

## 精英竞争和综合控制的多目标粒子群算法

陈飞<sup>1</sup>, 刘衍民<sup>2</sup>, 刘君<sup>3</sup>, 张娴子<sup>3</sup>

1. 贵州大学 数学与统计学院, 贵阳 550025
2. 遵义师范学院 数学学院, 贵州 遵义 563006
3. 贵州民族大学 数据科学与信息工程学院, 贵阳 550025

**摘要:**目的 多目标粒子群算法虽然极易实现且收敛速度快,但在平衡其收敛性和多样性方面仍需进一步改善。方法 针对上述问题,提出一种精英竞争和综合控制的多目标粒子群算法(ECMOPSO)。一方面,算法采用全局损害选择精英粒子集,然后将两两竞争引入多目标粒子群算法中,通过精英竞争选取优胜者粒子,将其与全局领导者融合形成更全面的社会综合信息,以增强种群中粒子之间信息的交互性,更好引导种群中的粒子飞行,提升算法全局探索能力;另一方面,结合全局损害和基于位移密度估计对外部存档进行维护,从而提高外部存档中非劣解的质量,平衡算法的收敛性和多样性。结果 将ECMOPSO算法与4个多目标粒子群算法和4个多目标进化算法在ZDT和UF系列基准测试问题上进行仿真实验,并采用Wilcoxon秩和检验和Friedman秩检验比较ECMOPSO算法与所选对比算法的整体性能。实验结果表明:相比其他几个对比算法,ECMOPSO算法的收敛能力、解的分布性以及稳定性都得到了一定的提升。结论 ECMOPSO算法可以很好地平衡收敛性和多样性,提升其整体性能,能有效求解大多数多目标优化问题。

**关键词:**多目标粒子群算法;精英竞争;综合控制;全局损害;基于位移密度估计

中图分类号:TP18 文献标识码:A doi:10.16055/j.issn.1672-058X.2024.0002.010

### Multi-objective Particle Swarm Optimization with Elite Competition and Comprehensive Control

CHEN Fei<sup>1</sup>, LIU Yanmin<sup>2</sup>, LIU Jun<sup>3</sup>, ZHANG Xianzi<sup>3</sup>

1. School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China
2. School of Mathematics, Zunyi Normal University, Guizhou Zunyi 563006, China
3. School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China

**Abstract: Objective** Although multi-objective particle swarm optimization is easy to implement and has a fast convergence speed, it still needs to be further improved in the aspect of balancing convergence and diversity.

**Methods** To solve the above problem, a multi-objective particle swarm optimization with elite competition and comprehensive control (ECMOPSO) was proposed. On one hand, the algorithm selected the elite particle set with global detriment, and then the pairwise competitions were introduced into the multi-objective particle swarm optimization. The winner particles were selected through elite competition, and the winner particles were fused with the global leaders to form more comprehensive social information, so as to enhance the information interaction between particles in the

收稿日期:2022-12-27 修回日期:2023-03-02 文章编号:1672-058X(2024)02-0074-12

基金项目:贵州省进化人工智能重点实验室([2022]059);贵州省数字经济重点人才计划(2022001)。

作者简介:陈飞(1999—),女,贵州遵义人,硕士研究生,从事优化理论及智能算法研究。

通讯作者:刘衍民(1978—),男,山东临沂人,教授,博士研究生导师,从事优化理论及智能算法研究。Email: yanmin7813@163.com.

引用格式:陈飞,刘衍民,刘君,等.精英竞争和综合控制的多目标粒子群算法[J].重庆工商大学学报(自然科学版),2024,41(2):74—85.

CHEN Fei, LIU Yanmin, LIU Jun, et al. Multi-objective particle swarm optimization with elite competition and comprehensive control[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2024, 41(2): 74—85.

population, better guide the flight of particles in the population, and improve the global exploration ability of the algorithm. On the other hand, global detriment and shifted-based density estimation were combined to maintain the external archive, so as to improve the quality of non-dominated solutions in the external archive and balance the convergence and diversity of the algorithm. **Results** The ECMOPSO algorithm, four multi-objective particle swarm optimization algorithms, and four multi-objective evolutionary algorithms were simulated for ZDT and UF series benchmark problems, and the Wilcoxon rank sum test and the Friedman rank test were used to compare the overall performance of ECMOPSO algorithm with the selected comparison algorithms. The experimental results showed that compared with other comparison algorithms, the convergence ability, solution distribution, and stability of the ECMOPSO algorithm have been improved to some extent. **Conclusion** ECMOPSO algorithm can balance convergence and diversity well, improve its overall performance, and can effectively solve most multi-objective optimization problems.

**Keywords:** multi-objective particle swarm optimization; elite competition; comprehensive control; global detriment; shifted-based density estimation

## 1 引言

由于多目标优化问题 (Multi-objective Optimization Problems, MOPs)<sup>[1]</sup> 的目标之间相互冲突, 一个目标的改进可能会导致其他目标退化, 所以 MOPs 中通常得到的是一组折衷解, 这个解被称为 Pareto 最优解<sup>[1]</sup>。粒子群算法 (Particle Swarm Optimization, PSO)<sup>[2]</sup> 具有收敛速度快、参数少和易实现等特点, 在求解单目标优化问题时有着良好的优化性能, 随之有学者将其扩展来求解 MOPs, 也取得了很好的效果。

Coello 等<sup>[3]</sup> 在 2002 年提出多目标粒子群算法 (Multi-Objective Particle Swarm Optimization, MOPSO), 该算法使用外部存档来存储算法在搜索过程中发现的所有非劣解, 为全局最优解 (Global Best Solution,  $g_{best}$ ) 的选取提供良好的候选者, 从而与个体最优解 (Personal Best Solution,  $p_{best}$ ) 共同指导种群中的粒子飞行, 这样可以加快算法的收敛速度。但是, 在优化过程中, MOPSO 算法全局搜索能力较弱, 易导致早熟收敛而陷入局部最优及多样性差等问题。因此, 许多研究人员针对这些问题提出了改进的 MOPSO 算法<sup>[4-8]</sup>。Han 等<sup>[4]</sup> 使用多目标梯度法对外部存档进行维护, 以提高进化过程中的收敛速度和局部开发能力, 但在解决某些多模态问题时并没有表现出较大优势; Li 等<sup>[5]</sup> 提出一种主导差分方法来判别解之间的优势关系以及设计一个基于  $L_p$  范数的密度估计器, 使算法不仅具有良好的收敛性和多样性, 而且具有较低的复杂度, 但在解决具有复杂 Pareto 前沿 (Pareto Front, PF) 问题时还有一定不足; Wu 等<sup>[6]</sup> 提出一种新的进化状态估计机制自适应地切换领导者的选择策略, 平衡算法在演化过程中

的开发与探索关系, 但在某些 MOPs 上性能不太稳定; Han 等<sup>[7]</sup> 提出一种基于解分布熵和种群间距信息混合框架的自适应算法, 在收敛速度和精度方面都有一定的提高, 但增加了一定的计算复杂度; Zhu 等<sup>[8]</sup> 利用分解方法将 MOPs 转换为一组子问题, 并且外部档案中的个体通过基于免疫的进化策略进行进化, 有利于加快收敛速度, 但其收敛性和多样性并未均衡。

综上所述, 以上各种算法在收敛性和多样性等方面都有一定的提升, 但是在有效平衡其收敛性和多样性方面仍需改进。为进一步解决上述问题, 本文提出一种精英竞争和综合控制的多目标粒子群算法 (ECMOPSO)。首先, 为了提高种群的搜索能力, 采用全局损害选择精英粒子集; 其次, 将 MOPSO 算法与两两竞争结合起来, 通过精英竞争选取优胜者粒子, 将其与  $g_{best}$  融合以便更好地引导种群中的粒子飞行; 最后, 结合全局损害和基于位移密度估计的综合平衡控制策略对外部存档进行维护, 有效平衡算法的收敛性和多样性。

## 2 相关工作

### 2.1 多目标优化问题

一般多目标优化问题<sup>[9]</sup> 可以表述如下:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) \\ \text{s. t.} \quad & g_i(\mathbf{x}) \geq 0, i = 1, 2, \dots, q \\ & h_j(\mathbf{x}) = 0, j = 1, 2, \dots, r \end{aligned}$$

