

利用重叠关系分类的自适应双种群框架求解约束多目标优化问题

黄维灿

广东工业大学数学与统计学院, 广东 广州

收稿日期: 2026年2月28日; 录用日期: 2026年3月27日; 发布日期: 2026年4月7日

摘要

在约束多目标优化问题(CMOPs)中, 现有的约束多目标进化算法往往难以有效平衡多个冲突目标与复杂约束之间的关系, 导致收敛性和多样性不足。本文提出一种自适应双种群进化算法(DPSCEA), 通过对约束Pareto前沿(CPF)与无约束Pareto前沿(UPF)的重叠位置关系进行分类, 分类后根据不同位置关系采取针对性演化策略。DPSCEA采用双种群框架: P_c 种群逼近CPF, P_u 种群逼近UPF。演化过程分为学习阶段(分类关系)和演化阶段(基于重叠类型自适应调整参数和进化算子策略)。并通过增强转移机制, 将 P_c 种群的非支配及精英个体迁移至 P_u 种群, 提升优秀目标种群的利用效率。试验结果表明, 与目前六种先进算法比较, DPSCEA在IGD和HV指标上表现出显著优势, 结果验证了DPSCEA在处理约束多目标优化方面的有效性和先进性。

关键词

约束多目标, 进化算法, 重叠关系, 自适应

An Adaptive Dual-Population Framework Based on Overlapping Relationship Classification for Solving Constrained Multi-Objective Optimization Problems

Weican Huang

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: February 28, 2026; accepted: March 27, 2026; published: April 7, 2026

Abstract

In constrained multi-objective optimization problems (CMOPs), existing constrained multi-objective

evolutionary algorithms often struggle to effectively balance multiple conflicting objectives with complex constraints, resulting in insufficient convergence and diversity. This paper proposes an adaptive dual-population evolutionary algorithm (DPSCEA), which classifies the overlapping positional relationships between the constrained Pareto front (CPF) and the unconstrained Pareto front (UPF), and then adopts targeted evolutionary strategies based on different positional relationships. DPSCEA uses a dual-population framework: the P_C population approximates the CPF, while the P_U population approximates the UPF. The evolutionary process is divided into a learning phase (classifying relationships) and an evolution phase (adaptively adjusting parameters and evolutionary operator strategies based on overlap type). Through an enhanced transfer mechanism, nondominated and elite individuals from the P_C population are migrated to the P_U population, improving the utilization efficiency of high-quality objective populations. Experimental results show that, compared with six state-of-the-art algorithms, DPSCEA exhibits significant advantages in IGD and HV metrics, and the results validate the effectiveness and superiority of DPSCEA in handling constrained multi-objective optimization.

Keywords

Constrained Multi-Objective, Evolutionary Algorithms, Overlapping Relationships, Adaptive

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

在工程设计、智能制造、能源系统规划以及复杂系统决策等实际应用中[1], 优化问题往往同时涉及多个相互冲突的目标, 并受到一系列等式或不等式约束的限制, 这种情况下的难题通常被称为是约束多目标优化问题(Constrained Multi-Objective Optimization Problems, CMOPs)。CMOPs 的数学定义如下:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_M(x)) \quad (1)$$

约束条件为:

$$\begin{aligned} g_n(x) &\leq 0, \quad n = 1, 2, 3, \dots, l \\ h_n(x) &= 0, \quad n = l+1, l+2, l+3, \dots, q \\ x &\in S \subseteq R^M \end{aligned} \quad (2)$$

其中 $x = (x_1, x_2, \dots, x_D)$ 为决策向量, $F: S \rightarrow R^M$ 为目标函数, $g_n(x)$ 和 $h_n(x)$ 分别为不等式和等式约束, l 为不等式约束数, $q-l$ 为等式约束数。

对于决策变量 x , 在第 n 个约束下的违反程度可以用如下公式计算:

$$cv_n(x) = \begin{cases} \max(0, g_n(x)), & n = 1, 2, \dots, l \\ \max(0, h_n(x) - \delta), & n = l+1, l+2, \dots, q \end{cases} \quad (3)$$

其中, δ 表示一个非常小的正值, 用于放松约束。

$$CV(x) = \sum_n cv_n(x) \quad (4)$$

对于一个解 $x = (x_1, x_2, \dots, x_D)$, 当且仅当 $CV(x) = 0$ 时, 称这个解为可行解。满足所有约束条件的解所组成的集合映射在目标空间所形成的图像称为可行域。由于 CMOPs 是由多个相互冲突的目标所组成的,

因此得到的可行解空间并不是单一的最优解,而是约束帕累托前沿。解决 CMOPs 的目标是找到一组可行 Pareto 最优解,形成带约束 Pareto 前沿(CPF) [2],实现良好的收敛和分布。此类问题要求同时优化多个冲突目标,同时满足一组约束,因此单一解无法实现目标之间的权衡。演化算法是一种基于种群的优化算法,具有强大的搜索能力,并能输出一组权衡解,适合求解各种复杂优化问题。因此研究人员倾向于使用演化算法求解 CMOPs,各种类型的约束多目标演化算法(CMOEAs)逐渐增多。

CMOEA 通常包括两个核心组件:多目标演化算法(MOEA)和约束处理技术(CHO)。MOEAs 在促进种群演化进程中发挥关键作用,根据其环境选择机制,可分为三类:基于指标的方法、基于分解的方法以及基于支配的方法。这些演化算法在无约束多目标问题中取得了优异结果,但缺乏处理约束的能力。如果直接将 MOEAs 应用于求解 CMOPs,很可能会得到一组不满足指定约束的解。为了处理约束,许多研究致力于设计 CHTs,这些技术可以在满足约束的解和不满足约束的解之间进行选择[3]。因此,它们通常嵌入到 MOEAs 的环境选择过程中,形成 CMOEAs 来求解 CMOPs。同时,为了评估设计的 CMOEAs 在求解问题方面的能力,近年来开发了大量基准测试问题集。

