

# 基于种群分区的多策略自适应多目标粒子群算法

张伟<sup>1</sup> 黄卫民<sup>1</sup>

**摘要** 在多目标粒子群优化算法中,平衡算法收敛性和多样性是获得良好分布和高精度 Pareto 前沿的关键,多数已提出的方法仅依靠一种策略引导粒子搜索,在解决复杂问题时算法收敛性和多样性不足.为解决这一问题,提出一种基于种群分区的多策略自适应多目标粒子群优化算法.采用粒子收敛性贡献对算法环境进行检测,自适应调整粒子的探索和开发过程;为准确制定不同性能的粒子的搜索策略,提出一种多策略的全局最优粒子选取方法和多策略的变异方法,根据粒子的收敛性评价指标,将种群划分为 3 个区域,将粒子性能与算法寻优过程结合,提升种群中各个粒子的搜索效率;为解决因选取的个体最优粒子不能有效指导粒子飞行方向,使算法停滞,陷入局部最优的问题,提出一种带有记忆区间的个体最优粒子选取方法,提升个体最优粒子选取的可靠性并加快粒子收敛过程;采用包含双性能测度的融合指标维护外部存档,避免仅根据粒子密度对外部存档维护时,删除收敛性较好的粒子,导致种群产生退化,影响粒子开发能力.仿真实验结果表明,与其他几种多目标优化算法相比,该算法具有良好的收敛性和多样性.

**关键词** 多目标优化,粒子群算法,种群分区,多策略

**引用格式** 张伟,黄卫民.基于种群分区的多策略自适应多目标粒子群算法.自动化学报,2022,48(10):2585-2599

**DOI** 10.16383/j.aas.c200307

## Multi-strategy Adaptive Multi-objective Particle Swarm Optimization Algorithm Based on Swarm Partition

ZHANG Wei<sup>1</sup> HUANG Wei-Min<sup>1</sup>

**Abstract** In the multi-objective particle swarm optimization algorithm, balancing the convergence and diversity of the algorithm is the key to obtain the Pareto front with good distribution and accuracy. Most of the proposed methods rely on only one strategy to guide the particle search, and the algorithm may lack convergence and diversity when solving complex problems. To solve this problem, a multi-strategy adaptive multi-objective particle swarm optimization based on swarm partition is proposed. Firstly, the algorithm detects environment by the convergence contribution of particles and adjusts the process of particle exploration and exploitation adaptively. Secondly, in order to accurately formulate the search strategy of particles with different performances, a multi-strategy global optimal particle selection method and a mutation method are proposed. According to the evaluation index of the convergence of the particles, the population is divided into three regions. Combining particle performance with the algorithm optimization process can improve the search efficiency of each particle. Thirdly, an individual optimal particle selection scheme with memory interval is proposed to solve the problem that the algorithm falls into local optimization because the selected individual optimal particles cannot guide the flight direction of particles effectively. That can improve the reliability of individual optimal particle selection, and accelerate the process of particle convergence. Finally, the fusion metric including particle convergence and diversity is used to maintain the external archive. It can avoid deleting the particles with good convergence and resulting in population degradation and affecting particle development capabilities, when external archive maintenance is just based on the particle density. Experimental results show that the proposed algorithm has good performance compared with some other multi-objective optimization algorithms.

**Key words** Multi-objective optimization, particle swarm optimization, swarm partition, multi-strategy

**Citation** Zhang Wei, Huang Wei-Min. Multi-strategy adaptive multi-objective particle swarm optimization algorithm based on swarm partition. *Acta Automatica Sinica*, 2022, 48(10): 2585-2599

收稿日期 2020-05-13 录用日期 2020-09-07  
Manuscript received May 13, 2020; accepted September 7, 2020  
国家自然科学基金(61703145),河南省高校科技创新团队(20-IRTSTHN019)资助  
Supported by National Natural Science Foundation of China (61703145) and Scientific and Technological Innovation Team of Colleges and Universities in Henan Province (20IRTSTHN019)  
本文责任编辑 魏庆来

在多数实际问题和工程应用中,决策者需要同时使多个目标在决策空间内尽可能得到最佳优化,

Recommended by Associate Editor WEI Qing-Lai  
1. 河南理工大学电气工程与自动化学院 焦作 454003  
1. School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003

即多目标优化问题 (Multi-objective optimization problems, MOPs)<sup>[4]</sup>. 在 MOPs 中, 使算法快速有效地收敛在 Pareto 最优解集中并使得到的非支配解在 Pareto 前沿中保持均匀分布是衡量多目标优化算法性能的重要指标之一, 即算法的收敛性和多样性. 过度强调算法的收敛性会导致粒子聚集, 陷入局部最优; 过度增强算法多样性会导致算法收敛速度慢, 影响最终非支配解集的收敛性<sup>[2]</sup>. 因此, 平衡算法的收敛性和多样性是研究 MOPs 的重要课题之一<sup>[3-5]</sup>.

粒子群优化算法因为易于实现和计算效率高等特点, 在单目标优化问题上得到广泛应用. 当粒子群优化应用于多目标优化问题时, 即多目标粒子群优化 (Multi-objective particle swarm optimization, MOPSO). 由于 MOPs 中没有真正意义上的最优解, 所以 MOPSO 算法的全局最优粒子 *gbest* 和个体最优粒子 *pbst* 需要采取特定的策略进行选取, 并且由于 MOPSO 算法较快的收敛速度, 在算法学习过程中, 易使所得到的非支配解集迅速丧失多样性<sup>[6]</sup>.

为解决上述问题, 学者们相继提出了一些改进的 MOPSO 算法. Coello 等<sup>[7]</sup> 提出一种基于自适应网格的多目标粒子群优化算法, 采用自适应网格技术对外部存档进行维护, 指导粒子更新, 并对粒子以及粒子的飞行区域实施变异. 虽然算法相比于传统多目标进化算法在 MOPs 问题上表现出一定优势, 但在解决多局部前沿的复杂 MOPs 问题时存在困难, 且非支配解多样性较差. Raquel 等<sup>[8]</sup> 提出一种基于拥挤距离的 MOPSO (An effective use of crowding distance in MOPSO, cdMOPSO) 算法, 采用目标空间的拥挤距离评估非支配解的密集程度, 并据此来选取全局最优粒子和保持外部存档中解的分布性, 但算法仍存在收敛性不足的问题. 为解决这个问题, Yen 等<sup>[9]</sup> 提出一种动态多子群的多目标粒子群优化算法, 算法将种群分为多个子群, 通过动态调整子群大小从种群角度平衡算法的探索能力和开发能力, 实验结果表明算法性能得到改善.

与上述通过支配关系确定搜索机制的 MOPSO 算法不同, Peng 等<sup>[10]</sup> 基于分解的多目标进化算法 (Multi-objective evolutionary algorithm based on decomposition, MOEA/D) 的分解框架, 将 MOPs 分解为多个单目标优化问题, 用粒子群优化的搜索方法替代遗传算子, 从而提出一种 MOPSO 算法, 虽然其维护了一个外部存档存储各单目标优化问题的全局最优粒子, 但并未对粒子群优化的搜索机制进行优化, 算法综合性能不足. Martinez 等<sup>[11]</sup> 提出一种多目标粒子群优化器, 算法提出了一种重

新初始化策略来保持群的多样性, 当粒子存在超过一定代数时将会重新初始化, 以此来提高种群多样性和避免陷入局部极值, 但由于用分解替换了占优关系, 算法在求解一些复杂 MOPs 时, 不能覆盖整个 Pareto 前沿. 为解决这个问题, Dai 等<sup>[12]</sup> 提出一种基于分解的多目标粒子群优化算法, 根据一系列预先定义的方向向量将整个目标空间分解为一系列子空间, 并让每个子空间都有至少一个解来维护解集的多样性, 进一步提升 MOPSO 算法解决 MOPs 的能力.

学者们还提出了其他改进方法提升 MOPSO 算法的性能. Hu 等<sup>[13]</sup> 提出一种基于平行单元坐标系的自适应多目标粒子群优化 (Adaptive MOPSO based on parallel cell coordinate, pccsAMOPSO) 算法, 将目标空间的 Pareto 前沿变换为平行单元坐标形式的二维网格, 采用种群的熵作为反馈信息, 设计具有自适应调节探索和开发过程的进化策略, 使算法的收敛性和多样性都得到了提升. Kumar 等<sup>[14]</sup> 提出一种自学习的多目标粒子群优化算法, 在粒子速度和位置更新上采用 4 种运算方式, 不同方式对应不同的学习来源, 独立提高每个粒子搜索的有效性. 但目前多数改进的 MOPSO 算法仅依靠一种搜索策略更新粒子的位置和速度, 没有考虑不同性能粒子的搜索情况, 在求解复杂多目标问题时, 算法收敛性和多样性不足.

为解决上述问题, 进一步提升 MOPSO 算法的性能, 本文提出一种基于种群分区的多策略自适应多目标粒子群优化 (Multi-strategy adaptive multi-objective particle swarm optimization based on swarm partition, spmsAMOPSO) 算法. 对 MOPSO 算法的改进在于: 1) 提出一种基于种群分区的多策略全局最优粒子选取方法, 并基于种群分区思想, 提出一种多策略的变异方法, 提升粒子探索和开发能力; 2) 为提升个体最优粒子选取的可靠性, 提出一种带有记忆区间的个体最优粒子选取方法; 3) 根据算法环境, 自适应调整粒子探索和开发过程, 并采用双性能测度的融合指标维护外部存档, 平衡算法的收敛性和多样性. 实验结果表明, 相比一些其他多目标优化算法, 本文算法在收敛性和多样性上具有一定优势.

## 1 多目标优化问题和粒子群优化算法

### 1.1 多目标优化问题

多目标优化问题的数学模型可描述为:

$$\min y(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})), \mathbf{x} \in \Omega^n \quad (1)$$

式中,  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  为问题的决策变量;  $\Omega^n$  为决策变量可行解空间;  $y: \Omega^n \rightarrow \mathbf{R}^M$  是决策空间到目标空间的映射;  $M$  为目标空间维数.

有关求解 MOPs 的 Pareto 最优定义如下<sup>[14]</sup>:

1) Pareto 支配: 对任意的  $\mathbf{x}_a, \mathbf{x}_b \in \Omega^n$ , 是问题的两个可行解. 式中,  $\mathbf{x}_a = (x_{a,1}, x_{a,2}, \dots, x_{a,n})$ ,  $\mathbf{x}_b = (x_{b,1}, x_{b,2}, \dots, x_{b,n})$ , 当且仅当:

$$\begin{cases} \forall i \in [1, M], & f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b) \\ \wedge \exists j \in [1, M], & f_j(\mathbf{x}_a) < f_j(\mathbf{x}_b) \end{cases} \quad (2)$$

则称  $\mathbf{x}_a$  支配  $\mathbf{x}_b$  (记为  $\mathbf{x}_a \prec \mathbf{x}_b$ ).

2) Pareto 最优解: 对任意的  $\mathbf{x}' \in \Omega^n$  为问题的可行解, 当且仅当:

$$\neg \exists \mathbf{x} \in \Omega^n, \mathbf{x} \prec \mathbf{x}' \quad (3)$$

则称  $\mathbf{x}'$  为 Pareto 最优解.

3) Pareto 最优解集: 对给定的 MOPs, Pareto 最优解集定义为:

$$P^* = \{\mathbf{x} \in \Omega^n | \neg \exists \mathbf{x}' \in \Omega^n, \mathbf{x}' \prec \mathbf{x}\} \quad (4)$$

4) Pareto 前沿: 由  $P^*$  的定义, Pareto 前沿 (Pareto front, PF) 可定义为:

$$PF = \{y = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})) | \mathbf{x} \in P^*\} \quad (5)$$

## 1.2 标准粒子群算法

粒子群算法是由 Kennedy 等<sup>[15]</sup>提出的一种基于种群搜索的进化计算技术. 种群中粒子  $i$  的速度  $\mathbf{v}_i$  和位置  $\mathbf{x}_i$  的更新方法分别为:

$$v_{i,d}^{t+1} = w_p v_{i,d}^t + c_1^t \text{rand}_1 (pbest_{i,d}^t - x_{i,d}^t) + c_2^t \text{rand}_2 (gbest_{i,d}^t - x_{i,d}^t) \quad (6)$$

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1} \quad (7)$$

式中,  $d = 1, 2, \dots, D$ ,  $D$  为问题的决策空间维数;  $v_{i,d}^t$  和  $x_{i,d}^t$  分别为  $t$  时刻粒子  $i$  第  $d$  维变量的速度和位置;  $pbest_{i,d}^t$  和  $gbest_{i,d}^t$  分别为  $t$  时刻粒子  $i$  的个体最优粒子和全局最优粒子的第  $d$  维变量;  $w_p$  为  $t$  时刻算法惯性权重,  $c_1^t$  和  $c_2^t$  为  $t$  时刻算法学习因子,  $\text{rand}_1$  和  $\text{rand}_2$  为  $[0, 1]$  区间上均匀分布的随机数.

