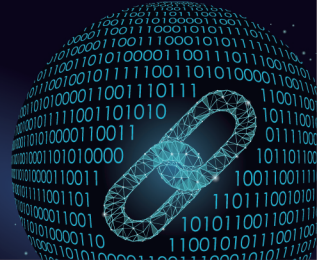




中国科技期刊卓越行动计划项目入选期刊

# 控制与决策

CONTROL AND DECISION



## 基于双重距离的多目标粒子群优化算法

慈雨, 荣淼, 彭晨

引用本文:

慈雨, 荣淼, 彭晨. 基于双重距离的多目标粒子群优化算法[J]. 控制与决策, 2024, 39(6): 1801–1809.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.2098>

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于R2指标和目标空间分解的高维多目标粒子群优化算法

R2 indicator and objective space partition based many-objective particle swarm optimizer

控制与决策. 2021, 36(9): 2085–2094 <https://doi.org/10.13195/j.kzyjc.2020.0113>

#### 具有重组学习和混合变异的动态多种群粒子群优化算法

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

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

#### 基于改进多目标优化算法的分布式数据中心负载调度

Multi-objective optimization of energy and performance management in distributed data centers

控制与决策. 2021, 36(1): 159–165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

#### 工序加工时间不确定的可重入调度问题的求解

Solution of reentrant scheduling problem with uncertain processing time

控制与决策. 2021, 36(5): 1127–1136 <https://doi.org/10.13195/j.kzyjc.2019.1036>

#### 基于树形结构无界存档的多目标粒子群算法

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

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

# 基于双重距离的多目标粒子群优化算法

慈雨, 荣淼<sup>†</sup>, 彭晨

(上海大学 机电工程与自动化学院, 上海 200444)

**摘要:** 多目标粒子群优化 (multi-objective particle swarm optimization, MOPSO) 算法在维护收敛性的同时搜索分布良好的最优解集较为费力. 为此, 提出一种基于双重距离的 MOPSO, 由种群的平均距离定义粒子的邻域空间, 邻域粒子数为粒子的等级, 数量越多, 粒子的等级越大. 当等级相同时, 算法结合粒子的拥挤距离选择最优粒子, 并更新外部归档集. 此外, 算法结合粒子的变异行为避免陷入局部最优. 在对比实验中, 该算法在收敛性和多样性上可取得较优结果. 最后, 将该算法应用到电力系统的环境/经济调度模型 (environmental/economic dispatch, EED), 也可获得性能较好的解集.

**关键词:** 平均距离; 拥挤距离; 邻域信息; 多目标粒子群优化; 多目标问题; 环境/经济调度

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.2098

引用格式: 慈雨, 荣淼, 彭晨. 基于双重距离的多目标粒子群优化算法[J]. 控制与决策, 2024, 39(6): 1801-1809.

## Multi-objective particle swarm optimization algorithm based on dual distances

CI Yu, RONG Miao<sup>†</sup>, PENG Chen

(School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China)

**Abstract:** Multi-objective particle swarm optimization (MOPSO) algorithm is laborious to choose well-distributed optimal solutions while maintaining convergence. In this paper, an improved MOPSO based on dual distance is proposed to alleviate the above issues. It defines the neighborhood of a particle according to the average distance of the population, and thus the number of neighbors is the level of the particle. The more there are neighbors, the higher the level is. The average distance of two particles is equal so that with combining the crowding distance of swarm, the optimal particle is chosen and the extend archive is updated. In addition, this algorithm combines the mutation behavior of particles to avoid local space. In the experiment, the proposed algorithm is compared with several state-of-the-art algorithms, demonstrating that this algorithm achieves good performance in distribution and diversity of swarm. Finally, the algorithm is applied to the environmental/economic scheduling model in power systems, which can obtain well-distributed optimal solutions.

**Keywords:** average distance; crowding distance; neighborhood information; MOPSO; MOP; environmental/economic dispatch

## 0 引言

多目标问题 (multi-objective problem, MOP) 一直受到众多学者的关注. 为了解决该问题, 基于生物行为的多目标优化算法涌现, 如人工蜂群算法<sup>[1]</sup>、萤火虫算法<sup>[2]</sup>、灰狼算法<sup>[3]</sup>等. MOPSO 是一种启发式算法, 由于其具有收敛速度快、参数灵活等优点, 被广泛应用于 MOPs. 应用过程中学者通常会考虑两个方面: 一是搜索未知空间, 二是平衡种群的多样性和

收敛性. Coello 等<sup>[4]</sup> 借助网格空间和轮盘赌法, 将 MOPSO 应用于求解 MOPs; Lin 等<sup>[5]</sup> 提出了新颖的多目标粒子群优化方法 (NMOPSO), 将适应度评估与更新粒子速度相结合, 以搜索稀疏区域; 冯茜等<sup>[6]</sup> 采用基于双档案的粒子群, 结合区域划分策略平衡种群的多样性; Dai 等<sup>[7]</sup> 将目标空间分解为多个子区域进行搜索, 得到较好的 Pareto 前沿 (pareto optimal front, PF), 同时利用拥挤距离保持种群的多样性.

收稿日期: 2022-12-04; 录用日期: 2023-04-15.

基金项目: 国家自然科学基金项目 (62103255, 61833011).

责任编委: 巩敦卫.

<sup>†</sup>通讯作者. E-mail: mrong707@shu.edu.cn.

\*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

拥挤距离来源于NSGA-II (non-dominated sorting genetic algorithm II)<sup>[8]</sup>,并得到了广泛的应用<sup>[9-12]</sup>. Li等<sup>[11]</sup>提出了一种知识引导的多目标粒子群优化融合学习策略(KGMOPSO),该策略使用一对称为最大拥挤距离和最小拥挤距离、尽可能保留分布良好的pareto最优解集(pareto optimal set, PS),但并未利用邻域信息;Li<sup>[12]</sup>采用随机模拟PSO,将外部档案中的粒子按照拥挤距离值降序排序,选择前10%的粒子形成内部最优存档,进而保持PS的多样性.

以增加种群的多样性为目标,算法优先选取拥挤距离最大的粒子作为全局最优粒子,进而探索空间的稀疏区域.多数情况下,MOPSO会结合拥挤距离避免粒子过度聚集,该过程产生的复杂度是能接受的.当大部分粒子被困在局部最优区域时,尽管可以去除最拥挤的粒子,但也会保留一些拥挤程度大的粒子,同时整个过程忽略了大量的邻居信息.受文献<sup>[13]</sup>的启发,本文提出一种具有双重距离的粒子群算法(MOPSO with average distance and crowding distance),称为ACMOPSO,主要内容如下:

1) ACMOPSO以种群的平均距离为基准,为粒子创建邻域空间,进而根据邻域粒子数量划分等级.等级越高的粒子其拥挤程度越高,等级越低的粒子其邻域越稀疏.仅仅在等级相同时算法才结合修正后的拥挤距离,在特定的粒子群体中选择或者删除粒子,优化了拥挤距离的使用范围.