其中:  $\mathbf{x} = (x_1, x_2, \dots, x_D)$  是决策空间  $\mathbf{X} \subset \mathbf{R}^D$  中的一个  $D$  维决策向量;  $\mathbf{F}(\mathbf{x})$  表示  $M$  维的目标向量;  $f_i(\mathbf{x})$  表示第  $i$  维的目标函数;  $g_i(\mathbf{x}) \geq 0$  ( $i = 1, 2, \dots, q$ ) 和  $h_j(\mathbf{x}) = 0$  ( $j = 1, 2, \dots, r$ ) 分别是不等式约束和等式约束。

## 2.2 粒子群算法

在 PSO 算法<sup>[2]</sup>中,每个粒子通过跟踪两个领导者  $p_{\text{best}}$  和  $g_{\text{best}}$  更新其自身速度。 $p_{\text{best}}$  记录了每个粒子在历史搜索过程中的最优位置,而  $g_{\text{best}}$  则记录了整个种群当前搜索到的最优位置。假设第  $i$  个粒子的速度矢量为  $\mathbf{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ , 位置矢量为  $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ , 其中  $i = 1, 2, \dots, N$ ,  $N$  为种群规模,  $D$  为决策空间的维数。则第  $i$  个粒子在第  $t+1$  次迭代都遵循式(1)和式(2)来更新其速度和位置。

$$\mathbf{V}_i(t+1) = \omega \mathbf{V}_i(t) + c_1 r_1 (p_{\text{best}_i}(t) - \mathbf{X}_i(t)) + c_2 r_2 (g_{\text{best}_i}(t) - \mathbf{X}_i(t)) \quad (1)$$

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \quad (2)$$

其中:  $t$  是迭代次数,  $\omega$  为惯性权重,  $c_1$  和  $c_2$  是分别用来控制  $p_{\text{best}}$  和  $g_{\text{best}}$  对粒子速度影响的加速度系数,  $r_1$  和  $r_2$  是  $[0, 1]$  中均匀生成的两个随机数。

## 2.3 竞争群优化器

竞争群优化器 (Competitive Swarm Optimizer, CSO) 是由 Cheng 等<sup>[10]</sup> 提出的一个受 PSO 算法启发且计算成本较低的算法, 它与传统 PSO 算法在本质上有很大不同。CSO 算法不再使用  $p_{\text{best}}$  和  $g_{\text{best}}$  来引导粒子飞行, 而是使用一种两两竞争学习来更新粒子的速度和位置。实验结果显示: CSO 算法比 PSO 算法可以更好地平衡收敛性和多样性。第  $t+1$  次迭代过程中, 第  $k$  轮竞争之后失败粒子的速度和位置更新公式如下:

$$\mathbf{V}_{l,k}(t+1) = r_1 \mathbf{V}_{l,k}(t) + r_2 (\mathbf{X}_{w,k}(t) - \mathbf{X}_{l,k}(t)) + \varphi r_3 (\bar{\mathbf{X}}_k(t) - \mathbf{X}_{l,k}(t))$$

$$\mathbf{X}_{l,k}(t+1) = \mathbf{X}_{l,k}(t) + \mathbf{V}_{l,k}(t+1)$$

其中:  $t$  是迭代次数;  $k = 1, 2, \dots, N/2$ ,  $N$  为种群规模;  $r_1$ ,  $r_2$  和  $r_3$  是  $[0, 1]$  中均匀生成的 3 个随机数;  $\mathbf{V}_{l,k}(t)$  和  $\mathbf{X}_{l,k}(t)$  分别是第  $t$  次迭代过程中第  $k$  轮竞争之后失败粒子的速度和位置;  $\mathbf{X}_{w,k}(t)$  和  $\bar{\mathbf{X}}_k(t)$  分别是第  $t$  次迭代过程中第  $k$  轮竞争之后优胜者粒子的位置和种群中所有粒子位置的均值;  $\varphi$  是用来控制  $\bar{\mathbf{X}}_k(t)$  影响的一个参数。

## 3 MOPSO 算法设计

为了更好平衡算法的收敛性和多样性, 使粒子能够在较高维度的搜索空间中跳出局部最优而收敛到全局最优, 本文提出一个改进的 ECMOPSO 算法。与原始 MOPSO 算法相比, ECMOPSO 算法提出了一种精英竞争的社会融合策略对粒子进行更新, 并利用一种综合平衡控制策略对外部存档进行维护, 有效地平衡了收敛性和多样性。

## 3.1 精英竞争的社会融合

在大多数元启发式算法中, 如何有效平衡收敛性和多样性是一个至关重要的问题。在标准的 MOPSO 算法中, 种群中的每个粒子向  $p_{\text{best}}$  和  $g_{\text{best}}$  这两个领导者学习, 在其“社会学习”部分只选择一个全局领导者。这种情况下, 粒子不能从各种样本中学到更多有用的经验, 从而降低了种群的多样性。

基于上述这个问题, ECMOPSO 算法结合了 CSO 算法中的两两竞争对种群中的粒子进行更新, 以实现收敛性和多样性之间的平衡。首先, 创建一个预定义大小为  $r$  的精英粒子集  $E$ , 以获得综合性能较好的非劣解。由于精英粒子集是用来提供候选粒子的, 用于两两竞争选择领导者来引导种群的搜索, 所以选择的精英粒子应该在收敛性和多样性之间保持良好的平衡。因此, 本文结合基于非支配排序和基于指标排序的方法来选择精英粒子。由于全局损害 (Global Detriment, GD)<sup>[11]</sup> 考虑解在不同目标上差异的绝对值累积, 是一个考虑解之间差异有多显著的方法, 所以选择它来作为  $E$  中精英粒子的选择原则, 最终选出排名前  $r$  的粒子作为精英粒子集。GD 的描述具体如下:

$$R_{\text{GD}}(NS_i) = \frac{1}{NM} \sum_{j=1}^N \sum_{m=1}^M \frac{\max(f_m(NS_i) - f_m(NS_j), 0)}{f_{m,\text{max}} - f_{m,\text{min}}}$$

其中:  $NS_i$  是第  $i$  个非劣解;  $N$  是非劣解的个数;  $M$  是目标个数;  $f_m(\cdot)$  为第  $m$  个目标的适应度函数;  $f_{m,\text{max}}$  和  $f_{m,\text{min}}$  分别是第  $m$  个目标上的最大值和最小值。

接下来, 从精英粒子集  $E$  中随机选出两个精英粒子竞争, 优胜者粒子  $e$  与  $g_{\text{best}}$  融合来引导种群中的粒子飞行, 这样有利于更好地实现收敛性和多样性之间的平衡。假设  $\mathbf{V}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$  为第  $i$  个粒子的速度向量,  $\mathbf{X}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$  为位置向量, 优胜者粒子位置向量为  $\mathbf{e}_i = (e_{i,1}, e_{i,2}, \dots, e_{i,D})$ , 其中:  $i = 1, 2, \dots, N$ ,  $N$  为种群中的粒子数;  $D$  为决策空间的维数。则第  $i$  个粒子在第  $t+1$  次迭代时的速度和位置更新如式(3)和式(2)所示。

$$\mathbf{V}_i(t+1) = \omega \mathbf{V}_i(t) + c_1 r_1 (p_{\text{best}_i}(t) - \mathbf{X}_i(t)) + c_2 r_2 ((g_{\text{best}_i}(t) + \mathbf{e}_i(t)) / 2 - \mathbf{X}_i(t)) \quad (3)$$

其中:  $t$  为迭代次数,  $\omega$  为惯性权重,  $p_{\text{best}}$  和  $g_{\text{best}}$  分别为个体最优解和全局最优解,  $c_1$  和  $c_2$  是分别用来控制  $p_{\text{best}}$  和  $g_{\text{best}}$  对粒子速度影响的加速度系数,  $r_1$  和  $r_2$  是  $[0, 1]$  中均匀生成的两个随机数。

## 3.2 综合平衡控制

在大多数多目标粒子群算法中, 随着算法迭代次

数的增加,为了能够保留搜索过程中发现的具有良好性能的非劣解,以及最后获得分布良好的 Pareto 前沿,需要对外部存档进行有效维护。

MOPSO<sup>[3]</sup>算法采用自适应网格策略对外部存档中的非劣解进行删除,该算法随机删除密度最大的粒子,这样很可能会删除一些性能相对较好的非劣解,降低了算法的收敛性。因此,本文提出 GD 和基于位移密度估计(Shifted-based Density Estimation, SDE)<sup>[12]</sup>的综合平衡控制策略(其值由  $R_{SDE}$  表示)来估计存档中解的密度,从而对自适应网格中密度大的解进行删除,确保外部存档中非劣解保持良好的分布。SDE 同时涵盖个体的收敛性信息和分布性信息,当度量外部存档中的非劣解  $NS_i$  和其他非劣解  $NS_j$  之间的相似度时,首先按照式(4)来根据  $NS_i$  和  $NS_j$  之间的收敛性比较移动  $NS_j$  在目标空间中的位置,即

$$f'_m(NS_j) = \begin{cases} f_m(NS_j), & \text{if } f_m(NS_i) < f_m(NS_j) \\ f_m(NS_i), & \text{if } f_m(NS_i) \geq f_m(NS_j) \end{cases} \quad (4)$$

