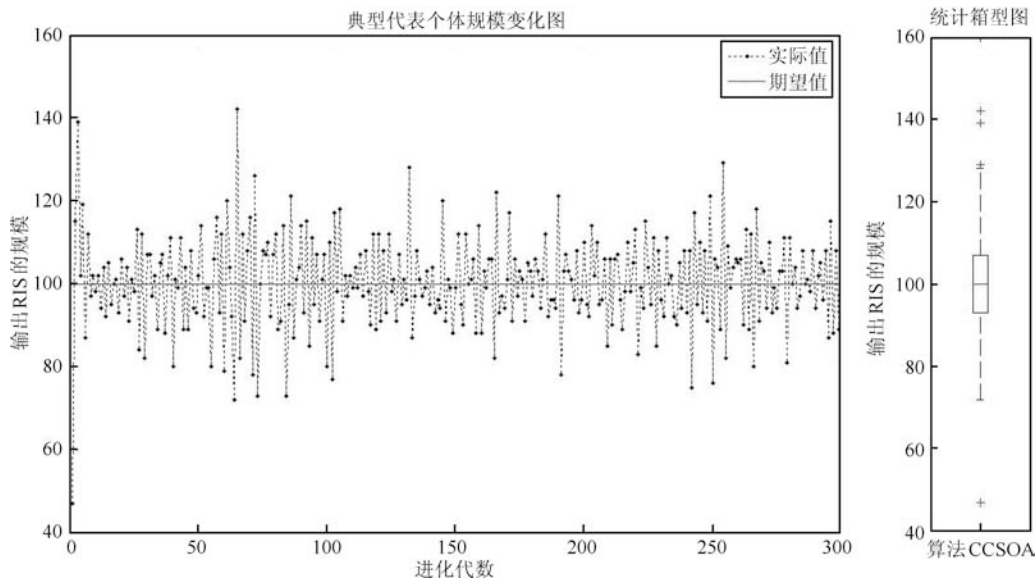


图 6 典型代表个体数量统计图和统计箱型图

Fig. 6 The statistic chart of the RIS scale and its corresponding box-plots

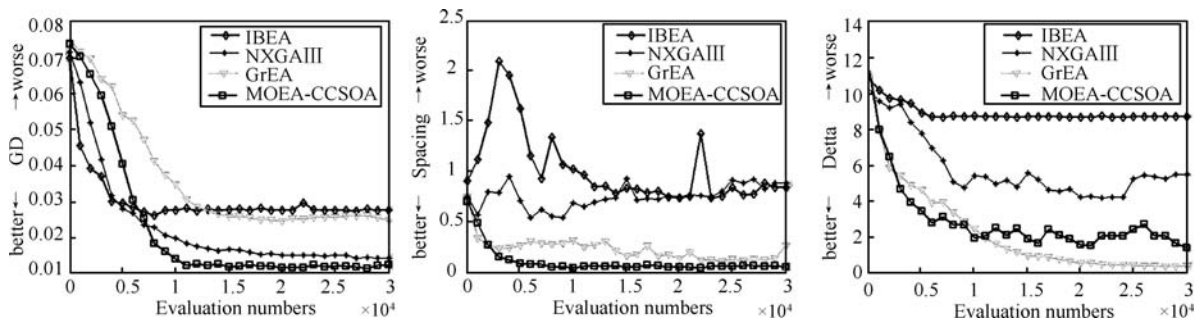


图 7 4 种对比算法在 10 维 DTLZ2 测试函数上 GD/Spacing/Delta 指标的进化轨迹图

Fig. 7 Evolutionary trajectories of GD/Spacing/Delta for the four algorithms on the ten-objective DTLZ2

表 1 指标的方差分析和多重比较结果  
Table 1 The results of variance analysis and multiple comparison for metric