2) 为验证ACMOPSO的综合性能,将该算法同典型的多目标优化算法(multi-objective evolutionary algorithms, MOEAs)作比较.本文也考虑了环境/经济调度问题模型(EED),以求证该方法与MOPSO和MOPSOCD(MOPSO with crowding distance)的性能关系.

## 1 相关内容

### 1.1 MOP

多目标优化问题由多个目标组成,并且目标之间往往存在冲突.假设MOP含有 $M$ 个目标函数,不失通用性可表示为

$$\begin{aligned} \min f(x) &= [f_1(x), f_2(x), \dots, f_M(x)]; \\ \text{s.t. } x &\in R^n. \end{aligned} \quad (1)$$

其中: $x=(x_1, \dots, x_k, \dots, x_n)$ ,  $R^n$ 为 $n$ 维决策空间,  $f_M(x)$ 为第 $M$ 个目标函数.在决策空间的第 $k$ 个维度上,  $x_k$ 为连续取值的,  $x_k^{\min}$ 、 $x_k^{\max}$ 分别为最小值和最大值.

### 1.2 MOPSO

在MOPSO中,种群是粒子的集合,其中每个粒子均代表MOP的解.假设种群的大小为 $N$ ,种群在第 $t+1$ 次迭代中,第 $i$  ( $i=1, 2, \dots, N$ )个粒子的速度计算公式和位置更新公式如下:

$$\begin{cases} v_i(t+1) = \omega v_i(t) + c_1 r_1 (x_{i,pbest} - x_i(t)) + \\ \quad c_2 r_2 (x_{gbest} - x_i(t)), \\ x_i(t+1) = x_i(t) + v_i(t+1). \end{cases} \quad (2)$$

其中: $v_i(t)$ 、 $v_i(t+1)$ 为粒子的速度,  $x_i(t)$ 、 $x_i(t+1)$ 为粒子的位置,将粒子 $i$ 的个体最佳位置记为 $x_{i,pbest}$ ,  $x_{gbest}$ 为种群最佳位置,  $r_1$ 、 $r_2$ 为 $[0,1]$ 的随机数,  $c_1$ 、 $c_2$ 为加速常数.

$\omega$ 主要体现了粒子的惯性运动.在算法搜索早期, $\omega$ 应取较大值,以增强算法的全局搜索能力;到后期,算法应专注于局部空间, $\omega$ 应取较小值.基于MOPSO<sup>[4]</sup>的参数,本文将 $\omega$ 的变化范围定为 $[0.2, 0.6]$ ,递减过程如下:

$$\omega(t+1) = \omega(t)\omega_d^t. \quad (3)$$

其中: $\omega(t)$ 、 $\omega(t+1)$ 分别为第 $t$ 次、第 $t+1$ 次迭代的惯性权重,  $\omega_d$ 为权重的递减率.设 $\omega(0) = 0.6$ ,当 $\omega_d$ 变化时得到的几种惯性权重的变化曲线如图1所示.当 $\omega_d = 0.99$ 时,惯性权重在整个迭代过程中起作用.

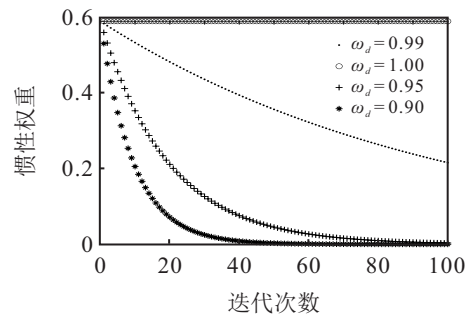


图1 惯性权重变化

MOPSO具有良好的收敛速度,但易陷入局部最优.因此,本文算法也加入变异概率(probability of mutation, PM),以控制粒子的变异行为.本文采用非线性函数<sup>[4]</sup>描述动态衰减过程,即

$$PM = \left(1 - \frac{t-1}{T-1}\right)^{\frac{1}{pm}}. \quad (4)$$

其中: $t$ 为已迭代次数; $T$ 为最大迭代次数; $pm$ 为变异算子,根据MOPSO的变异率,其值设为 $0.1$ <sup>[4]</sup>.

## 2 本文算法

### 2.1 ACMOPSO

ACMOPSO的总体过程主要有初始化、计算平均距离、环境选择、解集维护等,具体步骤如下.

step 1: 在决策空间中初始化输入种群 pop 及外部档案 rep, 指标均设为 0; 初始化参数主要包括: 最大迭代次数  $T$ 、种群最大粒子数  $n_{Rep}$ 、变异算子 pm、惯性权重的递减率  $\omega_d$ 、种群全局最佳粒子  $x_{gbest}$ 、平均距离的缩放因子  $Mu$ .

step 2: 将 pareto 最优解添加到 rep.

step 3: 判断当前迭代次数  $t$  是否大于  $T$ .

step 4: 得出粒子的拥挤距离并计算种群的平均距离.

step 5: 参照环境选择策略生成  $x_{gbest}$ .

step 6: 根据式(2)更新粒子群的速度和位置.

step 7: 在  $[0, 1]$  中随机生成 rand, 当  $rand \leq PM$  时, 粒子发生位置突变.

step 8: 挑选粒子历史最优解并添加到 rep 中.

step 9: 如果 rep 中的粒子数量超过上限  $n_{Rep}$ , 则可采取解集维护策略.

step 10: 根据式(3)计算  $\omega$ .

step 11: 根据式(4)计算 PM.

step 12:  $t \leftarrow t + 1$ , 并转到 step 3.

### 2.2 粒子的拥挤距离

目标值排序后, 第  $i$  个粒子的位置为  $x_i$ , 其拥挤距离  $d_i$  按下式计算:

$$d_i = \sum_{m=1}^M |f_m(x_{i+1}) - f_m(x_{i-1})|. \quad (5)$$

图2举例说明了双目标函数中拥挤距离的计算, 其中5个粒子的位置分别记为  $x_1 \sim x_5$ . 当第  $i$  个粒子为端点粒子时, 将该粒子的位置  $x_i$  与其相邻粒子的位置  $x_{neighbor}$  在每个目标上距离的倍数和作为端点的拥挤距离. 因此, 在图2中, 1号、5号粒子均是端点粒子, 对应的相邻粒子分别是2号、4号粒子. 端点粒子  $i$  的拥挤距离可通过下式计算:

$$d_i = 2 \cdot \sum_{m=1}^M |f_m(x_i) - f_m(x_{neighbor})|. \quad (6)$$

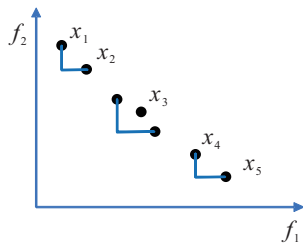


图2 拥挤距离的计算

图3为双重距离的计算示意图, 其中红色虚线框表示种群的平均距离, 蓝色实线表示个体的拥挤距离.

