

# 控制与决策

Control and Decision

## 基于自适应双阶段分级均衡的约束高维多目标进化算法

王林锋, 揭丽琳, 黎明, 何超, 李军华

引用本文:

王林锋, 揭丽琳, 黎明, 等. 基于自适应双阶段分级均衡的约束高维多目标进化算法[J]. *控制与决策*, 2025, 40(5): 1512-1522.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2024.0288>

---

## 您可能感兴趣的其他文章

### Articles you may be interested in

#### 基于弱关联的自适应高维多目标进化算法

A weak association-based adaptive evolutionary algorithm for manyobjective optimization

*控制与决策*. 2021, 36(8): 1804-1814 <https://doi.org/10.13195/j.kzyjc.2019.1723>

#### 基于分类的多策略预测方法求解动态多目标优化问题

[Classification-based multi-strategy prediction method for dynamic multiobjective optimization problems](#)

*控制与决策*. 2021, 36(7): 1569-1580 <https://doi.org/10.13195/j.kzyjc.2019.1320>

#### 基于向量角分解的高维多目标进化算法

Many-objective evolutionary algorithm based on vector angle decomposition

*控制与决策*. 2021, 36(3): 761-768 <https://doi.org/10.13195/j.kzyjc.2019.0925>

#### 基于分解的多目标多因子进化算法

A multiobjective multifactorial evolutionary algorithm based on decomposition

*控制与决策*. 2021, 36(3): 637-644 <https://doi.org/10.13195/j.kzyjc.2019.0525>

#### 一种求解约束多目标问题的协作进化算法

A collaborative evolutionary algorithm for solving constrained multi-objective problems

*控制与决策*. 2021, 36(11): 2656-2664 <https://doi.org/10.13195/j.kzyjc.2020.0791>

# 基于自适应双阶段分级均衡的约束高维多目标进化算法

王林锋, 揭丽琳<sup>†</sup>, 黎明, 何超, 李军华

(南昌航空大学 江西省图像处理与模式识别重点实验室, 南昌 330063)

**摘要:** 为了提高约束进化算法处理各种不同约束高维多目标优化问题的精确性和鲁棒性, 提出一种基于自适应双阶段分级均衡的约束高维多目标进化算法. 该算法将多阶段优化方法与混合约束处理技术相结合, 首先通过动态个体支配关系设计分段时机, 并自适应地切换进化过程的目标优化和约束处理两个阶段; 然后依据种群进化信息构建混合分级均衡准则, 利用自适应随机排序法在不可行状态选择个体, 并在半可行状态下定义半可行性准则以选择个体, 从而保持可行解与不可行解的动态均衡, 提高种群的收敛性、分布性和多样性. 在标准测试函数集 C\_DTLZ、DC\_DTLZ 和 MW 的大量实验结果表明, 所提出的算法对不同目标维数以及狭窄、离散或互不连通可行域的约束高维多目标问题均能取得较好的收敛性能和稳定性, 相对于 MOEA/D-FCHT、MOEA/D-2WA、PPS、ToP 和 Trip 五种先进方法, 具有更高的收敛精度和更好的鲁棒性.

**关键词:** 约束高维多目标; 多阶段优化; 约束处理技术; 分段时机; 分级均衡准则; 进化算法

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2024.0288

引用格式: 王林锋, 揭丽琳, 黎明, 等. 基于自适应双阶段分级均衡的约束高维多目标进化算法 [J]. 控制与决策, 2025, 40(5): 1512-1522.

## Constrained many-objective evolutionary algorithm based on adaptive two-stage hierarchical equilibrium

WANG Lin-feng, JIE Li-lin<sup>†</sup>, LI Ming, HE Chao, LI Jun-hua

(Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang 330063, China)

**Abstract:** To enhance the precision and robustness of constrained evolutionary algorithms for handling various constrained many-objective problems, this paper proposes a constrained many-objective evolutionary algorithm based on adaptive two-stage hierarchical equilibrium. The algorithm combines a multi-stage optimization approach with hybrid constraint handling techniques. Firstly, it designs segmentation timing based on dynamic individual dominance relationships and adaptively switches between the objective function-focused phase and the constraint handling-focused phase. Then, it constructs a hybrid hierarchical equilibrium criterion based on population evolution information, employing an adaptive random ranking method to select individuals in infeasible situations and defining a semi-feasibility criterion for selecting individuals in semi-feasible situations. This approach maintains a dynamic balance between feasible and infeasible solutions, improving population convergence, distribution, and diversity. Extensive experiments on standard test function sets C\_DTLZ, DC\_DTLZ, and MW show that the proposed algorithm achieves better convergence performance and stability across different objective dimensions and various constrained high-dimensional multi-objective problems with narrow, discrete, or disconnected feasible regions, and demonstrates that the proposed algorithm has higher convergence accuracy and better robustness compared to five advanced methods, namely MOEA/D-FCHT, MOEA/D-2WA, PPS, ToP, and Trip.

**Keywords:** constrained many-objective; multi-stage optimization; constraint handling techniques; segmentation timing; hierarchical equilibrium criterion; evolutionary algorithm

## 0 引言

在科学研究和工程应用中, 所面临的优化问题

通常包含多种约束条件及多个相互冲突和耦合的目标. 当目标函数超过 3 个时, 问题被称为约束高维多

收稿日期: 2024-03-20; 录用日期: 2024-09-28.

基金项目: 国家自然科学基金项目 (62066031, 62272209, 62261039); 江西省自然科学基金项目 (20242BAB25094); 江西省图像处理与模式识别重点实验室开放基金项目 (ET202108415).

责任编辑: 刘士新.

<sup>†</sup>通信作者. E-mail: jielilin@nchu.edu.cn.

目标优化问题 (CMaOPs)<sup>[1]</sup>, 广泛应用于航空航天、车间调度、工程设计和决策科学等领域<sup>[2-5]</sup>. 然而, 由于约束条件的复杂性和目标数量的增加, 导致可行域变得狭窄或互不连通, 并且目标空间拓扑复杂, 这使得算法难以有效地跨越不可行域, 在庞大目标空间中获得良好的收敛性和分布性. 因此, 研究 CMaOPs 对于高效地解决工程问题具有重要的理论和实践指导意义.

近年来, 尽管对约束单目标和多目标优化问题的研究已相当广泛, 但专注于 CMaOPs 的研究成果却相对较少<sup>[6-7]</sup>. 约束多目标进化算法 (CMOEA) 主要采用惩罚函数法<sup>[8]</sup>、多目标优化法<sup>[9]</sup>、目标与约束分离法<sup>[10]</sup> 以及其他混合法<sup>[11]</sup> 来处理约束条件. 惩罚函数法通过引入静态或动态惩罚因子将约束问题转化为无约束问题<sup>[12]</sup>, 虽然简单易行, 但确定合适的惩罚因子却非常困难. 多目标优化方法则通过将约束条件视为附加目标, 从而将问题转化为无约束的多目标优化问题, 再利用多目标进化算法求解. 文献 [13] 中的 MOEA/D-2WA 通过将约束违反程度转化为目标函数的一部分, 并构建不可行和可行的权重类型, 改进了基于分解的多目标进化算法, 有助于获得全局最优解, 但在目标数量增加时计算复杂度显著上升. 目标与约束分离法则包括 Deb 约束支配准则<sup>[14]</sup>、 $\varepsilon$  约束准则<sup>[15]</sup> 和随机排序方法<sup>[16]</sup>, 其中 Deb 约束支配准则是最广泛应用的方法之一, 但由于其偏好可行解而忽略不可行解, 容易导致算法陷入局部最优. 文献 [17] 中的基于分解的约束多目标进化算法 (C-MOEA/D) 通过将多目标优化转换为单目标优化, 提高了收敛速度, 但仍受限于 Deb 约束支配准则. 相比之下, 混合法通过有机融合不同的约束处理机制, 提高了种群的收敛性和搜索精度. 例如, 文献 [18] 提出了一种自适应权衡模式, 在不可行阶段采用多目标优化方法处理约束条件, 而在半可行阶段则采用自适应惩罚函数法来应对约束; 文献 [19] 则开发了混合约束处理技术 (HCT), 并将其与差分进化算法 (DE) 结合形成 ECO-HCT, 但这些方法在处理不可行状态时仍有不足. 此外, 引入模糊集理论的文献 [20] 提出了一种模糊约束处理技术 (FCHT), 并将其嵌入 C-MOEA/D 框架中, 形成 MOEA/D-FCHT, 但该算法性能高度依赖于模糊度和权重参数的选择.

多阶段优化方法与不同约束处理机制的融合研究相对有限, 文献 [21] 提出的两阶段搜索框架 (ToP) 在第 1 阶段考虑所有约束和单一的目标, 第 2 阶段则同时考虑所有约束和目标. 然而, 该算法在第 1 阶段使用约束支配准则处理约束条件, 在一定程度上影响了算法的收敛精度. 文献 [22] 提出了一种基于

Push 和 Pull 策略的搜索框架 (PPS), 虽然在收敛性和多样性方面有所改善, 但在处理复杂约束问题时解集往往过于集中或分散. 上述研究主要集中在两阶段搜索空间信息和单一约束处理技术的应用上, 缺乏对分段时机及其与混合约束准则融合方法的深入探讨, 加之多阶段方法中分段时机不当会导致算法陷入局部最优或停滞, 因此现有 CMOEAs 在处理 CMaOPs 时依然面临收敛性和分布性不足的问题.