测试函数	M	GD 指标多重比较						Spacing 指标多重比较						Delta 指标多重比较					
		D---A		D---B		D---C		D---A		D---B		D---C		D---A		D---B		D---C	
		P	T	P	T	P	T	P	T	P	T	P	T	P	T	P	T	P	T
DTLZ1	3	<1% <sub>00</sub>	↑	0.64	↔	0.489	↔	<1% <sub>00</sub>	↑	0.985	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑
	4	0.874	↔	0.912	↔	0.986	↔	<1% <sub>00</sub>	↑	0.377	↔	0.947	↔	0.657	↔	<1% <sub>00</sub>	↓	0.001	↓
	5	0.494	↔	0.502	↔	0.077	↔	<1% <sub>00</sub>	↑	0.043	↓	0.997	↔	0.945	↔	<1% <sub>00</sub>	↓	0.326	↔
	6	0.179	↔	0.191	↔	0.996	↔	<1% <sub>00</sub>	↑	0.192	↔	1.000	↔	0.251	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓
	8	0.001	↓	0.001	↓	0.007	↓	0.999	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	0.995	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓
	10	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	1.000	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	0.988	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	0.002	↑
DTLZ2	3	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	4	<1% <sub>00</sub>	↑	0.002	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	5	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.978	↔	0.999	↔
	6	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.241	↔	0.005	↓	0.602	↔
	8	<1% <sub>00</sub>	↑	0.097	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.042	↑	<1% <sub>00</sub>	↓
	10	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓
DTLZ3	3	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.773	↔	0.001	↑	0.002	↑	0.994	↔	0.012	↓	<1% <sub>00</sub>	↑	0.219	↔
	4	0.996	↔	<1% <sub>00</sub>	↑	0.861	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	1.000	↔	0.007	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓
	5	0.017	↓	<1% <sub>00</sub>	↑	0.518	↔	0.163	↔	0.248	↔	0.586	↔	0.148	↓	<1% <sub>00</sub>	↑	0.722	↔
	6	0.018	↓	1.000	↔	0.170	↔	0.825	↔	0.004	↓	0.372	↔	0.273	↔	<1% <sub>00</sub>	↑	0.002	↓
	8	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	0.002	↓	0.997	↔	<1% <sub>00</sub>	↓	0.200	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.998	↔
	10	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	0.317	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.998	↔
DTLZ4	3	0.001	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.013	↑	<1% <sub>00</sub>	↑
	4	<1% <sub>00</sub>	↑	0.176	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.021	↑	<1% <sub>00</sub>	↑
	5	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.009	↑
	6	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.005	↑
	8	<1% <sub>00</sub>	↑	0.198	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	10	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.110	↔	1.000	↔
DTLZ5	3	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	0.020	↓	<1% <sub>00</sub>	↓	0.075	↔	<1% <sub>00</sub>	↑
	4	0.953	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	5	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	6	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.095	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.016	↑
	8	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.012	↑	0.019	↑	0.026	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	10	<1% <sub>00</sub>	↑	0.645	↔	<1% <sub>00</sub>	↑	0.001	↑	0.638	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
DTLZ6	3	0.746	↔	0.098	↔	0.908	↔	<1% <sub>00</sub>	↓	0.001	↓	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↓	0.206	↔	<1% <sub>00</sub>	↓
	4	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	0.604	↔	0.025	↔	0.963	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.010	↑
	5	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.204	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	0.991	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.033	↑
	6	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.877	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	8	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	1.000	↔	<1% <sub>00</sub>	↑	0.095	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	10	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.073	↔	<1% <sub>00</sub>	↑	0.912	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
DTLZ7	3	0.231	↔	<1% <sub>00</sub>	↑	0.175	↔	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.149	↔	<1% <sub>00</sub>	↑
	4	0.040	↑	0.987	↔	0.001	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.005	↑	<1% <sub>00</sub>	↑
	5	1.000	↔	0.071	↔	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	6	0.665	↔	0.001	↑	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑
	8	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.005	↑
	10	<1% <sub>00</sub>	↓	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	<1% <sub>00</sub>	↑	0.006	↓
分类统计 (T)	↑	----	24	----	26	----	17	----	32	----	24	----	24	----	31	----	30	----	25
	↔	----	9	----	11	----	15	----	7	----	8	----	11	----	6	----	5	----	8
	↓	----	9	----	5	----	10	----	3	----	10	----	7	----	5	----	7	----	9