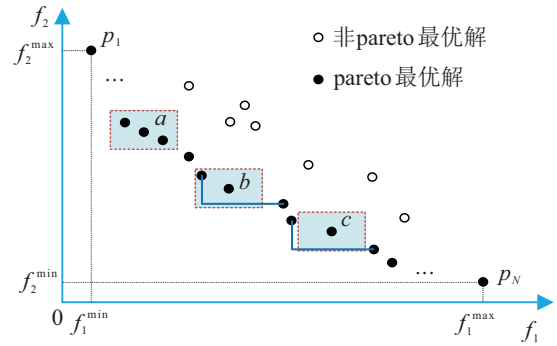


图3 双重距离计算

以往根据拥挤距离的选择机制,  $b$  与  $c$  之间存在3种关系, 描述如下:

1)  $d_b > d_c$ , 表明  $b$  周围的粒子更稀疏, 优先被考虑为全局粒子, 而维护解集时  $c$  优先被删除;

2)  $d_b < d_c$ , 表明  $b$  周围的粒子比  $c$  更密集, 算法的处理过程与 1) 相反;

3)  $d_b = d_c$ , 表明它们的密集度相近, 在环境选择和维护解集中均被随机处理.

图3中也可能存在一种情况:  $b$  周围粒子的密度更高,  $c$  周围的粒子分布更均匀. 若先在  $b$  上进行具体的修正, 则在解集的分布上可能会得到更好的结果. 为了更全面地考虑问题, 本文引入种群的平均距离.

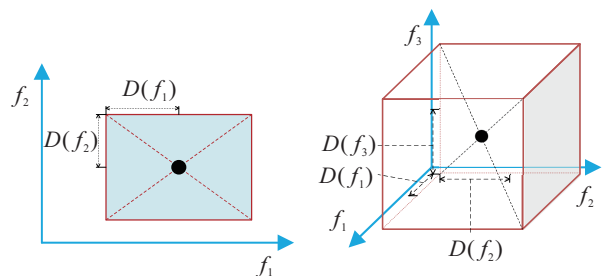
### 2.3 种群的平均距离

在上述讨论的基础上, 本小节给出平均距离  $D(f)$  的计算方法, 作为粒子群的等级分层准则. 粒子群在第  $m$  个目标上的平均距离可由下式计算:

$$D(f_m) = \frac{f_m^{\max} - f_m^{\min}}{2N}. \quad (7)$$

其中:  $f_m^{\max}$  和  $f_m^{\min}$  分别为第  $m$  个目标上的最大值和最小值, 种群的大小为  $N$ , 为了简化平均距离的实现,  $f_m^{\max} - f_m^{\min}$  取半后参与运算.

在图3中, 为了更好地抽象种群的平均距离, 将其用虚线矩形框表示, 并分布在每个粒子周围, 其第  $m$  个轴上的边长为  $2 \cdot D(f_m)$ , 对应于第  $m$  个目标的平均距离. 双目标和多目标 MOPs 中, 在种群的平均距离作用下, 粒子的邻域空间如图4所示.



(a) 粒子的二维邻域空间 (b) 粒子的三维邻域空间

图4 粒子的邻域空间大小

在双目标问题中,粒子的邻域为矩形空间.在具有3个目标的目标MOPs中,邻域为以粒子为中心的立方体.

在每次迭代中,算法基于目标函数计算种群的平均距离,随之确定粒子的邻域空间.较小的空间会降低挖掘有用信息的能力,而较大的范围会增加计算复杂度.因此,有必要确定一个可接受的邻域大小,以使得粒子的等级更好地起作用.鉴于上述考虑,算法引入一个缩放因子 $Mu$ ,缩放后的计算式平均距离为 $Mu \cdot D(f)$ .

如图5(a)所示,当 $Mu$ 变大时,矩形框被放大,粒子的邻域粒子数量增加. $a$ 、 $b$ 、 $c$ 周围的邻域粒子数均为4.在判断每个粒子的拥挤程度时,同一等级的粒子过多,没有办法区别出拥挤程度相似的粒子,却增加了算法的计算量.如图5(b)所示,当 $Mu$ 取很小值时,矩形框会缩小.极端的情况是,矩形框中仅有粒子本身,算法只依靠每个粒子的拥挤距离从外部存档中选择或删除粒子.显然,粒子的邻域信息是无效的,同时也增加了算法的计算量.

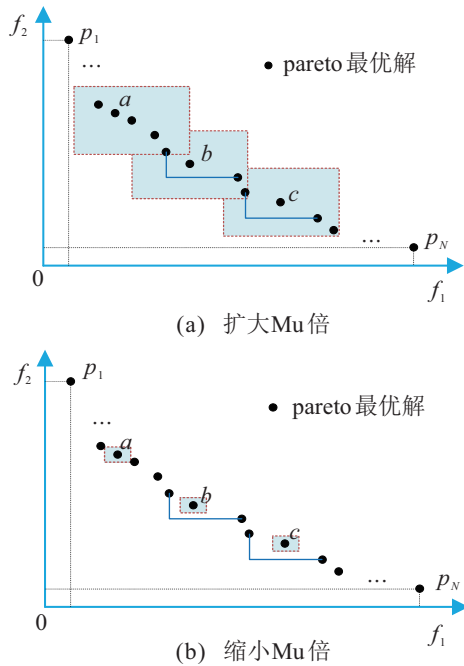


图5 平均距离的缩放

为了给粒子添加新的属性,代表其邻域粒子数量,算法给出了由种群平均距离计算邻域粒子数的过程.该过程的输入量有外部档案 $rep$ 、缩放因子 $Mu$ 、目标总数 $M$ ,具体步骤如下.

step 1: 统计 $rep$ 的粒子数 $N$ .

step 2: 根据第 $m$ 个目标值,将 $rep$ 的粒子升序排序,得到索引数组 $rank[N]$ .

step 3: 第 $m$ 个目标的最大值和最小值分别记作

$f_m^{\max}$ 、 $f_m^{\min}$ .

step 4: 计算种群在第 $m$ 个目标中平均距离缩放值 $Mu \cdot D(f_m)$ .

step 5: 依据索引数组 $rank[i]$ 遍历 $rep$ ,运行条件为 $i \leq N$ .

step 6: 计算相邻粒子在第 $m$ 个目标的差值的绝对值,记为 $f_{abs}$ 的第 $m$ 维度值.

step 7: 当 $f_{abs}$ 在第 $m$ 维度值小于 $Mu \cdot D(f_m)$ 时,记为两个粒子在第 $m$ 个目标中互为邻域粒子.

step 8:  $i \leftarrow i + 1$ ,并转到step 5.

step 9: 当 $m \leq M$ 时, $m \leftarrow m + 1$ ,并转到step 2;否则,转到step 10.

step 10: 遍历 $rep$ ,统计粒子在 $M$ 维邻域空间中本身之外的粒子数.

以粒子的邻域粒子数为标准,通过ACMOPSO对粒子进行等级划分.例如,在图3中粒子 $a$ 和 $b$ 的等级分别为2和1,而 $c$ 的等级为0.