## 2 基于种群分区的多策略自适应多目标粒子群优化算法

MOPSO 算法因其特殊的搜索机制和快速的收敛速度, 需要对算法的参数调整、全局最优粒子选取、个体最优粒子选取、外部存档维护以及粒子变异操作等部分制定策略以引导粒子搜索<sup>[7]</sup>. 本文在全局最优粒子选取和变异操作部分, 提出一种基于

种群分区的多策略改进方法, 并提出一种带有记忆区间的个体最优粒子选取方法, 结合自适应参数调整以及使用融合指标的外部存档维护策略, 进一步提升 MOPSO 算法性能. 本文 spmsAMOPSO 算法的整体结构如图 1 所示, 通过算法的参数调整、最优粒子选取、变异以及外部存档维护等部分之间的协同, 共同指导粒子寻优过程, 提升算法综合性能.

### 2.1 自适应调整惯性权重和学习因子

对于 MOPSO 算法, 由于算法迭代过程中, 其环境是实时变化的, 为提升粒子探索和开发能力, 本文根据算法环境变化调整惯性权重  $w_p$  以及学习因子  $c_1$  和  $c_2$ , 达到算法参数精细化调控的目的.

当算法搜索到的非支配解需要加入外部存档时, 计算其对外部存档中解的支配程度, 得到粒子的收敛性贡献<sup>[16]</sup>:

$$Cv(\mathbf{x}_{n,i}, rep) = \max_{\mathbf{x}_r \in rep} \rho(\mathbf{x}_{n,i}, \mathbf{x}_r) \quad (8)$$

式中,  $\mathbf{x}_{n,i}$  为算法寻优所得到的第  $i$  个非劣解;  $\mathbf{x}_r$  为外部存档中存储的非支配解;  $rep$  为算法外部存档;  $\rho(\mathbf{x}_{n,i}, \mathbf{x}_r)$  为  $\mathbf{x}_{n,i}$  对  $\mathbf{x}_r$  的支配程度, 计算方法如下:

$$\rho(\mathbf{x}_{n,i}, \mathbf{x}_r) = \begin{cases} \frac{1}{M} \sum_{j=1}^M \frac{|f_j(\mathbf{x}_{n,i}) - f_j(\mathbf{x}_r)|}{(f_{j,\max} - f_{j,\min})}, & \mathbf{x}_{n,i} \prec \mathbf{x}_r \\ 0, & \text{其他} \end{cases} \quad (9)$$

式中,  $M$  为目标个数;  $f_{j,\max}$  和  $f_{j,\min}$  分别为外部存档中非支配解在第  $j$  个目标上的最大值和最小值. 若  $t$  时刻算法搜索到  $n_p$  个非支配解, 则算法的环境检测参数  $A_p$  定义为:

$$A_p^t = \max_{i=1,2,\dots,n_p} Cv(\mathbf{x}_{n,i}, rep) \quad (10)$$

粒子对外部档案的收敛性贡献反映算法当前的收敛状态, 当  $A_p$  较大时, 种群处于探索阶段, 应保持其全局搜索能力, 随着  $A_p$  逐渐减小, 种群逐渐达到收敛, 应加强局部搜索, 当  $A_p$  趋于 0 时, 表明种群达到收敛稳定状态.

由文献 [17] 可知, 较小的惯性权重  $w_p$  和学习因子  $c_1$  以及较大的学习因子  $c_2$  有利于算法收敛, 较大的  $w_p$  和  $c_1$  以及较小的  $c_2$  有利于增强算法多样性. 为避免参数调整影响算法搜索连贯性, 本文采用增量式参数调节方法:

$$\xi^t = \begin{cases} \xi^{t-1}, & \Delta A_p = 0 \\ \xi^{t-1} + \gamma \Delta A_p, & \Delta A_p > 0 \\ \xi^{t-1} - \gamma \Delta A_p, & \Delta A_p < 0 \end{cases} \quad (11)$$

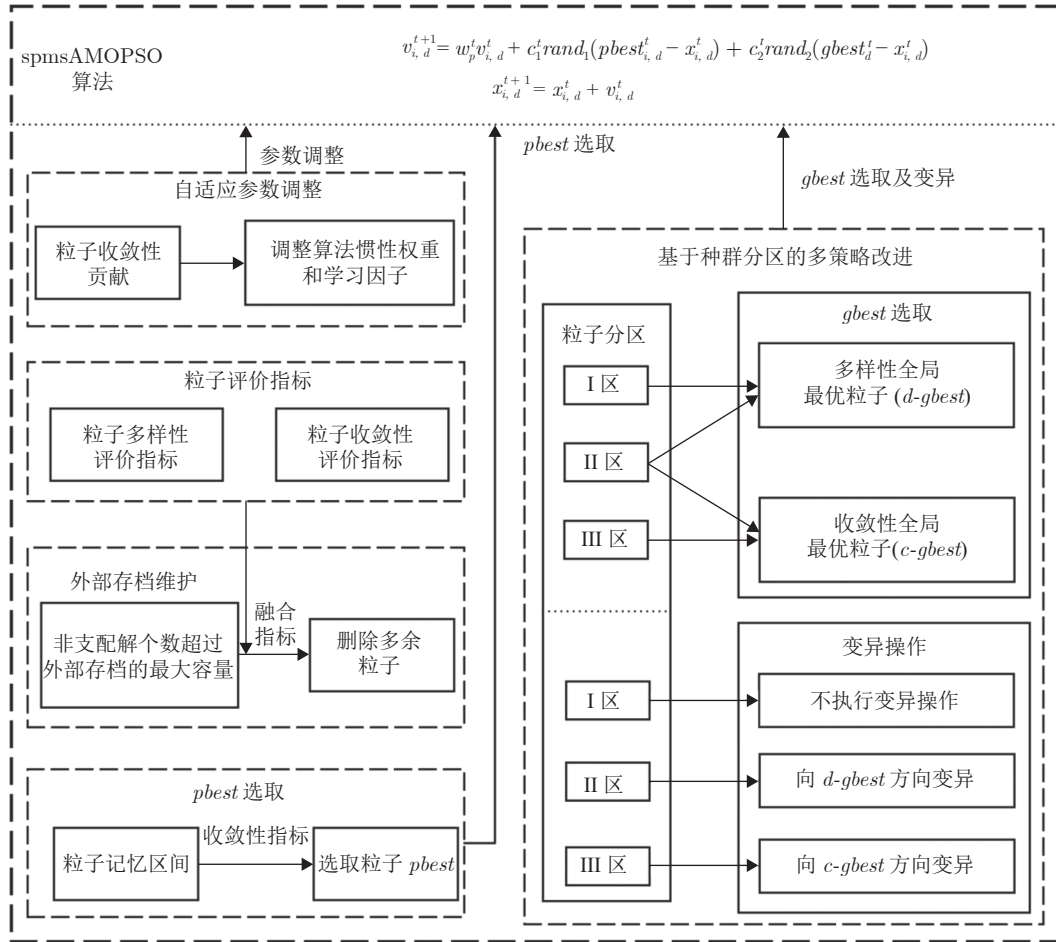


图 1 spmsAMOPSO 算法整体框图  
Fig.1 Frame of spmsAMOPSO algorithm

式中,  $\xi^t = [w_p^t, c_1^t, c_2^t]^T$ , 由  $t$  时刻  $w_p$ 、 $c_1$  和  $c_2$  的值构成;  $\gamma = [\lambda_{w_p}, \lambda_{c_1}, -\lambda_{c_2}]^T$ , 由  $w_p$ 、 $c_1$  和  $c_2$  的增量步长因子构成;  $\Delta A_p$  为收敛性贡献增量, 计算方法为  $\Delta A_p = A_p^t - A_p^{t-1}$ ; 设置  $w_p$  的上下限分别为 0.9 和 0.4,  $c_1$  和  $c_2$  的变化范围均为区间  $[0.5, 2.5]$ <sup>[18]</sup>.

### 2.2 基于种群分区的多策略全局最优粒子选取

全局最优粒子 ( $gbest$ ) 选取是影响 MOPSO 算法收敛性和多样性的重要问题. 本文采用不同的评价指标分别选取促进算法收敛的全局最优粒子 ( $c-gbest$ ) 和增强算法多样性全局最优粒子 ( $d-gbest$ ), 并根据粒子性能将种群划分为多个区域, 对不同区域粒子制定不同的  $gbest$  选取策略.

#### 2.2.1 $c-gbest$ 和 $d-gbest$ 的选取方法

为保证算法的收敛性, 在算法寻优过程中选取外部存档中收敛程度最好的粒子做为算法的  $c-gbest$ .

本文采用粒子的平均排名和全局损害作为粒子的收敛性评价指标.

粒子平均排名 (Average ranking, AR) 的计算方法为:

$$AR(\mathbf{x}_i) = \sum_{m=1}^M r_m(\mathbf{x}_i) \quad (12)$$

式中,  $r_m(\mathbf{x}_i)$  为粒子  $i$  在第  $m$  个目标变量上的排名;  $M$  为目标个数.

粒子全局损害 (Global detriment, GD) 的计算方法如下:

$$GD(\mathbf{x}_i) = \sum_{\mathbf{x}_i \neq \mathbf{x}_j} \sum_{m=1}^M \max(f_m(\mathbf{x}_i) - f_m(\mathbf{x}_j), 0) \quad (13)$$

式中,  $f_m(\mathbf{x}_i)$  为粒子  $i$  在第  $m$  个目标上的适应度值;  $GD(\mathbf{x}_i)$  表示粒子  $i$  比其他粒子在目标上的劣势.

由于通过 AR 指标选取最优粒子时容易使粒子聚集在 Pareto 前沿的边缘, 而 GD 指标容易排除

Pareto 前沿中的端点粒子<sup>[6]</sup>. 将上述两个指标结合, 改善评价方式缺陷:

$$FR(\mathbf{x}_i) = \eta_1 AR(\mathbf{x}_i) + \eta_2 GD(\mathbf{x}_i) \quad (14)$$

式中,  $\eta_1$  和  $\eta_2$  为区间  $[0, 1]$  之间的常数, 用于确定指标 AR 和 GD 的权重, 其取值分别为:  $\eta_1 = 0.4$ 、 $\eta_2 = 0.6$ , 具体讨论分析见第 3.2.1 节. Fusion ranking (FR) 指标有利于目标解的序列和个体之间的相对信息结合, 从而更全面地评估解空间内个体的收敛程度. 由式 (12) ~ (14) 计算外部存档中各粒子的融合排序 FR 指标, 选取 FR 指标最小的粒子作为算法的  $c$ - $gbest$ .

为保证非支配解集的多样性, 在算法寻优过程中应引导粒子向外部存档中非支配解较为稀疏的区域飞行, 本文采用粒子的拥挤距离作为算法多样性全局最优粒子 ( $d$ - $gbest$ ) 的选取标准.

粒子  $i$  与粒子  $j$  之间的欧氏距离为:

$$dis(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{m=1}^M (f_m(\mathbf{x}_i) - f_m(\mathbf{x}_j))^2} \quad (15)$$

为避免个别极端粒子影响粒子密度估计, 本文采用局部距离作为粒子的密度评价指标, 取粒子  $i$  距离最近的  $S$  个粒子之间距离的平均值作为粒子  $i$  的拥挤距离, 即:

$$d_s(\mathbf{x}_i) = rank(dis(\mathbf{x}_i)) \quad (16)$$

$$LD(\mathbf{x}_i) = \frac{1}{S} \sum_{s=1}^S d_s(\mathbf{x}_i) \quad (17)$$

局部拥挤距离 (Locality crowding distance, LD) 为反映粒子分布情况的指标, 粒子的 LD 指标越大, 表明粒子分布越稀疏. 根据式 (15) ~ (17) 计算外部存档中各粒子的 LD 指标, 选取 LD 指标最大的粒子作为算法的  $d$ - $gbest$ .

### 2.2.2 种群分区及各区域粒子的 $gbest$ 选取方法

为增强粒子探索和开发能力, 并根据算法环境自适应调整种群探索和开发过程, 扩大粒子搜索区域的同时保持算法收敛性和多样性的平衡. 本文通过粒子收敛程度将种群中粒子划分为 3 个区域, 根据不同区域粒子的特征, 提出一种多策略的  $gbest$  选取方法.