相关说明: 表中字母 A、B、C、D 分别指代算法 IBEA、GrEA、NSGAIII 和 MOEA-CCSOA; 算法 X1 与算法 X2 针对某个指标的比较中(表中表示为 X1--X2), P 为方差分析得到的显著性值, T 为比较结果: ↑ 表示 X1 显著优于 X2, ↓ 表示 X1 在对比指标上显著比 X2 要差, ↔ 表示 X1 和 X2 的对比结果无显著差别,显著性水平为 0.05.

### 4.3.5 综合分析

1) 进化轨迹分析以 10 维目标的 DTLZ2 测试函数为例, 图 7 给出了 4 种对比算法在 GD、Spacing 和 Delta 指标的进化轨迹并行图. 从 GD 指标对比图中可以看出, 当评价次数小于 4000 次时, 算法 IBEA 的进化效果最好, 但较为早熟, 评价次数大于 7000 次后, 算法 MOEA-CCSOA 在收敛性方面均优于对比算法; 从 Spacing 指标对比图中可以看出, 当评价次数大于 2500 次后, MOEA-CCSOA 均优于表现次优的 GrEA, 说明通过 CCSOA 算法对存档种群进行维护, 提高了存档种群分布的均匀性; 从 Delta 指标对比图中可以看出, 当评价次数大约小于 1200 次时, 算法 MOEA-CCSOA 在提高种群分布的宽广性方面表现最好, 之后的轨迹图客观地表明出算法 MOEA-CCSOA 在 10 维 DTLZ2 上比 GrEA 要差些, 但仍比其他两个对比算法优异.

2) 方差分析与多重比较为了避免用有限容量样本推断总体所可能造成的抽样误差, 我们对各个算法独立运行 30 次得到的对应指标样本数据进行了单因素方差分析和均值的多重比较<sup>[29]</sup>. 并在表 1 列出了对比结果. 从表中的对比数据可知, 针对收敛特性: 算法 MOEA-CCSOA 在维度为 3、5、6、和 10 的 DTLZ2 上, 在维度为 3、5、6 和 8 的 DTLZ4 上, 以及维度为 6 和 8 的 DTLZ5 上都显著优于对比算法; 针对解集均匀性: 算法 MOEA-CCSOA 在 DTLZ2、DTLZ4 和 DTLZ7 的所有测试维度上以及 4 维和 5 维的 DTLZ5 上都显著最优; 在解集宽广性方面: 算法 MOEA-CCSOA 在维度为 3 和 4 的 DTLZ2 上、在维度为 3、4、5、6 和 8 的 DTLZ4 上, 在维度为 4、5、6、8 和 10 的 DTLZ5 上, 在维度为 4、5、6、8 和 10 的 DTLZ6 上以及在维度为 4、5、6 和 8 的 DTLZ7 上都表现出最好的宽广性. 而表 1 中对 T 值进行分类统计的结果也表明: 针对 42 个测试函数, 算法 MOEA-CCSOA 与其他对比算法单独对比时, 显著较优 (↑) 的函数个数明显多于显著较差 (↓) 的数量, 具有一定的优势. 在不同的测试函数上, 各算法各有优劣, 总的来说, 与当前优秀算法 IBEA、NSGAIII 和 GrEA 进行对比, 采用了基于张角拥挤控制策略的算法 MOEA-CCSOA 在高维多目标优化问题上不仅表现出有较好的收敛效果, 而且解集分布的均匀性和宽广性也有较为明显的优势, 达到了算法设计的目的.