目前已有许多研究者设计出不同的 CMOEAs,但通病是无法有效平衡目标与约束。仅依赖罚函数的 CMOEAs 需精细参数调整[4],基于约束支配原则(CDP)的方法可能过度优先可行性,忽略目标信息[5]。近期有研究者研究探索带约束 Pareto 前沿(CPF)与无约束 Pareto 前沿(UPF)关系以指导演化[6]。UPF 提供优秀目标信息,如果采用不动态的方法去评估两者关系,其优秀信息利用可能会受限。双种群框架分别维护 CPF 和 UPF 种群,实现双向辅助,但通常存在权重分配不合理的情况,未能适应不同位置关系[7]。两阶段算法先探索景观后利用潜力区域,但该方法可能忽略自适应算子和参数调整,导致算力浪费[8]。

基于这些观察,本文提出自适应双种群演化算法(DPSCEA),利用 CPF-UPF 重叠关系情况来求解 CMOPs。DPSCEA 采用双种群: P_C 种群逼近 CPF, P_U 种群逼近 UPF。种群进化阶段分为学习阶段(使用 GA 和 DE 生成子代,分类关系)和演化阶段(基于关系自适应策略,如参数调整和多样性阈值)。从 P_C 种群转移非支配及精英个体至 P_U 种群,提升目标利用。主要贡献包括:

- (1) 新型自适应双种群框架,通过聚类分类 CPF-UPF 重叠大小情况,实现针对性演化策略;
- (2) 集成自适应 GA 和 DE 变体、参数调整及多样性监控,平衡探索与利用;
- (3) 增强转移机制,选择 P_C 种群非支配及单目标精英个体辅助 P_U 种群进化。

2. 提出算法

2.1. 算法框架

算法 1: DPSCEA 总体框架

输入: 种群大小 N 、最大评估次数 FES 、学习阶段评估次数占最大评估次数百分比
 输出: 最终帕累托前沿解集

- 1: 初始化 P_C 种群(逼近 CPF)和 P_U 种群(逼近 UPF), 每种群规模 N , 决策向量 $x \in R^D$
- 2: 计算所有个体的目标向量 $f(x)$ 和约束违反度 $CV(x) = \sum \max(0, g_i(x)) + \sum |h_j(x)|$
- 3: $inLearning \leftarrow true$, $cnt \leftarrow 0$, $t \leftarrow 2N$
- 4: while $t < FES_{max}$ do
- 5: if $inLearning$ then
- 6: 执行学习阶段(算法 2)
- 7: else
- 8: 执行演化阶段(算法 4)
- 9: end if
- 10: $t \leftarrow t + 2N$
- 11: end while
- 12: 从 P_C 种群提取所有非支配可行解作为最终 PF

算法 1 展示了算法的整体框架。本文提出的自适应双种群演化算法(DPSCEA)旨在利用 CPF 和 UPF 之间的重叠关系来求解 CMOPs, 常用的分类情况如图 1 所示。算法采用双种群框架: P_C 种群专注于逼近约束 Pareto 前沿, P_U 种群专注于逼近无约束 Pareto 前沿。演化过程分为两个阶段: 学习阶段和演化阶段。

2.2. 学习阶段

算法 2: 学习阶段算法

输入: 初始种群 P_C 种群和 P_U 种群, 学习阶段评估次数占最大评估次数百分比

输出: 学习阶段结束后的 P_C 种群和 P_U 种群和 flag (重叠类型)

```

1: while inLearning do
2:    $\epsilon_{\text{current}} = \left(1 - \frac{\text{Problem.FE}}{\text{maxLearnFE}}\right)^{\text{cp}}$  (其中  $\epsilon_0 = \max \text{CV of initial PC}$ ,  $\text{cp}=5$ )
3:   // GA + DE_current_to_rand_1 生成子代(各一半)
4:   for pop  $\in \{P_C, P_U\}$  do
5:     gaOff = GA_TournamentSelection(pop, Fitness, N/2)
6:     deOff = DE_current_to_rand_1(pop, N/2, F, CR)
7:     offspring  $\leftarrow$  [gaOff; deOff]
8:   end for
9:   // 环境选择
10:  [P_C, Fitness_P_C] = FirstStageEnvSelect([P_C; offspring], N,  $\epsilon_{\text{current}}$ )
11:  [P_U, Fitness_P_U] = FirstStageEnvSelect([P_U; offspring], N)
12:  // 自适应 F、CR
13:  if success_rate > 0.5 then
14:    F  $\leftarrow$  min(F + 0.05, 0.9), CR  $\leftarrow$  min(CR + 0.05, 0.9)
15:  else
16:    F  $\leftarrow$  max(F - 0.05, 0.4), CR  $\leftarrow$  max(CR - 0.05, 0.1)
17:  end if
18:  if cnt > T 和 (目标均值变化 <  $\epsilon$  and flag 稳定) or Problem.FE  $\geq$  maxLearnFE then
19:    inLearning  $\leftarrow$  false
20:  end if
21:  flag = ClusteringClassification (P_C, P_U)
22: end while