其中: $i, j = 1, 2, \dots, n, i \neq j, n$  为当前外部存档中非劣解的个数; $m = 1, 2, \dots, M, M$  表示目标个数; $f_m(NS_i)$  和  $f_m(NS_j)$  分别为非劣解  $NS_i$  和  $NS_j$  的第  $m$  个目标值; $f'_m(NS_j)$  表示解  $NS_j$  位移之后的目标值。移动位置后,根据式(5)中的欧氏距离来度量解  $NS_i$  和  $NS_j$  之间的相似度, $d_{i,j}$  表示  $NS_i$  与  $NS_j$  位移后的距离。再通过式(6)计算非劣解  $NS_i$  的密度  $D(NS_i)$ :

$$d_{i,j} = \sum_{m=1}^M |f'_m(NS_j) - f_m(NS_i)| \quad (5)$$

$$D(NS_i) = \min(d_{i,1}, d_{i,2}, \dots, d_{i,n}) \quad (6)$$

下面描述外部存档维护的综合平衡控制(Comprehensive Balance Control, CBC)指标(其值用  $R_{CBC}$  表示)。如式(7)所示, $R_{CBC}$  由  $R_{SDE}$  计算得到的密度和  $R_{GD}$  之比组成,可以更全面地评估外部存档中的非劣解, $R_{CBC}$  的值越大说明解的收敛性和多样性越好。

$$R_{CBC}(NS_i) = \frac{D(NS_i)}{R_{GD}(NS_i)} \quad (7)$$

为了验证  $R_{CBC}$  指标维护外部存档时的性能,图 1 显示了一个双目标的例子来说明分别采用指标  $R_{SDE}$ 、 $R_{GD}$  和  $R_{CBC}$  对密度最大的粒子进行删除时的效果。假设最密集的网格中有 6 个非劣解: $N1(0.054\ 7, 0.713\ 1)$ 、 $N2(0.055\ 0, 0.711\ 1)$ 、 $N3(0.055\ 1, 0.711\ 0)$ 、 $N4(0.055\ 3, 0.710\ 2)$ 、 $N5(0.055\ 6, 0.708\ 1)$  和  $N6(0.055\ 9, 0.705\ 8)$ , 表 1 分别列出了由指标  $R_{SDE}$ 、 $R_{GD}$  和  $R_{CBC}$  计算所获得的排序值,将要被删除解的值以粗体显示。从表中可以看

出: $N2$  和  $N3$  获得了相同的  $R_{SDE}$  值,这里随机删除其中一个,用  $R_{GD}$  评估时选择将极值点  $N6$  删除,这将导致非劣解分布不均匀;而  $R_{CBC}$  则结合了这两个指标的优点,选择相对拥挤的解  $N3$  删除。从图 1 可以直观地观察到:用  $R_{SDE}$  和  $R_{CBC}$  来删除粒子时都获得了较均匀的分布,但  $R_{CBC}$  更能精准地删除综合性能不好的粒子,更有效地平衡算法的收敛性和多样性。

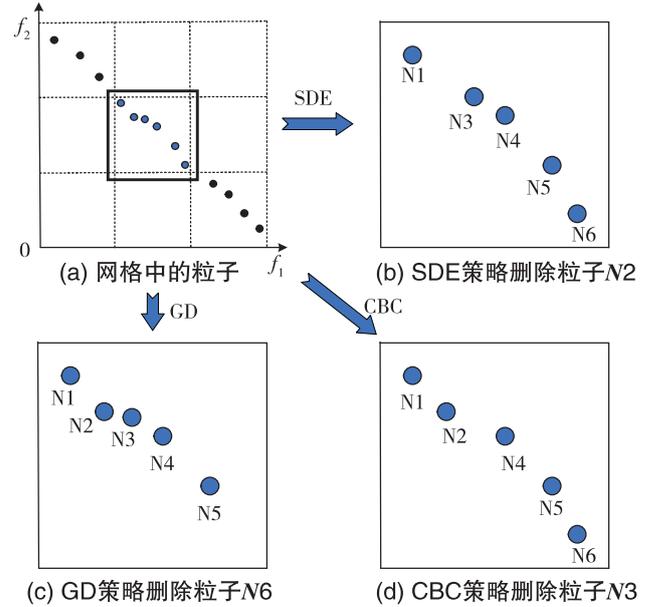


图 1 分别由  $R_{SDE}$ 、 $R_{GD}$  和  $R_{CBC}$  策略删除外部存档中粒子的图解  
Fig. 1 Illustrations of deleting particles in the external archive by  $R_{SDE}$ ,  $R_{GD}$  and  $R_{CBC}$ , respectively

表 1 由  $R_{SDE}$ 、 $R_{GD}$  和  $R_{CBC}$  获得的排序值

指标	N1	N2	N3	N4	N5	N6
$R_{SDE}$	0.000 3	<b>0.000 1</b>	<b>0.000 1</b>	0.000 3	0.000 3	0.002 3
$R_{GD}$	0.220 3	0.127 0	0.136 3	0.150 6	0.186 0	<b>0.263 9</b>
$R_{CBC}$	0.001 4	0.000 8	<b>0.000 7</b>	0.002 0	0.001 6	0.008 7

### 3.3 算法的主要流程

算法的具体步骤如下:

**Step1** 设置 ECMOPSO 算法的相关参数,初始化种群中粒子的位置。

**Step2** 对粒子进行变异操作,并计算其适应度值。

**Step3** 结合非支配排序和指标排序选取精英粒子集  $E$ 。

**Step4** 建立外部存档,存储非劣解。若外部存档没有达到最大存储数,则继续引入非劣解;若达到最大存储数,则利用综合平衡控制对外部存档进行维护。

**Step5** 选取  $p_{best}$  和  $g_{best}$ , 并利用两两竞争选择优胜者粒子  $e$ 。

**Step6** 通过式(3)和式(2)更新粒子的速度和位置。

**Step7** 若当前迭代次数没有达到最大迭代次数,则返回 Step2;否则终止迭代,输出最优解集。

ECMOPSO 算法的流程图如图 2 所示:

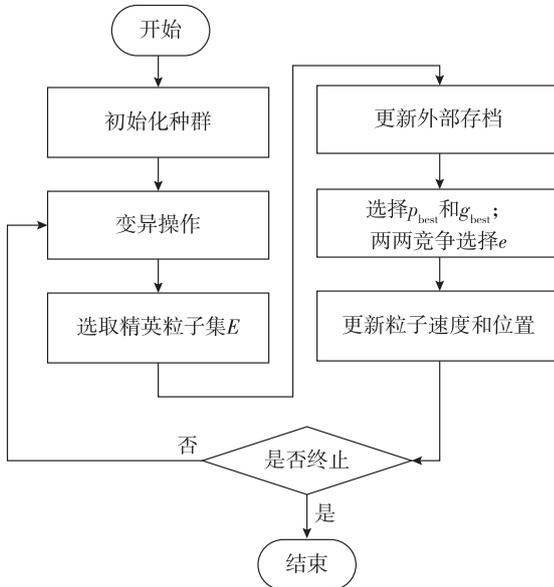


图 2 ECMOPSO 算法流程图

Fig. 2 Flow chart of ECMOPSO algorithm

## 4 实验仿真分析

### 4.1 实验参数设置

为了对 ECMOPSO 算法的性能更客观地评价,使用了 ZDT<sup>[13]</sup> 和 UF<sup>[14]</sup> 两个不同基准测试组的 15 个测试问题来评估算法的性能。其中包括 ZDT 测试组的 5 个双目标测试问题以及 UF 测试组的 7 个双目标测试问题和 3 个三目标测试问题。这些测试问题具有不同的特性和复杂的 Pareto 前沿特征,比如凹凸性、多模态和不规则 Pareto 前沿形状等,可以很好地验证算法的可靠性和效率。对于双目标测试问题,ZDT1—ZDT3 和 UF1—UF7 的决策变量个数设置为 30,ZDT4 和 ZDT6 的决策变量个数设置为 10。对于三目标测试问题,UF8—UF10 的决策变量个数设置为 30。另外,选取了 8 个算法进行对比,其中有 4 个先进的多目标粒子群算法:SMPSO<sup>[15]</sup>、dMOPSO<sup>[16]</sup>、MPSOD<sup>[17]</sup>、NMPSO<sup>[18]</sup>,以及 4 个具有竞争力的多目标进化算法:NSGAI<sup>[19]</sup>、MOEADD<sup>[20]</sup>、SPEAR<sup>[21]</sup>、DGEA<sup>[22]</sup>。为了保证算法性能比较的公平性,所有对比算法设置的相关参数都与原始参考文献一致,每个算法的主要参数设置如表 2 所示。