根据种群中各粒子的 FR 指标大小, 得到每个粒子基于指标 FR 的种群内排名:

$$FRs^t = rank(FR(pop^t)) \quad (18)$$

根据每个粒子的群内排名 FRs, 确定各粒子的所属区域:

$$region(\mathbf{x}_i) = \begin{cases} \text{I 区}, & FRs(\mathbf{x}_i) < \alpha \\ \text{II 区}, & \alpha \leq FRs(\mathbf{x}_i) < \beta \\ \text{III 区}, & FRs(\mathbf{x}_i) \geq \beta \end{cases} \quad (19)$$

式中,  $\alpha$  和  $\beta$  均为正常数, 通过确定  $\alpha$  和  $\beta$  的值, 将种群划分为 3 个区域, 分区参数选取分别为:  $\alpha = 0.2N$ ,  $\beta = 0.8N$ ,  $N$  为种群粒子个数, 分区参数的取值对算法影响分析见第 3.2.2 节. 图 2 给出了种群各区域粒子的划分情况.

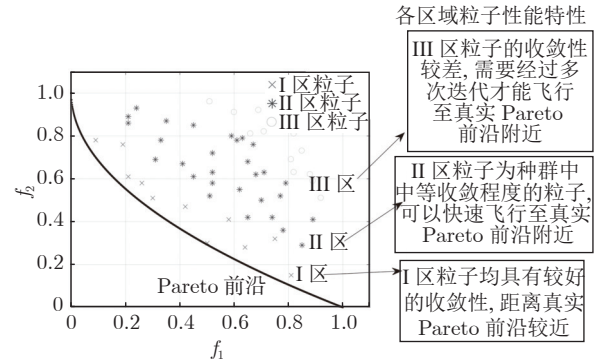


图 2 种群中各粒子所属区域

Fig.2 Location of each particles in the population

根据各区域内粒子的不同特征, 制定一种多策略的  $gbest$  选取方案:

1) 对于 I 区粒子, 其在种群中具有较好的收敛性, 为保持算法搜索到非支配解的多样性, 使 I 区粒子选取  $d$ - $gbest$  作为全局最优粒子指导其飞行.

2) 对于 II 区粒子, 为平衡算法的收敛性和多样性, 使种群能够探索更多区域, 使 II 区粒子根据算法环境进行全局最优粒子的选取, 由第 2.1 节中的粒子收敛贡献度对算法当前环境进行估计, 根据算法环境确定粒子选取  $c$ - $gbest$  或  $d$ - $gbest$  作为  $gbest$  的概率:

$$gbest_i^t = \begin{cases} c\text{-}gbest^t, & rand < Ap^t \\ d\text{-}gbest^t, & \text{其他} \end{cases} \quad (20)$$

由式 (20) 使 II 区粒子能够根据算法当前环境进行自适应探索和开发. 当  $Ap$  较大时, 为保证算法收敛性, 使粒子有较大概率选取  $c$ - $gbest$  作为  $gbest$  指导其飞行; 当  $Ap$  逐渐减小时, 为增强算法的多样性, 应使粒子有较大概率选取  $d$ - $gbest$  作为  $gbest$ .

3) 对于 III 区粒子, 由于其在种群中性能较差, 对种群中的其他粒子的飞行没有太大贡献, 为提升粒子的开发能力, 保证算法收敛性, 使 III 区粒子选取  $c$ - $gbest$  作为全局最佳粒子指导其飞行.

通过上述多策略的  $gbest$  选取方法, 使算法能够根据不同区域的粒子性能和算法环境, 选取  $c$ -

$gbest$  或  $d-gbest$  作为全局最优粒子, 有效提升算法的收敛性和多样性, 增强粒子探索和开发能力.

### 2.3 基于种群分区的多策略高斯变异

由于 MOPSO 算法具有快速的收敛过程, 算法易发生早熟收敛, 过早结束搜索过程, 陷入局部最优. 为解决这个问题, 在粒子位置更新时, 对部分粒子加入扰动, 使算法跳出局部最优, 扩大粒子搜索区域<sup>[19]</sup>. 基于种群分区, 针对不同区域粒子的性能, 提出一种多策略的变异方法:

1) 若粒子  $i$  位于 I 区, 表明粒子  $i$  距离真实 Pareto 前沿较近, 为避免种群产生退化, 保留种群中的精英个体, 粒子  $i$  不进行变异操作;

2) 若粒子  $i$  位于 II 区, 则其具有较好的性能, 可以通过较少的迭代次数飞行至真实 Pareto 前沿附近; 若  $\delta < p_m$ , 将粒子  $i$  向  $d-gbest$  方向进行变异, 增强算法多样性. 变异方法如下:

$$x_{i,d} = x_{i,d} + \delta(x_{\max,d} - x_{\min,d})(d-gbest_d - x_{i,d}) \times \exp\left(-\left(\frac{d-gbest_d - x_{i,d}}{\sigma}\right)^2\right) \quad (21)$$

3) 若粒子  $i$  位于 III 区, 则粒子性能较差, 距离真实 Pareto 前沿较远, 优先考虑增强粒子的收敛性; 若  $\delta < (1 - p_m)$ , 将粒子  $i$  向  $c-gbest$  方向进行变异:

$$x_{i,d} = x_{i,d} + \delta(x_{\max,d} - x_{\min,d})(c-gbest_d - x_{i,d}) \times \exp\left(-\left(\frac{c-gbest_d - x_{i,d}}{\sigma}\right)^2\right) \quad (22)$$

式中,  $\delta$  为区间  $[0, 1]$  之间的随机数;  $d$  为区间  $[1, D]$  之间的随机整数,  $D$  为问题的决策空间维数;  $x_{\max,d}$  和  $x_{\min,d}$  分别为  $d$  维决策空间上的最大值和最小值;  $p_m$  为粒子变异因子, 取值为  $p_m = 0.25 + (t/2T)$ ;  $\sigma$  为高斯函数方差, 根据算法迭代次数更新:  $\sigma = \sigma_0^2 \exp(-t/T)$ ; 方差  $\sigma$  控制变异范围, 随迭代次数增加, 使粒子的变异范围越来越小, 保证算法迭代后期粒子的开发能力.

上述变异方法具有以下 2 个优点: 1) 算法能够根据粒子位置进行不同程度的变异, 使算法跳出局部最优的同时避免对粒子搜索产生太大干扰, 保持算法搜索连贯性. 2) 根据不同区域粒子的性能差异, 制定不同的变异策略, 扩大粒子的搜索区域, 增强粒子的探索和开发能力.

### 2.4 带有记忆区间的个体最优粒子选取方法

合理选择个体最优粒子 ( $pbest$ ) 有助于提高种群对局部空间的开发能力. 多数 MOPSO 算法均采

用一个外部库来存储粒子的  $pbest$ , 当粒子位置  $x_i$  受  $pbest_i$  支配时,  $pbest_i$  保持, 当  $x_i$  支配  $pbest_i$  时, 将  $pbest_i$  更新为粒子当前位置  $x_i$ , 两者无支配关系时, 采用随机方法进行选取. 这种方法虽然计算简单, 但当  $x_i$  与  $pbest_i$  无支配关系时, 选取的  $pbest$  不能有效指导粒子的飞行方向, 易导致算法停滞<sup>[20]</sup>.

基于上述问题, 本文提出一种带有记忆区间的  $pbest$  选取方法, 在算法迭代过程中, 每个粒子具有一个外部存储库, 存储粒子最近几次迭代所经过的位置, 形成记忆区间:

$$Me_i^t = [x_i^{t-u}, x_i^{t-u+1}, \dots, x_i^{t-1}, x_i^t] \quad (23)$$

式中,  $x_i^t$  为粒子  $i$  第  $t$  次迭代时的位置;  $u$  为粒子的记忆区间长度.

根据第 2.2.1 节中粒子的融合排序方法, 计算记忆区间中各位置的 FR 指标, 选取 FR 指标最小的位置作为粒子  $i$  下一次迭代时的  $pbest$ :

$$pbest_i^{t+1} = \min_{i=1,2,\dots,n} (FR(Me_i^t)) \quad (24)$$

当记忆区间饱和时, 需要制定策略对记忆区间进行维护, 图 3 以 2 个粒子为例, 给出了粒子的记忆区间更新过程.

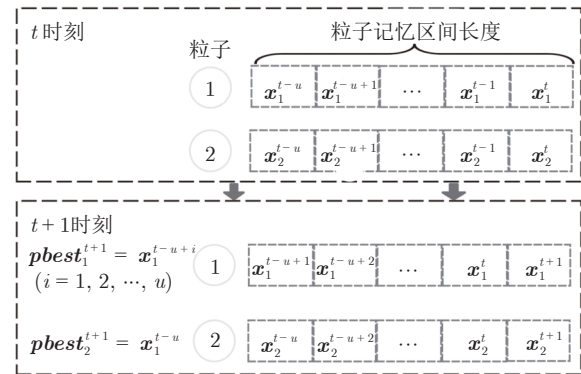


图 3 粒子记忆区间更新过程

Fig. 3 Update process of particle memory interval

由图 3 可以看出, 粒子记忆区间更新存在两种情况: 1) 若存储位置达到记忆区间最大值, 则将记忆中最早一次迭代时的位置遗忘, 即  $x_i^{t-u}$ . 2) 若  $x_i^{t-u}$  为最优的粒子位置, 则按时间顺序向后, 将相邻的一个位置遗忘. 即:

$$Me_i^{t+1} = \begin{cases} [x_i^{t-u}, x_i^{t-u+2}, \dots, x_i^t], & pbest_i^{t+1} = x_i^{t-u} \\ [x_i^{t-u+1}, x_i^{t-u+2}, \dots, x_i^t], & \text{其他} \end{cases} \quad (25)$$

### 2.5 基于双性能测度融合指标的外部存档维护

算法在迭代过程中将非支配解存入外部存档

中, 随着算法迭代次数增加, 非支配解个数随之增加, 当达到外部存档的最大容量时, 需要对外部存档进行维护. 多数 MOPSO 算法在进行外部存档维护时, 采用粒子的分布性能指标作为判断粒子性能的标准, 将密度最大的粒子进行删除. 这种方法虽然能够保持外部档案中非支配解的多样性, 但可能会删除外部存档中收敛性较好的非支配解, 导致种群产生退化, 影响粒子开发能力<sup>[21]</sup>. 因此, 在进行外部存档维护时不应将粒子密度作为外部存档维护的唯一性能指标, 还应考虑到粒子的收敛性.

根据粒子的收敛性指标  $FR$  和粒子的分布性指标  $LD$ , 采用融合的评价指标判断外部存档中粒子的综合性能:

$$CR(x_i) = \frac{FR(x_i)}{LD(x_i)} \quad (26)$$

由式 (12) 和式 (14) 可知, 粒子  $FR$  指标越小, 粒子收敛性越好,  $LD$  指标越大, 粒子分布性越好. 由式 (26) 可以看出, 粒子的综合排序 (Comprehensive ranking, CR) 指标不仅能够反映粒子的收敛程度而且包含了粒子的分布情况, 粒子的 CR 指标越小, 表明粒子的综合性能越好. 因此, 当算法搜索到的非支配解个数超过外部存档的最大容量时, 将外部存档中 CR 指标较大的粒子删除.

### 3 实验分析

#### 3.1 性能评价指标

为有效评价算法的收敛性和多样性, 本文采用反世代距离 (Inverted generational distance, IGD)、间隔指标 (Spacing, SP)、出错比率 (Error ratio, ER) 和计算时间, 分别对算法性能进行评价.

1) IGD: IGD 是度量真实 Pareto 前沿与算法获得的近似 Pareto 前沿之间的距离指标<sup>[13]</sup>. IGD 值越小, 表明算法获得的近似 Pareto 前沿的收敛性和多样性越好, 越接近真实 Pareto 前沿, 其计算方法为:

$$IGD(PF, PF^*) = \frac{1}{n} \sum_{i=1}^n \min dis(PF_i, PF^*) \quad (27)$$

式中,  $PF$  为多目标算法所求得 Pareto 前沿;  $PF^*$  为真实 Pareto 前沿的一组采样点;  $n$  为 Pareto 前沿中非支配解个数.

2) SP: 用于度量每个解到其他解最小距离的标准差, 是衡量一个范围内邻近解差异的重要指标<sup>[22]</sup>. SP 值越小, 表明解集分布越均匀, 其计算方法为:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (28)$$

式中,  $d_i$  为第  $i$  个非支配解与其他解之间最小的欧氏距离;  $\bar{d}$  为所有  $d_i$  的平均值.

3) ER: 用来说明最优解的百分比<sup>[7]</sup>. ER 越小, 所得非支配解在真实 Pareto 前沿中占比越高, 其计算方法为:

$$ER = \frac{1}{n} \sum_{i=1}^n e_i \quad (29)$$

式中,  $e_i$  取值方法为: 若第  $i$  个非支配解在真实 Pareto 前沿中, 则  $e_i = 0$ , 否则  $e_i = 1$ .

4) 计算时间  $t$ : 计算时间是多目标优化算法的重要性能评价指标之一<sup>[23]</sup>, 相同实验平台下算法的计算时间能够表征各算法的计算复杂度.