```

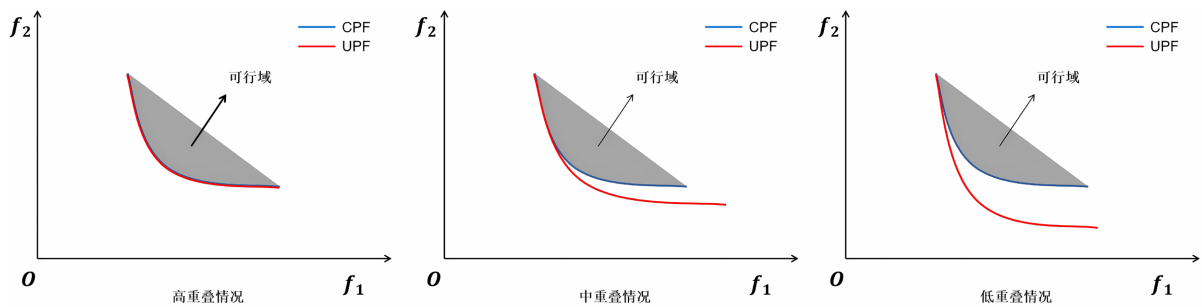


Figure 1. Classification of CPF-UPF relationship types

图 1. CPF-UPF 关系类型分类

算法 2 为学习阶段算法, 通过特定的学习策略演化两个种群, 以测量 CPF 和 UPF 之间的关系。 P_C 种群和 P_U 种群同时使用遗传算法(GA)和差分演化(DE)各生成子代的一半, 随后将父代种群和子代种群进行交叉合并, 在合并种群中进行环境选择, 基于目标函数值与约束违反程度计算综合适应度, 得到最优种群个体, 更新父代种群。并通过计算个体之间的两两欧式距离, 得出当前种群的多样性值。同时, 根据种群的多样性值自适应调整缩放因子 F 和交叉概率 CR, 优化种群在学习阶段的探索。当种群进化到满足

学习阶段终止条件时, P_C 种群和 P_U 种群基于可行性和支配关系确定 CPF-UPF 关系类型如图 1 所示(高重叠、中重叠或低重叠), 学习阶段结束, 进入演化阶段。

算法 3 确定 CPF-UPF 关系类型。通过非支配排序得出 P_C 种群的第一前沿非支配解, 若 P_C 种群的第一前沿非支配解的个数未超过 P_C 种群的 10%, 则计算 P_C 种群的第一前沿非支配个体之间两两欧式距离, 得出个体之间欧式距离的平均值; 若 P_C 种群的第一前沿非支配解的个数超过 P_C 种群的 10%, 通过计算 P_C 种群的第一前沿非支配个体之间两两欧式距离, 选出两两欧式距离最大的 10% 个体, 同时得出该种群大小 10% 个体欧式距离的平均值, 该距离的一半记为 r , 以两两欧式距离最大的 10% 个体为圆心, r 为半径得出圆的大小, 计算 P_U 种群中在圆中的个体数, 将在圆中个体数/ P_C 种群个体数, 得出重叠比例, 当重叠比例大于 0.8 时, 认定为高重叠情况; 当重叠比例小于 0.2 时, 认定为低重叠情况; 当重叠比例位于 0.2~0.8 之间, 认定为中等重叠情况。

算法 3: CPF-UPF 重叠关系分类

```

输入: 大小为 N 的  $P_C$  种群和  $P_U$  种群
输出: flag  $\in$  {1: high, 2: medium, 3: low}
1:  $P_{C\_nd}$   $\leftarrow$  非支配解 of  $P_C$  (CV 支配优先)
2:  $P_{U\_nd}$   $\leftarrow$  非支配解 of  $P_U$  (仅目标支配)
3: if  $P_{U\_nd}$  全可行 then flag  $\leftarrow$  1
4: else if  $P_{U\_nd}$  全不可行 then flag  $\leftarrow$  3
5: else
6:   combined  $\leftarrow$  [ $P_{C\_nd}$ .objs;  $P_{U\_nd}$ .objs], labels  $\leftarrow$  [1...; 2...]
7:   [idx]  $\leftarrow$  kmeans(combined, M)
8:   for each cluster k do
9:     overlap_k  $\leftarrow$  min(count_ $P_C$ , count_ $P_U$ ) / max(count_ $P_C$ , count_ $P_U$ )
10:  end for
11:  overlap  $\leftarrow$  mean(overlap_k)
12:  if overlap > 0.8 then flag  $\leftarrow$  1
13:  else if overlap < 0.2 then flag  $\leftarrow$  3
14:  else flag  $\leftarrow$  2
15: end if

```

2.3. 演化阶段

在演化阶段, 根据学习阶段所学习到的关系, 设计相应的演化策略来提高目标信息的利用效率, 更好地引导种群进化, 其核心逻辑是基于种群多样性和种群间重叠度, 动态调整策略参数和进化算子, 平衡算法的探索与开发能力。在演化阶段自适应调整策略参数 P , 其调整逻辑由当前种群的多样性值进行决定。 P_C 种群采用与学习阶段相同的固定混合策略生成子代, 不随种群关系变化, 保证进化稳定性。 P_U 种群的子代生成策略会根据种群间重叠度和当前多样性值进行自适应切换。

DPSCEA 算法在 LIRCMOP5 测试问题的运行如图 2 所示, 在学习阶段结束后, 得到 CPF 与 UPF 的重叠比例大于 0.8, 因此判断为高重叠情况, 在演化阶段采取高重叠情况的演化策略来引导种群进化。两个种群搜索空间位置关系高度重合, 避免冗余搜索, 采取强化优质基因利用, 以最优个体为引导, 提升局部求精效率。在演化阶段结束后, 对种群混合后进行环境选择, 可得到完整的最终帕累托前沿解集。