表 2 ECMOPSO 与几个对比算法的参数设置

Table 2 Parameter settings of ECMOPSO and several comparison algorithms

算法名称	参数设置
SMPSO	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
dMOPSO	$T_a = 2, \theta = 5$
MPSOD	$\omega \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5], p_c = 0.9, F = 0.5, CR = 0.5, p_m = 1/n, \eta_m = 20, \eta_c = 20$
NMPSO	$\omega \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
NSGAI	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
MOEADD	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 30, T = 20, \delta = 0.9$
SPEAR	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
DGEA	$R = 10$
ECMOPSO	$\omega = 0.4, c_1 = c_2 = 2, r = 30$

所有算法种群规模设置为 200,外部存档大小为 200,最大函数评估次数为 10 000。对于参数  $r$ ,即精英粒子集大小,将会在 4.5 节进行参数分析。各算法在每个测试问题上都是独立运行 30 次,并且所有算法的实验数据都是在 Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz Windows 7 系统中运用 PlatEMO 平台<sup>[23]</sup>和 MATLAB R2021b 实现的。

### 4.2 性能评价指标

为了评估算法的性能,采用反世代距离(Inverted Generational Distance, IGD<sup>[24]</sup>,其值用  $R_{IGD}$  表示)和超体

积(Hypervolume, HV<sup>[25]</sup>,其值用  $R_{HV}$  表示)作为算法的性能评价指标。

(1)  $R_{IGD}$  是一种综合性指标,用来衡量算法得到的 Pareto 最优解集与真实 PF 之间的距离,可以很好地检验算法的收敛性和多样性。一个算法的  $R_{IGD}$  值越小,就说明算法的收敛性和多样性越好。 $R_{IGD}$  的计算公式为

$$R_{IGD}(P, S) = \frac{1}{|P|} \sum_{x \in P} dist(x, S)$$

其中: $S$  表示算法得到的 Pareto 最优解集, $P$  表示在 Pareto 前沿上均匀分布的解集, $dist(x, S)$  是  $P$  中的解  $x$

与  $S$  之间的最小欧氏距离。

(2)  $R_{HV}$  也是一个综合的性能指标,表示算法获得的 Pareto 最优解集与参照点围成的目标区域的体积。该指标可以估计算法所得到解集的收敛性和多样性,一个算法的  $R_{HV}$  值越大,就说明算法的收敛性和多样性越好。假设  $Z=(Z_1, Z_2, \dots, Z_m)$  是目标空间中由所有 Pareto 最优解支配的一个参考点,那么  $R_{HV}$  的计算公式为

$$R_{HV}(S) = \delta(\cup_{x \in S} [f_1(x), z_1] \times [f_m(x), z_m])$$

其中: $S$  表示算法得到的 Pareto 最优解集, $\delta$  表示勒贝

格测度。

### 4.3 实验结果及数据分析

表 3 和表 4 分别给出了 ECMOPSO 算法与其他 8 个对比算法在 15 个测试问题上  $R_{IGD}$  值和  $R_{HV}$  值的均值(Mean)和标准差(Std),每个测试问题的最佳  $R_{IGD}$  值和  $R_{HV}$  值以粗体显示。此外,在  $\alpha=0.05$  的显著性水平下,采用 Wilcoxon 秩和检验,以显示检验结果之间的显著性差异。表中的符号“+”、“-”和“ $\approx$ ”分别表示其他算法的结果明显优于 ECMOPSO 算法、明显差于 ECMOPSO 算法以及与 ECMOPSO 算法在统计上相似。

表 3 ECMOPSO 与 8 个对比算法在 15 个测试问题上的  $R_{IGD}$  值  
Table 3  $R_{IGD}$  values of ECMOPSO and eight comparison algorithms on fifteen test problems

测试问题	$R_{IGD}$	SMPSO	dMOPSO	MPSOD	NMPSO	NSGAIII	MOEADD	SPEAR	DGEA	ECMOPSO
ZDT1	Mean	9.312 8e-2	5.702 1e-2	1.053 9e-1	2.932 0e-2	1.007 5e-1	1.207 1e-1	1.704 1e-1	1.117 1e+0	<b>2.880 7e-3</b>
	Std	(8.76e-2)	(1.72e-2)	(4.48e-2)	(1.09e-2)	(1.58e-2)	(1.84e-2)	(2.27e-2)	(2.91e-1)	<b>(1.31e-4)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT2	Mean	5.352 3e-2	4.389 9e-2	1.700 7e-1	1.936 6e-2	2.018 4e-1	1.533 8e-1	3.751 7e-1	8.798 2e-1	<b>3.442 8e-3</b>
	Std	(1.07e-1)	(1.26e-2)	(1.02e-1)	(6.61e-3)	(3.66e-2)	(2.65e-2)	(1.38e-1)	(3.45e-1)	<b>(1.24e-3)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT3	Mean	1.828 6e-1	3.503 5e-2	1.8725 e-1	8.742 9e-2	8.796 7e-2	1.824 7e-1	1.565 4e-1	9.667 7e-1	<b>3.461 8e-3</b>
	Std	(9.84e-2)	(6.31e-3)	(4.18e-2)	(2.79e-2)	(1.49e-2)	(1.50e-2)	(2.01e-2)	(2.18e-1)	<b>(3.00e-4)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT4	Mean	1.052 5e+1	5.485 3e+0	3.609 3e+1	1.718 1e+1	2.502 8e+0	<b>9.414 2e-1</b>	2.025 0e+0	6.476 4e+0	1.477 8e+0
	Std	(5.31e+0)	(6.11e+0)	(6.75e+0)	(1.02e+1)	(9.78e-1)	<b>(3.49e-1)</b>	(6.56e-1)	(3.85e+0)	(1.53e+0)
	Wilcoxon	-	-	-	-	-	$\approx$	-	-	-
ZDT6	Mean	<b>1.917 2e-3</b>	5.954 3e-3	2.756 3e-2	2.313 5e-3	1.446 5e+0	5.829 9e-1	1.050 9e+0	2.284 4e-2	2.384 2e-3
	Std	<b>(5.28e-5)</b>	(7.44e-3)	(2.87e-2)	(2.03e-4)	(2.84e-1)	(1.72e-1)	(2.05e-1)	(1.09e-1)	(8.60e-5)
	Wilcoxon	+	-	-	+	-	-	-	-	-
UF1	Mean	3.814 4e-1	6.568 5e-1	2.871 1e-1	1.330 8e-1	1.350 4e-1	1.551 8e-1	1.388 4e-1	6.094 1e-1	<b>1.245 0e-1</b>
	Std	(9.05e-2)	(9.66e-2)	(5.02e-2)	(2.61e-2)	(3.81e-2)	(3.36e-2)	(2.47e-2)	(1.43e-1)	<b>(8.69e-3)</b>
	Wilcoxon	-	-	-	$\approx$	$\approx$	-	-	-	-
UF2	Mean	1.018 4e-1	9.576 5e-2	1.129 5e-1	8.302 7e-2	8.161 2e-2	7.534 8e-2	7.512 9e-2	1.801 7e-1	<b>7.428 5e-2</b>
	Std	(8.97e-3)	(6.88e-3)	(1.01e-2)	(7.26e-3)	(5.33e-3)	(6.19e-3)	(1.06e-2)	(2.18e-2)	<b>(4.41e-3)</b>
	Wilcoxon	-	-	-	-	-	$\approx$	$\approx$	-	-
UF3	Mean	4.458 6e-1	3.300 9e-1	5.004 7e-1	3.577 4e-1	4.795 8e-1	4.676 6e-1	4.329 1e-1	5.671 3e-1	<b>2.744 6e-1</b>
	Std	(5.40e-2)	(6.15e-3)	(1.80e-2)	(6.03e-2)	(1.08e-2)	(1.32e-2)	(1.47e-2)	(3.36e-2)	<b>(1.74e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
UF4	Mean	1.112 9e-1	1.372 2e-1	9.905 4e-2	<b>6.289 6e-2</b>	9.585 7e-2	9.099 8e-2	8.495 7e-2	1.202 9e-1	1.361 3e-1
	Std	(8.05e-3)	(4.87e-3)	(4.98e-3)	<b>(6.16e-3)</b>	(2.50e-3)	(4.09e-3)	(2.04e-3)	(1.22e-2)	(1.50e-2)
	Wilcoxon	+	$\approx$	+	+	+	+	+	+	+