针对上述问题, 本文结合多阶段优化方法与混合约束处理技术的优势, 提出一种基于自适应双阶段分级均衡的约束高维多目标进化算法 (CMaOEA-ATHE), 主要贡献如下:

1) 提出自适应双阶段优化策略, 动态平衡目标优化与约束满足. 第 1 阶段忽略约束条件, 全面搜索高维目标空间, 使种群快速收敛到无约束 Pareto 前沿. 第 2 阶段采用混合约束处理技术帮助种群跨越不可行区域, 收敛到可行且非支配解区域.

2) 构建混合分级均衡准则, 提高种群的分布性和收敛性. 在不可行状态下, 利用自适应随机排序法选择个体; 在半可行状态下, 定义半可行性准则选择个体, 从而有针对性地选择不同状态下的个体参与进化, 增强全局搜索能力和局部搜索精度.

3) 综合优化方法的应用: 通过自适应调整和混合约束处理技术, 有效解决了提高精确性、鲁棒性与保持算法稳定性之间的潜在冲突. 该策略确保了算法在解决 CMaOPs 时表现得更加高效和稳健.

## 1 基础知识和相关工作

### 1.1 约束高维多目标优化问题

不失一般性, 一个具有  $n$  个决策变量和  $m$  个目标函数以及  $q$  个约束条件的约束优化问题 (最小化问题) 的数学模型<sup>[21]</sup> 可表示如下:

$$\begin{aligned} \underset{\mathbf{x} \in \Omega \subset R^n}{\text{minimize}} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})). \quad (1) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, p; \\ & h_j(\mathbf{x}) = 0, \quad j = p + 1, p + 2, \dots, q; \\ & \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \Omega. \end{aligned}$$

其中:  $F$  为  $m$  个相互冲突的目标函数向量, 当  $m \geq 4$  时, 称式 (1) 为约束高维多目标优化问题;  $\mathbf{x}$  为一个候选解,  $g_j(\mathbf{x})$  为第  $j$  个不等式约束,  $h_j(\mathbf{x})$  为第  $j$  个等式约束;  $\Omega \subset R^n$  为决策变量空间.

为度量候选解  $\mathbf{x}$  对约束条件的满足程度, 定义约束违反度函数, 则每个解  $\mathbf{x}$  的约束违反度<sup>[22]</sup> 为

$$CV = \sum_{j=1}^q CV_j(\mathbf{x}). \quad (2)$$

其中:  $CV_j(\mathbf{x})$  为第  $j$  个约束条件的约束违反度, 其公式定义为

$$CV_j(\mathbf{x}) = \begin{cases} \max\{0, g_j(\mathbf{x})\}, & 1 \leq j \leq p; \\ \max\{0, |h_j(\mathbf{x})| - \delta\}, & p+1 \leq j \leq q. \end{cases} \quad (3)$$

这里  $\delta$  是一个足够小的正数, 用来放松等式约束. 当  $CV = 0$  时, 该候选解  $\mathbf{x}$  为可行解; 反之, 当  $CV > 0$  时, 该候选解  $\mathbf{x}$  为不可行解.

## 1.2 相关定义

**定义 1 (可行解)** 当且仅当  $g_j(\mathbf{x}) \leq 0 (j \in \{1, 2, \dots, p\})$ ,  $h_j(\mathbf{x}) = 0 (j \in \{p+1, p+2, \dots, q\})$  时, 称  $\mathbf{x}$  为 CMOEAs 的可行解.

**定义 2 (不可行解)** 不能同时满足  $g_j(\mathbf{x}) \leq 0 (j \in \{1, 2, \dots, p\})$  和  $h_j(\mathbf{x}) = 0 (j \in \{p+1, p+2, \dots, q\})$  的解  $\mathbf{x}$ , 称其为不可行解.

**定义 3 (可行域)** 由所有可行解组成的集合称为 CMOEAs 的可行域.

**定义 4 (不可行域)** 由所有不可行解组成的集合称为 CMOEAs 的不可行域.

**定义 5 (Pareto 最优解)** 若在决策空间中的解不被任何其他解 Pareto 支配, 则称为 Pareto 最优解.

**定义 6 (Pareto 最优解集)** 由决策空间中所有 Pareto 最优解构成的集合称为 Pareto 最优解集.

## 1.3 双阶段优化策略

在 CMOEAs 中, 目标优化与约束满足之间的平衡至关重要, 尤其在两阶段 CMOEAs 中, 如何有效切换阶段是一个关键问题. 为此, 研究者们提出了多种双阶段优化策略. 例如, 文献 [21] 提出的双阶段搜索框架 (ToP) 根据当前种群的可行率及前 1/3 可行解与归一化目标值的最大差值来切换阶段; 文献 [22] 中 Push 和 Pull 机制的搜索框架 (PPS) 通过理想点与最低点之间的最大变化率来决定推与拉的搜索过程切换; 其他 CMOEAs 采用预设的固定分段时机作为双阶段优化策略.

尽管这些策略取得了一定进展, 但仍存在局限性: ToP 中的单一参考向量可能导致解集过于集中, 影响后续阶段的可行区域探索, 甚至导致算法收敛于局部最优或停滞; PPS 的推拉机制可能会降低在不连续或不相交可行区域的收敛性, 影响可行区域的探索; 固定分段时机的策略可能浪费计算资源.

## 2 基于自适应双阶段分级均衡的约束高维多目标进化算法

有效求解 CMOEAs 的关键在于动态平衡目标优化与约束满足. CMOEAs-ATHE 提出了一种自适应

双阶段优化策略, 通过动态个体支配关系指导搜索, 使进化过程能够在目标优化与约束处理之间自适应切换, 从而提升搜索和收敛性能. 此外, 基于种群的进化信息, 将其分为不可行、半可行、可行 3 种状态, 构建了混合分级平衡准则. 在约束处理阶段, 利用混合约束处理机制帮助种群跨越不可行区域, 确保算法在可行且非支配区域的优良收敛性和分布性.

### 2.1 自适应双阶段优化策略设计

在 CMOEAs-ATHE 中, 双阶段优化的切换时机显著影响优化性能, 决定了能否有效求解 CMOEAs 或收敛到最优可行解. 因此, 目标优化与约束处理之间的阶段切换至关重要. 现有的分段时机选择指导原则通常基于特定问题的实证研究, 缺乏广泛适用性. 因此, 设计一个自适应调整分段时机的动态更新策略显得尤为重要.

本文提出一种基于当前种群进化状态的双阶段动态优化策略, 以避免 CMOEAs-ATHE 收敛至局部最优. 随着搜索的进行, 种群中的大多数个体有可能逼近 Pareto 前沿. 为此, 本文分析了 CMOEAs-ATHE 在进化过程中的支配关系变化, 并据此设计自适应双阶段优化策略, 依据非支配解的比例指导搜索过程. 数学表达式为

$$r_n(t) = \frac{N_n(t)}{N}. \quad (4)$$

其中:  $t$  为当前迭代次数,  $N_n(t)$  为第  $t$  代种群非支配解的数目,  $N$  为种群规模,  $r_n(t)$  始终在区间  $(0, 1)$  内变化.

为动态调整 CMOEAs-ATHE 的分段时机, 并使其随进化状态自适应变化, 利用  $r_n(t)$  判断种群中的大多数解是否接近 Pareto 前沿, 从而设计如下自适应双阶段优化策略.

**阶段 1:** 当种群的非支配解比例低于阈值  $v(t)$  时, 种群主要集中于目标优化过程  $\lambda_1$ , 此阶段利用基于 Pareto 支配关系的多目标进化算法 (MOEAs) 有效进行目标空间搜索, 快速逼近无约束的 Pareto 前沿.

**阶段 2:** 当种群的非支配解比例较高于阈值  $v(t)$  时, 种群进入收敛阶段, 主要集中于约束搜索过程  $\lambda_2$ , 利用 C-MOEAs/D 的混合约束处理机制帮助种群跨越不可行区域, 收敛至可行且非支配区域.

基于上述分析, 所设计的自适应双阶段优化策略可表示为

$$\lambda = \begin{cases} \lambda_1, & 0 < r_n(t) \leq v(t); \\ \lambda_2, & \text{otherwise.} \end{cases} \quad (5)$$

其中:  $\lambda$ 表示搜索过程,  $v(t)$ 为自适应调节的阈值. 为确保进化初期以较高概率进行目标优化, 而在后期以较高概率进行约束搜索, 设计如下动态调整方法:

$$v(t) = \varphi(0) \times \exp(-\kappa t/T). \quad (6)$$

其中:  $\varphi(0)$ 为初始阈值, 设定为 1;  $\kappa$ 为时间常数;  $T$ 为最大迭代次数. 显然,  $v(t)$ 的取值始终在区间  $[\varphi(0)e^{-\kappa}, \varphi(0)]$ 内变化, 并随迭代次数的增加而减少, 确保了进化初期无约束搜索更为重要, 而后期约束搜索更为重要, 从而精确控制双阶段搜索的切换.