## 2.4 环境选择和解集维护策略

全局最优粒子不仅要引领种群探索未知区域,而且使PF的分布性更佳,因此具备两个条件:pareto最优解和邻域粒子数量较少.将所有粒子依据邻域粒子数排序后,等级最低的粒子优先被考虑作为全局最优解 $x_{gbest}$ .如果全局最优解不唯一,则算法选择拥挤距离最大的粒子.

解集维护策略的过程与环境选择策略比较相似:粒子的等级越高,表示该粒子的密度越大.因此,该算法首先根据外部档案 $rep$ 与上限值 $nRep$ 的差值确定循环删除的次数.在每次删除的过程中,等级最高并且拥挤距离最小的粒子最先被删除.

## 2.5 算法复杂度

算法的计算复杂度主要来源于邻域粒子数的统计、个体拥挤距离的计算、全局最优粒子的选取以及外部档案的维护等部分.

假设有 $M$ 个目标函数和 $N$ 个粒子.在计算种群的平均距离时,算法的计算代价主要是依靠目标值对粒子两两比较排序的过程,计算复杂度为 $O(MN^2)$ .在有序的粒子群中,算法统计每个粒子的邻域粒子数所消耗的代价为 $O(MN)$ .在选取种群的全局最优粒子和维护外部档案中,算法的计算代价主要是在重组的种群中得出粒子的拥挤距离.假设重组的种群大小为 $K$ ,则该部分的计算复杂度为 $O(MK \log K)^{[10]}$ .在第2.3节中,平均距离的缩放因子能影响具有相同等级的粒子数量,则 $K < N$ .因此,ACMOPSO的整

体计算复杂度为 $O(MN^2)$ .

### 3 实验准备

#### 3.1 基准函数和性能指标

本节研究ACMOPSO在ZDT<sup>[14]</sup>和DTLZ<sup>[15]</sup>上的性能. 由于ZDT5为离散问题, 本文不考虑该问题. 它们的PF表现出各种各样的特征, 如凸性、凹性等.

本文采用两种指标评估算法: 一是从收敛性和分布性综合评价的反世代距离(inverted generational distance, IGD<sup>[16]</sup>), 其值越小表示综合性能越好; 二是仅评价粒子之间差异程度的纯多样性度量(pure diversity, PD<sup>[17]</sup>), 其值越大表示算法的多样性越优.

#### 3.2 对比算法及参数设置

本文考虑了包括拥挤距离、帕累托占优或分解等机制的两组对比算法: 一组是基于MOPSO的算法, 即MOPSO<sup>[4]</sup>、MOPSOCD<sup>[10]</sup>、MMOPSO(MOPSO with multiple search strategies)<sup>[18]</sup>; 另一组是典型MOEAs, 如NSGA-II<sup>[8]</sup>、rNSGA-II(reference solution-based NSGA-II)<sup>[19]</sup>以及MOEAD(multi-objective evolutionary algorithm based on decomposition)<sup>[20]</sup>. 6种比较算法的源代码均由PlatEMO<sup>[21]</sup>提供, 主要参数及取值如表1所示, 表1中 $n$ 为变量维数.

表1 对比算法的实验参数

算法	参数设置
MOPSO	$\omega = 0.4, \text{mutrate} = 0.5$
MOPSOCD	$\omega = 0.4, \text{PMUT} = 0.5$
MMOPSO	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.0], p_c = 0.9, p_m = 1/n, \eta_c = \eta_m = 20, \delta = 0.9$
NSGA-II	$p_c = 0.9, p_m = 1/n, \eta_c = \eta_m = 20$
rNSGA-II	$p_c = 0.9, p_m = 1/n, \eta_c = \eta_m = 20, \delta = 0.1$
MOEAD	$\text{CR} = 1, F = 0.5, \eta = 20, P_m = 1/n$

ACMOPSO的参数如下: 迭代次数为100, 惯性权重 $\omega$ 初始化为0.6, 其衰减速率 $\omega_d$ 为0.99.  $c_1$ 、 $c_2$ 分别为1.0、1.5.  $\text{Mu}$ 由后续实验确定. 所有算法的种群大小均为100.

基准问题的变量长度均参考PlatEMO平台的推荐值, 设置如下: ZDT1~ZDT3为30, ZDT4、ZDT6均为10, DTLZ2~DTLZ6为12, DTLZ1为7, DTLZ7为22. 本实验环境为Matlab r2018b, 所有算法独立运行20次, 并统计指标的平均值(标准差). 在显著性水平为0.05的情况下, 采用Wilcoxon秩和检验来检验结果. 在问题的结果上, ‘+’ ‘-’ ‘=’分别表示ACMOPSO显著优于、显著劣于或等同于其他算法.

## 4 结果分析

### 4.1 Mu分析

如第2.3节所述, 在ACMOPSO中引入 $\text{Mu}$ 来缩放粒子的邻域空间, 进而影响算法的寻优效果. 该算法在 $\text{Mu} \in \{0.01, 0.1, 0.5, 1, 2, 3, 4, 5\}$ 下独立运行20次的IGD平均值如表2所示.

表2 不同Mu时, ACMOPSO的IGD平均值(标准差)

problem	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Mu=0.01	4.783 5e-3 (1.60e-4)	5.212 2e-3 (8.92e-4)	6.283 0e-3 (4.72e-4)	9.151 2e-1 (3.75e-1)	3.810 6e-3 (9.94e-4)
Mu=0.1	4.790 0e-3 (2.39e-4)	5.140 4e-3 (3.99e-4)	6.214 6e-3 (4.08e-4)	8.800 3e-1 (2.67e-1)	3.779 9e-3 (1.61e-3)
Mu=0.5	4.382 7e-3 (1.34e-4)	4.577 9e-3 (2.55e-4)	5.524 1e-3 (3.68e-4)	8.060 3e-1 (2.81e-1)	3.198 8e-3 (5.65e-4)
Mu=1.0	<b>4.142 4e-3</b> <b>(1.19e-4)</b>	4.091 2e-3 (2.70e-4)	5.367 1e-3 (9.77e-5)	6.106 8e-1 (7.55e-2)	2.767 4e-3 (2.46e-4)
Mu=2.0	4.149 5e-3 (5.99e-5)	<b>4.066 3e-3</b> <b>(9.88e-5)</b>	<b>5.224 1e-3</b> <b>(1.17e-4)</b>	<b>6.100 4e-1</b> <b>(7.02e-2)</b>	<b>2.724 8e-3</b> <b>(3.62e-4)</b>
Mu=3.0	5.059 5e-3 (1.95e-4)	4.887 9e-3 (2.03e-4)	6.635 7e-3 (2.30e-4)	8.828 8e-1 (6.19e-2)	3.906 7e-3 (3.95e-4)
Mu=4.0	6.129 2e-3 (2.11e-4)	6.202 4e-3 (1.84e-4)	7.673 6e-3 (1.47e-4)	1.215 5e+0 (1.75e-1)	5.260 3e-3 (6.92e-4)
Mu=5.0	7.405 0e-3 (2.67e-4)	7.067 3e-3 (2.34e-4)	9.083 7e-3 (7.48e-4)	1.563 5e+0 (5.56e-2)	5.780 6e-3 (9.86e-4)