续表(表3)

测试问题	$R_{IGD}$	SMP SO	dMOP SO	MPSOD	NMPSO	NSG AIII	MOEADD	SPEAR	DGEA	ECMOP SO
UF5	Mean	2.829 8e+0	3.194 9e+0	2.756 2e+0	1.686 8e+0	1.615 1e+0	1.434 7e+0	<b>1.107 1e+0</b>	3.061 8e+0	1.424 9e+0
	Std	(5.59e-1)	(2.75e-1)	(2.66e-1)	(4.03e-1)	(3.91e-1)	(2.43e-1)	<b>(1.88e-1)</b>	(7.08e-1)	(3.08e-1)
	Wilcoxon	-	-	-	-	≈	≈	+	-	-
UF6	Mean	1.250 9e+0	2.344 9e+0	1.390 4e+0	6.933 8e-1	6.963 5e-1	8.156 1e-1	6.614 8e-1	2.488 7e+0	<b>5.772 6e-1</b>
	Std	(4.17e-1)	(5.44e-1)	(3.30e-1)	(1.22e-1)	(1.32e-1)	(1.21e-1)	(9.41e-2)	(5.78e-1)	<b>(5.24e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
UF7	Mean	3.508 8e-1	3.839 6e-1	2.513 3e-1	1.866 3e-1	2.038 3e-1	1.910 0e-1	1.728 5e-1	7.014 9e-1	<b>9.410 7e-2</b>
	Std	(1.33e-1)	(7.93e-2)	(6.38e-2)	(1.74e-1)	(6.16e-2)	(7.47e-2)	(4.94e-2)	(9.26e-2)	<b>(1.11e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
UF8	Mean	3.873 2e-1	3.414 9e-1	5.472 8e-1	4.775 8e-1	3.459 5e-1	3.318 5e-1	3.229 3e-1	7.279 7e-1	<b>2.765 4e-1</b>
	Std	(5.15e-2)	(3.37e-2)	(3.01e-2)	(1.01e-1)	(3.79e-2)	(2.17e-2)	(1.95e-2)	(1.26e-1)	<b>(3.88e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
UF9	Mean	5.525 1e-1	5.796 5e-1	6.601 2e-1	4.604 8e-1	5.153 1e-1	5.290 6e-1	4.637 8e-1	7.515 7e-1	<b>1.473 5e-1</b>
	Std	(3.74e-2)	(3.88e-2)	(3.81e-2)	(5.27e-2)	(4.58e-2)	(6.04e-2)	(4.87e-2)	(1.39e-1)	<b>(1.99e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
UF10	Mean	2.649 4e+0	1.320 4e+0	4.211 9e+0	1.435 7e+0	2.393 0e+0	3.443 0e+0	1.784 8e+0	4.738 9e+0	<b>1.270 3e+0</b>
	Std	(4.07e-1)	(2.12e+0)	(4.16e-1)	(3.60e-1)	(4.77e-1)	(5.31e-1)	(3.09e-1)	(7.65e-1)	<b>(6.00e-1)</b>
	Wilcoxon	-	≈	-	≈	-	-	-	-	-
+/-/≈	2/13/0	0/13/2	1/14/0	2/11/2	1/12/2	1/11/3	2/12/1	1/14/0	-	-
Best/All	1/15	0/15	0/15	1/15	0/15	1/15	1/15	0/15	11/15	-

表 4 ECMOP SO 与 8 个对比算法在 15 个测试问题上的  $R_{HV}$  值Table 4  $R_{HV}$  values of ECMOP SO and eight comparison algorithms on fifteen test problems

测试问题	$R_{HV}$	SMP SO	dMOP SO	MPSOD	NMPSO	NSG AIII	MOEADD	SPEAR	DGEA	ECMOP SO
ZDT1	Mean	6.051 7e-1	6.527 2e-1	5.720 0e-1	6.896 9e-1	5.893 8e-1	5.595 5e-1	4.986 8e-1	1.614 9e-2	<b>7.213 6e-1</b>
	Std	(1.05e-1)	(1.95e-2)	(5.85e-2)	(1.28e-2)	(1.96e-2)	(2.35e-2)	(2.42e-2)	(3.71e-2)	<b>(1.44e-4)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT2	Mean	3.932 6e-1	3.783 1e-1	2.502 6e-1	4.339 7e-1	2.089 0e-1	2.470 9e-1	8.947 7e-2	2.957 3e-2	<b>4.453 2e-1</b>
	Std	(9.98e-2)	(1.96e-2)	(1.02e-1)	(9.77e-3)	(2.99e-2)	(2.56e-2)	(6.11e-2)	(9.92e-2)	<b>(3.14e-3)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT3	Mean	5.205 3e-1	5.977 7e-1	4.676 7e-1	5.722 8e-1	5.433 6e-1	4.728 7e-1	5.052 5e-1	3.969 8e-2	<b>6.005 6e-1</b>
	Std	(6.83e-2)	(1.79e-2)	(4.14e-2)	(9.92e-3)	(1.03e-2)	(1.73e-2)	(3.00e-2)	(5.82e-2)	<b>(7.13e-4)</b>
	Wilcoxon	-	≈	-	-	-	-	-	-	-
ZDT4	Mean	0.000 0e+0	3.947 5e-2	0.000 0e+0	0.000 0e+0	4.353 8e-4	3.312 3e-2	0.000 0e+0	0.000 0e+0	<b>6.919 6e-2</b>
	Std	(0.00e+0)	(9.21e-2)	(0.00e+0)	(0.00e+0)	(2.38e-3)	(4.47e-2)	(0.00e+0)	(0.00e+0)	<b>(5.93e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	-
ZDT6	Mean	<b>3.900 3e-1</b>	3.859 9e-1	3.673 2e-1	3.897 5e-1	0.000 0e+0	1.970 8e-2	1.468 7e-3	3.799 0e-1	3.894 2e-1
	Std	<b>(5.86e-5)</b>	(7.41e-3)	(1.97e-2)	(1.76e-4)	(0.00e+0)	(2.13e-2)	(8.04e-3)	(5.05e-2)	(1.62e-4)
	Wilcoxon	+	≈	-	+	-	-	-	-	-

续表(表4)