## 5 结语

本文通过提出张角的概念, 设计出一种基于张角的拥挤控制策略 CCSOA, 并作为环境选择算子融合到进化多目标优化算法中, 得到算法 MOEA-CCSOA. 算法 CCSOA 不仅能够加强种群边界极端

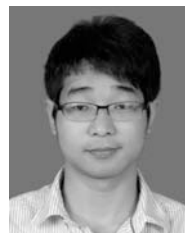
个体和稀疏区域的被支配个体的进化, 而且能够有效筛选出在高维目标空间分布较为均匀的个体进入下一代, 有效提高种群的多样性维持能力. 通过对 3、4、5、6、8 和 10 维目标在 DTLZ 系列测试函数集上进行反复的实验, 并与现有表现优异的高维多目标优化算法 IBEA、NSGAIII 和 GrEA 进行对比分析的结果表明, 本文提出的算法 CCSOA 在高维目标下仍能够较好地维持种群分布的均匀性和宽广性, 种群的多样性维持能力和算法的收敛效果均得到有效的提高.

本文提出的拥挤控制策略在高维多目标优化中取得了一定的效果, 但高维目标优化的相关工作仍有待进一步研究, 未来将主要研究以下几个方面: 1) 在约束条件下或者动态环境下改进文章的拥挤控制策略; 2) 研究高维目标下新的占优策略; 3) 应用 MOEA-CCSOA 解决某个领域的实际问题.

## References

- 1 Coello C C, Lamont G B, Van Veldhuizen D A. *Evolutionary Algorithms for Solving Multi-objective Problems* (2nd edition). New York: Springer, 2007. 3–57
- 2 Gong M Guo, Jiao L C, Yang D D, Ma W P. Research on evolutionary multi-objective optimization algorithms. *Journal of Software*, 2009, **20**(2): 271–289
- 3 Jiao Li-Cheng, Shang Rong-Hua, Ma Wen-Ping, Gong Mao-Guo, Li Yang-Yang, Liu Fang. *Multi-Objective Optimization Immune Algorithm, the Theory and Application*. Beijing: Science Press, 2010. 3–27  
(焦李成, 尚荣华, 马文萍, 公茂果, 李阳阳, 刘芳. 多目标优化免疫算法、理论和应用. 北京: 科学出版社, 2010. 3–27)
- 4 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
- 5 Yang S X, Li M Q, Liu X H, Zheng J H. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(5): 721–736
- 6 Cui Xun-Xue. *Multiobjective Evolutionary Algorithms and Their Applications*. Beijing: National Defence Industry Press, 2006.  
(崔逊学. 多目标进化算法及其应用. 北京: 国防工业出版社, 2006.)
- 7 Zheng Jin-Hua. *Multiobjective Evolutionary Algorithms and Their Application*. Beijing: Science Press, 2007.  
(郑金华. 多目标进化算法及其应用. 北京: 科学出版社, 2007.)
- 8 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
- 9 Corne D W, Jerram N R, Knowles J D, Oates M J. PESA-II: region-based selection in evolutionary multiobjective optimization. In: *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*. San Francisco, USA: Morgan Kaufmann Publishers, 2001. 283–290
- 10 Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In: *Proceedings of the 2002 International Conference on Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems*. Berlin: Springer, 2002. 95–100