从表2的比较结果中可以看出: 当 $\text{Mu}$ 从2增加到5时, IGD的均值和标准差值逐渐减小, 说明整体性能逐渐恶化; 当 $\text{Mu} = 5$ 时, 算法的IGD结果最大, 性能最差; 当 $\text{Mu}$ 设置在0.01~1时, 随着 $\text{Mu}$ 的增大, IGD的值逐渐增大, 说明算法的性能越来越好; 当 $\text{Mu} = 0.01$ 时, 算法的IGD值较高, 说明综合性能较差. 这说明当 $\text{Mu}$ 取较小值时, 种群的平均距离并不会提高算法的性能. 虽然当 $\text{Mu} = 0.01$ 和 $\text{Mu} = 0.1$ 时, ACMOPSO的IGD值变小, 但差异不显著. 可以推断: 当 $\text{Mu}$ 小于0.01时, ACMOPSO的IGD不会有显著的下降趋势; 当 $\text{Mu}$ 非常大或非常小时, 种群的平均距离对拥挤距离都不能起到很好的影响.

实验结果表明: 当 $\text{Mu} \in \{0.01, 0.1, 0.5, 1, 2, 3, 4, 5\}$ 时, ACMOPSO的IGD均值呈现先减小后增大的变化; 当 $\text{Mu} = 2$ 时, ACMOPSO的结果在IGD上表现最好. 因此, 在后续的实验中将 $\text{Mu}$ 设为2.

### 4.2 与典型MOPSO比较

表3给出了本文算法及典型MOPSO得到的IGD的均值和标准差, 表4给出了各算法得到的PD的均值和标准差. 最优的结果用粗体标记. 对比算法在ZDT6上得到的PF如图6(a)~图6(c)所示, 图6(g)为ACMOPSO得到的近似PF.

表3 ACMOPSO与对比算法的IGD性能

problem	MOPSO	MOPSOCD	MMOPSO	NSGA-II	rNSGA-II	MOEAD	ACDMOPSO
ZDT1	9.925 1e-01 (2.21e-01)–	1.181 3e-02 (3.42e-02)–	4.877 2e-03 (2.12e-04)=	1.215 6e-02 (1.54e-03)–	3.003 4e-01 (1.30e-02)–	1.542 0e-01 (5.58e-02)–	<b>4.149 5e-03</b> <b>(5.98e-05)</b>
ZDT2	1.766 0e+00 (4.01e-01)–	4.405 8e-03 (4.27e-04)=	3.111 6e-02 (1.15e-01)–	3.577 5e-02 (5.77e-02)–	3.350 8e-01 (1.76e-02)–	5.257 1e-01 (7.86e-02)–	<b>4.066 3e-03</b> <b>(9.88e-05)</b>
ZDT3	9.828 2e-01 (2.56e-01)–	2.474 3e-02 (3.86e-02)–	5.431 2e-03 (1.18e-04)=	1.400 8e-02 (1.00e-02)–	4.005 2e-01 (4.88e-02)–	1.447 2e-01 (4.55e-02)–	<b>5.224 1e-03</b> <b>(1.16e-04)</b>
ZDT4	1.733 1e+01 (4.43e+00)–	1.443 7e+01 (6.68e+00)–	2.224 6e+00 (1.95e+00)–	2.433 2e-01 (1.53e-01)+	3.375 9e-01 (1.03e-01)+	4.819 8e-01 (1.59e-01)+	<b>6.100 4e-01</b> <b>(7.02e-02)</b>
ZDT6	2.137 7e-01 (3.32e-01)–	4.309 3e-03 (1.33e-03)–	4.281 6e-03 (2.02e-04)–	5.843 9e-02 (2.86e-02)–	2.679 2e-01 (3.35e-02)–	8.281 2e-02 (3.29e-02)–	<b>2.724 8e-03</b> <b>(3.62e-04)</b>
DTLZ1	1.257 4e+01 (4.93e+00)–	1.786 3e+01 (4.36e+00)–	2.209 0e+00 (2.20e+00)=	<b>1.195 7e-01</b> <b>(1.25e-01)+</b>	3.318 7e-01 (4.07e-01)+	2.513 3e-01 (3.66e-01)+	2.287 9e+00 (6.25e-01)
DTLZ2	1.037 2e-01 (1.82e-02)–	9.934 8e-02 (8.02e-03)–	7.267 3e-02 (3.28e-03)=	6.978 7e-02 (3.15e-03)–	5.072 4e-01 (1.32e-02)–	5.513 9e-02 (3.72e-04)=	<b>5.253 9e-02</b> <b>(2.39e-03)</b>
DTLZ3	1.813 8e+02 (3.59e+01)–	1.086 4e+02 (3.86e+01)=	7.606 5e+01 (3.52e+01)+	<b>7.839 9e+00</b> <b>(4.05e+00)+</b>	8.067 7e+00 (4.59e+00)+	1.733 6e+01 (1.88e+01)+	1.036 9e+02 (1.01e+01)
DTLZ4	3.314 7e-01 (6.86e-02)–	2.951 9e-01 (4.16e-02)–	7.3303 e-02 (1.74e-03)=	1.633 6e-01 (1.99e-01)–	4.597 7e-01 (1.31e-01)–	6.060 3e-01 (3.47e-01)–	<b>7.316 2e-02</b> <b>(3.63e-03)</b>
DTLZ5	1.432 4e-02 (3.46e-03)–	2.193 8e-02 (5.90e-03)–	6.608 2e-03 (3.82e-04)–	6.103 3e-03 (3.30e-04)–	3.974 0e-01 (1.61e-02)–	3.247 2e-02 (3.87e-04)–	<b>4.279 3e-03</b> <b>(1.95e-04)</b>
DTLZ6	3.193 6e+00 (8.92e-01)–	4.530 8e-02 (1.25e-01)–	6.870 7e-03 (6.58e-04)–	6.047 7e-03 (3.95e-0 4)–	2.307 4e-01 (4.14e-02)–	3.177 6e-02 (1.69e-03)–	<b>4.357 4e-03</b> <b>(2.32e-04)</b>
DTLZ7	4.368 6e+00 (8.39e-01)–	8.867 9e-02 (8.35e-03)=	2.818 4e-01 (2.93e-01)–	9.852 4e-02 (1.09e-02)–	1.289 6e+00 (2.18e-01)–	1.500 6e-01 (6.59e-03)–	<b>7.154 9e-02</b> <b>(3.67e-03)</b>
+/-/=	0/12/0	0/9/3	1/6/5	3/9/0	3/9/0	3/8/1	