测试问题	$R_{HV}$	SMPSO	dMOPSO	MPSOD	NMPSO	NSGAIII	MOEADD	SPEAR	DGEA	ECMOPSO
UF1	Mean	2.649 1e-1	5.508 4e-2	3.381 7e-1	5.153 7e-1	5.176 8e-1	4.927 9e-1	5.140 2e-1	1.099 9e-1	<b>5.273 8e-1</b>
	Std	(7.17e-2)	(5.08e-2)	(5.07e-2)	(3.63e-2)	(4.46e-2)	(4.05e-2)	(3.05e-2)	(8.33e-2)	<b>(1.50e-2)</b>
	Wilcoxon	-	-	-	≈	≈	-	≈	-	
UF2	Mean	6.067 1e-1	6.154 5e-1	5.787 8e-1	6.192 3e-1	6.155 9e-1	6.209 3e-1	6.247 7e-1	5.024 6e-1	<b>6.344 4e-1</b>
	Std	(8.62e-3)	(6.11e-3)	(1.04e-2)	(8.22e-3)	(6.98e-3)	(7.56e-3)	(9.70e-3)	(2.80e-2)	<b>(4.21e-3)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	
UF3	Mean	2.170 6e-1	3.096 3e-1	1.734 6e-1	2.830 2e-1	1.845 2e-1	1.999 2e-1	2.231 7e-1	1.217 6e-1	<b>3.740 4e-1</b>
	Std	(5.15e-2)	(9.53e-3)	(1.59e-2)	(5.67e-2)	(1.12e-2)	(1.15e-2)	(8.52e-3)	(1.64e-2)	<b>(1.88e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	
UF4	Mean	2.928 8e-1	2.532 7e-1	3.080 0e-1	<b>3.604 0e-1</b>	3.126 8e-1	3.188 6e-1	3.252 7e-1	2.811 0e-1	2.571 5e-1
	Std	(9.25e-3)	(4.58e-3)	(6.00e-3)	<b>(8.40e-3)</b>	(2.99e-3)	(4.21e-3)	(2.71e-3)	(1.43e-2)	(1.61e-2)
	Wilcoxon	+	≈	+	+	+	+	+	+	
UF5	Mean	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	<b>1.345 3e-5</b>	0.000 0e+0	0.000 0e+0
	Std	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	<b>(7.37e-5)</b>	(0.00e+0)	(0.00e+0)
	Wilcoxon	≈	≈	≈	≈	≈	≈	≈	≈	
UF6	Mean	1.143 2e-3	0.000 0e+0	0.000 0e+0	1.837 0e-2	1.224 4e-2	2.888 1e-3	1.665 9e-2	0.000 0e+0	<b>1.996 2e-2</b>
	Std	(6.26e-3)	(0.00e+0)	(0.00e+0)	(2.67e-2)	(1.73e-2)	(6.71e-3)	(2.03e-2)	(0.00e+0)	<b>(1.09e-2)</b>
	Wilcoxon	-	-	-	≈	-	-	≈	-	
UF7	Mean	2.078 3e-1	1.726 1e-1	2.612 8e-1	3.877 5e-1	3.179 2e-1	3.342 7e-1	3.565 7e-1	2.173 5e-2	<b>4.408 9e-1</b>
	Std	(1.03e-1)	(5.62e-2)	(6.09e-2)	(1.16e-1)	(6.35e-2)	(6.65e-2)	(4.50e-2)	(2.23e-2)	<b>(1.70e-2)</b>
	Wilcoxon	-	-	-	≈	-	-	-	-	
UF8	Mean	1.739 0e-1	2.628 2e-1	5.452 3e-2	2.880 1e-1	2.307 8e-1	1.565 3e-1	1.821 9e-1	2.481 7e-2	<b>3.470 6e-1</b>
	Std	(4.51e-2)	(2.32e-2)	(1.71e-2)	(5.17e-2)	(3.81e-2)	(3.31e-2)	(3.40e-2)	(2.82e-2)	<b>(1.05e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	
UF9	Mean	2.126 9e-1	2.329 4e-1	1.116 7e-1	3.218 9e-1	2.389 2e-1	2.084 4e-1	2.750 6e-1	8.241 3e-2	<b>6.099 5e-1</b>
	Std	(3.11e-2)	(1.76e-2)	(2.52e-2)	(5.49e-2)	(4.10e-2)	(4.72e-2)	(4.50e-2)	(5.59e-2)	<b>(1.83e-2)</b>
	Wilcoxon	-	-	-	-	-	-	-	-	
UF10	Mean	0.000 0e+0	<b>9.113 8e-2</b>	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	0.000 0e+0	3.079 1e-2
	Std	(0.00e+0)	<b>(2.49e-2)</b>	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(5.35e-2)
	Wilcoxon	-	+	-	-	-	-	-	-	
+/-/≈		2/12/1	1/10/4	1/13/1	2/9/4	1/12/2	1/13/1	1/11/3	1/13/1	—
Best/All		1/15	1/15	0/15	1/15	0/15	0/15	1/15	0/15	11/15

从表 3 和表 4 可以看出:本文提出的 ECMOPSO 算法综合性能明显优于对比的 8 个算法。从 Wilcoxon 秩和检验结果来看:ECMOPSO 算法与对比算法 SMPSO、dMOPSO、MPSOD、NMPSO、NSGAIII、MOEADD、SPEAR

和 DGEA 在 15 次比较中, 分别有 13、13、14、11、12、11、12、14 次的表现显著优于这些算法, 有 0、2、0、2、2、3、1、0 次得到了相似的结果。此外, 从表 3 中的最佳值可以看出: ECMOPSO 在 15 个测试问题上获得了 11 个最佳  $R_{IGD}$  值, 而算法 SMPSO、dMOPSO、MPSOD、NMPSO、NSGAIII、MOEADD、SPEAR 和 DGEA 获得的最佳  $R_{IGD}$  值个数分别为 1、0、0、1、0、1、1、0。表 4 中最佳  $R_{HV}$  值所得结果与  $R_{IGD}$  相似, ECMOPSO 算法在 15 个测试问题上也获得了 11 个最佳值。所有统计结果都证明 ECMOPSO 算法和所选的算法比较仍然具有很强的竞争力。

为进一步比较 ECMOPSO 算法与所选对比算法的整体性能, 还采用 Friedman 秩检验计算所有算法的平均排名。从表 5 可以看出: 无论是  $R_{IGD}$  值还是  $R_{HV}$  值, 提出的 ECMOPSO 算法最终排名都是第一, 这说明在和其他 8 个算法比较时, ECMOPSO 算法在这些测试问题上的整体性能较为显著。

表 5 ECMOPSO 算法和所有对比算法  $R_{IGD}$  值和  $R_{HV}$  值的 Friedman 秩检验

Table 5 Friedman rank test of  $R_{IGD}$  values and  $R_{HV}$  values of ECMOPSO and all comparison algorithms

算法	$R_{IGD}$		$R_{HV}$	
	Friedman test	Rank	Friedman test	Rank
SMPSO	5.73	7	5.43	6
dMOPSO	5.40	6	4.77	4
MPSOD	7.00	8	6.70	8
NMPSO	3.40	2	3.10	2
NSGAIII	4.93	5	5.03	5
MOEADD	4.67	4	5.43	7
SPEAR	3.87	3	4.67	3
DGEA	8.27	9	7.90	9
ECMOPSO	1.73	1	1.97	1

从  $R_{IGD}$ 、 $R_{HV}$ 、Wilcoxon 秩和检验和 Friedman 秩检验结果可以得出结论: 所提出的 ECMOPSO 算法与对比算法相比有更好的综合性能, 在求解 MOPs 时可以获得更好的收敛性和多样性。

#### 4.4 图形比较分析

为了比较各算法的收敛性和分布性, 观察它们是否真正收敛到近似 Pareto 前沿, 图 3 和图 4 分别展示 ECMOPSO 算法与其他 8 个对比算法在测试问题 ZDT3

和 UF9 上的 Pareto 最优解集分布。从图中可以看出: ECMOPSO 算法在收敛性和分布性上的表现相对于其他算法来说都更有优势。这说明精英竞争和综合平衡控制策略能更好地平衡收敛性和多样性。

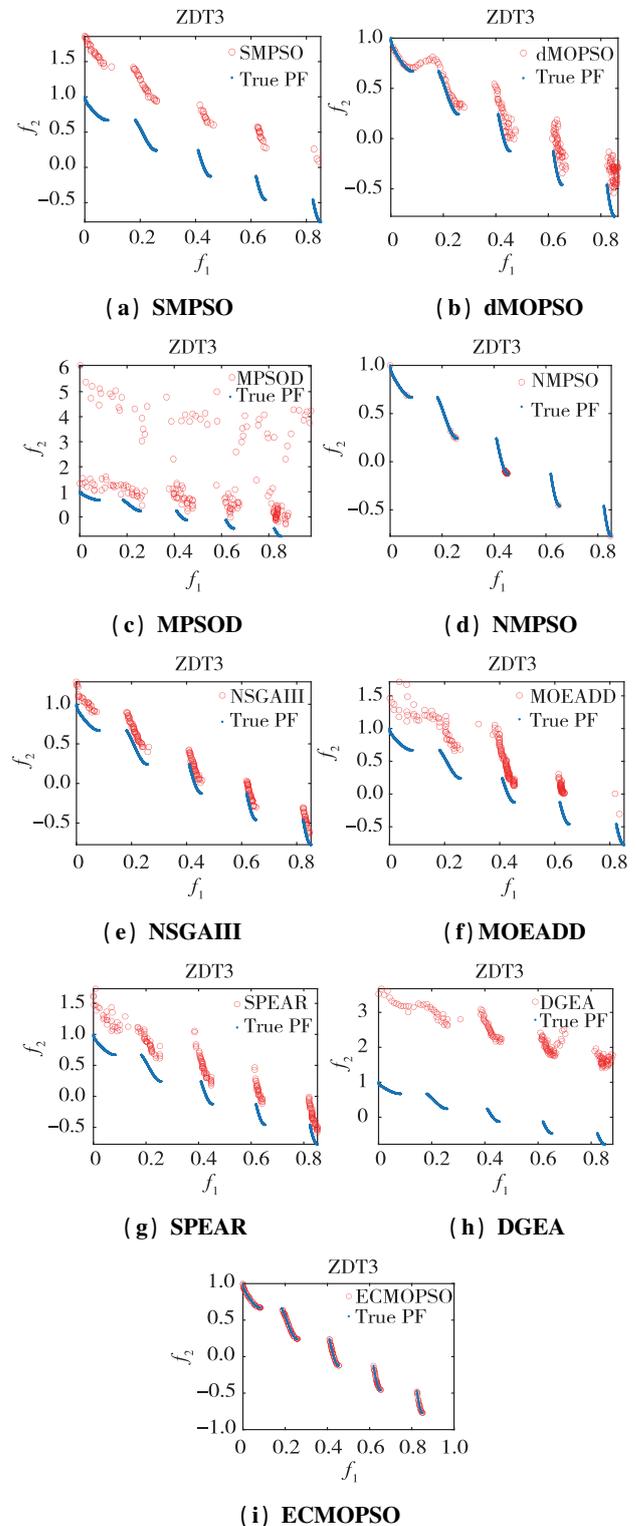


图 3 9 个算法在 ZDT3 测试问题上的近似 PF  
Fig. 3 Approximation PF of nine algorithms on ZDT3 test problems