#### 3.2 参数取值和分析

本节通过本文算法在不同参数取值方案下对 ZDT3 和 DTLZ2 测试函数的实验, 说明参数的取值方法. 实验基本参数设置: 种群粒子数量为 100, 外部存档最大容量为 200, 粒子记忆区间长度为  $u = 5$ , 粒子拥挤距离中  $S = 4$ , 变异策略中高斯函数的方差初始值  $\sigma_0 = 0.8$ .

##### 3.2.1 融合收敛性指标 $FR$ 的权重 $\eta_1$ 和 $\eta_2$

由式 (14) 所描述的融合收敛性指标  $FR$  的权重  $\eta_1$  和  $\eta_2$  的取值对粒子的收敛性评价产生影响, 进而影响算法性能, 合适的  $\eta_1$  和  $\eta_2$  取值能够有效对粒子进行分区, 提升算法开发能力. 本节通过多种参数取值下的实验, 说明  $\eta_1$  和  $\eta_2$  的取值原因. 选取 7 种参数取值方案:

Case 1:  $\eta_1 = 0.2, \eta_2 = 0.8$ ;

Case 2:  $\eta_1 = 0.3, \eta_2 = 0.7$ ;

Case 3:  $\eta_1 = 0.4, \eta_2 = 0.6$ ;

Case 4:  $\eta_1 = 0.5, \eta_2 = 0.5$ ;

Case 5:  $\eta_1 = 0.6, \eta_2 = 0.4$ ;

Case 6:  $\eta_1 = 0.7, \eta_2 = 0.3$ ;

Case 7:  $\eta_1 = 0.8, \eta_2 = 0.2$ .

分别对两目标问题 ZDT3 和三目标问题 DTLZ2 进行实验, 测试 ZDT3 问题时, 每 20 次迭代进行一次 IGD 指标评价, 测试 DTLZ2 问题时, 每 30 次迭代进行一次 IGD 指标评价, 实验结果如图 4 所示.

由图 4 可以看出, 在 7 种参数取值方案中, 算法在 Case 3 ~ Case 5 的取值方案下, 性能较好, 即  $\eta_1$  和  $\eta_2$  的取值大小接近时, 算法能够获得良好的性能. 这是由于粒子平均排名 AR 和粒子全局损害

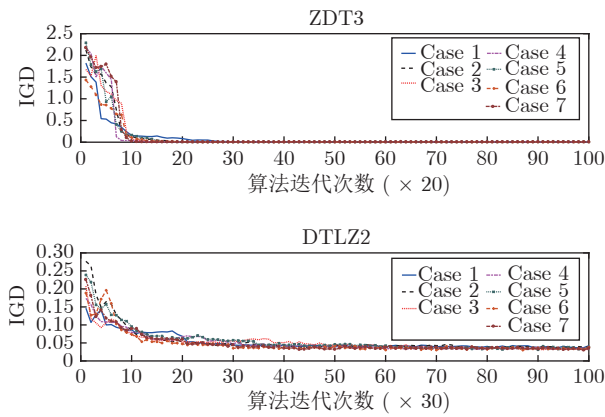


图4  $\eta_1$  和  $\eta_2$  的不同取值方案下 IGD 指标变化  
Fig.4 IGD metric changes of different  $\eta_1$  and  $\eta_2$  value schemes

GD 的计算方法造成的. 粒子收敛性融合指标 FR 由 AR 和 GD 的加权和构成, 种群中粒子个数为 100 时, 粒子的 AR 指标略大于 GD 指标, 当  $\eta_1$  的取值远大于  $\eta_2$  时, GD 指标在粒子收敛性评价中几乎被忽略, 即  $FR \approx AR$ . 粒子的收敛性评价将由 AR 主导, 导致粒子收敛性评价不全面, 算法性能下降, 反之亦然. 为保证本文算法性能, 本文中  $\eta_1$  和  $\eta_2$  的取值分别为 0.4 和 0.6.

### 3.2.2 分区参数 $\alpha$ 和 $\beta$

第 2.2.2 节通过分区参数  $\alpha$  和  $\beta$  将种群划分为 3 个区域, I 区和 III 区最优粒子的选取可以保证算法基本的收敛性和多样性, II 区粒子能够根据算法当前环境进行自适应最优粒子的选取, 提升粒子的探索和开发能力. 因此,  $\alpha$  和  $\beta$  的取值将影响算法的收敛速度和综合性能. 图 5 给出本文算法解决 ZDT3 和 DTLZ2 问题时, 3 种  $\alpha$  和  $\beta$  取值方案下 IGD 指标下降趋势, 取值方案分别为: Case 1:  $\alpha = 0.1N$ ,  $\beta = 0.9N$ ; Case 2:  $\alpha = 0.2N$ ,  $\beta = 0.8N$ ; Case 3:  $\alpha = 0.3N$ ,  $\beta = 0.7N$ . 式中  $N$  为粒子总个数.

由图 5 可以看出, 当  $\alpha$  和  $\beta$  的取值使 II 区粒子数量较多时, 算法收敛速度较慢, 当 II 区粒子数量较少时算法收敛速度较快. 由 ZDT3 测试问题的 Case 3 可以看出, 当 II 区粒子数量较少时, 算法的 IGD 指标下降陡峭, 并且算法迭代过程中 IGD 指标下降无明显波动, 表明算法在搜索过程中由于 II 区粒子数量不足, 导致粒子的探索能力不足; 由 DTLZ2 测试问题可以看出, 随着 II 区粒子的增多, 算法获得的最终 IGD 指标逐渐减小, 算法的综合性能逐渐提升.

为避免算法收敛速度过快, 保证种群中粒子的探索和开发能力, 应使 II 区粒子保持较高数量. 通

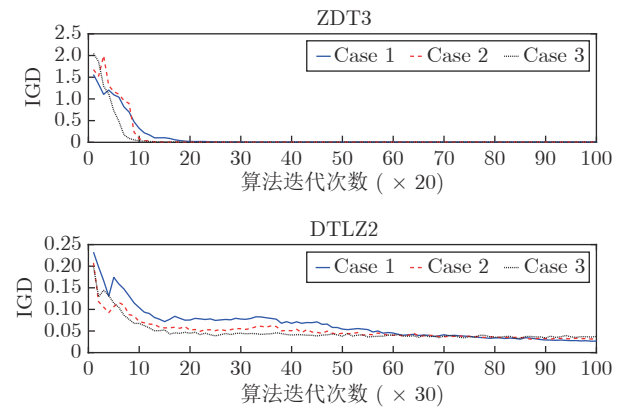


图5 不同分区参数取值方案下 IGD 指标变化  
Fig.5 IGD metric changes of different partition parameter value schemes

过大量实验验证,  $\alpha$  和  $\beta$  的取值使 I 区、II 区和 III 区粒子数分别约为粒子总数的 1/5、3/5 和 1/5, 能够使算法保持合适的收敛速度并获得良好的综合性能.

### 3.3 实验结果及比较分析

为验证本文算法的有效性, 分别对两目标问题 (ZDT 系列函数) 和三目标问题 (DTLZ 系列函数) 进行实验测试, 并设置 8 种多目标算法进行对比实验, 对比算法包括: 基于聚类的多目标粒子群优化算法 (MOPSO based on clustering, clusterMOPSO)<sup>[23]</sup>、cdMOPSO<sup>[8]</sup>、pccsAMOPSO<sup>[13]</sup>、非支配排序遗传算法 (Nondominated sorting genetic algorithm II, NSGA-II)<sup>[24]</sup>、增强 Pareto 进化算法 (Strength Pareto evolutionary algorithm, SPEA2)<sup>[25]</sup>、MOEA/D<sup>[26]</sup>、基于竞争机制的多目标粒子群优化算法 (Competitive mechanism based MOPSO, CMOPSO)<sup>[27]</sup> 和嵌入基于分解归档方法的 SPEA2 算法 (Decomposition-based archiving approach embedded in SPEA2, SPEA2 + DA-A)<sup>[28]</sup> 算法. 各算法对测试问题的最大评价次数均为 1000 次, 实验结果均为各算法在同一测试问题上独立运行 30 次的统计结果. 算法运行环境均为 Inter Core i5-5257U 2.70 GHz CPU, 8 GB 内存, Windows10 操作系统, Matlab 2016b. 各对比算法参数设置均与原文献保持一致.

表 1 ~ 6 分别给出了包括本文 spmsAMOPSO 算法在内的多种多目标优化算法在 12 个测试问题上的 IGD、SP 以及 ER 性能指标的平均值和标准差. 表 7 给出了几种算法计算所需的时间, 其中粗体数据为各评价指标下所得的最优结果.

由表 1 ~ 2 可以看出, 本文算法在 12 个测试问



表 1 本文算法与其他多目标粒子群算法的 IGD 评价指标对比  
Table 1 Results of IGD metric of the proposed algorithm and MOPSOs

测试函数	spmsAMOPSO		clusterMOPSO <sup>[23]</sup>		cdMOPSO <sup>[8]</sup>		pccsAMOPSO <sup>[13]</sup>		CMOPSO <sup>[27]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	$2.34 \times 10^{-3}$	$5.46 \times 10^{-6}$	$1.25 \times 10^{-2}$	$1.78 \times 10^{-3}$	$4.24 \times 10^{-3}$	$2.58 \times 10^{-4}$	$3.83 \times 10^{-3}$	$2.67 \times 10^{-4}$	$3.82 \times 10^{-3}$	$2.15 \times 10^{-5}$
ZDT2	$1.98 \times 10^{-3}$	$6.18 \times 10^{-6}$	$1.78 \times 10^{-2}$	$5.09 \times 10^{-3}$	$4.28 \times 10^{-3}$	$1.14 \times 10^{-4}$	$3.81 \times 10^{-3}$	$5.81 \times 10^{-5}$	$3.86 \times 10^{-3}$	$2.83 \times 10^{-5}$
ZDT3	$8.83 \times 10^{-4}$	$1.02 \times 10^{-5}$	$1.05 \times 10^{-1}$	$7.05 \times 10^{-2}$	$3.06 \times 10^{-3}$	$7.13 \times 10^{-5}$	$4.91 \times 10^{-3}$	$6.56 \times 10^{-4}$	$4.50 \times 10^{-3}$	$2.83 \times 10^{-5}$
ZDT4	$2.32 \times 10^{-3}$	$3.96 \times 10^{-6}$	$3.99 \times 10^0$	$2.61 \times 10^0$	$5.91 \times 10^{-1}$	$4.52 \times 10^{-1}$	$5.79 \times 10^{-3}$	$2.98 \times 10^{-4}$	$3.70 \times 10^{-2}$	$4.59 \times 10^{-2}$
ZDT6	$1.60 \times 10^{-3}$	$5.20 \times 10^{-7}$	$4.39 \times 10^{-1}$	$2.37 \times 10^{-2}$	$2.99 \times 10^{-3}$	$1.54 \times 10^{-4}$	$4.68 \times 10^{-3}$	$7.67 \times 10^{-4}$	$3.09 \times 10^{-3}$	$2.61 \times 10^{-5}$
DTLZ1	$1.23 \times 10^{-2}$	$1.23 \times 10^{-3}$	$4.15 \times 10^1$	$2.29 \times 10^1$	$2.75 \times 10^1$	$9.34 \times 10^0$	$1.36 \times 10^{-1}$	$1.12 \times 10^{-1}$	$4.44 \times 10^{-2}$	$7.83 \times 10^{-2}$
DTLZ2	$3.62 \times 10^{-3}$	$8.56 \times 10^{-5}$	$1.26 \times 10^{-1}$	$1.68 \times 10^{-2}$	$1.02 \times 10^{-1}$	$1.34 \times 10^{-2}$	$6.14 \times 10^{-2}$	$1.89 \times 10^{-3}$	$4.40 \times 10^{-3}$	$3.61 \times 10^{-5}$
DTLZ3	$3.28 \times 10^{-3}$	$9.62 \times 10^{-5}$	$4.76 \times 10^1$	$2.87 \times 10^1$	$4.46 \times 10^1$	$1.02 \times 10^1$	$2.19 \times 10^{-1}$	$1.85 \times 10^{-1}$	$4.24 \times 10^{-3}$	$1.71 \times 10^{-4}$
DTLZ4	$7.58 \times 10^{-3}$	$3.45 \times 10^{-3}$	$2.28 \times 10^{-1}$	$8.45 \times 10^{-2}$	$1.02 \times 10^{-1}$	$3.66 \times 10^{-2}$	<b><math>4.21 \times 10^{-3}</math></b>	$3.37 \times 10^{-3}$	$4.41 \times 10^{-3}$	$7.58 \times 10^{-5}$
DTLZ5	$9.76 \times 10^{-3}$	$6.42 \times 10^{-4}$	$1.62 \times 10^{-4}$	$2.48 \times 10^{-3}$	$6.05 \times 10^{-3}$	$7.29 \times 10^{-4}$	$1.18 \times 10^{-2}$	$2.48 \times 10^{-3}$	<b><math>4.40 \times 10^{-3}</math></b>	$5.66 \times 10^{-5}$
DTLZ6	$3.62 \times 10^{-3}$	$3.47 \times 10^{-5}$	$7.38 \times 10^{-2}$	$1.55 \times 10^{-1}$	$5.23 \times 10^{-3}$	$3.90 \times 10^{-4}$	$5.04 \times 10^{-3}$	$2.27 \times 10^{-4}$	$4.09 \times 10^{-3}$	<b><math>1.83 \times 10^{-5}</math></b>
DTLZ7	$4.23 \times 10^{-3}$	$1.37 \times 10^{-4}$	$4.02 \times 10^{-2}$	$2.06 \times 10^{-3}$	$5.78 \times 10^{-2}$	$8.49 \times 10^{-3}$	$4.27 \times 10^{-2}$	$9.51 \times 10^{-4}$	$4.43 \times 10^{-3}$	<b><math>4.01 \times 10^{-5}</math></b>