DPSCEA 算法在 LIRCMOP9 测试问题的运行如图 3 所示, 在学习阶段结束后, 得到 CPF 与 UPF 的重叠比例约为 0.5, 因此判断为中等重叠情况, 在演化阶段采取中等重叠情况的演化策略来引导种群进化。中等重叠场景中兼顾探索广度与利用深度, 采用多策略融合模式, 同时平衡探索与协同, 积极强化 P_C 种群优质基因的迁移利用, 同时平衡探索与协同。在演化阶段结束后, 对种群混合后进行环境选择, 可得到完整的最终帕累托前沿解集。

算法 4: 演化阶段

输入: 大小为 N 的 P_C 种群和 P_U 种群, $flag$
 输出: 最终帕累托前沿解集

- 1: // 增强转移机制
- 2: $nd_P_C \leftarrow P_C$ 中非支配可行解($FrontNo==1$)
- 3: $elite_P_C \leftarrow$ 每个目标上排序前 10% 的个体
- 4: $transfer_set \leftarrow nd_P_C \cup elite_P_C$ (去重)
- 5: $proportion \leftarrow \{0.05 \text{ if } flag=1, 0.10 \text{ if } flag=2, 0.15 \text{ if } flag=3\}$
- 6: 随机选取 $proportion \times N$ 个 $transfer_set$ 个体加入 P_U (替换最差个体)
- 7: // P_C 种群: 固定混合策略
- 8: $P_{C_off} \leftarrow GA_TournamentSelection(P_C, N/2) + DE_current_to_rand_1(P_C, N/2, F, CR)$
- 9: // P_U 种群: 根据 $flag$ 概率混合策略
- 10: switch $flag$
- 11: case 1 (high):
- 12: with prob 0.6: $GA + DE_transfer(P_C \rightarrow P_U)$
- 13: with prob 0.2: $DE_current_to_other_pbest_1(P_U, P_C_pbest, p, F, CR)$
- 14: else: DE_best_1 或 DE_rand_1 (依多样性)
- 15: case 2 (medium):
- 16: with prob 0.5: $GA + DE_transfer + DE_current_to_other_pbest_1$ (各 1/3)
- 17: with prob 0.25: $GA + DE_transfer$
- 18: else: $DE_current_to_other_pbest_1$
- 19: case 3 (low):
- 20: with prob 0.7: $DE_current_to_other_pbest_1$
- 21: with prob 0.15: $GA + DE_transfer$
- 22: else: DE_best_1 或 DE_rand_1
- 23: end switch
- 24: // 环境选择
- 25: $P_C \leftarrow SecondStageEnvSelect([P_C; P_{C_off}; P_{U_off}], N)$
- 26: $P_U \leftarrow SecondStageEnvSelect([P_U; P_{U_off}; P_{C_off}], N)$