本文提出的自适应双阶段优化策略具有以下特点: 1) 灵活性: 该策略不依赖于固定分段时机或频繁的阶段切换, 提升搜索效率, 简化 CMOEs 求解过程; 2) 自适应性: 实时监控种群进化状态, 动态调整阶段切换, 以有效利用搜索空间信息; 3) 高效性: 在目标优化与约束处理之间自适应切换, 提高搜索和收敛性能, 有效克服多阶段优化方法的局限性.

### 2.2 构建混合分级均衡准则

传统 CMOEs 通常采用单一约束处理技术, 未能有效处理不可行解, 限制了种群的分布性和收敛性. 为此, 本文提出一种基于动态可行性的混合分级均衡准则, 将种群按可行解比例分为不可行、半可行和可行 3 种状态 (见图 1), 通过适当的约束处理技术动态平衡可行解与不可行解的关系, 从而提升算法的分布性和收敛性. 可行解比例的计算公式如下:

$$r_f(t) = \frac{N_f(t)}{N}. \quad (7)$$

其中:  $N_f(t)$ 是第  $t$ 代种群中可行解的数目,  $r_f(t)$ 的取值始终在区间  $[0, 1]$ 内变化.

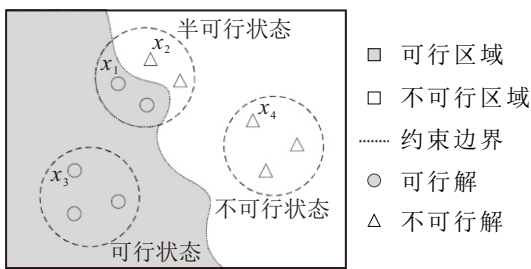


图1 混合分级均衡准则种群示意图

1) 不可行状态. 当种群中尚未找到可行解, 即  $r_f(t) = 0$ 时, 采用基于惩罚的边界交叉方法 (PBI)<sup>[23]</sup>聚合目标函数, 将目标函数值转化为标量值. 聚合函数表达式如下:

$$g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = d_1 + \theta d_2, \\ d_1 = \frac{\|(F(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{w}\|}{\|\mathbf{w}\|}, \\ d_2 = \left\| F(\mathbf{x}) - \left( \mathbf{z}^* + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|} \right) \right\|. \quad (8)$$

其中:  $w$ 是目标空间中均匀分布的权重向量,  $\mathbf{z}^*$ 代表理想点,  $\theta$ 是惩罚因子,  $d_1$ 和  $d_2$ 是两个距离参数.  $g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*)$ 值越小, 解  $\mathbf{x}$ 越优.

本文利用自适应随机排序方法 (DSR)<sup>[24]</sup>动态调整个体的选择, 提高种群探索未知领域的的能力, 使其向更优可行方向逼近. 任意两个解  $\mathbf{x}$ 和  $\mathbf{y}$ 的 DSR 定义如下:

$$\mathbf{x} \prec_{DSR} \mathbf{y} \Leftrightarrow \begin{cases} g^{pbi}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) < g^{pbi}(\mathbf{y}|\mathbf{w}, \mathbf{z}^*), \\ CV(\mathbf{x}) = CV(\mathbf{y}) = 0 \text{ or } \text{rand} < D(t); \\ CV(\mathbf{x}) < CV(\mathbf{y}), \text{ otherwise.} \end{cases} \quad (9)$$

$$D(t) = \mu(1 - t/T_{max}). \quad (10)$$

其中:  $D(t)$ 是自适应概率参数;  $\mu$ 是固定常数, 用于确定阈值边界. 可以看出,  $D(t)$ 的取值始终在区间  $(0, \mu]$ 内变化, 且随着迭代次数的增加,  $D(t)$ 减少, 确保选择优势不可行解的概率增大, 从而充分利用部分不可行解的优良性质.

2) 半可行状态. 当种群由可行解和不可行解组成, 即  $0 < r_f(t) < 1$ 时, 结合候选解的目标函数值与约束违反度构造半可行性准则. 设定  $r_f^{max}$ 为种群中可行解比例的最大值, 且  $r_f^{max} = 1$ ;  $r_f^{min}$ 为某一阈值, 且  $r_f^{min} = \alpha * r_f^{max}$ , 可通过实验分析确定. 半可行性准则基于以下 2 条准则.

**准则 1** 当可行解比例较低, 即  $0 < r_f(t) \leq r_f^{min}$ 时, 保留目标函数值更优且约束违反度更低的不可行解. 具体步骤如下.

step 1: 对种群个体  $\mathbf{x}$ 的各目标函数值进行归一化, 并计算归一化目标值的算术平均值  $\bar{f}_{nor}(\mathbf{x})$ <sup>[25]</sup>:

$$f_i^{min} = \min_{\mathbf{x} \in P} f_i(\mathbf{x}), \quad f_i^{max} = \max_{\mathbf{x} \in P} f_i(\mathbf{x}); \quad (11)$$

$$f_i^{nor}(\mathbf{x}) = \frac{f_i(\mathbf{x}) - f_i^{min}}{f_i^{max} - f_i^{min}}, \quad i = 1, 2, \dots, m; \quad (12)$$

$$\bar{f}_{nor}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m f_i^{nor}(\mathbf{x}). \quad (13)$$

其中:  $f_i^{min}$ 和  $f_i^{max}$ 分别为第  $i$ 个目标函数的最小值和最大值;  $f_i^{nor}(\mathbf{x})$ 为个体  $\mathbf{x}$ 归一化的第  $i$ 个目标函数值;  $\bar{f}_{nor}(\mathbf{x})$ 为  $m$ 个目标函数值的算术平均, 用来表征个体  $\mathbf{x}$ 的总体目标值;  $P$ 为当前种群.

step 2: 对种群个体  $\mathbf{x}$ 的各约束违反值归一化, 并计算归一化约束违反值的算术平均值  $\bar{c}_{nor}(\mathbf{x})$ <sup>[25]</sup>:

$$CV_j^{nor}(\mathbf{x}) = \frac{CV_j(\mathbf{x})}{\max_{\mathbf{x} \in P} CV_j(\mathbf{x})}, \quad j = 1, 2, \dots, q; \quad (14)$$

$$\bar{c}_{\text{nor}}(\mathbf{x}) = \frac{1}{q} \sum_{i=1}^q \text{CV}_j^{\text{nor}}(\mathbf{x}). \quad (15)$$

其中:  $\text{CV}_j(\mathbf{x})$  为第  $j$  个约束违反值,  $\text{CV}_j^{\text{nor}}(\mathbf{x})$  为个体  $\mathbf{x}$  归一化的第  $j$  个约束违反值;  $\bar{c}_{\text{nor}}(\mathbf{x})$  为  $q$  个约束违反值的算术平均, 用来度量个体  $\mathbf{x}$  的总体约束违反度.

step 3: 定义半可行性准则函数, 以度量个体  $\mathbf{x}$  相对于其他个体的优劣. 表达式为

$$F_{\text{mod}}(\mathbf{x}) = r_f(t) \times \bar{f}_{\text{nor}}(\mathbf{x}) + (1 - r_f(t)) \times \bar{c}_{\text{nor}}(\mathbf{x}), \quad (16)$$

其中  $r_f(t)$  用于动态控制目标函数值和约束违反值的权重.

**准则 2** 当可行解比例较高, 即  $r_f^{\min} < r_f(t) < r_f^{\max}$  时, 利用下式的约束支配准则, 引导种群快速逼近可行 Pareto 最优前沿:

$$\mathbf{x} <_{\text{CDP}} \mathbf{y} \Leftrightarrow \begin{cases} \bar{c}_{\text{nor}}(\mathbf{x}) = 0, \bar{c}_{\text{nor}}(\mathbf{y}) > 0; \\ \bar{f}_{\text{nor}}(\mathbf{x}) < \bar{f}_{\text{nor}}(\mathbf{y}), \\ \bar{c}_{\text{nor}}(\mathbf{x}) = \bar{c}_{\text{nor}}(\mathbf{y}) = 0; \\ \bar{c}_{\text{nor}}(\mathbf{x}) < \bar{c}_{\text{nor}}(\mathbf{y}). \end{cases} \quad (17)$$

其中可行解  $\mathbf{x}$  优于不可行解  $\mathbf{y}$ . 当  $\mathbf{x}$ 、 $\mathbf{y}$  均为可行解且  $\bar{f}_{\text{nor}}(\mathbf{x}) < \bar{f}_{\text{nor}}(\mathbf{y})$  时, 解  $\mathbf{x}$  优于解  $\mathbf{y}$ ; 若  $\mathbf{x}$ 、 $\mathbf{y}$  均为不可行解且  $\bar{c}_{\text{nor}}(\mathbf{x}) < \bar{c}_{\text{nor}}(\mathbf{y})$  时, 解  $\mathbf{x}$  优于解  $\mathbf{y}$ ; 否则, 解  $\mathbf{y}$  优于解  $\mathbf{x}$ .

3) 可行状态. 当种群完全由可行解组成时, 比较解仅基于目标函数值. 采用 PBI 方法聚合目标函数以进行比较:

$$\begin{cases} \mathbf{x} < \mathbf{y}, g^{\text{pbi}}(\mathbf{x}|\mathbf{w}, \mathbf{z}^*); \\ \mathbf{x} > \mathbf{y}, \text{otherwise.} \end{cases} \quad (18)$$

当解  $\mathbf{x}$  的 PBI 值小于解  $\mathbf{y}$  的 PBI 值时, 解  $\mathbf{x}$  优于解  $\mathbf{y}$ ; 否则, 解  $\mathbf{y}$  优于解  $\mathbf{x}$ .