表4 ACMOPSO与对比算法的PD性能

problem	MOPSO	MOPSOCD	MMOPSO	NSGA-II	rNSGA-II	MOeAD	ACMOPSO
ZDT1	1.842 3e+03 (3.37e+02)+	1.838 0e+03 (3.60e+02)=	<b>1.843 4e+03</b> <b>(3.03e+02)+</b>	1.668 3e+03 (2.55e+02)=	5.797 2e+01 (3.10e+01)–	6.536 2e+02 (1.54e+02)–	1.824 3e+03 (1.27e+02)
ZDT2	2.995 5e+02 (5.31e+02)–	<b>1.497 4e+03</b> <b>(3.40e+02)+</b>	1.632 0e+03 (4.66e+02)=	1.412 0e+03 (3.53e+02)–	5.500 6e+01 (3.73e+01)–	4.739 2e+01 (5.31e+01)–	1.718 2e+03 (1.08e+02)
ZDT3	1.468 5e+03 (3.21e+02)–	1.741 2e+03 (2.64e+02)=	1.642 6e+03 (2.86e+02)–	1.698 0e+03 (3.39e+02)=	1.296 4e+02 (6.46e+01)–	7.691 7e+02 (2.30e+02)–	<b>1.778 5e+03</b> <b>(8.67e+01)</b>
ZDT4	3.370 1e+02 (6.73e+02)–	1.304 1e+03 (1.02e+03)–	1.133 6e+03 (5.56e+02)–	8.551 4e+02 (2.72e+02)–	1.031 3e+02 (7.57e+01)–	8.325 4e+02 (2.44e+02)–	<b>1.622 3e+03</b> <b>(1.56e+02)</b>
ZDT6	1.406 8e+03 (5.50e+02)–	<b>1.632 7e+03</b> <b>(3.18e+02)=</b>	1.371 1e+03 (1.79e+02)–	1.209 9e+03 (2.08e+02)–	2.180 9e+02 (3.19e+01)–	8.063 9e+02 (1.85e+02)–	1.622 8e+03 (6.67e+01)
DTLZ1	<b>1.821 2e+06</b> <b>(6.60e+05)+</b>	1.705 5e+06 (7.27e+05)=	3.959 2e+05 (4.04e+05)–	9.025 9e+04 (3.15e+04)–	1.363 8e+05 (1.05e+05)–	1.332 7e+05 (9.69e+04)–	1.741 9e+06 (1.45e+05)
DTLZ2	<b>2.161 5e+05</b> <b>(1.38e+04)=</b>	1.980 1e+05 (1.63e+04)–	1.843 3e+05 (1.69e+04)–	1.959 2e+05 (1.88e+04)=	8.186 1e+03 (2.48e+03)–	1.433 7e+05 (5.11e+03)–	2.148 4e+05 (1.16e+04)
DTLZ3	9.061 2e+06 (5.89e+06)–	4.857 3e+06 (3.40e+06)–	9.363 2e+06 (5.05e+06)=	5.623 8e+05 (6.21e+05)–	3.946 3e+05 (4.21e+05)–	2.914 3e+06 (3.28e+06)–	<b>1.007 8e+07</b> <b>(4.34e+06)</b>
DTLZ4	5.580 6e+03 (6.64e+03)–	7.616 3e+03 (9.30e+03)–	<b>1.848 5e+05</b> <b>(1.82e+04)+</b>	1.475 3e+05 (7.77e+04)+	3.215 2e+04 (1.77e+04)–	2.904 8e+04 (5.98e+04)–	1.342 2e+05 (1.42e+04)
DTLZ5	6.936 2e+04 (8.24e+03)–	<b>9.457 9e+04</b> <b>(1.93e+04)+</b>	7.304 3e+04 (1.05e+04)–	7.007 9e+04 (1.04e+04)–	3.963 6e+03 (1.37e+03)–	4.462 6e+04 (2.89e+03)–	8.379 9e+04 (2.83e+03)
DTLZ6	6.736 7e+04 (3.03e+04)–	8.093 1e+04 (1.48e+04)=	7.002 5e+04 (8.66e+03)–	7.372 6e+04 (6.11e+03)–	2.625 3e+04 (6.11e+03)–	4.607 3e+04 (7.03e+03)–	<b>9.662 3e+04</b> <b>(4.58e+03)</b>
DTLZ7	1.456 8e+05 (8.49e+04)–	2.069 9e+05 (1.43e+04)=	1.460 4e+05 (5.49e+04)–	1.899 8e+05 (1.94e+04)=	6.016 5e+03 (1.54e+03)–	1.052 7e+05 (1.14e+04)–	<b>2.084 7e+05</b> <b>(1.78e+04)</b>
+/-/=	2/9/1	2/4/6	2/8/2	1/7/4	0/12/0	0/12/0	

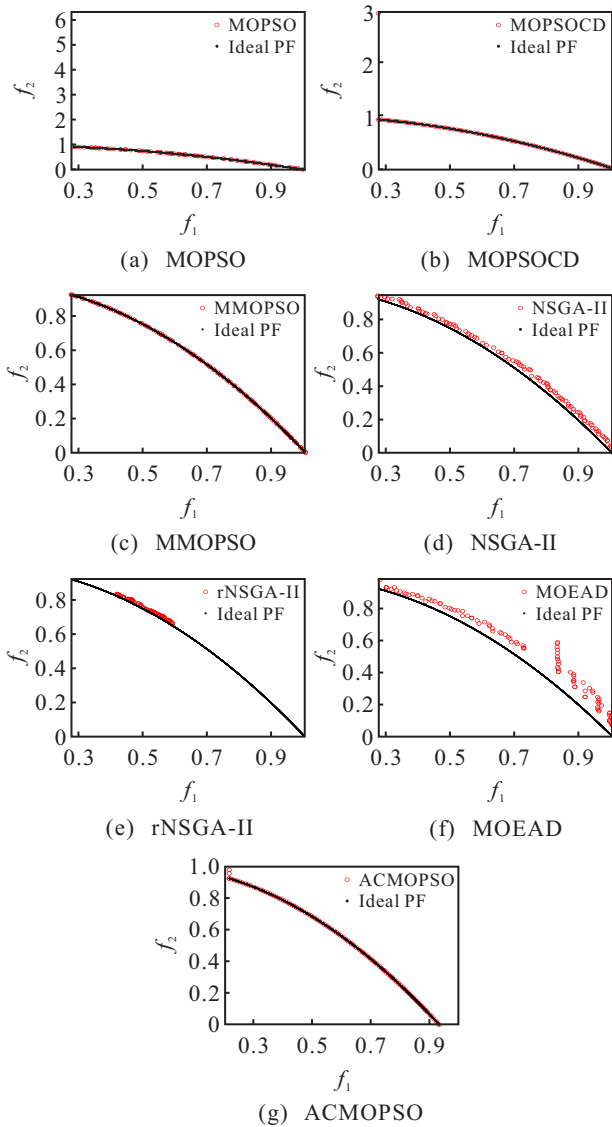


图6 在ZDT6中,各算法在IGD最优处获得的PF

从表3可以看出:在大多数测试问题上,尤其是双目标测试问题上,ACMOPSO的综合性能明显优于其他算法.在ZDT上,该算法得到的数据离散度最小,其IGD值分别为 $4.1495 \times 10^{-3}$ 、 $4.0663 \times 10^{-3}$ 、 $5.2241 \times 10^{-3}$ 、 $6.1004 \times 10^{-1}$ 以及 $2.7248 \times 10^{-3}$ .特别是在ZDT4上,ACMOPSO的IGD平均值比MOPSO和MOPSOCD低100倍.