表 2 本文算法与其他多目标进化算法的 IGD 评价指标对比  
Table 2 Results of IGD metric of the proposed algorithm and multi-objective genetic algorithms

测试函数	spmsAMOPSO		NSGA-II <sup>[24]</sup>		SPEA2 <sup>[25]</sup>		MOEA/D <sup>[26]</sup>		SPEA2 + DAA <sup>[28]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	$2.34 \times 10^{-3}$	$5.46 \times 10^{-6}$	$5.74 \times 10^{-3}$	$3.39 \times 10^{-4}$	$4.15 \times 10^{-3}$	$1.77 \times 10^{-4}$	$4.03 \times 10^{-3}$	$5.59 \times 10^{-5}$	$3.92 \times 10^{-3}$	$5.05 \times 10^{-5}$
ZDT2	$1.98 \times 10^{-3}$	$6.18 \times 10^{-6}$	$5.36 \times 10^{-3}$	$2.02 \times 10^{-4}$	$4.17 \times 10^{-3}$	$2.56 \times 10^{-4}$	$3.85 \times 10^{-3}$	$4.34 \times 10^{-5}$	$4.02 \times 10^{-3}$	$1.07 \times 10^{-4}$
ZDT3	$8.83 \times 10^{-4}$	$1.02 \times 10^{-5}$	$5.83 \times 10^{-3}$	$2.02 \times 10^{-4}$	$3.16 \times 10^{-3}$	$5.96 \times 10^{-3}$	$8.42 \times 10^{-2}$	$7.02 \times 10^{-3}$	$8.46 \times 10^{-3}$	$9.45 \times 10^{-3}$
ZDT4	$2.32 \times 10^{-3}$	$3.96 \times 10^{-6}$	$2.53 \times 10^1$	$7.21 \times 10^0$	$2.49 \times 10^1$	$7.25 \times 10^{-5}$	$4.86 \times 10^{-3}$	$8.41 \times 10^{-4}$	—	—
ZDT6	$1.60 \times 10^{-3}$	$5.20 \times 10^{-7}$	$1.65 \times 10^0$	$9.80 \times 10^{-1}$	$5.32 \times 10^{-3}$	$2.65 \times 10^{-8}$	$3.99 \times 10^{-3}$	$6.02 \times 10^{-5}$	—	—
DTLZ1	$1.23 \times 10^{-2}$	$1.23 \times 10^{-3}$	$1.41 \times 10^1$	$6.65 \times 10^0$	$3.77 \times 10^1$	<b><math>1.45 \times 10^{-4}</math></b>	$6.04 \times 10^{-1}$	$2.89 \times 10^{-1}$	$1.51 \times 10^{-2}$	$1.48 \times 10^{-3}$
DTLZ2	$3.62 \times 10^{-3}$	$8.56 \times 10^{-5}$	$1.06 \times 10^{-1}$	$8.38 \times 10^{-3}$	$8.22 \times 10^{-2}$	$2.83 \times 10^{-7}$	$6.24 \times 10^{-1}$	$6.44 \times 10^{-5}$	$3.81 \times 10^{-2}$	$3.03 \times 10^{-4}$
DTLZ3	$3.28 \times 10^{-3}$	$9.62 \times 10^{-5}$	$1.64 \times 10^1$	$7.56 \times 10^0$	$4.87 \times 10^1$	$0.00 \times 10^0$	$6.52 \times 10^{-1}$	$2.59 \times 10^{-1}$	—	—
DTLZ4	$7.58 \times 10^{-3}$	$3.45 \times 10^{-3}$	$7.30 \times 10^{-2}$	$5.09 \times 10^{-2}$	$7.29 \times 10^{-2}$	$1.42 \times 10^{-7}$	$2.70 \times 10^{-1}$	$6.83 \times 10^{-3}$	—	—
DTLZ5	$9.76 \times 10^{-3}$	$6.42 \times 10^{-4}$	$8.05 \times 10^{-3}$	$1.63 \times 10^{-3}$	$1.41 \times 10^{-2}$	$3.54 \times 10^{-5}$	$5.94 \times 10^{-1}$	<b><math>8.43 \times 10^{-8}</math></b>	—	—
DTLZ6	$3.62 \times 10^{-3}$	$3.47 \times 10^{-5}$	$1.47 \times 10^0$	$6.09 \times 10^{-1}$	$2.49 \times 10^{-1}$	$5.67 \times 10^{-5}$	$6.17 \times 10^{-1}$	$1.01 \times 10^{-4}$	—	—
DTLZ7	$4.23 \times 10^{-3}$	$1.37 \times 10^{-4}$	$6.14 \times 10^{-1}$	$1.29 \times 10^{-3}$	$6.24 \times 10^{-2}$	$9.50 \times 10^{-3}$	$6.57 \times 10^{-1}$	$9.87 \times 10^{-4}$	$3.69 \times 10^{-2}$	$5.02 \times 10^{-4}$

表 3 本文算法与其他多目标粒子群算法的 SP 评价指标对比  
Table 3 Results of SP metric of the proposed algorithm and MOPSOs

测试函数	spmsAMOPSO		clusterMOPSO <sup>[23]</sup>		cdMOPSO <sup>[8]</sup>		pccsAMOPSO <sup>[13]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	$4.09 \times 10^{-3}$	$5.57 \times 10^{-5}$	$7.25 \times 10^{-2}$	$6.17 \times 10^{-4}$	$8.56 \times 10^{-2}$	$1.43 \times 10^{-2}$	$1.33 \times 10^{-2}$	$5.42 \times 10^{-3}$
ZDT2	$3.19 \times 10^{-3}$	$5.58 \times 10^{-5}$	$1.77 \times 10^{-2}$	$1.95 \times 10^{-3}$	$1.91 \times 10^{-2}$	$6.42 \times 10^{-4}$	$1.06 \times 10^{-2}$	$4.63 \times 10^{-3}$
ZDT3	$4.35 \times 10^{-3}$	$4.54 \times 10^{-5}$	$7.78 \times 10^{-2}$	$4.36 \times 10^{-2}$	$5.97 \times 10^{-1}$	$2.25 \times 10^{-1}$	$1.25 \times 10^{-1}$	$6.48 \times 10^{-2}$
ZDT4	$3.84 \times 10^{-3}$	$9.84 \times 10^{-5}$	$9.19 \times 10^{-3}$	$1.32 \times 10^{-3}$	$8.49 \times 10^{-3}$	$5.85 \times 10^{-4}$	$1.09 \times 10^{-2}$	$2.78 \times 10^{-3}$
ZDT6	$3.11 \times 10^{-3}$	$3.91 \times 10^{-5}$	$2.69 \times 10^{-2}$	$1.67 \times 10^{-2}$	$9.83 \times 10^{-3}$	$4.65 \times 10^{-4}$	$1.10 \times 10^{-2}$	$5.89 \times 10^{-3}$
DTLZ1	$2.24 \times 10^{-2}$	$3.36 \times 10^{-4}$	$9.32 \times 10^{-2}$	$1.74 \times 10^{-2}$	$4.12 \times 10^{-2}$	$5.64 \times 10^{-3}$	$5.79 \times 10^0$	$4.83 \times 10^{-1}$
DTLZ2	$3.60 \times 10^{-2}$	$1.74 \times 10^{-4}$	$7.06 \times 10^{-2}$	$3.42 \times 10^{-3}$	$6.91 \times 10^{-2}$	$4.03 \times 10^{-3}$	$6.08 \times 10^{-2}$	$2.54 \times 10^{-3}$
DTLZ3	$4.12 \times 10^{-2}$	$2.33 \times 10^{-3}$	$7.93 \times 10^{-1}$	$2.06 \times 10^{-2}$	$5.88 \times 10^{-2}$	$3.96 \times 10^{-3}$	$6.09 \times 10^{-1}$	<b><math>7.45 \times 10^{-4}</math></b>
DTLZ4	$3.80 \times 10^{-2}$	$8.63 \times 10^{-4}$	$6.81 \times 10^{-1}$	$9.94 \times 10^{-3}$	$4.76 \times 10^{-2}$	$2.84 \times 10^{-3}$	$5.97 \times 10^{-1}$	$6.33 \times 10^{-2}$
DTLZ5	$7.52 \times 10^{-2}$	$5.83 \times 10^{-3}$	$8.93 \times 10^{-2}$	$7.52 \times 10^{-3}$	<b><math>2.42 \times 10^{-2}</math></b>	$6.22 \times 10^{-3}$	$1.02 \times 10^{-1}$	$3.74 \times 10^{-2}$
DTLZ6	$3.47 \times 10^{-2}$	$2.14 \times 10^{-3}$	$6.80 \times 10^{-2}$	$1.83 \times 10^{-2}$	$4.65 \times 10^{-2}$	$2.73 \times 10^{-3}$	$5.86 \times 10^{-2}$	<b><math>1.26 \times 10^{-4}</math></b>
DTLZ7	$9.47 \times 10^{-2}$	$4.81 \times 10^{-2}$	$3.90 \times 10^{-1}$	$1.29 \times 10^{-2}$	$5.97 \times 10^{-1}$	$2.13 \times 10^{-1}$	$1.08 \times 10^{-1}$	$3.14 \times 10^{-2}$

表 4 本文算法与其他多目标进化算法的 SP 评价指标对比  
Table 4 Results of SP metric of the proposed algorithm and multi-objective genetic algorithms

测试函数	spmsAMOPSO		NSGA-II <sup>[24]</sup>		SPEA2 <sup>[25]</sup>		MOEA/D <sup>[26]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	$4.09 \times 10^{-3}$	$5.57 \times 10^{-5}$	$5.83 \times 10^{-2}$	$9.39 \times 10^{-3}$	$3.73 \times 10^{-2}$	$2.67 \times 10^{-3}$	$4.85 \times 10^{-3}$	$7.19 \times 10^{-4}$
ZDT2	$3.19 \times 10^{-3}$	$5.58 \times 10^{-5}$	$7.24 \times 10^{-3}$	$7.41 \times 10^{-3}$	$1.09 \times 10^{-2}$	$1.04 \times 10^{-3}$	$4.36 \times 10^{-3}$	$7.41 \times 10^{-4}$
ZDT3	$4.35 \times 10^{-3}$	$4.54 \times 10^{-5}$	$9.22 \times 10^{-2}$	$8.42 \times 10^{-3}$	$6.07 \times 10^{-1}$	$1.03 \times 10^0$	$1.02 \times 10^{-1}$	$9.33 \times 10^{-3}$
ZDT4	$3.84 \times 10^{-3}$	$9.84 \times 10^{-5}$	$2.83 \times 10^{-2}$	$6.13 \times 10^{-3}$	$4.06 \times 10^{-2}$	$1.59 \times 10^{-2}$	$7.52 \times 10^{-3}$	$6.93 \times 10^{-4}$
ZDT6	$3.11 \times 10^{-3}$	$3.91 \times 10^{-5}$	$3.43 \times 10^{-2}$	$4.33 \times 10^{-3}$	$4.72 \times 10^{-2}$	$7.74 \times 10^{-3}$	$1.88 \times 10^{-2}$	$5.52 \times 10^{-3}$
DTLZ1	$2.24 \times 10^{-2}$	$3.36 \times 10^{-4}$	$7.21 \times 10^{-1}$	$6.34 \times 10^{-2}$	$2.92 \times 10^{-1}$	$3.62 \times 10^{-2}$	$9.83 \times 10^{-1}$	$2.33 \times 10^{-1}$
DTLZ2	$3.60 \times 10^{-2}$	$1.74 \times 10^{-4}$	$4.67 \times 10^{-2}$	$7.71 \times 10^{-3}$	$5.38 \times 10^{-2}$	$3.42 \times 10^{-3}$	$8.14 \times 10^{-2}$	$8.49 \times 10^{-3}$
DTLZ3	$4.12 \times 10^{-2}$	$2.33 \times 10^{-3}$	$8.26 \times 10^{-2}$	$2.09 \times 10^{-3}$	$6.34 \times 10^{-2}$	$5.12 \times 10^{-3}$	$1.79 \times 10^{-1}$	$3.36 \times 10^{-2}$
DTLZ4	$3.80 \times 10^{-2}$	$8.63 \times 10^{-4}$	$7.14 \times 10^{-2}$	$1.97 \times 10^{-3}$	$4.38 \times 10^{-2}$	$6.54 \times 10^{-3}$	$9.25 \times 10^{-2}$	$5.61 \times 10^{-3}$
DTLZ5	$7.52 \times 10^{-2}$	$5.83 \times 10^{-3}$	$6.37 \times 10^{-2}$	$1.74 \times 10^{-3}$	$3.15 \times 10^{-2}$	$5.31 \times 10^{-3}$	$8.02 \times 10^{-2}$	$4.39 \times 10^{-3}$
DTLZ6	$3.47 \times 10^{-2}$	$2.14 \times 10^{-3}$	$7.03 \times 10^{-2}$	$1.56 \times 10^{-3}$	$5.11 \times 10^{-2}$	$4.89 \times 10^{-3}$	$3.93 \times 10^{-2}$	$3.31 \times 10^{-3}$
DTLZ7	$9.47 \times 10^{-2}$	$4.81 \times 10^{-2}$	$4.19 \times 10^{-1}$	$7.96 \times 10^{-3}$	$2.96 \times 10^{-1}$	$5.29 \times 10^{-3}$	$1.85 \times 10^{-1}$	$7.93 \times 10^{-2}$