根据前文分析, 在进化初期应保留稀疏区域的不可行解, 中期偏重目标值更优且约束违反值更低的不可行解, 后期则偏重约束违反值低的解. 因此, 本文采用混合分级均衡准则, 动态选择不可行解参与进化, 实现可行解和不可行解的动态均衡.

### 2.3 CMAOEA-ATHE 算法流程

CMAOEA-ATHE 的总体流程如算法 1 所示. 初始种群和空存档生成后, 算法进入主循环. 在阶段 1, 采用改进的 NSGA-II-SDR 算法<sup>[26]</sup> 进行搜索; 在阶段 2, 将混合分级均衡准则嵌入基于分解的多目标进化算法 (MOEA/D) 中, 以使种群跨越不可行区域并收敛到可行且非支配区域. 通过基于收敛性与分布性的外部存档更新策略, 进一步提升解集的性能. 如

果存档的大小为  $N$ , 则更新存档为前  $N$  个解; 否则, 根据解的 CV 值进行排序, 选择前  $N$  个解作为更新后的存档. 如果存档的大小超过  $N$ , 则使用强度 Pareto 进化算法 (SPEA2)<sup>[27]</sup> 的适应度计算方法对存档中的解进行评价, 并选择前  $N$  个解作为更新后的存档.

#### 算法 1 CMAOEA-ATHE 算法框架

输入: 种群规模  $N$ , 最大迭代次数  $T$ ;

输出: 最终的非支配解集  $P^*$ .

- 1) 初始化种群  $P$  和生成初始存档  $A$ ,  $A = \emptyset$ ;
- 2) for  $t = 1 : T$
- 3) 通过式 (4) 和 (6) 计算当前种群的非支配解比例  $r_n(t)$  和  $v(t)$ ;
- 4) /\*自适应切换机制\*/
- 5) if  $r_n(t) \leq v(t)$
- 6)  $O \leftarrow$  由选定的 MOEA (NSGA-II-SDR) 生成子代;
- 7)  $A \leftarrow$  更新存档 ( $A, P \cup O, N$ );
- 8)  $P \leftarrow$  通过选定的 MOEA (NSGA-II-SDR) 进化下一代解;
- 9) else
- 10) 将  $P$  与  $A$  交换;
- 11)  $P \leftarrow$  由改进的 C-MOEA/D 进化;
- 12) if  $r_f(t) = 0$  /\*不可行状态\*/
- 13)  $P \leftarrow$  根据式 (9) 获得最优的解;
- 14) else if  $0 < r_f(t) < 1$  /\*半可行状态\*/
- 15)  $P \leftarrow$  通过式 (16) 和 (17) 获得最优的解;
- 16) else /\*可行状态\*/
- 17)  $P \leftarrow$  通过式 (18) 获得最优的解;
- 18) end if
- 19) end if
- 20) end for
- 21) 输出  $P^*$

### 2.4 计算时间复杂度分析

时间复杂度是评价算法性能的重要指标之一, 本文的算法复杂度主要分为阶段 1、阶段 2 和外部存档更新过程.

在阶段 1 中, 使用的 MOEA 为 NSGA-II-SDR, 其时间复杂度为  $O(MN^2)$ . 其中:  $M$  为目标数,  $N$  为种群规模. 在阶段 2 中, 基于可行性的动态约束准则被嵌入 MOEA/D, 计算复杂度为  $O(MTN)$ . CHT 对计算复杂度的影响很小, 因此阶段二的时间复杂度仍为  $O(MTN)$ . 在外部存档更新过程中, 适应度计算方法与 SPEA2 相同, 存档更新的时间复杂度为  $O(MN^3)$ . 综上, CMAOEA-ATHE 的整体时间复杂度为  $O(MN^3)$ .

### 3 实验仿真与结果分析

为验证 CMaOEA-ATHE 算法的有效性, 选取 MOEA/D-2WA<sup>[13]</sup>、MOEA/D-FCHT<sup>[20]</sup>、TOP<sup>[21]</sup>、PPS<sup>[22]</sup> 和 TriP<sup>[28]</sup> 五种当前国内外性能优异的算法进行对比分析. 所有算法在 Intel Xeon 8 G 内存和 Win10 64 位操作系统的 Platemo<sup>[29]</sup> 平台上运行.

#### 3.1 实验设置

选用 C\_DTLZ、DC\_DTLZ<sup>[30]</sup> 和 MW 测试集<sup>[31]</sup> 作为基准测试问题, 分别测试不同目标数下的算法性能. 使用反向世代距离 (IGD)<sup>[32]</sup> 和超体积度量 (HV)<sup>[33]</sup> 分别度量算法的收敛性和多样性.

1) 种群规模: 所有算法的种群规模均为 100.

2) 终止条件: 采用函数的最大评价次数作为终止条件. 双目标的问题最大评价次数设为 200 000; 超过 3 目标的问题设为 400 000.

3) 统计方法: 在每个测试问题上, 所有算法独立运行 30 次, 并采用 Wilcoxon 秩和检验方法<sup>[34]</sup> 进行结果分析. 实验结果以均值和标准差表述.

4) 其他参数设置: 经过多次实验对比, CMaOEA-ATHE 算法设置  $\kappa = 0.5$ ,  $\alpha = 0.5$ ; 其他算法参数设置与其原文相同.

#### 3.2 算法在 C\_DTLZ 测试集的对比分析

本节给出了 CMaOEA-ATHE 与其他算法在 C\_DTLZ 测试集上的对比实验结果. 表 1 和表 2 分

别展示了 6 种算法 IGD 和 HV 指标的均值和标准差. 其中: 加粗数据为最优值, “+” “=” 和 “-” 分别表示对比算法获得的结果优于、无差别于和劣于 CMaOEA-ATHE.

C1\_DTLZ1 主要测试算法收敛到 Pareto 前沿的性能, 其可行域仅为靠近前沿的狭小区域, 如何在高维目标空间中搜索到均匀分布的最优可行解就显得尤其重要. CMaOEA-ATHE 的 IGD 指标在 3、5 目标 C1\_DTLZ1 问题上取得最优, 而在其他目标上次好, 说明 CMaOEA-ATHE 具有较好的收敛精度和稳定性. C2\_DTLZ2 主要测试算法跨越孤立可行域、处理不连续前沿面的能力, 其不可行解对进化的作用较大. CMaOEA-ATHE 的 HV 指标在 C2\_DTLZ2 问题的 3、5、8 和 10 目标上均取得最优, 并且在 IGD 指标上也具有一定的优势. C3\_DTLZ4 主要测试算法收敛到可行域边界的能力, 其 Pareto 前沿是由可行边界构成且连续的. 对于 C3\_DTLZ4 问题的 3、5、8 和 10 目标, CMaOEA-ATHE 所求出的 IGD 和 HV 指标均取得最优, 说明所求最优解显著优于其他对比算法.

为直观展现 CMaOEA-ATHE 的收敛性能, 本文给出了 6 种算法在 8 目标 C3\_DTLZ4 问题上获得的非支配解分布, 如图 2 所示. 仿真结果表明, CMaOEA-ATHE 算法较好地保持了收敛性和分布性, 获得的

表1 6种算法在 C\_DTLZ 上获得的 IGD 值统计结果 (均值和标准差)