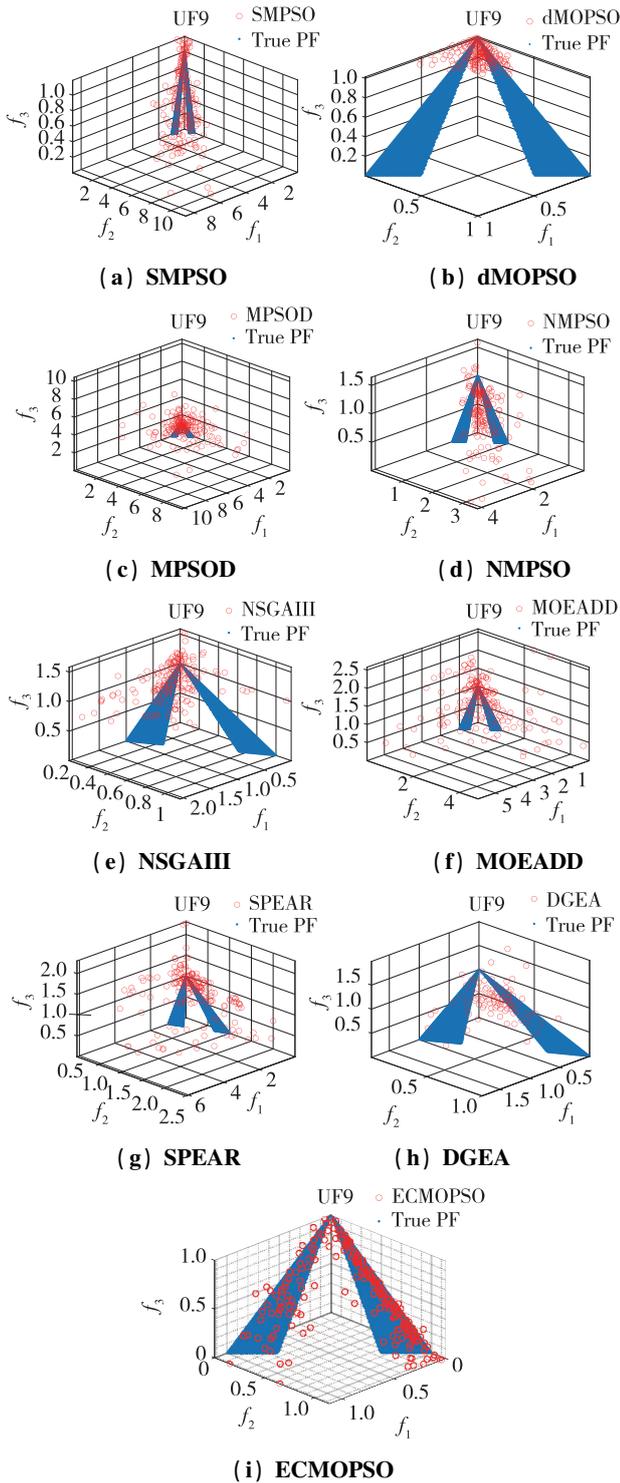


图 4 9 个算法在 UF9 测试问题上的近似 PF

Fig. 4 Approximation PF of nine algorithms on UF9 test problems

为了能够直观地比较各算法在 ZDT 和 UF 系列测试问题上的稳定性,图 5 展示了 ECMOPSO 算法与其他 8 个算法在一些测试问题上独立运行 30 次所得到的  $R_{IGD}$  值分布统计箱型图。其中箱型图横坐标上的 1、2、3、4、5、6、7、8、9 依次表示算法 SMPSO、dMOPSO、MPSOD、NMPSO、NSGAIII、MOEADD、SPEAR、DGEA 和

ECMOPSO,纵坐标表示算法的  $R_{IGD}$  值。从这些图中可以观察到:图 5 展示了与表 3 一致的对比结果,在这些测试问题上,ECMOPSO 算法无论在解的结果还是算法的稳定性上都比其他 8 个算法更为显著。

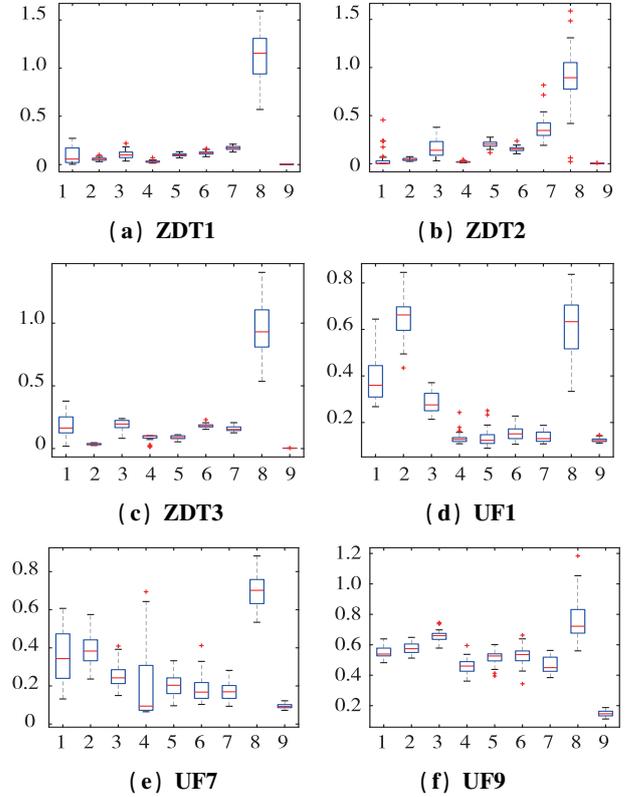
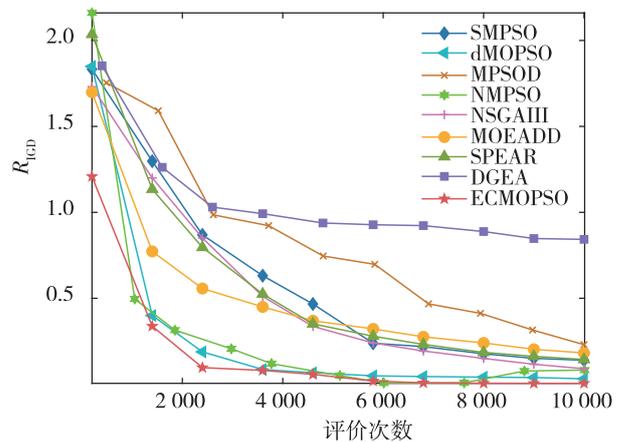


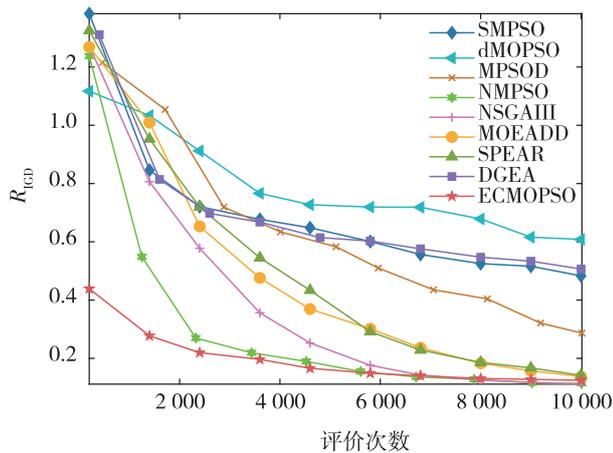
图 5 9 个算法在不同测试问题上  $R_{IGD}$  值的箱型统计图

Fig. 5 Box statistics of the  $R_{IGD}$  values of nine algorithms on different test problems