ACMOPSO根据粒子群的平均距离对粒子进行等级划分,并从每个类别中去除拥挤度较大的粒子,使PF分布更加均匀.如图6所示,ACMOPSO得到的近似PF的分布比其比较算法得到的近似PF的分布要平滑得多.MOPSO得到的近似PF的均匀性相对较差,这可能与档案维护策略有关.

拥挤距离可以充分利用邻域信息获得合适的搜索方向,提高当前种群的多样性.MOPSOCD和ACMOPSO都使用拥挤距离挖掘未知区域的粒子.从表4可以看出:ACMOPSO的优势可能在于多样

性的提高.在双目标问题(ZDT1、ZDT2、ZDT6)和多目标问题(DTLZ1、DTLZ2、DTLZ4、DTLZ5)上,ACMOPSO得到的PD结果分别为 $1.8243 \times 10^3$ 、 $1.7182 \times 10^3$ 、 $1.6228 \times 10^3$ 、 $1.7419 \times 10^6$ 、 $2.1484 \times 10^5$ 、 $1.3422 \times 10^5$ 、 $8.3799 \times 10^4$ ,均为最优值.

结合表3与表4,ACMOPSO在多样性上仅略低于比较算法,但分布较好.例如,在ZDT1和ZDT2上,ACMOPSO算法比MOPSOCD和MMOPSO算法产生的IGD平均值更小.但在PD均值方面,ACMOPSO低于但接近于MOPSOCD和MMOPSO.

### 4.3 与典型MOEAs相比较

表3给出了ACMOPSO和典型MOEAs计算的IGD均值和标准差,PD的均值和标准差如表4所示.各算法在ZDT6上得到的近似PF如图6(d)~图6(f)所示.

IGD的结果表明:12个测试问题中,ACMOPSO在8个测试实例上获得了最好的结果,尤其在ZDT问题上.同时,与NSGA-II和rNSGA-II相比,ACMOPSO能得到最小的IGD结果,说明平均距离对拥挤距离有着积极的影响,该影响能够提升算法的效果.在ZDT1、ZDT3、ZDT6和DTLZ4上,ACMOPSO算法得到的IGD均值要比其他算法小得多.从图6可以推断,本文算法得到的近似PF几乎是均匀分布的.同时不可忽视的是,ACMOPSO在某些问题上存在劣势,例如DTLZ3.

从表3和表4可以看出:在ZDT1~ZDT3、ZDT6、DTLZ2、DTLZ4、DTLZ5和DTLZ7上,ACMOPSO得到的指标明显优于MOEAs.结果表明,该算法在大多数情况下在多样性和分布性上具有竞争力.在图6中,MOEAs得到的近似PF收敛性均比ACMOPSO差.其中:rNSGA-II搜索到的近似PF分布相对分散,MOEAD得到的近似PF是断开的.

## 5 应用案例

上述内容探讨了ACMOPSO在基准函数上的性能.进一步观察其对现实优化问题的处理能力,本节考虑了含约束的EED模型<sup>[22-23]</sup>.本节采用惩罚因子处理约束,将原约束问题转化为无约束MOP.由于没有理想PF做参考,IGD替换为超体积(hypervolume, HV<sup>[24]</sup>)指标.在PD最优值时,给出3种算法得到的近似PF.

环境经济调度的目的是在满足供需平衡和产量定额的条件下,使燃煤发电机组的成本最小化.电力系统的成本包括能源的消耗量和污染物的排放

量. 假设该系统有  $n$  台燃煤发电机, 系统的输出功率  $p = [p_1, \dots, p_k, \dots, p_n]$ , 总燃料成本 (\$/h) 可表示为  $F(p)$ ,  $E(p)$  为电力系统总排放量 (t/h). 由于篇幅有限, 文中仅给出如下简要模型, 详细模型及系数请参阅文献[23]:

$$\begin{aligned} & \min [F(p), E(p)]; \\ & \text{s.t. } p_k^{\min} \leq p_k \leq p_k^{\max}, h(p) = 0, g(p) \geq 0. \end{aligned} \quad (8)$$

其中:  $p_k^{\min}$ 、 $p_k^{\max}$  为第  $k$  台发电机的输出限额,  $h(p)$  为系统的功率平衡约束,  $g(p)$  为系统的旋转备用容量约束.

### 5.1 约束处理

静态惩罚函数高效且易理解, 常被用于处理 MOP 中的约束条件. 本文将约束违反程度构造为惩罚项, 并将其纳入基于惩罚因子的目标函数. 将式(8)的原问题转化为如下无约束问题:

$$\begin{aligned} \min [f_1, f_2] &= [F(p) + M_F M(p), E(p) + M_E M(p)], \\ \text{where } M(p) &= |h(p)|^2 + [\min\{g(p), 0\}]^2, \end{aligned} \quad (9)$$

其中  $M_F$  和  $M_E$  分别为燃料成本和污染排放量的惩罚因子.

### 5.2 参数设置

实验采用 6 台发电机的 IEEE30 总线系统, 故  $n = 6$ . 系统总负荷为 283.4 MW. 惩罚因子  $M_F = 10$  和  $M_E = 0.0001$  同文献[23]. MOPSO 是 ACMOPSO 的基础算法, ACMOPSO 也包含了 MOPSOCD 的拥挤距离机制, 因此将二者作为对比算法. 种群的大小仍为 100. HV 的参考点为 (0,0). HV 值越小, PF 的综合性能越优. ACMOPSO、MOPSO 和 MOPSOCD 的参数设置可参阅第 3 节.

### 5.3 讨论

3 种算法独立运行 20 次, 统计结果如表 5 所示.

表 5 3 种算法在 EED 上的测试结果

	PD 的平均值 (标准差)	HV 的平均值 (标准差)
MOPSO	2.042 9e+04 (3.73e+03)	1.222 4e-03 (4.25e-05)
MOPSOCD	2.153 8e+04 (3.72e+03)	1.153 3e-03 (1.09e-05)
ACMOPSO	2.226 9e+04 (2.86e+03)	1.091 0e-03 (5.91e-06)

图 7 给出了具有最佳 PD 值的近似 PF. ACMOPSO 得到的 PD 均值均优于 MOPSO 和 MOPSOCD, 表明平均距离定义的邻域空间能够提升算法的多样性. 在 HV 方面, ACMOPSO 的均值最小, 其综合性能优于对比算法. 由此可以推断, 种群的平均距离明显增强了拥挤距离的使用效果. 从图 7 可以看出, 与 MOPSO 和 MOPSOCD 相比, ACMOPSO 得

到的粒子群分布较为均匀.