problem	M	MOEA/D-2WA	MOEA/D-FCHT	PPS	ToP	TriP	CMaOEA-ATHE
C1_DTLZ1	3	2.057 8e-2 (2.05e-4)=	2.256 6e-2 (4.59e-6)-	2.632 9e-2 (8.43e-4)-	1.458 1e-1 (1.32e-1)-	2.153 0e-2 (1.66e-4)-	<b>2.053 9e-2</b> <b>(2.55e-5)</b>
	5	6.786 1e-2 (1.35e-4)-	9.703 3e-2 (9.18e-3)-	8.149 6e-2 (2.07e-3)-	1.722 9e-1 (9.33e-2)-	7.282 6e-2 (8.08e-4)-	<b>6.752 1e-2</b> <b>(4.59e-4)</b>
	8	<b>1.078 4e-1</b> <b>(3.80e-4)+</b>	2.111 8e-1 (2.61e-2)-	1.251 8e-1 (2.17e-3)-	1.282 9e-1 (5.07e-3)-	1.327 9e-1 (1.65e-3)-	1.086 5e-1 (4.54e-5)
	10	1.535 8e-1 (4.87e-4)-	2.941 5e-1 (1.13e-1)-	1.463 3e-1 (6.86e-3)+	<b>1.446 9e-1</b> <b>(4.63e-3)+</b>	1.666 7e-1 (4.55e-3)-	1.531 2e-1 (5.68e-4)
C2_DTLZ2	3	4.935 9e-2 (4.22e-5)-	5.346 9e-2 (5.47e-4)-	5.553 1e-2 (1.54e-3)-	5.835 5e-2 (2.06e-3)-	<b>4.586 2e-2</b> <b>(5.63e-4)+</b>	4.928 5e-2 (1.74e-5)
	5	<b>1.826 5e-1</b> <b>(3.44e-5)+</b>	2.377 6e-1 (5.44e-3)-	2.160 1e-1 (6.37e-3)-	2.395 5e-1 (5.37e-3)-	1.958 7e-1 (3.61e-3)-	1.827 1e-1 (3.60e-5)
	8	2.796 2e-1 (1.83e-3)=	3.751 2e-1 (4.53e-2)-	4.204 7e-1 (4.67e-2)-	6.543 3e-1 (2.11e-1)-	3.382 7e-1 (4.33e-3)-	<b>2.789 5e-1</b> <b>(2.38e-3)</b>
	10	4.244 8e-1 (1.81e-2)+	6.023 5e-1 (6.62e-2)-	6.022 0e-1 (9.49e-2)-	8.571 8e-1 (1.81e-1)-	<b>4.018 3e-1</b> <b>(4.17e-3)+</b>	4.347 7e-1 (3.26e-5)
C3_DTLZ4	3	1.165 7e-1 (1.37e-1)-	7.067 3e-1 (6.28e-2)-	1.427 5e-1 (6.21e-2)-	1.419 4e-1 (7.14e-3)-	1.068 1e-1 (2.31e-3)-	<b>9.134 2e-2</b> <b>(2.05e-4)</b>
	5	3.885 6e-1 (1.35e-1)-	9.780 2e-1 (9.96e-2)-	5.089 8e-1 (2.76e-2)-	5.223 1e-1 (2.37e-2)-	4.547 6e-1 (2.10e-2)-	<b>3.664 6e-1</b> <b>(1.18e-1)</b>
	8	6.561 9e-1 (1.06e-1)-	1.156 7e+0 (5.92e-2)-	8.070 4e-1 (2.50e-2)-	1.377 2e+0 (2.43e-1)-	9.813 0e-1 (8.84e-2)-	<b>5.973 3e-1</b> <b>(7.41e-2)</b>
	10	8.192 4e-1 (1.02e-1)-	1.252 7e+0 (6.33e-2)-	9.335 0e-1 (3.35e-2)-	1.753 2e+0 (1.49e-1)-	1.342 5e+0 (1.27e-1)-	<b>7.616 8e-1</b> <b>(6.03e-2)</b>
+ / - / =		3/7/2	0/12/0	1/11/0	1/11/0	2/10/0	

表2 6种算法在 C\_DTLZ 上获得的 HV 值统计结果 (均值和标准差)

problem	M	MOEA/D-2WA	MOEA/D-FCHT	PPS	ToP	TriP	CMAOEA-ATHE
C1_DTLZ1	3	8.410 5e-1 (2.32e-3)-	7.999 0e-1 (9.51e-5)-	8.218 5e-1 (3.47e-3)-	5.500 6e-1 (2.39e-1)-	8.380 9e-1 (9.44e-4)-	<b>8.414 2e-1</b> <b>(4.86e-4)</b>
	5	<b>9.699 9e-1</b> <b>(7.28e-4)+</b>	8.546 1e-1 (2.02e-2)-	9.460 8e-1 (3.37e-3)-	6.815 8e-1 (2.09e-1)-	9.674 5e-1 (4.52e-4)+	9.636 6e-1 (7.15e-3)
	8	9.698 7e-1 (1.00e-2)-	6.954 8e-1 (8.79e-2)-	9.724 9e-1 (4.23e-3)-	9.521 9e-1 (1.11e-2)-	<b>9.930 1e-1</b> <b>(5.44e-4)+</b>	9.874 9e-1 (2.68e-3)
	10	<b>9.936 6e-1</b> <b>(3.77e-3)+</b>	4.923 6e-1 (2.26e-1)-	9.757 7e-1 (7.61e-3)-	9.591 8e-1 (7.95e-3)-	9.925 1e-1 (3.75e-3)+	9.811 7e-1 (1.26e-2)
C2_DTLZ2	3	5.150 8e-1 (1.49e-4)-	4.873 7e-1 (1.59e-3)-	4.998 4e-1 (3.60e-3)-	4.734 8e-1 (5.10e-3)-	5.141 9e-1 (1.67e-3)-	<b>5.152 8e-1</b> <b>(4.36e-5)</b>
	5	7.235 6e-1 (5.78e-4)=	5.902 2e-1 (8.90e-3)-	6.157 0e-1 (1.53e-2)-	4.791 2e-1 (1.34e-2)-	6.936 9e-1 (8.69e-3)-	<b>7.237 0e-1</b> <b>(5.12e-4)</b>
	8	7.989 5e-1 (4.25e-3)-	5.030 5e-1 (3.79e-2)-	5.437 9e-1 (2.73e-2)-	3.219 2e-1 (1.17e-1)-	7.874 5e-1 (1.12e-2)-	<b>8.022 8e-1</b> <b>(4.55e-3)</b>
	10	8.667 7e-1 (4.73e-3)-	4.042 5e-1 (5.13e-2)-	5.265 3e-1 (5.28e-2)-	2.160 6e-1 (9.21e-2)-	8.042 5e-1 (1.41e-2)-	<b>8.709 3e-1</b> <b>(2.77e-3)</b>
C3_DTLZ4	3	7.865 0e-1 (4.89e-2)-	5.226 1e-1 (5.91e-2)-	7.645 9e-1 (1.89e-2)-	7.553 3e-1 (5.56e-3)-	7.836 7e-1 (1.41e-3)-	<b>7.961 1e-1</b> <b>(2.51e-4)</b>
	5	9.405 0e-1 (2.91e-2)-	5.897 3e-1 (8.81e-2)-	8.993 6e-1 (8.86e-3)-	8.234 8e-1 (1.87e-2)-	9.061 3e-1 (8.99e-3)-	<b>9.470 0e-1</b> <b>(2.31e-2)</b>
	8	9.819 2e-1 (1.64e-2)-	7.461 2e-1 (6.50e-2)-	9.140 2e-1 (1.41e-2)-	4.174 7e-1 (1.78e-1)-	8.291 7e-1 (9.04e-2)-	<b>9.889 7e-1</b> <b>(5.37e-3)</b>
	10	9.793 3e-1 (3.17e-2)-	7.286 6e-1 (7.77e-2)-	9.206 5e-1 (1.56e-2)-	2.190 1e-1 (7.79e-2)-	6.447 8e-1 (1.19e-1)-	<b>9.950 8e-1</b> <b>(4.18e-3)</b>
+ / - / =		2/9/1	0/12/0	0/12/0	0/12/0	3/9/0	

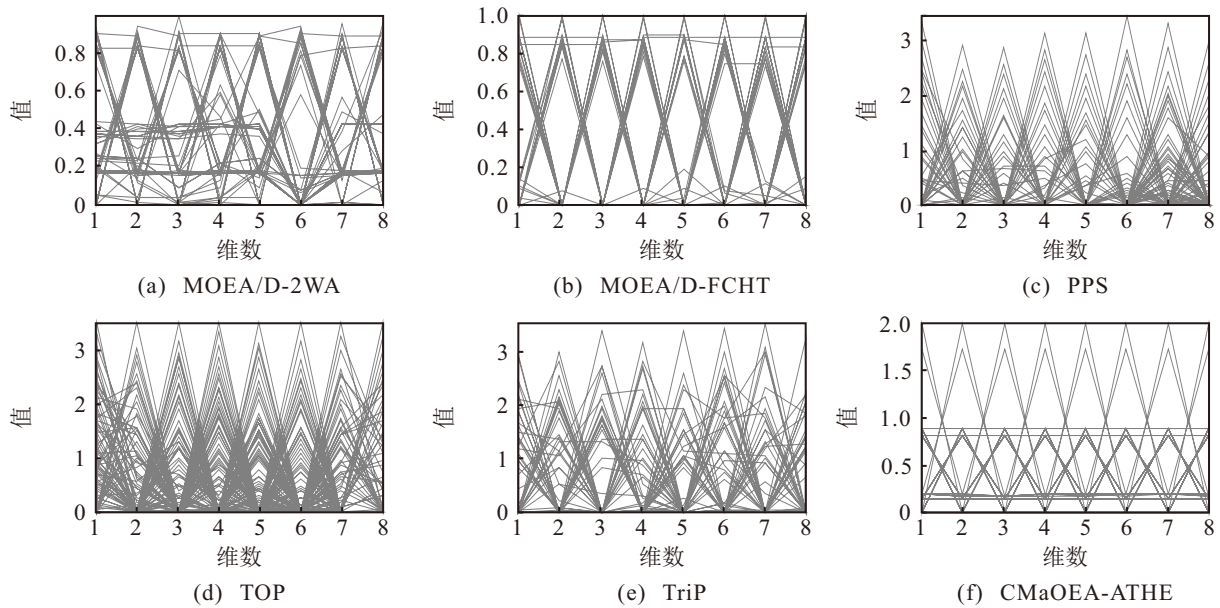


图2 C3\_DTLZ4 问题 8 目标上获得的非支配解集

非支配解集显著优于其他对比算法。

综合表 1、表 2 和图 2 可知, CMAOEA-ATHE 在 C\_DTLZ 测试集上性能显著, 且在处理不同目标维数的约束多目标问题时普适性较好。

### 3.3 算法在 DC\_DTLZ 测试集上的对比分析

为进一步探讨算法在 DC\_DTLZ 测试集上的性能, 本节中给出了对比算法在 DC\_DTLZ 测试集上的 IGD 值和 HV 值的统计结果, 如表 3 和表 4 所示。