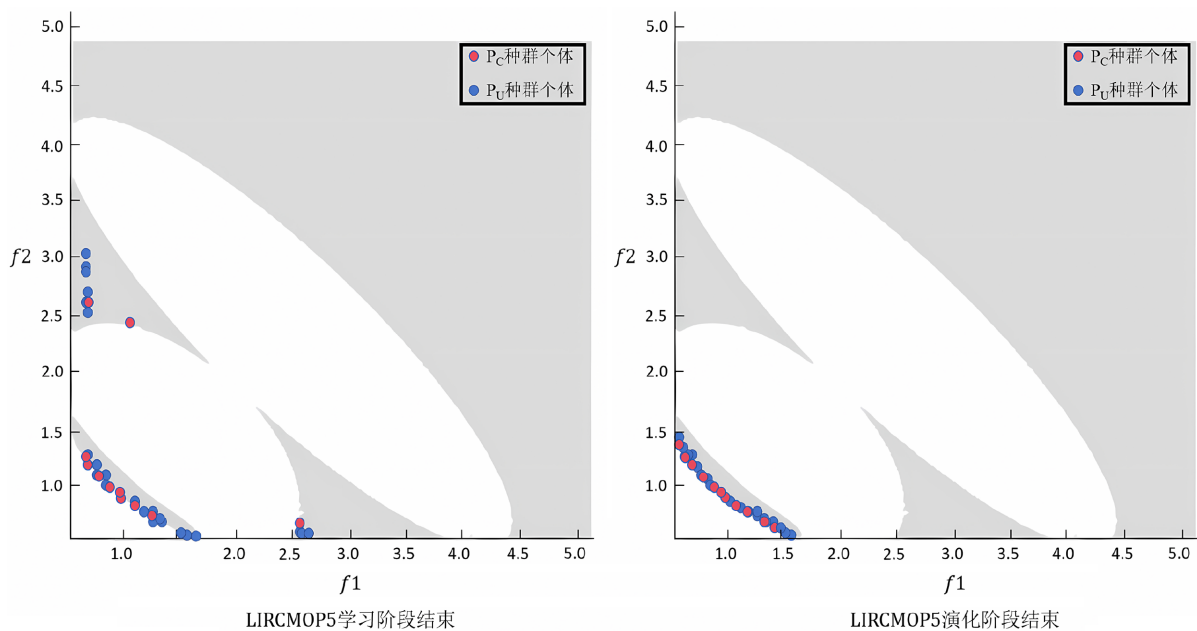


Figure 2. Performance of DPSCEA on the LIRC MOP5 test problem

图 2. DPSCEA 在 LIRC MOP5 测试问题的运行情况

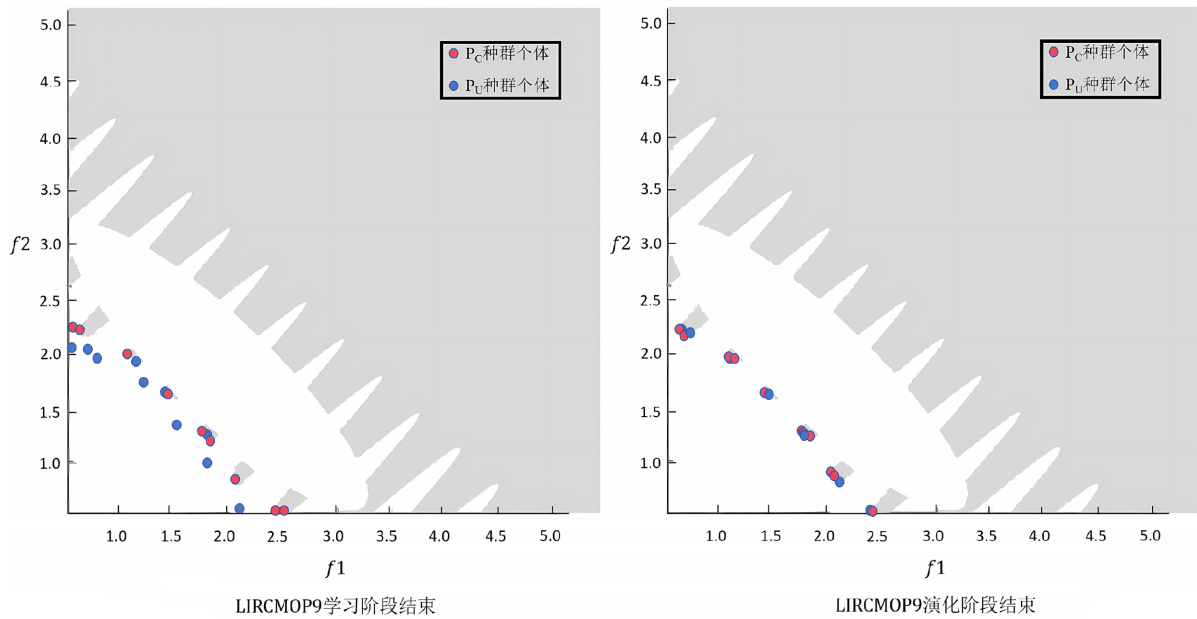


Figure 3. Performance of DPSCEA on the LIRC MOP9 test problem
图 3. DPSCEA 在 LIRC MOP9 测试问题的运行情况

DPSCEA 算法在 LIRC MOP3 测试问题的运行如图 4 所示，在学习阶段结束后，得到 CPF 与 UPF 的重叠比例为 0，因此判断为低重叠情况，在演化阶段采取低重叠情况的演化策略来引导种群进化，低重叠场景：两种群搜索空间差异显著，优先利用 P_C 种群优质基因，深度挖掘 P_C 种群的优化潜力，结合多样性状态动态调整搜索方向，且进行局部求精。在演化阶段结束后，对种群混合后进行环境选择，可得到完整的最终帕累托前沿解集。

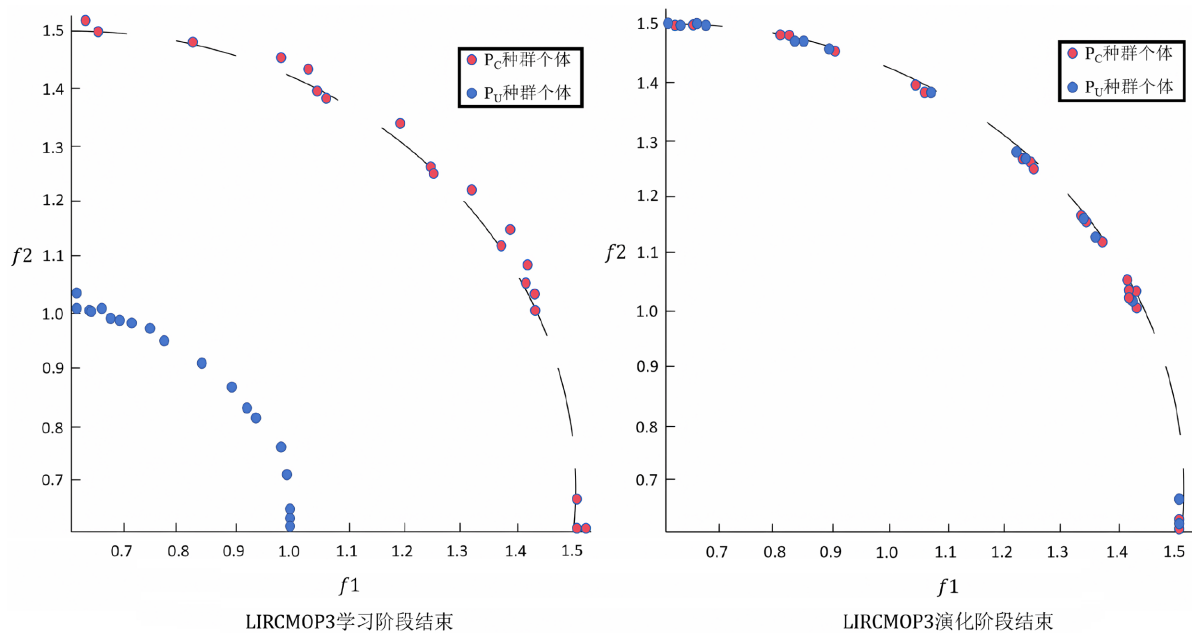


Figure 4. Performance of DPSCEA on the LIRC MOP3 test problem
图 4. DPSCEA 在 LIRC MOP3 测试问题的运行情况