- 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 Bi Xiao-Jun, Zhang Yong-Jian, Shen Ji-Hong. High-dimensional multi-objective multi-directional co-evolutionary algorithm. *Control and Decision*, 2014, **29**(10): 1737–1743  
(毕晓君, 张永建, 沈继红. 高维多目标多方向协同进化算法. 控制与决策, 2014, **29**(10): 1737–1743)
- 13 Deb K, Mohan M, Mishra S. Evaluating the  $\epsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 2005, **13**(4): 501–525
- 14 Sinha A, Saxena D K, Deb K, Tiwari A. Using objective reduction and interactive procedure to handle many-objective optimization problems. *Applied Soft Computing*, 2013, **13**(1): 415–427
- 15 Jaimes A L, Coello C A C, Chakraborty D. Objective reduction using a feature selection technique. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. New York: ACM, 2008. 673–680
- 16 Zitzler E, Kúnzli S. Indicator-based selection in multiobjective search. In: Proceedings of the 8th International Conference. Parallel Problem Solving from Nature-PPSN VIII. Birmingham, UK: Springer, 2004. 832–842
- 17 Gong Dun-Wei, Ji Xin-Fang, Sun Xiao-Yan. Solving many-objective optimization problems using set-based evolutionary algorithms. *Chinese Journal of Electronics*, 2014, **42**(1): 77–83  
(巩敦卫, 季新芳, 孙晓燕. 基于集合的高维多目标优化问题的进化算法. 电子学报, 2014, **42**(1): 77–83)
- 18 Jiao L C, Wang H D, Shang R H, Liu F. A co-evolutionary multi-objective optimization algorithm based on direction vectors. *Information Sciences*, 2013, **228**: 90–112
- 19 Reynoso-Meza G, Sanchis J, Blasco X, Martínez M. Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. *Applications of Evolutionary Computation*. Berlin: Springer-Verlag, 2010. 532–541
- 20 Tang L X, Wang X P. A Hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(1): 20–45
- 21 Zhan Wei. Research on Manifold Learning Algorithm for High-Dimensional Multi-Objective Optimization Problems [Ph. D. dissertation], China University of Geosciences, China, 2013.  
(詹炜. 求解高维多目标优化问题的流形学习算法研究 [博士学位论文], 武汉: 中国地质大学, 中国, 2013.)
- 22 Agrawal R B, Deb K, Agrawal R B. Simulated binary crossover for continuous search space. *Complex Systems*, 1995, **9**: 115–148
- 23 Hinterding R. Gaussian mutation and self-adaption for numeric genetic algorithms. In: Proceedings of the 1995 IEEE Conference on Evolutionary Computation. Perth, WA, Australia: IEEE, 1995, **1**: 384–389
- 24 Liu Y Z, Li S F. A new differential evolutionary algorithm with neighborhood search. *Information Technology Journal*, 2011, **10**: 573–578
- 25 Deb K, Thiele L, Laumanns M, Zitzler E. *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London: Springer, 2005. 105–145
- 26 Van Veldhuizen D A, Lamont G B. On measuring multi-objective evolutionary algorithm performance. In: Proceedings of the 2000 Congress on Evolutionary Computation. La Jolla, CA: IEEE, 2000, **1**: 204–211
- 27 Zhou A M, Jin Y C, Zhang Q F, Sendhoff B, Tsang E. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: Proceedings of the 2006 IEEE Congress on Evolutionary Computation. Vancouver, BC: IEEE, 2006. 892–899
- 28 Tan K C, Yang Y J, Goh C K. A distributed cooperative coevolutionary algorithm for multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2006, **10**(5): 527–549
- 29 Tamhane A C. Multiple comparisons in model I one-way ANOVA with unequal variances. *Communications in Statistics*, 1977, **6**(1): 15–32



**陈振兴** 福建师范大学数学与计算机科学学院硕士研究生. 主要研究方向为计算智能与数据挖掘.

E-mail: czx\_ky@yeah.net

(**CHEN Zhen-Xing** Master student at the College of Mathematics and Computer Science, Fujian Normal University. His research interest covers computational intelligence and data mining.)



**严宣辉** 福建师范大学数学与计算机科学学院副教授. 主要研究方向为人工智能与数据挖掘. 本文通信作者.

E-mail: yan@fjnu.edu.cn

(**YAN Xuan-Hui** Associate professor at the College of Mathematics and Computer Science, Fujian Normal University. His research interest covers artificial intelligence and data mining. Corresponding author of this paper.)



**吴坤安** 福建师范大学数学与计算机科学学院硕士研究生. 主要研究方向为计算智能与数据挖掘.

E-mail: wka2010@163.com

(**WU Kun-An** Master student at the College of Mathematics and Computer Science, Fujian Normal University. His research interest covers computational intelligence and data mining.)



**白猛** 福建师范大学数学与计算机科学学院硕士研究生. 主要研究方向为计算智能与数据挖掘.

E-mail: feilong5720@163.com

(**BAI Meng** Master student at the College of Mathematics and Computer Science, Fujian Normal University. His research interest covers computational intelligence and data mining.)