DC1\_DTLZ1 和 DC1\_DTLZ3 的可行域在 Pareto 前沿呈现不连续分布, 主要用于测试算法跨越孤立可行域及处理不连续前沿面的能力。CMAOEA-ATHE 的 IGD 指标在 DC1\_DTLZ1 的 5 和 8 目标上获得最优性能, 而在 3 和 10 目标上次好; 在 DC1\_DTLZ3 的 5、8 和 10 目标上获得最优值, 而在 3 目标上略低于 TriP。DC2\_DTLZ1 和 DC2\_DTLZ3 问题的目标空间中存在不可行的屏障, 主要测试算法跨越不可行区域并收敛到 Pareto 前沿的能力。在 DC2\_DTLZ1

表3 6种算法在DC\_DTLZ上获得的IGD值统计结果(均值和标准差)

problem	M	MOEA/D-2WA	MOEA/D-FCHT	PPS	ToP	TriP	CMAOEA-ATHE
DC1_DTLZ1	3	2.588 8e-2 (6.01e-3)-	1.995 8e-2 (1.09e-5)-	4.052 3e-2 (5.83e-2)-	1.934 6e-2 (4.68e-3)=	<b>1.246 8e-2</b> (1.32e-4)+	1.875 2e-2 (3.83e-3)
	5	5.471 8e-2 (3.50e-5)-	7.847 1e-2 (2.68e-3)-	8.779 5e-2 (6.60e-2)-	7.829 7e-1 (1.87e+0)-	6.028 7e-2 (2.05e-3)-	<b>5.449 0e-2</b> (3.48e-4)
	8	8.537 3e-2 (2.92e-4)-	1.471 7e-1 (1.82e-2)-	1.169 7e-1 (9.66e-3)-	8.189 9e-1 (1.77e+0)-	1.411 0e-1 (5.44e-2)-	<b>8.420 3e-2</b> (7.33e-5)
	10	<b>1.032 3e-1</b> (9.12e-4)+	1.494 7e+0 (2.81e+0)-	1.358 8e-1 (9.68e-3)-	7.086 2e-1 (1.86e+0)-	1.469 6e-1 (1.41e-2)-	1.053 8e-1 (4.62e-4)
DC1_DTLZ3	3	4.509 0e-2 (4.16e-5)=	5.416 4e-2 (5.73e-4)-	4.240 5e-1 (5.73e-1)-	3.063 7e-1 (5.44e-1)-	<b>3.674 0e-2</b> (5.82e-4)+	4.509 7e-2 (1.86e-5)
	5	1.807 0e-1 (2.85e-4)=	3.254 7e-1 (1.71e-1)-	1.789 7e+0 (2.02e+0)-	4.158 4e+0 (2.38e+0)-	2.909 0e-1 (7.09e-2)-	<b>1.806 9e-1</b> (2.29e-4)
	8	4.339 9e-1 (1.40e-1)=	7.740 6e-1 (2.25e-1)-	9.721 4e-1 (4.70e-1)-	1.493 7e+1 (7.92e+0)-	4.421 3e+0 (3.60e+0)-	<b>4.018 7e-1</b> (2.07e-2)
	10	4.794 9e-1 (4.85e-3)=	5.533 3e+0 (4.95e+0)-	3.786 4e+0 (5.03e+0)-	1.956 8e+1 (5.11e+0)-	8.951 3e+0 (7.48e+0)-	<b>4.763 6e-1</b> (1.65e-3)
DC2_DTLZ1	3	3.013 6e-2 (3.64e-2)=	2.068 7e-2 (1.77e-4)-	7.582 4e-2 (6.87e-2)-	NaN (NaN)	2.259 5e-2 (3.88e-4)-	<b>2.056 3e-2</b> (1.72e-5)
	5	7.455 2e-2 (2.46e-2)=	7.607 1e-2 (2.60e-3)-	8.987 7e-2 (2.10e-2)-	NaN (NaN)	7.8169e-2 (3.20e-3)-	<b>6.807 7e-2</b> (6.46e-5)
	8	1.236 0e-1 (2.62e-2)-	1.455 8e-1 (1.70e-2)-	1.644 9e-1 (3.01e-2)-	NaN (NaN)	1.7627e-1 (4.37e-2)-	<b>1.087 0e-1</b> (2.94e-6)
	10	1.972 4e-1 (5.81e-2)=	2.166 1e-1 (2.94e-2)-	1.704 6e-1 (2.66e-2)=	NaN (NaN)	2.234 2e-1 (4.90e-2)-	<b>1.540 7e-1</b> (4.01e-5)
DC2_DTLZ3	3	4.597 0e-1 (2.06e-1)=	1.042 9e-1 (1.52e-1)+	4.820 6e-1 (1.97e-1)+	NaN (NaN)	<b>5.772 9e-2</b> (8.52e-4)+	5.606 7e-1 (3.83e-4)
	5	6.065 8e-1 (7.58e-2)=	4.597 0e-1 (2.18e-1)=	4.973 6e-1 (2.15e-1)-	NaN (NaN)	5.977 0e-1 (1.65e-1)-	<b>4.571 8e-1</b> (2.03e-1)
	8	6.361 5e-1 (1.66e-1)-	7.292 4e-1 (1.98e-1)-	7.957 4e-1 (1.70e-1)-	9.718 8e-1 (1.19e-1)-	8.085 9e-1 (1.52e-1)-	<b>3.869 8e-1</b> (2.88e-5)
	10	7.783 1e-1 (1.41e-1)-	1.100 1e+0 (3.21e-1)-	9.081 7e-1 (1.48e-1)-	1.225 7e+0 (0.00e+0)=	8.742 3e-1 (1.91e-1)-	<b>5.466 1e-1</b> (1.20e-1)
+/-/=		1/6/9	1/14/1	1/14/1	0/14/2	3/13/0	

表4 6种算法在DC\_DTLZ上获得的HV值统计结果(均值和标准差)

problem	M	MOEA/D-2WA	MOEA/D-FCHT	PPS	ToP	TriP	CMAOEA-ATHE
DC1_DTLZ1	3	6.017 9e-1 (2.01e-2)-	5.924 8e-1 (2.06e-4)-	5.654 3e-1 (7.45e-2)-	5.934 5e-1 (1.82e-2)-	<b>6.300 4e-1</b> (9.60e-4)+	6.255 9e-1 (1.28e-2)
	5	<b>7.448 9e-1</b> (2.54e-4)+	6.790 5e-1 (1.55e-2)-	6.839 6e-1 (1.34e-1)-	1.780 7e-1 (2.07e-1)-	7.446 9e-1 (3.58e-3)=	7.447 5e-1 (9.76e-4)
	8	6.737 2e-1 (5.14e-3)=	5.109 9e-1 (6.15e-2)-	<b>7.579 5e-1</b> (1.74e-2)+	1.703 1e-1 (1.47e-1)-	6.779 8e-1 (1.62e-1)+	6.733 8e-1 (9.19e-4)
	10	6.246 1e-1 (1.66e-2)-	8.342 5e-2 (1.16e-1)-	<b>7.179 9e-1</b> (2.85e-2)+	1.890 7e-1 (9.44e-2)-	6.962 7e-1 (3.57e-2)+	6.523 5e-1 (5.30e-3)
DC1_DTLZ3	3	<b>4.728 2e-1</b> (7.42e-5)=	4.407 9e-1 (3.46e-3)-	3.059 7e-1 (1.33e-1)-	3.036 0e-1 (1.73e-1)-	4.717 6e-1 (1.20e-3)-	4.728 1e-1 (4.24e-5)
	5	7.458 0e-1 (1.58e-3)=	4.955 9e-1 (2.04e-1)-	1.112 2e-1 (1.79e-1)-	7.168 9e-3 (2.21e-2)-	4.782 7e-1 (1.54e-1)-	<b>7.464 3e-1</b> (1.23e-3)
	8	7.961 2e-1 (1.90e-1)=	3.236 5e-1 (2.38e-1)-	1.306 4e-1 (8.31e-2)-	3.354 0e-3 (1.28e-2)-	4.487 1e-2 (1.05e-1)-	<b>8.544 9e-1</b> (1.20e-2)
	10	<b>9.216 6e-1</b> (8.35e-3)=	6.988 9e-2 (1.39e-1)-	5.547 6e-2 (7.45e-2)-	0.000 0e+0 (0.00e+0)-	5.778 6e-3 (1.98e-2)-	9.201 2e-1 (6.84e-3)
DC2_DTLZ1	3	8.174 7e-1 (9.22e-2)=	8.404 4e-1 (1.08e-3)-	6.827 6e-1 (1.80e-1)-	NaN (NaN)	8.360 2e-1 (1.42e-3)-	<b>8.416 5e-1</b> (2.14e-4)
	5	9.577 0e-1 (4.92e-2)-	9.620 3e-1 (4.25e-3)-	9.405 0e-1 (5.32e-2)-	NaN (NaN)	9.625 9e-1 (2.88e-3)-	<b>9.706 4e-1</b> (2.10e-4)
	8	9.362 4e-1 (7.30e-2)-	9.200 5e-1 (3.71e-2)-	9.094 3e-1 (6.34e-2)-	NaN (NaN)	9.113 0e-1 (5.95e-2)-	<b>9.898 7e-1</b> (9.80e-5)
	10	9.439 2e-1 (6.06e-2)-	6.466 5e-1 (1.96e-1)-	9.463 7e-1 (5.25e-2)-	NaN (NaN)	9.009 8e-1 (5.80e-2)-	<b>9.971 7e-1</b> (5.07e-5)
DC2_DTLZ3	3	1.182 3e-1 (2.24e-1)=	5.017 1e-1 (1.37e-1)+	1.031 7e-1 (2.00e-1)+	NaN (NaN)	<b>5.563 5e-1</b> (3.18e-3)+	8.062 7e-3 (3.54e-5)
	5	6.045 8e-2 (1.37e-1)=	<b>4.788 2e-1</b> (2.43e-1)=	2.897 0e-1 (3.07e-1)-	NaN (NaN)	1.399 0e-1 (2.46e-1)-	3.332 1e-1 (3.66e-1)
	8	2.773 6e-1 (4.03e-1)-	4.897 6e-1 (2.36e-1)-	1.547 3e-1 (2.24e-1)-	3.161 6e-2 (8.33e-3)-	1.904 8e-1 (2.41e-1)-	<b>8.863 4e-1</b> (3.52e-4)
	10	1.877 8e-1 (3.82e-1)-	1.490 8e-1 (1.91e-1)-	1.472 3e-1 (1.39e-1)-	8.891 1e-3 (0.00e+0)=	2.848 2e-1 (2.65e-1)-	<b>8.144 3e-1</b> (3.25e-1)
+/-/=		1/6/9	1/14/1	3/13/0	0/9/7	4/11/1	