2.4. 计算复杂度

DPSCEA 的计算过程由学习阶段与进化阶段的迭代更新组成, 其计算复杂度与种群规模 N 、目标函数数量 M 、决策变量维度 D 及总迭代次数 G 密切相关。在算法的迭代过程中, 种群初始化、子代生成、环境选择、跨种群精英迁移等操作的计算复杂度分别为 $O(N^2 \cdot M)$ 、 $O(N \cdot (D + \log N))$ 、 $O(N^2 \cdot M)$ 、 $O(N^2 \cdot M)$ 。根据种群规模、目标数量及迭代次数的取值, 此迭代一次的计算复杂度为:

$$T_{iter} = O(N^2 \cdot M) + O(N \cdot (D + \log N))a \quad (5)$$

对于总迭代次数为 G 的优化问题, 算法的总计算复杂度为:

$$T_{total} = O(N^2 \cdot M) + O(G \cdot (N^2 \cdot M + N \cdot (D + \log N))) \quad (6)$$

在多目标优化场景中(M 为常数、 $D \ll N^2$), 可简化为:

$$T_{total} = O(G \cdot N^2 \cdot M) \quad (7)$$

3. 实验分析

3.1. 算法对比与测试问题

为了更好地评估本文算法的性能, 选取了目前最为先进的约束多目标进化算法作为对比: APSEA [9]、BiCo [10]、CCMO [11]、CTAEA [12]、URCMO [7]、ToP [13]。测试平台为 PlatEMOv4.11 [14]。测试问题集选取 LIR-CMOP 测试集[15]和 ZXH-CF 测试集[16]。LIR-CMOP 测试集共有 14 个测试问题, 其中 LIR-CMOP1-LIR-CMOP12 为双目标问题, LIR-CMOP13 和 LIR-CMOP14 为三目标问题。ZXH-CF 测试集共有 16 个测试问题, 其中 ZXH-CF1-ZXH-CF12 为三目标问题, ZXH-CF13-ZXH-CF16 为双目标问题。为确保公平, 对于每个测试问题, 将种群规模 N 设置为 100, 最大评估次数为 100,000 次, 每个算法进行 30 次独立运算, 取运算后的平均值进行比较。DPSCEA 超参数汇总如表 1 所示。

Table 1. Summary of DPSCEA hyperparameters

表 1. DPSCEA 超参数汇总

参数	值	描述
初始 F	0.6	DE 缩放因子(学习阶段自适应)
初始 CR	0.2	DE 交叉概率(学习阶段自适应)
cp	5	ε 衰减指数
T	3	flag 稳定性窗口
初始 p	0.2	DE_current_to_other_pbest 比例
多样性阈值	$0.8/0.2 \times \text{initial_diversity}$	p 自适应调整阈值
转移比例	0.05 (high)/0.10 (medium)/0.15 (low)	精英迁移比例

3.2. 性能指标

评价不同算法性能的指标采用反向世代距离(IGD) [17]和超体积(HV) [18]。

反向世代距离(IGD)的定义:

$$\text{IGD}(P, Q) = \frac{\sum_{X \in P} \text{dis}(X, Q)}{|P|} \quad (8)$$

其中,集合 Q 为算法求得的非支配解集, P 为真实的 Parato 最优解集, $|P|$ 为解集 P 中解的个数, $\text{dis}(X, Q)$ 表示 P 中的解 X 到解集 Q 的欧式距离的最小值。IGD 值越小, 表明算法的综合性能越好。

超体积(HV)的定义:

$$\text{HV} = \varrho \left(\bigcup_{i=1}^{|S|} v_i \right) \quad (9)$$

其中, ϱ 为勒贝格测度, $|S|$ 表示算法求得的非支配解集的数量, v_i 表示解集中第 i 个非支配解与参考点构成的超体。HV 越大, 表明算法的综合性能越好。

3.3. 实验结果与分析

本文算法在 LIR-CMOP 测试集和 ZXH-CF 测试集上与对比算法进行性能评估。实验结果均经过显著性水平为 0.05 的 Wilcoxon 秩和检验, 实验结果如表格所示。表格中对每个基准函数的最优值进行加粗显示, 并在底部列出了 Wilcoxon 秩和检验的统计结果, 符号“+”、“-”和“=”分别表示算法显著优于、劣于或统计学上与本文算法持平。

3.3.1. LIR-CMOP 测试集结果分析

针对 LIR-CMOP 测试集, 本文将 DPSCEA 与六种主流算法进行比较, 其 IGD 和 HV 指标的实验结果如表 2, 表 3 所示。

Table 2. Comparison of IGD index of DPSCEA and other algorithms on the LIR-CMOP test set