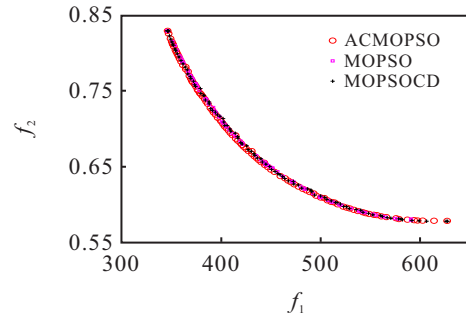


图 7 EED 中, 算法在 PD 值最优处的 PF

## 6 结论

本文结合种群的平均距离和粒子的拥挤距离, 提出了带双重距离的多目标粒子群优化算法 (ACMOPSO). 在 ACMOPSO 中, 粒子的邻域空间由种群的平均距离决定, 并且根据邻域内粒子数将粒子划分为若干等级. 在此基础上: 一方面选取当前种群中拥挤度最小的粒子作为全局最优粒子, 增加种群的多样性; 另一方面, 删除等级最高并且拥挤度最大的粒子, 维护解集的分布性.

在实验部分, 本文首先讨论了适合后续实验的 Mu 值, 其次探讨了 ACMOPSO 与典型优化算法的性能. 实验结果表明该算法在多样性和收敛性方面具有优越性. 此外, 本文将所提出的算法应用于 EED 问题, 得到了性能较优的解集. 然而, 在求解复杂的 MOPs 时, 所提算法的性能并不能令人满意. 因此, 后续工作可以考虑具有更复杂目标 (例如多模态多目标<sup>[25]</sup>) 以及多种特征<sup>[26]</sup> 的测试问题.

### 参考文献 (References)

- [1] Wang H Q, Wei J H, Wen S J, et al. Improved artificial bee colony algorithm and its application in classification[J]. Journal of Robotics and Mechatronics, 2018, 30(6): 921-926.
- [2] Cheng Z W, Song H H, Wang J Q, et al. Hybrid firefly algorithm with grouping attraction for constrained optimization problem[J]. Knowledge-Based Systems, 2021, 220: 106937.
- [3] 李全耀, 沈艳霞. 一种基于教与学的混合灰狼优化算法[J]. 控制与决策, 2022, 37(12): 3190-3196. (Li Q Y, Shen Y X. A hybrid gray wolf optimization algorithm based on the teaching-learning optimization[J]. Control and Decision, 2022, 37(12): 3190-3196.)
- [4] Coello C A C, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]. Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu, 2002: 1051-1056.
- [5] Lin Q Z, Liu S B, Zhu Q L, et al. Particle swarm optimization with a balanceable fitness estimation

- for many-objective optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 32-46.
- [6] 冯茜, 李擎, 王耀祖, 等. 约束多目标粒子群算法在烧结配矿优化中的应用[J]. 控制理论与应用, 2022, 39(5): 923-932.  
(Feng Q, Li Q, Wang Y Z, et al. Application of constrained multi-objective particle swarm optimization to sinter proportioning optimization[J]. Control Theory & Applications, 2022, 39(5): 923-932.)
- [7] Dai C, Wang Y P, Ye M. A new multi-objective particle swarm optimization algorithm based on decomposition[J]. Information Sciences, 2015, 325: 541-557.
- [8] 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.
- [9] 胡成玉, 余果, 颜雪松, 等. 基于改进多目标优化算法的分布式数据中心负载调度[J]. 控制与决策, 2021, 36(1): 159-165.  
(Hu C Y, Yu G, Yan X S, et al. Multi-objective optimization of energy and performance management in distributed data centers[J]. Control and Decision, 2021, 36(1): 159-165.)
- [10] Raquel C R, Naval P C. An effective use of crowding distance in multiobjective particle swarm optimization[C]. Proceedings of the 7th annual conference on genetic and evolutionary computation. Washington DC, 2005: 257-264.
- [11] Li W, Meng X, Huang Y, et al. Knowledge-guided multiobjective particle swarm optimization with fusion learning strategies[J]. Complex & Intelligent Systems, 2021, 7(3): 1223-1239.
- [12] Li C. Multi-objective optimization of space adaptive division for environmental economic dispatch[J]. Sustainable Computing: Informatics and Systems, 2021, 30: 100500.
- [13] 崔志华, 张茂清, 常宇, 等. 基于平均距离聚类的 NSGA-II[J]. 自动化学报, 2021, 47(5): 1171-1182.  
(Cui Z H, Zhang M Q, Chang Y, et al. NSGA-II with average distance clustering[J]. Acta Automatica Sinica, 2021, 47(5): 1171-1182.)
- [14] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.
- [15] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu, 2002: 825-830.
- [16] Jiang S Y, Yang S X. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts[J]. IEEE Transactions on Cybernetics, 2015, 46(2): 421-437.
- [17] Wang H D, Jin Y C, Yao X. Diversity assessment in many-objective optimization[J]. IEEE Transactions on Cybernetics, 2017, 47(6): 1510-1522.
- [18] Lin Q Z, Li J Q, Du Z H, et al. A novel multi-objective particle swarm optimization with multiple search strategies[J]. European Journal of Operational Research, 2015, 247(3): 732-744.
- [19] Said L B, Bechikh S, Ghédira K. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(5): 801-818.
- [20] Zhang Q F, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- [21] 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.
- [22] 帅茂杭, 熊国江, 胡晓, 等. 基于改进多目标骨干粒子群算法的电力系统环境经济调度[J]. 控制与决策, 2022, 37(4): 997-1004.  
(Shuai M H, Xiong G J, Hu X, et al. Economic emission dispatch of power system based on improved bare-bone multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2022, 37(4): 997-1004.)
- [23] 张子泳, 仇梦林, 李莎. 基于多目标粒子群算法的电力系统环境经济调度研究[J]. 电力系统保护与控制, 2017, 45(10): 1-10.  
(Zhang Z Y, Zhang M L, Li S. Environmental/economic power dispatch based on multi-objective particle swarm constraint optimization algorithm[J]. Power System Protection and Control, 2017, 45(10): 1-10.)
- [24] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271.
- [25] 岳彩通, 梁静, 瞿博阳, 等. 多模态多目标优化综述[J]. 控制与决策, 2021, 36(11): 2577-2588.  
(Yue C T, Liang J, Qu B Y, et al. A survey on multimodal multiobjective optimization[J]. Control and Decision, 2021, 36(11): 2577-2588.)
- [26] 于海波, 朱秦娜, 康丽, 等. 带偏向性轮盘赌的多算子协同粒子群优化算法[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2022.1486.  
(Yu H B, Zhu Q N, Kang L, et al. A multi-operator collaborative particle swarm optimization algorithm with biased roulette[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2022.1486.)

## 作者简介

慈雨(1999—), 男, 硕士生, 从事智能优化算法的研究, E-mail: ciyu\_81@foxmail.com;

荣淼(1992—), 女, 讲师, 博士, 从事智能优化与控制方法等研究, E-mail: mrong707@shu.edu.cn;

彭晨(1972—), 男, 教授, 博士, 从事网络化控制、火电能效优化等研究, E-mail: c.peng@i.shu.edu.cn.