问题的 3、5、8 和 10 目标上, CMaOEA-ATHE 的 IGD 和 HV 指标均获得最优性能; 在 DC2\_DTLZ3 问题中求得的 IGD 指标相对较大, 且在 HV 指标上也具有一定优势, 其最优解优于其他对比算法. 结果表明, CMaOEA-ATHE 在 DC\_DTLZ 测试集中的稳定性、分布性和收敛性能明显优于其他 5 种算法, 尤其在处理具有不连续可行域或复杂 Pareto 前沿的高维多目标优化问题时展现出很强的竞争力.

### 3.4 算法在 MW 测试集的对比分析

深入分析 CMaOEA-ATHE 在不同测试问题中的性能, 表 5 给出了对比算法在 MW 测试集上的 IGD 值统计结果, 图 3 则展示了 6 种算法在 10 目标 MW14 问题上获得的非支配解分布.

分析表 5 和图 3 可知, 对于狭窄、离散或不规则可行域的 MW 测试问题, CMaOEA-ATHE 在 13 例问题中取得了 8 个最优的 IGD 指标值, 超过算例的

表5 6种算法在 MW 上获得的 IGD 值统计结果 (均值和标准差)

problem	M	MOEA/D-2WA	MOEA/D-FCFT	PPS	ToP	TriP	CMaOEA-ATHE
MW2	2	1.994 2e-2 (6.51e-3)=	1.591 7e-1 (1.04e-1)-	1.472 9e-1 (9.01e-2)-	1.417 0e-1 (1.11e-1)-	1.704 8e-2 (8.00e-3)=	<b>1.696 0e-2</b> <b>(8.95e-3)</b>
	3	4.137 0e-2 (4.76e-4)-	5.727 6e-2 (2.07e-3)-	5.446 6e-2 (2.13e-3)-	2.258 0e-1 (2.29e-1)-	4.552 3e-2 (6.58e-3)-	<b>4.112 4e-2</b> <b>(8.07e-6)</b>
MW4	5	1.350 6e-1 (6.35e-4)-	1.854 4e-1 (1.12e-2)-	1.873 3e-1 (8.39e-3)-	2.270 1e-1 (3.23e-2)-	1.699 1e-1 (4.73e-3)-	<b>1.347 6e-1</b> <b>(3.96e-4)</b>
	8	<b>2.159 9e-1</b> <b>(3.75e-4)+</b>	3.697 3e-1 (4.74e-2)-	2.928 5e-1 (7.84e-3)-	3.500 8e-1 (1.08e-1)-	3.644 7e-1 (1.75e-2)-	2.170 8e-1 (1.25e-4)
	10	<b>3.052 1e-1</b> <b>(1.21e-2)+</b>	4.809 2e-1 (8.76e-2)-	3.232 5e-1 (1.88e-2)-	3.497 9e-1 (1.35e-2)-	4.475 5e-1 (2.59e-2)-	3.094 5e-1 (1.42e-4)
MW8	3	5.128 1e-2 (3.78e-3)=	1.669 3e-1 (7.97e-2)-	1.425 8e-1 (5.30e-2)-	1.822 8e-1 (2.15e-1)-	<b>4.801 9e-2</b> <b>(1.87e-3)+</b>	5.106 8e-2 (2.14e-3)
	5	2.034 4e-1 (1.87e-3)-	3.168 2e-1 (5.51e-2)-	2.940 6e-1 (5.31e-2)-	5.428 5e-1 (3.84e-1)-	2.318 3e-1 (1.14e-2)-	<b>2.024 1e-1</b> <b>(8.64e-4)</b>
	8	3.944 7e-1 (2.27e-3)-	6.002 4e-1 (1.12e-1)-	5.446 9e-1 (2.97e-2)-	6.677 0e-1 (3.11e-1)-	4.796 4e-1 (7.66e-3)-	<b>3.915 3e-1</b> <b>(1.65e-3)</b>
	10	4.826 9e-1 (2.19e-3)-	5.760 5e-1 (5.31e-2)-	6.190 6e-1 (3.58e-2)-	5.691 7e-1 (8.02e-3)-	5.927 9e-1 (9.93e-3)-	<b>4.763 9e-1</b> <b>(4.84e-3)</b>
MW14	3	2.122 2e-1 (9.22e-4)=	2.046 5e-1 (5.87e-3)+	1.439 8e-1 (2.41e-2)+	2.061 4e-1 (7.01e-2)+	<b>1.059 2e-1</b> <b>(2.57e-3)+</b>	2.109 4e-1 (5.21e-3)
	5	6.951 0e-1 (1.76e-2)-	7.611 8e-1 (6.42e-2)-	1.102 8e+0 (3.72e-1)-	2.008 6e+0 (4.16e-1)-	<b>4.411 0e-1</b> <b>(1.69e-2)+</b>	6.811 0e-1 (2.95e-2)
	8	9.665 6e-1 (2.63e-2)-	1.355 7e+0 (6.51e-2)-	1.238 1e+0 (4.65e-2)-	1.227 1e+0 (3.14e-2)-	1.210 3e+0 (3.74e-2)-	<b>9.598 7e-1</b> <b>(1.11e-2)</b>
	10	1.827 9e+0 (4.89e-2)-	1.933 1e+0 (1.76e-1)-	1.637 9e+0 (3.01e-2)=	1.651 8e+0 (1.93e-2)-	1.607 4e+0 (1.77e-2)=	<b>1.599 9e+0</b> <b>(9.78e-2)</b>
+ / - / =		2/5/6	1/12/0	1/12/0	1/12/0	3/8/2	

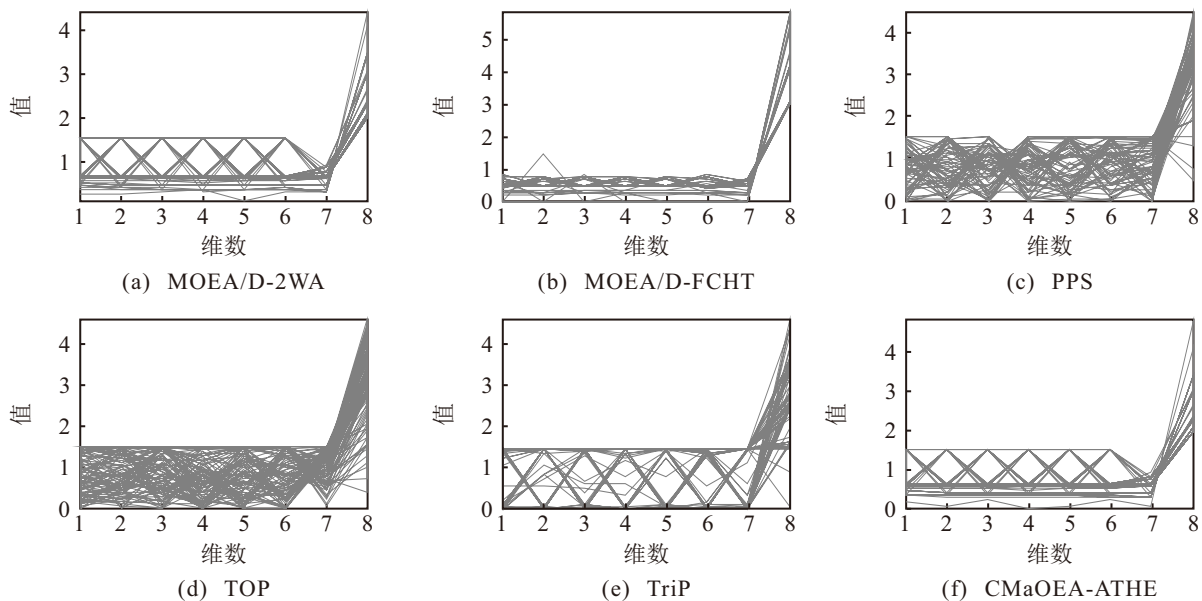


图3 MW14 问题 10 目标上获得的非支配解集

61%,且在保持种群的收敛性、多样性和分布性方面表现出明显优势.特别是随着约束复杂性的增加,CMaOEA-ATHE的优势更加突出,原因在于该方法有效地结合了自适应双阶段优化策略和混合分级均衡准则的优点.