表 2. DPSCEA 与其他算法在 LIR-CMOP 测试集的 IGD 指标比较

Problem	APSEA	BiCo	CTAEA	CCMO	ToP	URCMO	DPSCEA
LIRCMOP1	2.9376e-1 (3.05e-2) -	2.4374e-1 (2.82e-2) =	3.1941e-1 (8.19e-2) -	2.7602e-1 (6.33e-2) =	3.2474e-1 (1.72e-2) -	1.6969e-1 (6.35e-2) +	2.3080e-1 (1.01e-1)
LIRCMOP2	2.5408e-1 (2.95e-2) -	2.0622e-1 (2.41e-2) =	2.7007e-1 (7.27e-2) -	2.4577e-1 (4.47e-2) -	2.8094e-1 (1.47e-2) -	8.7651e-2 (7.11e-2) +	2.1805e-1 (3.98e-2)
LIRCMOP3	2.9805e-1 (4.70e-2) -	2.4437e-1 (2.83e-2) +	3.4166e-1 (4.16e-2) -	3.0361e-1 (4.73e-2) -	3.3999e-1 (8.13e-3) -	1.2626e-1 (6.28e-2) +	2.5816e-1 (9.63e-2)
LIRCMOP4	2.8779e-1 (3.97e-2) -	2.3333e-1 (2.18e-2) +	3.0578e-1 (7.65e-2) -	2.7124e-1 (4.43e-2) =	3.1468e-1 (1.04e-2) -	1.4247e-1 (7.63e-2) +	2.4760e-1 (8.21e-2)
LIRCMOP5	1.1872e+0 (1.65e-1) -	1.2239e+0 (5.88e-3) -	1.2027e+0 (1.63e-1) -	3.0959e-1 (6.35e-2) -	1.1244e+0 (2.53e-1) -	6.5381e-1 (5.23e-1) -	2.8007e-2 (1.22e-2)
LIRCMOP6	1.1471e+0 (3.68e-1) -	1.3454e+0 (1.86e-4) -	1.3462e+0 (8.19e-4) -	3.1751e-1 (9.17e-2) -	1.2930e+0 (2.05e-1) -	4.2106e-1 (4.52e-1) -	6.6647e-2 (6.63e-2)
LIRCMOP7	1.0906e-1 (3.88e-2) -	9.1660e-1 (7.78e-1) -	4.8845e-1 (6.25e-1) -	1.1396e-1 (2.88e-2) -	1.4287e+0 (5.78e-1) -	4.8293e-2 (4.81e-2) -	1.1704e-2 (1.12e-2)
LIRCMOP8	1.8102e-1 (6.33e-2) -	1.3551e+0 (6.02e-1) -	7.8220e-1 (6.58e-1) -	2.0022e-1 (6.53e-2) -	1.5587e+0 (3.84e-1) -	4.6723e-2 (4.90e-2) -	8.2798e-3 (3.24e-4)
LIRCMOP9	8.5093e-1 (1.65e-1) -	9.9812e-1 (1.20e-1) -	6.0004e-1 (1.49e-1) -	5.6786e-1 (1.12e-1) -	5.7674e-1 (1.29e-1) -	4.3682e-1 (9.10e-2) -	1.9716e-1 (6.09e-2)
LIRCMOP10	8.4532e-1 (1.56e-1) -	9.9163e-1 (5.07e-2) -	4.9910e-1 (1.39e-1) -	2.2146e-1 (6.91e-2) -	4.4861e-1 (9.32e-2) -	2.5900e-1 (1.27e-1) -	5.2741e-2 (5.42e-2)

续表

LIRCMOP11	6.4376e-1 (1.61e-1) -	8.4795e-1 (1.49e-1) -	2.5452e-1 (8.90e-2) -	1.1940e-1 (6.23e-2) -	4.1064e-1 (1.02e-1) -	1.8494e-1 (1.41e-1) -	2.9957e-2 (3.72e-2)
LIRCMOP12	5.0813e-1 (1.71e-1) -	7.2852e-1 (2.41e-1) -	3.7804e-1 (1.62e-1) -	2.7325e-1 (1.43e-1) -	3.0855e-1 (8.30e-2) -	2.0099e-1 (7.65e-2) -	8.9617e-2 (5.03e-2)
LIRCMOP13	1.3158e+0 (1.91e-3) -	1.3189e+0 (2.63e-3) -	1.0923e-1 (2.04e-3) -	9.3916e-2 (1.09e-3) =	1.3341e+0 (5.77e-2) -	1.0416e-1 (2.58e-3) -	9.3973e-2 (1.47e-3)
LIRCMOP14	1.2725e+0 (2.08e-3) -	1.2752e+0 (1.75e-3) -	1.1118e-1 (1.77e-3) -	9.5954e-2 (1.02e-3) +	1.3029e+0 (4.87e-3) -	9.7883e-2 (9.51e-4) -	9.6438e-2 (1.04e-3)
+/-/=	0/14/0	2/10/2	0/14/0	1/10/3	0/14/0	4/10/0	

Table 3. Comparison of HV index of DPSCEA and other algorithms on the LIR-CMOP test set
表 3. DPSCEA 与其他算法在 LIR-CMOP 测试集的 HV 指标比较