表 5 本文算法与其他多目标粒子群算法的 ER 评价指标对比  
Table 5 Results of ER metric of the proposed algorithm and MOPSOs

测试函数	spmsAMOPSO		clusterMOPSO <sup>[29]</sup>		cdMOPSO <sup>[8]</sup>		pccsAMOPSO <sup>[19]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	$6.73 \times 10^{-4}$	$1.34 \times 10^{-4}$	$1.06 \times 10^{-2}$	$1.00 \times 10^{-2}$	$9.90 \times 10^{-2}$	$1.25 \times 10^{-3}$	$8.12 \times 10^{-3}$	$1.43 \times 10^{-2}$
ZDT2	$3.75 \times 10^{-3}$	$1.17 \times 10^{-5}$	$4.92 \times 10^{-1}$	$8.72 \times 10^{-2}$	$1.30 \times 10^0$	$1.21 \times 10^{-1}$	$9.00 \times 10^{-3}$	$2.36 \times 10^{-3}$
ZDT3	$8.95 \times 10^{-3}$	$1.37 \times 10^{-3}$	$2.66 \times 10^{-1}$	$9.30 \times 10^{-2}$	$2.89 \times 10^{-1}$	$7.24 \times 10^{-2}$	$2.60 \times 10^{-2}$	$8.32 \times 10^{-3}$
ZDT4	$2.70 \times 10^{-2}$	$8.35 \times 10^{-3}$	$4.42 \times 10^{-1}$	$1.02 \times 10^{-1}$	$9.34 \times 10^{-2}$	$6.83 \times 10^{-2}$	<b><math>2.37 \times 10^{-2}</math></b>	<b><math>4.74 \times 10^{-3}</math></b>
ZDT6	$1.60 \times 10^{-3}$	$1.24 \times 10^{-2}$	$2.98 \times 10^{-2}$	$7.43 \times 10^{-3}$	$6.77 \times 10^{-3}$	$3.43 \times 10^{-3}$	$1.10 \times 10^{-2}$	<b><math>4.64 \times 10^{-4}</math></b>
DTLZ1	$6.81 \times 10^{-2}$	<b><math>4.24 \times 10^{-3}</math></b>	<b><math>4.83 \times 10^{-2}</math></b>	$6.52 \times 10^{-3}$	$6.22 \times 10^{-1}$	$3.81 \times 10^{-2}$	$9.90 \times 10^0$	$7.13 \times 10^{-1}$
DTLZ2	<b><math>4.74 \times 10^{-1}</math></b>	<b><math>3.52 \times 10^{-3}</math></b>	$1.19 \times 10^0$	$7.62 \times 10^{-1}$	$8.32 \times 10^{-1}$	$3.53 \times 10^{-2}$	$9.10 \times 10^{-1}$	$5.31 \times 10^{-2}$
DTLZ3	<b><math>8.15 \times 10^{-2}</math></b>	$2.47 \times 10^{-2}$	$4.07 \times 10^{-1}$	$5.26 \times 10^{-2}$	$7.21 \times 10^{-1}$	$2.12 \times 10^{-2}$	$9.53 \times 10^{-2}$	$3.64 \times 10^{-2}$
DTLZ4	$8.02 \times 10^{-2}$	<b><math>1.14 \times 10^{-3}</math></b>	$8.24 \times 10^{-2}$	$3.01 \times 10^{-3}$	$2.42 \times 10^{-1}$	$9.71 \times 10^{-2}$	$6.33 \times 10^{-1}$	$3.60 \times 10^{-2}$
DTLZ5	<b><math>1.43 \times 10^{-2}</math></b>	$3.90 \times 10^{-3}$	$4.81 \times 10^{-1}$	$5.32 \times 10^{-2}$	$1.99 \times 10^{-1}$	$9.73 \times 10^{-2}$	$2.72 \times 10^{-2}$	$4.74 \times 10^{-3}$
DTLZ6	$1.73 \times 10^{-1}$	$5.69 \times 10^{-2}$	<b><math>4.62 \times 10^{-2}</math></b>	<b><math>6.39 \times 10^{-3}</math></b>	$3.48 \times 10^{-1}$	$2.15 \times 10^{-3}$	$1.92 \times 10^{-1}$	$8.13 \times 10^{-2}$
DTLZ7	<b><math>2.18 \times 10^{-1}</math></b>	<b><math>7.64 \times 10^{-2}</math></b>	$4.77 \times 10^{-1}$	$5.07 \times 10^{-1}$	$5.89 \times 10^{-1}$	$4.33 \times 10^{-1}$	$3.16 \times 10^{-1}$	$9.57 \times 10^{-2}$

表 6 本文算法与其他多目标进化算法的 ER 评价指标对比  
Table 6 Results of ER metric of the proposed algorithm and multi-objective genetic algorithms

测试函数	spmsAMOPSO		NSGA-II <sup>[24]</sup>		SPEA2 <sup>[25]</sup>		MOEA/D <sup>[26]</sup>	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
ZDT1	<b><math>6.73 \times 10^{-4}</math></b>	<b><math>1.34 \times 10^{-4}</math></b>	$8.06 \times 10^{-3}$	$5.82 \times 10^{-3}$	$3.00 \times 10^{-3}$	$6.75 \times 10^{-3}$	$7.65 \times 10^{-2}$	$2.44 \times 10^{-3}$
ZDT2	<b><math>3.75 \times 10^{-3}</math></b>	<b><math>1.17 \times 10^{-5}</math></b>	$5.74 \times 10^{-1}$	$3.41 \times 10^{-2}$	$9.11 \times 10^{-1}$	$4.68 \times 10^{-2}$	$6.53 \times 10^{-1}$	$2.78 \times 10^{-2}$
ZDT3	<b><math>8.95 \times 10^{-3}</math></b>	<b><math>1.37 \times 10^{-3}</math></b>	$2.09 \times 10^{-2}$	$4.12 \times 10^{-2}$	$1.49 \times 10^{-1}$	$6.92 \times 10^{-2}$	$4.87 \times 10^{-2}$	$6.47 \times 10^{-3}$
ZDT4	$2.70 \times 10^{-2}$	$8.35 \times 10^{-3}$	$3.49 \times 10^{-2}$	$7.66 \times 10^{-3}$	$8.45 \times 10^{-2}$	$2.93 \times 10^{-2}$	$7.60 \times 10^{-1}$	$5.36 \times 10^{-2}$
ZDT6	<b><math>1.60 \times 10^{-3}</math></b>	$1.24 \times 10^{-2}$	$9.93 \times 10^{-3}$	$7.64 \times 10^{-4}$	$8.02 \times 10^{-3}$	$3.24 \times 10^{-3}$	$1.73 \times 10^{-2}$	$2.12 \times 10^{-3}$
DTLZ1	$6.81 \times 10^{-2}$	<b><math>4.24 \times 10^{-3}</math></b>	$2.80 \times 10^{-1}$	$1.55 \times 10^{-2}$	$8.31 \times 10^{-2}$	$4.85 \times 10^{-3}$	$4.64 \times 10^{-1}$	$3.06 \times 10^{-2}$
DTLZ2	<b><math>4.74 \times 10^{-1}</math></b>	<b><math>3.52 \times 10^{-3}</math></b>	$7.06 \times 10^{-1}$	$5.44 \times 10^{-2}$	$9.47 \times 10^{-1}$	$2.13 \times 10^{-1}$	$2.59 \times 10^0$	$7.42 \times 10^{-1}$
DTLZ3	<b><math>8.15 \times 10^{-2}</math></b>	$2.47 \times 10^{-2}$	$8.60 \times 10^{-1}$	$3.75 \times 10^{-2}$	$9.83 \times 10^{-1}$	$3.41 \times 10^{-2}$	$2.17 \times 10^{-1}$	<b><math>4.54 \times 10^{-3}</math></b>
DTLZ4	$8.02 \times 10^{-2}$	<b><math>1.14 \times 10^{-3}</math></b>	$4.85 \times 10^{-1}$	$4.21 \times 10^{-2}$	<b><math>6.13 \times 10^{-2}</math></b>	$1.90 \times 10^{-3}$	$1.36 \times 10^{-1}$	$6.29 \times 10^{-2}$
DTLZ5	<b><math>1.43 \times 10^{-2}</math></b>	$3.90 \times 10^{-3}$	$9.43 \times 10^{-2}$	$4.16 \times 10^{-3}$	$8.32 \times 10^{-2}$	<b><math>2.73 \times 10^{-3}</math></b>	$v1.03 \times 10^{-1}$	$6.51 \times 10^{-2}$
DTLZ6	$1.73 \times 10^{-1}$	$5.69 \times 10^{-2}$	$5.84 \times 10^{-2}$	$7.91 \times 10^{-3}$	$4.36 \times 10^{-1}$	$2.26 \times 10^{-2}$	$2.53 \times 10^{-1}$	$9.06 \times 10^{-2}$
DTLZ7	<b><math>2.18 \times 10^{-1}</math></b>	<b><math>7.64 \times 10^{-2}</math></b>	$4.34 \times 10^{-1}$	$2.64 \times 10^{-1}$	$3.92 \times 10^{-1}$	$8.70 \times 10^{-2}$	$3.76 \times 10^{-1}$	$2.92 \times 10^{-1}$

表 7 不同算法对多目标测试问题的运行时间 (s)  
Table 7 Computational time of different algorithms for multi-objective test problems (s)

函数	spmsAMOPSO	clusterMOPSO <sup>[23]</sup>	cdMOPSO <sup>[8]</sup>	pccsAMOPSO <sup>[13]</sup>	NSGA-II <sup>[24]</sup>	SPEA2 <sup>[25]</sup>	MOEA/D <sup>[26]</sup>
ZDT1	105.26	108.87	<b>101.72</b>	121.95	129.17	135.23	152.45
ZDT2	<b>104.72</b>	114.95	112.74	108.34	133.15	126.65	139.13
ZDT3	<b>111.23</b>	136.38	135.81	124.57	132.40	137.98	135.02
ZDT4	<b>115.17</b>	132.09	129.69	122.19	125.67	130.61	147.61
ZDT6	<b>122.48</b>	131.62	124.23	126.62	133.52	128.74	149.55
DTLZ1	<b>210.77</b>	230.82	226.47	219.83	248.13	232.47	280.73
DTLZ2	218.93	233.49	<b>212.73</b>	215.63	250.26	238.69	289.36
DTLZ3	<b>212.61</b>	228.34	217.51	218.26	250.98	242.81	281.75
DTLZ4	218.34	230.25	220.98	<b>216.44</b>	247.37	241.33	276.84
DTLZ5	<b>219.15</b>	234.16	228.21	221.92	255.91	250.62	286.17
DTLZ6	<b>216.37</b>	236.59	225.40	225.73	245.69	247.97	288.33
DTLZ7	<b>215.42</b>	243.52	224.64	232.41	246.38	238.11	295.42

题的结果中, 获得 10 次 IGD 指标的最优平均值和 6 次标准差最优值. 虽然在 DTLZ4 测试问题中本文算法的 IGD 指标略逊于 pccsAMOPSO 算法 (其结果为最优值, 本文算法结果为次优值), 但优于其中 5 种多目标算法. 这是由于本文算法在进行最优粒子选取时采用分区策略, 将部分性能相似的粒子划分为一个区域, 对一个区域的粒子实现一种搜索引导策略, 而 pccsAMOPSO 算法在进行全局最优粒子选取时, 对种群中粒子逐个进行选取, 而本文算法根据粒子当前性能进行分区所获得的分区较为模糊, 在对复杂多目标问题进行优化时, 综合性能表现稍显不足. 但本文算法通过种群分区策略, 提出一种新的变异方法, 并结合带有记忆区间的个体最优粒子选取和融合指标的外部存档维护等方法, 使本文算法最终仍具有良好的性能. 在 DTLZ5 测试问题中本文算法性能相较于 CMOPSO 算法和 cdMOPSO 算法具有一些劣势 (综合性能上 CMOPSO 算法最优, cdMOPSO 算法次优), 但优于其余 6 种多目标优化算法. 对其余测试问题中, 本文算法均取得 IGD 指标的最优值, 各算法在 12 个测试问题中的 IGD 最优值个数表明, 本文算法在综合性能上优于其他 8 种多目标优化算法.