综合分析表1~表5以及图2、图3的结果,CMaOEA-ATHE在处理大多数不规则可行域或复杂Pareto前沿的CMaOPs时表现出优异性能.该算法有效保持了种群的收敛性、多样性和分布性,展现了良好的鲁棒性,表明了自适应双阶段策略和混合分级均衡准则的有效性.因此,CMaOEA-ATHE是一种高效的约束高维多目标进化算法,能有效解决复杂CMaOPs并获得满意的非支配解集.

## 4 结论

本文结合多阶段优化方法和混合约束处理技术的优势,提出了一种基于自适应双阶段分级均衡的约束高维多目标进化算法.主要创新包括:提出了自适应双阶段优化策略,能够在进化过程中动态调整目标优化和约束处理的切换,充分利用搜索空间信息;构建了混合分级均衡准则,通过自适应随机排序法和定义的半可行性准则,针对性地选择不同状态下的个体参与进化,以实现可行和不可行个体的动态平衡,从而提高种群的收敛性、多样性和分布性.在标准测试集C\_DTLZ、DC\_DTLZ和MW上进行了大量对比实验,结果表明相较于其他5种性能较好的算法,本文算法具有更高的收敛精度和更好的鲁棒性,对不同目标维数及复杂可行域的CMaOPs均具有良好的收敛性能和稳定性.

未来研究将继续保持算法的直观性和实用性,并探索更高级的自适应机制和更复杂的种群管理策略,以进一步提升其在实际应用中的价值和参考意义.

## 参考文献 (References)

- [1] 顾清华, 莫明慧, 卢才武, 等. 求解约束高维多目标问题的分解约束支配NSGA-II优化算法[J]. 控制与决策, 2020, 35(10): 2466-2474.  
(Gu Q H, Mo M H, Lu C W, et al. Decomposition-based constrained dominance principle NSGA-II for constrained many-objective optimization problems[J]. Control and Decision, 2020, 35(10): 2466-2474.)
- [2] Zhang Y C, Ma M C, Yang X Y, et al. Disturbance-observer-based fixed-time control for 6-DOF spacecraft rendezvous and docking operations under full-state constraints[J]. Acta Astronautica, 2023, 205: 225-238.
- [3] 朱家政, 张宏立, 王聪, 等. 基于深度强化学习的模糊作业车间调度问题[J]. 控制与决策, 2024, 39(2): 595-603.

- (Zhu J Z, Zhang H L, Wang C, et al. Fuzzy job shop scheduling problem based on deep reinforcement learning[J]. Control and Decision, 2024, 39(2): 595-603.)
- [4] 周鲜成, 郑梓亮, 杨堃, 等. 城市物流配送中时间-位置依赖型多目标绿色车辆路径问题研究[J]. 控制与决策, 2025, 40(2): 413-422.  
(Zhou X C, Zheng Z L, Yang K, et al. Research on time-and-site dependent multi-objective green vehicle routing problem in urban logistics distribution[J]. Control and Decision, 2025, 40(2): 413-422.)
- [5] Li X, Lai J G, Tang R L. A hybrid constraints handling strategy for multiconstrained multiobjective optimization problem of microgrid economical/environmental dispatch[J]. Complexity, 2017, 2017(1): 6249432.
- [6] 吕莉, 潘宁康, 肖人彬, 等. 求解带约束优化问题的混合式多策略萤火虫算法[J]. 控制与决策, 2024, 39(8): 2551-2559.  
(Lv L, Pan N K, Xiao R B, et al. Hybrid multi-strategy firefly algorithm for solving optimization problems with constraints[J]. Control and Decision, 2024, 39(8): 2551-2559.)
- [7] 王飞梦, 孙靖, 戴红伟. 区间约束违反度引导的区间约束多目标进化算法[J]. 控制与决策, 2024, 39(12): 4083-4092.  
(Wang F M, Sun J, Dai H W. An interval constraint violation degree guided interval constrained multiobjective evolutionary algorithm[J]. Control and Decision, 2024, 39(12): 4083-4092.)
- [8] Ma Z W, Wang Y. Shift-based penalty for evolutionary constrained multiobjective optimization and its application[J]. IEEE Transactions on Cybernetics, 2023, 53(1): 18-30.
- [9] Harada T. Parallel cooperative multiobjective coevolutionary algorithm for constrained multiobjective optimization problems[J]. Applied Soft Computing, 2024, 153: 111290.
- [10] Qiao K J, Chen Z L, Qu B Y, et al. A dual-population evolutionary algorithm based on dynamic constraint processing and resources allocation for constrained multi-objective optimization problems[J]. Expert Systems with Applications, 2024, 238: 121707.
- [11] Ning W K, Guo B L, Yan Y Y, et al. Constrained multi-objective optimization using constrained non-dominated sorting combined with an improved hybrid multi-objective evolutionary algorithm[J]. Engineering Optimization, 2017, 49(10): 1645-1664.
- [12] Dang Q L. Multiple dynamic penalties based on decomposition for constrained optimization[J]. Expert Systems with Applications, 2022, 206: 117820.
- [13] Jiao R W, Zeng S Y, Li C H, et al. Two-type weight adjustments in MOEA/D for highly constrained many-objective optimization[J]. Information Sciences, 2021, 578: 592-614.
- [14] 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.
- [15] Qiao K J, Liang J, Yu K J, et al. Self-adaptive resources allocation-based differential evolution for constrained evolutionary optimization[J]. *Knowledge-Based Systems*, 2022, 235: 107653.
- [16] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2000, 4(3): 284-294.
- [17] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [18] Wang Y, Cai Z X, Zhou Y R, et al. An adaptive tradeoff model for constrained evolutionary optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 80-92.
- [19] Peng H, Xu Z Z, Qian J Y, et al. Evolutionary constrained optimization with hybrid constraint-handling technique[J]. *Expert Systems with Applications*, 2023, 211: 118660.
- [20] Han D, Du W L, Jin Y C, et al. A fuzzy constraint handling technique for decomposition-based constrained multi- and many-objective optimization[J]. *Information Sciences*, 2022, 597: 318-340.
- [21] Liu Z Z, Wang Y. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 870-884.
- [22] Fan Z, Li W J, Cai X Y, et al. Push and pull search for solving constrained multi-objective optimization problems[J]. *Swarm and Evolutionary Computation*, 2019, 44: 665-679.
- [23] Pang L M, Ishibuchi H, Shang K. Use of two penalty values in multiobjective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Cybernetics*, 2023, 53(11): 7174-7186.
- [24] Li X, An Q, Zhang J, et al. A novel two-stage constraints handling framework for real-world multi-constrained multi-objective optimization problem based on evolutionary algorithm[J]. *Applied Intelligence*, 2021, 51(11): 8212-8229.
- [25] Jiao L C, Luo J J, Shang R H, et al. A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems[J]. *Applied Soft Computing*, 2014, 14: 363-380.
- [26] Tian Y, Cheng R, Zhang X Y, et al. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(2): 331-345.
- [27] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength pareto evolutionary algorithm[J]. Technical Report Gloriosastrasse, 2001, 103: 95-100.
- [28] 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.
- [29] Ming F, Gong W Y, Wang L, et al. A tri-population based co-evolutionary framework for constrained multi-objective optimization problems[J]. *Swarm and Evolutionary Computation*, 2022, 70: 101055.
- [30] Li K, Chen R Z, Fu G T, et al. Two-archive evolutionary algorithm for constrained multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(2): 303-315.
- [31] Ma Z W, Wang Y. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(6): 972-986.
- [32] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multiobjective optimizers: An analysis and review[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117-132.
- [33] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. *Evolutionary Computation*, 2011, 19(1): 45-76.
- [34] Bosman P A N, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 174-188.

## 作者简介

王林锋 (1997-), 男, 硕士生, 主要研究方向为进化计算, E-mail: wanglf520qq@163.com;

揭丽琳 (1987-), 女, 副教授, 博士, 主要研究方向为进化计算、模式识别, E-mail: jielilin@nchu.edu.cn;

黎明 (1965-), 男, 教授, 博士, 主要研究方向为进化计算、模式识别, E-mail: liming@nchu.edu.cn;

何超 (1992-), 男, 讲师, 博士, 主要研究方向为进化算法、图像处理, E-mail: 71368@nchu.edu.cn;

李军华 (1974-), 男, 教授, 博士, 主要研究方向为进化算法、智能控制, E-mail: jhlee126@126.com.