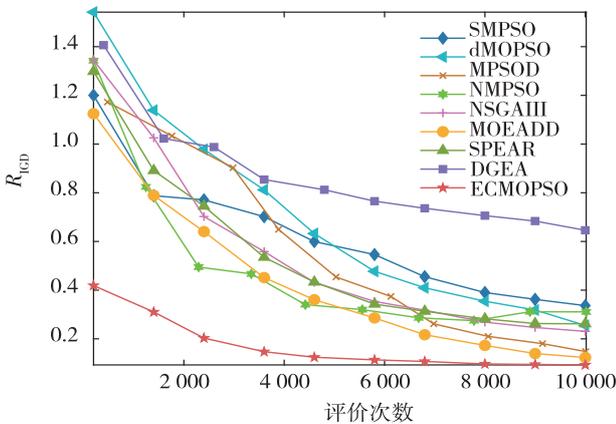
除了解集的质量,算法的收敛速度也是一个重要的性能指标。图 6 为 ECMOPSO 算法和 8 个算法在 ZDT3、UF7 和 UF9 上评价 10 000 次得到的  $R_{IGD}$  值收敛轨迹。从图中可以看出,ECMOPSO 算法具有很好的收敛速度,明显优于其他 8 个对比算法。



(a) ZDT3



(b) UF1



(c) UF7

图 6 ECMOPSO 和 8 个对比算法在 ZDT3、UF1 和 UF7 测试问题上的  $R_{IGD}$  收敛轨迹

Fig. 6  $R_{IGD}$  convergence trajectories of ECMOPSO and eight comparison algorithms on ZDT3, UF1 and UF7 test problems

综上所述,从 PF 图、箱型图和收敛轨迹图来看,所提出的 ECMOPSO 算法与所选的几个有竞争力的算法相比,ECMOPSO 算法的综合性能要优于其他算法,说明所提出的精英竞争和综合平衡控制策略可以更好地平衡算法的收敛性和多样性。

#### 4.5 参数分析

本节将对参数  $r$  进行灵敏度分析。在提出的 ECMOPSO 算法中,精英粒子集的大小  $r$  对算法的性能有一定的影响:如果  $r$  太小,可能会导致种群过早收敛,但如果  $r$  太大,可能会降低种群收敛速度。因此,合适大小的  $r$  可以很好地平衡收敛性和多样性。

图 7 展示了 ECMOPSO 算法在测试问题 UF1、UF3、UF6、UF7 和 UF8 上不同大小精英粒子集的  $R_{IGD}$  值,该算法在每个测试问题上都独立运行了 30 次。从图中可以看出,当精英粒子集的大小为 30 时,ECMOPSO 算法的性能最佳。

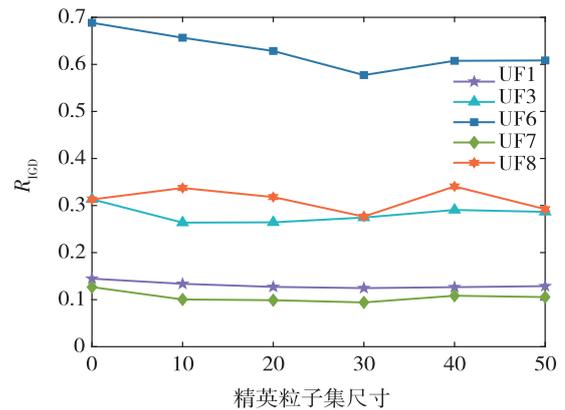


图 7 ECMOPSO 在一些测试问题上不同大小精英粒子集的  $R_{IGD}$  值

Fig. 7 The  $R_{IGD}$  values of different sizes of elite particle sets calculated by ECMOPSO on some test problems

## 5 结论

提出一种精英竞争和综合控制的多目标粒子群算法,有效平衡了算法的收敛性和多样性。ECMOPSO 算法将两两竞争引入多目标粒子群算法中,采用全局损害选择精英粒子集,然后通过精英竞争选择优胜者粒子,将其与全局领导者融合来引导种群中的粒子飞行,提升了算法的全局探索能力;并且结合全局损害和基于位移密度估计对外部存档进行维护,从而提高外部存档中非劣解的质量,平衡算法的收敛性和多样性;最后采用 Wilcoxon 秩和检验和 Friedman 秩检验来比较 ECMOPSO 算法与所选对比算法的整体性能,将 ECMOPSO 算法与 8 个对比算法在 ZDT 和 UF 测试问题上进行仿真实验。实验结果表明:ECMOPSO 算法具有更好的综合性能,可以更好地平衡收敛性和多样性,能有效地求解大多数多目标优化问题。

#### 参考文献(References):

- [1] LUO Q, WU G, JI B, et al. Hybrid multi-objective optimization approach with Pareto local search for collaborative truck-drone routing problems considering flexible time windows [J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(8): 13011—13025.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//Proceedings of the International Conference on Neural Networks. Piscataway, 1995: 1942—1948.
- [3] 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. CEC'02 IEEE, 2002: 1051—1056.
- [4] HAN H, LU W, ZHANG L, et al. Adaptive gradient

- multiobjective particle swarm optimization[J]. IEEE Transactions on Cybernetics, 2018, 48(11): 3067—3079.
- [5] LI L, CHANG L, GU T, et al. On the norm of dominant difference for many-objective particle swarm optimization [J]. IEEE Transactions on Cybernetics, 2021, 51(4): 2055—2067.
- [6] WU B, HU W, HU J, et al. Adaptive multiobjective particle swarm optimization based on evolutionary state estimation [J]. IEEE Transactions on Cybernetics, 2021, 51(7): 3738—3751.
- [7] HAN H, LU W, QIAO J. An adaptive multi-objective particle swarm optimization based on multiple adaptive methods[J]. IEEE Transactions on Cybernetics, 2017, 47(9): 2754—2767.
- [8] ZHU Q, LIN Q, CHEN W, et al. An external archive-guided multi-objective particle swarm optimization algorithm[J]. IEEE Transactions on Cybernetics, 2017, 47(9): 2794—2808.
- [9] 王世华, 李娜娜, 刘衍民. 基于双决策和快速分层的多目标粒子群算法[J]. 重庆工商大学学报(自然科学版), 2022, 39(1): 62—70.  
WANG Shi-hua, LI Na-na, LIU Yan-min. Multi-objective particle swarm optimization based on double decision and fast stratification [J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2022, 39(1): 62—70.
- [10] CHENG R, JIN Y. A competitive swarm optimizer for large scale optimization [J]. IEEE Transactions on Cybernetics, 2015, 45(2): 191—204.
- [11] HUANG W, ZHANG W. Multi-objective optimization based on an adaptive competitive swarm optimizer [J]. Information Sciences, 2022, 583(1): 266—287.
- [12] LI M, YANG S, LIU X. Shift-based density estimation for Pareto-based algorithms in many-objective optimization [J]. IEEE Transactions on Evolutionary Computation, 2013, 18(3): 348—365.
- [13] ZITZLER E, DEB K, THIELE L. Comparison of multi-objective evolutionary algorithms: Empirical results [J]. Evolutionary Computation, 2000, 8(2): 173—195.
- [14] ZHANG Q, ZHOU A, ZHAO S, et al. Multi-objective optimization test instances for the CEC 2009 special session and competition [C]//University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-objective Optimization Algorithms, Technical Report, 2008, 264: 1—30.
- [15] NEBRO A J, DURILLO J J, GARCIA-NIETO J, et al. SMPSO: A new PSO-based metaheuristic for multi-objective optimization [C]//Proceedings of the IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making (MCDM). IEEE, 2009: 66—73.
- [16] MARTINE Z S Z, COELLO C A. A multi-objective particle swarm optimizer based on decomposition [C]//Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. 2011: 69—76.
- [17] DAI C, WANG Y, YE M. A new multi-objective particle swarm optimization algorithm based on decomposition [J]. Information Sciences, 2015, 325(1): 541—557.
- [18] LIN Q, LIU S, ZHU Q, 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.
- [19] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: Solving problems with box constraints [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577—601.
- [20] LI K, DEB K, ZHANG Q, et al. An evolutionary multi-objective optimization algorithm based on dominance and decomposition [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(5): 694—716.
- [21] JIANG S, YANG S. A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(3): 329—346.
- [22] HE C, CHENG R, YAZDANI D. Adaptive offspring generation for evolutionary large-scale multi-objective optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2): 786—798.
- [23] 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.
- [24] COELLO C C A, CORTES N C. Solving multi-objective optimization problems using an artificial immune system [J]. Genetic Programming and Evolvable Machines, 2005, 6(2): 163—190.
- [25] ZITZLER E, THIELE L. Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257—271.