由表 3 和表 4 可以得到, 本文算法在 12 个测试问题的结果中, 获得 11 次 SP 指标的最优平均值和 7 次标准差最优值. 在 DTLZ5 测试问题中, 本文算法的 SP 指标略逊于 cdMOPSO 和 NSGA-II 算法 (cdMOPSO 和 NSGA-II 算法分别获得最优值和次优值), 但优于 pccsAMOPSO、clusterMOPSO、

MOEA/D 和 SPEA2 等 4 种多目标优化算法, 在其余测试问题中, 本文算法的 SP 指标均取得最优值. 实验结果表明, 相比于其余多目标优化算法, 本文算法获得的 Pareto 前沿中非支配解分布均匀, 算法具有良好的多样性.

SP 指标的实验结果表明, 本文算法获得的非支配解集具有良好的多样性, 这是由于本文算法通过对算法环境的检查自适应地调整了算法惯性权重和学习因子, 并采用种群分区策略, 对种群中收敛性较好的粒子制定特殊的搜索引导策略和变异方法, 提升粒子的探索效率, 有效增强算法多样性.

由表 5 ~ 6 可以看出, 在 12 个测试问题的结果中, 本文算法共获得 8 次 ER 评价指标的最优值, 在求解 ZDT4、DTLZ1 和 DTLZ4 测试问题时取得次优值 (MOEA/D、clusterMOPSO 和 SPEA2 算法分别取得最优值). 在 DTLZ6 测试问题中, 本文算法的 ER 指标略逊于 clusterMOPSO 和 NSGA-II 算法, 这是由于本文算法在最优粒子选取, 变异以及外部存档维护等方面均采用不同的策略对算法的收敛性和多样性进行平衡, 即牺牲了粒子的部分局部开发能力换取粒子的全局开发能力, 使算法取得相较于其他算法更好的多样性, 因此在部分复杂多目标优化问题中收敛性略显不足. 但相比于 cdMOPSO 等其余 4 种多目标优化算法, 本文算法在收敛性上仍然具有一定优势. 综合各算法的 ER 评价指标实验结果, 本文 spmsAMOPSO 算法获得的非支配解集能够较好地逼近真实 Pareto 前沿, 相比于其他几种多目标优化算法, 具有良好的

收敛性。

ER 指标的实验结果表明, 本文算法相较于其他几种多目标优化算法具有更好的收敛性, 这是由于本文算法通过在全局最优粒子选取和变异中加入种群分区, 对种群中性能较差的粒子实行特殊的寻优策略, 增强劣势粒子的利用率, 有效提升算法收敛性; 在个体最优粒子选取中, 加入粒子的记忆机制, 提升个体最优粒子选取的可靠性, 增强个体最优粒子对粒子收敛过程的指导作用。

由表 7 可以看出, 本文算法在求解多数测试问题时均具有较好的实时性, 仅在 ZDT1 和 DTLZ2 问题时劣于 cdMOPSO 算法, 在 DTLZ4 问题时略逊于 pccsAMOPSO。虽然本文算法采用了分区机制和粒子记忆区间, 增加了算法的计算复杂度, 但有效的种群划分使粒子的全局最优粒子选取以及变异更有效率, 并且个体最优粒子的可靠选取使粒子能够更快收敛至 Pareto 前沿附近, 使算法运行时间相较其他同类型优化算法更短。

为详细说明各算法所得非支配解集的收敛性和多样性, 图 6~8 分别给出了 6 种算法在求解 ZDT3、DTLZ2 和 DTLZ7 测试问题时的非支配解集和真实 Pareto 前沿。由实验结果可以看出, 在三种测试问题中, cdMOPSO、pccsAMOPSO、NSGA-II、SPEA2 和 MOEA/D 等多目标优化算法均不同程度的出现了收敛性和多样性不足的问题。同样是在

DTLZ2 测试问题中, cdMOPSO、NSGA-II 和 SPEA2 算法均表现出收敛性不足的缺点, 虽然 pccsAMOPSO 和 MOEA/D 算法具有较好的收敛性, 但算法多样性较差, 而本文 spmsAMOPSO 算法得到的 Pareto 前沿中, 非支配解不仅较好地收敛在真实 Pareto 前沿附近, 而且具有良好的分布性, 算法在收敛性和多样性上均有较好表现。

为直观展示各算法多次实验数据的分布情况, 图 9 分别给出了 7 种多目标优化算法在测试 ZDT3、DTLZ2 和 DTLZ7 问题时 IGD、SP 以及 ER 指标的箱形图, 其中横轴坐标 1~7 分别代表 spmsAMOPSO、clusterMOPSO、cdMOPSO、pccsAMOPSO、NSGA-II、SPEA2 和 MOEA/D 算法。由图 9 可以看出, 本文算法在求解 ZDT3、DTLZ2 和 DTLZ7 测试问题时, IGD、SP 以及 ER 指标相较于其他 6 种多目标优化算法均具有较大优势, 并且本文算法的多次测试结果数据无明显偏态, 出现的异常数据较少, 各项性能指标的中值和最大偏差均优于其他多目标优化算法。实验结果表明, 本文算法不仅具有良好的综合性能, 并且测试结果稳定。

综合实验结果表明, 本文算法得到的非支配解集能够有效地收敛在真实 Pareto 前沿, 并具有良好的分布性。这是由于本文算法通过对种群进行分区, 制定多策略的搜索引导方案, 有效平衡了算法的收敛性和多样性, 提升了粒子的探索和开发能力及寻

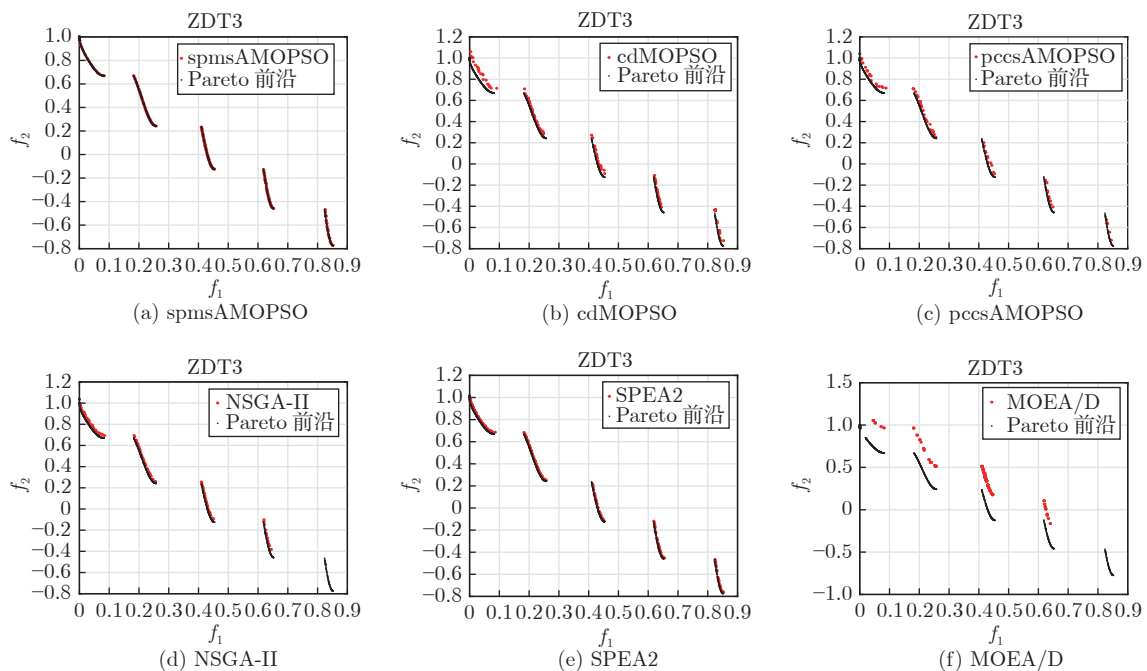


图 6 不同多目标优化算法对 ZDT3 函数的 Pareto 前沿

Fig. 6 Pareto front of ZDT3 function of different multi-objective optimization algorithms

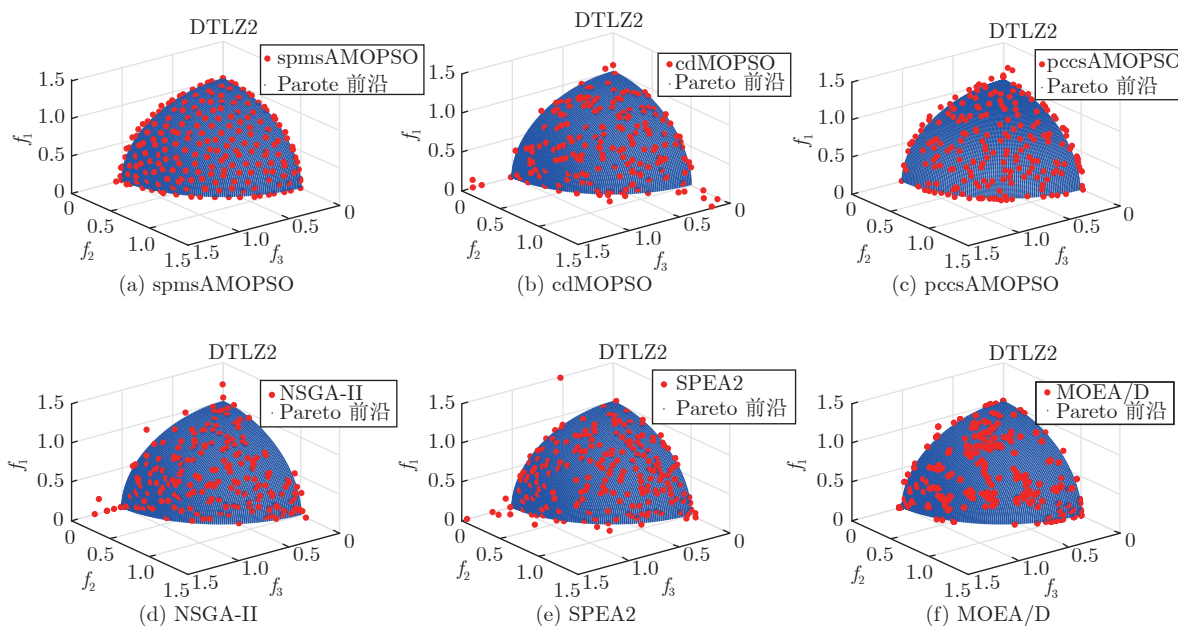


图 7 不同多目标优化算法对 DTLZ2 函数的 Pareto 前沿

Fig.7 Pareto front of DTLZ2 function of different multi-objective optimization algorithms

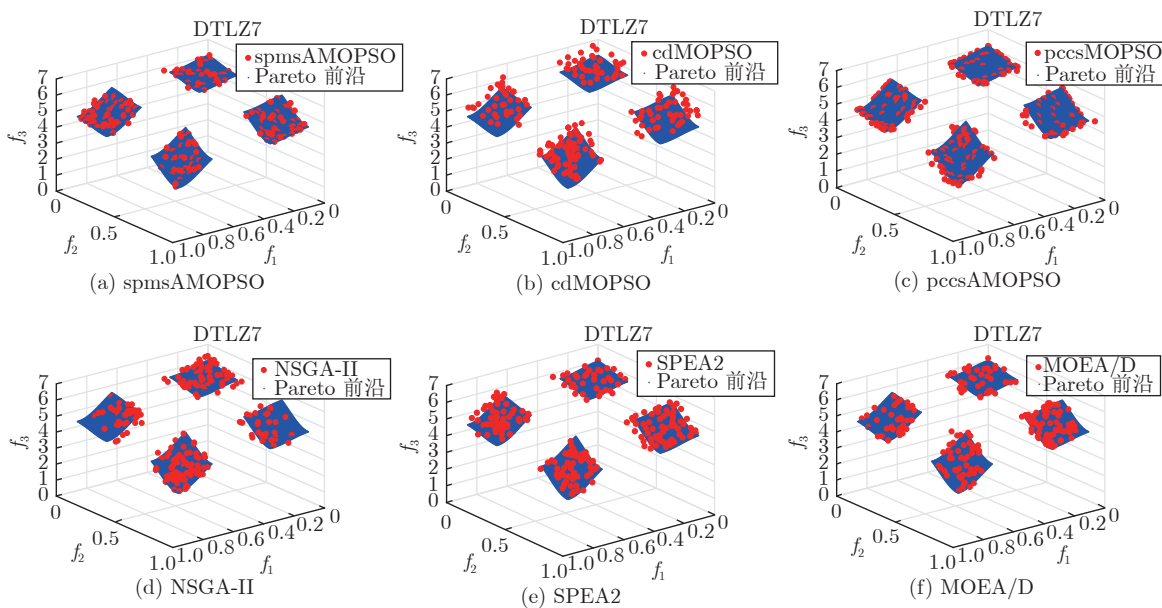


图 8 不同多目标优化算法对 DTLZ7 函数的 Pareto 前沿

Fig.8 Pareto front of DTLZ7 function of different multi-objective optimization algorithms

优效率, 并且使算法具有较好综合性能的同时降低了算法的计算复杂度.

### 4 结束语

本文提出了一种基于种群分区的多策略自适应多目标粒子群优化算法. 算法通过基于种群分区的多策略改进, 确定算法的全局最优粒子选取和变异

方法, 将粒子性能与算法寻优过程结合, 提升种群中各个粒子的搜索效率; 提出带有记忆区间的个体最优粒子选取方法, 提升个体最优粒子选取的可靠性, 避免因个体最优粒子不能有效指导粒子的飞行, 使算法停滞, 陷入局部最优; 采用双性能测度指标的外部存档维护策略, 能够有效保持外部存档中非支配解良好的分布性, 避免进行外部存档维护时,

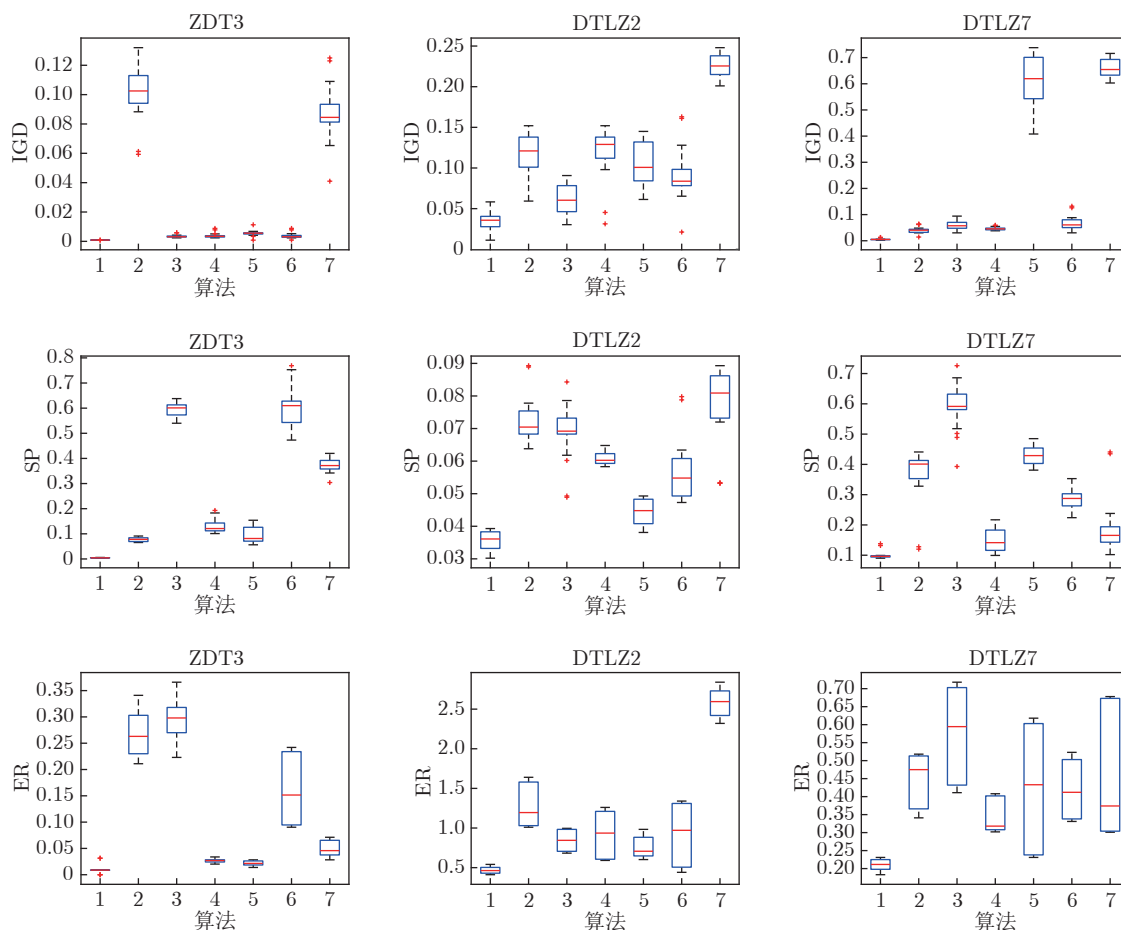


图9 7种多目标优化算法在测试ZDT3、DTLZ2和DTLZ7问题时IGD、SP以及ER指标的箱形图

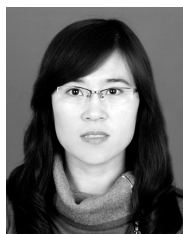
Fig.9 Box plots of IGD, SP and ER metric on ZDT3, DTLZ2 and DTLZ7 problems of multi-objective optimization algorithms

删除收敛性较好的粒子, 导致种群产生退化, 影响粒子开发能力. 为验证算法有效性, 采用ZDT和DTLZ系列测试函数进行仿真实验, 并与其他几种多目标优化算法进行对比, 通过各算法对同一测试问题的IGD、SP、ER以及计算时间等评价指标的实验结果, 分别说明本文算法的收敛性, 多样性和实时性. 综合实验结果表明, 本文算法具有较显著的收敛性和多样性优势, 且具有良好的实时性.

### References

- Qiao Jun-Fei, Li-Fei, Yang Cui-Li. An NSGA II algorithm based on uniform distribution strategy. *Acta Automatica Sinica*, 2019, **45**(7): 1325-1334  
(乔俊飞, 李霏, 杨翠丽. 一种基于均匀分布策略的NSGA II算法. *自动化学报*, 2019, **45**(7): 1325-1334)
- Cheng S, Zhao L L, Jiang X Y. Many-objective particle swarm optimization based on adaptive fuzzy dominance. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2017, **14**(1): 56-63
- Feng Q, Li Q, Chen P, Hen W, Xue Z E, Yin L, Ge C. Multi-objective particle swarm optimization algorithm based on adaptive angle division. *IEEE Access*, 2019, **7**: 87916-87930
- Liu X F, Zhan Z H, Gao Y, Zhang J, Kwong S, Zhang J. Co-evolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2019, **23**(4): 587-602
- Yu Wei-Wei, Xie Cheng-Wang, Bi Ying-Zhou, Xia Xue-Wen, Li Xiong, Ren Ke-Yan, et al. Many-objective particle swarm optimization based on adaptive fuzzy dominance. *Acta Automatica Sinica*, 2018, **44**(12): 2278-2289  
(余伟伟, 谢承旺, 闭应洲, 夏雪文, 李雄, 任柯燕, 等. 一种基于自适应模糊支配的高维多目标粒子群算法. *自动化学报*, 2018, **44**(12): 2278-2289)
- Li L, Wang W L, Li W K, Xu X L, Zhao Y W. A novel ranking-based optimal guides selection strategy in MOPSO. *Procedia Computer Science*, 2016, **91**: 1001-1010
- Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization. *IEEE Trans on Evolutionary Computation*, 2004, **8**(3): 256-279
- Raquel C R, Naval P C. An effective use of crowding distance in multiobjective particle swarm optimization. In: *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. New York, USA: 2005. 257-264
- Yen G G, Wen F, Leong. Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2009, **39**(4): 890-911
- Peng W, Zhang Q F. A decomposition-based multi-objective

- particle swarm optimization algorithm for continuous optimization problems. In: Proceedings of the 2008 IEEE International Conference on Granular Computing. Hangzhou, China: 2008. 534–537
- 11 Martínez S Z, Coello C A C. A multi-objective particle swarm optimizer based on decomposition. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. Dublin, Ireland: 2011. 69–76
  - 12 Dai C, Wang Y, Ye M. A new multi-objective particle swarm optimization algorithm based on decomposition. *Information Sciences*, 2015, **325**: 541–557
  - 13 Hu W, Yen G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Transactions on Evolutionary Computation*, 2015, **19**(1): 1–18
  - 14 Kumar R S, Kondapaneni K, Dixit V, Goswami A, Thakur L S, Tiwari M K. Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach. *Computers and Industrial Engineering*, 2016, **99**: 29–40
  - 15 Kennedy J, Eberhart R C. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway, USA: 1995. 1942–1948
  - 16 Han Min, He Yong. Adaptive multi-objective particle swarm optimization with Gaussian chaotic mutation and elite learning. *Control and Decision*, 2016, **31**(8): 1372–1378  
(韩敏, 何泳. 基于高斯混沌变异和精英学习的自适应多目标粒子群算法. 控制与决策, 2016, **31**(8): 1372–1378)
  - 17 Rajani, Kumar D, Kumar V. Impact of controlling parameters on the performance of MOPSO algorithm. *Procedia Computer Science*, 2020, **167**: 2132
  - 18 Luo J P, Huang X W, Yang Y, Li X, Wang Z K, Feng J Q. A many-objective particle swarm optimizer based on indicator and direction vectors for many-objective optimization. *Information Sciences*, 2020, **514**: 166
  - 19 Lin Q Z, Liu S B, Zhu Q L, Tang C Y, Song R Z, Chen J Y, et al. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Trans on Evolutionary Computation*, 2018, **22**(1): 32–46
  - 20 El-Shorbagy M A, Elhoseny M, Hassanien A E, Ahmed S H. A novel PSO algorithm for dynamic wireless sensor network multiobjective optimization problem. *Transactions on Emerging Telecommunications Technologies*, 2018, **30**(11): 1–14
  - 21 Mohamed A M, Enrico Z. An adaptive particle swarm optimization method for multi-objective system reliability optimization. *Journal of Risk and Reliability*, 2019, **233**: 990–1001
  - 22 Helwig S, Branke J, Mostaghim S. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2013, **17**(2): 259
  - 23 Chakraborty P, Das S, Roy G G, Abraham A. On convergence of the multi-objective particle swarm optimizers. *Information Sciences*, 2011, **181**: 1411–1425
  - 24 Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(2): 182–197
  - 25 Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103, Swiss Federal Institute of Technology, Switzerland, 2001
  - 26 Zhang Q F, Li H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 2007, **11**(6): 712–731
  - 27 Zhang X Y, Zheng X T, Cheng R, Qiu J F, Jin Y C. A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. *Information Sciences*, 2018, **427**: 63–76
  - 28 Zhang Y, Gong D W, Sun J Y, Qu B Y. A decomposition-based archiving approach for multi-objective evolutionary optimization. *Information Sciences*, 2018, **430**: 397–413



**张伟** 河南理工大学电气工程与自动化学院教授。2016年获北京工业大学博士学位。主要研究方向为智能特征建模, 神经网络设计和污水处理过程的优化控制。本文通信作者。

E-mail: zwei1563@126.com

**(ZHANG Wei** Professor at the School of Electrical Engineering and Automation, Henan Polytechnic University. She received her Ph.D. degree from Beijing University of Technology in 2016. Her research interest covers intelligent feature modeling, design of neural networks, and optimization control for wastewater treatment process. Corresponding author of this paper.)



**黄卫民** 河南理工大学电气工程与自动化学院硕士研究生。2019年获中原工学院学士学位。主要研究方向为多目标优化问题及神经网络结构优化和算法设计。

E-mail: hweimin97520@163.com

**(HUANG Wei-Min** Master student at the School of Electrical Engineering and Automation, Henan Polytechnic University. He received his bachelor degree from Zhongyuan University of Technology in 2019. His research interest covers multi-objective optimization problem and neural network structure optimization and algorithm design.)