Problem	APSEA	BiCo	CTAEA	CCMO	ToP	URCMO	DPSCEA
LIRCMOP1	1.1535e-1 (1.02e-2) -	1.2877e-1 (9.32e-3) =	1.1576e-1 (2.12e-2) =	1.2097e-1 (1.92e-2) =	1.0752e-1 (8.71e-3) -	1.5936e-1 (2.64e-2) +	1.5169e-1 (2.74e-2)
LIRCMOP2	2.3467e-1 (1.49e-2) -	2.4976e-1 (1.37e-2) =	2.3173e-1 (3.61e-2) =	2.3769e-1 (2.38e-2) =	2.1947e-1 (1.14e-2) -	3.1184e-1 (3.95e-2) +	2.8039e-1 (3.07e-2)
LIRCMOP3	1.0448e-1 (1.29e-2) -	1.1708e-1 (9.42e-3) =	9.6015e-2 (9.68e-3) -	1.0460e-1 (1.57e-2) -	9.1002e-2 (5.50e-3) -	1.5837e-1 (2.40e-2) +	1.2435e-1 (2.17e-2)
LIRCMOP4	1.9467e-1 (1.93e-2) =	2.1423e-1 (1.08e-2) +	1.8859e-1 (3.01e-2) -	2.0158e-1 (1.82e-2) =	1.8232e-1 (9.08e-3) -	2.5369e-1 (3.41e-2) +	2.2576e-1 (2.73e-2)
LIRCMOP5	5.1167e-3 (2.80e-2) -	0.0000e+0 (0.00e+0) -	4.5998e-3 (2.52e-2) -	1.5167e-1 (2.39e-2) -	1.8103e-2 (5.72e-2) -	1.2141e-1 (1.22e-1) -	2.7916e-1 (7.33e-3)
LIRCMOP6	2.0397e-2 (3.83e-2) -	0.0000e+0 (0.00e+0) -	0.0000e+0 (0.00e+0) -	1.1249e-1 (1.47e-2) -	3.7907e-3 (1.44e-2) -	1.1163e-1 (6.32e-2) -	1.7680e-1 (1.80e-2)
LIRCMOP7	2.5187e-1 (1.19e-2) -	1.1904e-1 (1.21e-1) -	1.8356e-1 (1.01e-1) -	2.4999e-1 (8.71e-3) -	3.9957e-2 (9.12e-2) -	2.7633e-1 (1.87e-2) -	2.9183e-1 (6.08e-3)
LIRCMOP8	2.3721e-1 (1.30e-2) -	4.9455e-2 (9.13e-2) -	1.3637e-1 (1.01e-1) -	2.3238e-1 (1.31e-2) -	2.0446e-2 (6.26e-2) -	2.7683e-1 (1.91e-2) -	2.9380e-1 (2.03e-4)
LIRCMOP9	1.8670e-1 (8.72e-2) -	1.1548e-1 (5.29e-2) -	2.9841e-1 (7.33e-2) -	3.2598e-1 (4.67e-2) -	3.0296e-1 (7.89e-2) -	3.9180e-1 (4.55e-2) -	5.0947e-1 (1.28e-2)
LIRCMOP10	1.1985e-1 (1.10e-1) -	6.0721e-2 (1.31e-2) -	4.1200e-1 (1.03e-1) -	5.7895e-1 (4.34e-2) -	4.4968e-1 (9.93e-2) -	5.6191e-1 (1.04e-1) -	6.8410e-1 (2.35e-2)
LIRCMOP11	2.9949e-1 (1.09e-1) -	2.0110e-1 (5.89e-2) -	5.8063e-1 (5.33e-2) -	6.3288e-1 (4.00e-2) -	4.1403e-1 (7.80e-2) -	5.6916e-1 (1.06e-1) -	6.7813e-1 (2.15e-2)
LIRCMOP12	4.1126e-1 (7.95e-2) -	3.0635e-1 (1.07e-1) -	4.3812e-1 (6.52e-2) -	5.0092e-1 (5.13e-2) -	4.6466e-1 (4.39e-2) -	5.2120e-1 (3.73e-2) -	5.7789e-1 (2.70e-2)
LIRCMOP13	1.5705e-4 (1.48e-4) -	1.0804e-4 (1.13e-4) -	5.4508e-1 (1.73e-3) -	5.5382e-1 (1.69e-3) +	1.8501e-3 (9.92e-3) -	5.3217e-1 (2.95e-3) -	5.5057e-1 (2.09e-3)
LIRCMOP14	6.0922e-4 (2.40e-4) -	4.2525e-4 (3.37e-4) -	5.4581e-1 (9.60e-4) -	5.5414e-1 (1.38e-3) +	1.4687e-4 (2.27e-4) -	5.4932e-1 (1.74e-3) -	5.5173e-1 (1.24e-3)
+/-/=	0/13/1	1/10/3	0/12/2	2/9/3	0/14/0	4/10/0	

3.3.2. ZXH_CF 测试集结果分析

针对 LIR-CMOP 测试集, 本文将 DPSCEA 与六种主流算法进行比较, 其 IGD 和 HV 指标的实验结果如表 4 和表 5 所示。

Table 4. Comparison of IGD index of DPSCEA and other algorithms on the ZXH_CF test set
表 4. DPSCEA 与其他算法在 ZXH_CF 测试集的 IGD 指标比较

Problem	APSEA	BiCo	CTAEA	CCMO	ToP	URCMO	DPSCEA
ZXH_CF1	4.7106e-2 (1.13e-3) +	4.7074e-2 (1.26e-3) +	4.8881e-2 (1.88e-3) +	4.6786e-2 (9.59e-4) +	9.3376e-2 (1.28e-2) -	5.4501e-2 (2.23e-3) -	4.9351e-2 (1.13e-3)
ZXH_CF2	1.0748e-1 (1.75e-1) -	1.5061e-1 (2.11e-1) -	1.1452e-1 (1.41e-1) -	1.7137e-1 (2.89e-1) -	2.4289e-1 (1.32e-1) -	6.0067e-2 (9.96e-4) -	5.8429e-2 (1.97e-3)
ZXH_CF3	6.3613e-2 (1.29e-3) +	6.2378e-2 (1.31e-3) +	7.4706e-2 (2.56e-3) -	6.2392e-2 (1.41e-3) +	2.2986e-1 (8.22e-2) -	7.6983e-2 (2.16e-3) -	6.6485e-2 (1.94e-3)
ZXH_CF4	1.6564e-1 (1.27e-1) =	1.1404e-1 (8.14e-2) =	1.1644e-1 (7.87e-2) -	1.7737e-1 (1.96e-1) =	1.0733e+0 (2.70e-1) -	1.2299e-1 (9.93e-2) -	7.7671e-2 (6.86e-2)
ZXH_CF5	6.2186e-2 (7.02e-2) -	4.8683e-2 (4.13e-2) -	8.9430e-2 (1.63e-1) -	1.0425e-1 (2.05e-1) =	1.5223e-1 (1.01e-1) -	3.5775e-2 (1.38e-3) -	3.2305e-2 (1.30e-3)
ZXH_CF6	3.1122e-2 (6.38e-4) -	3.1443e-2 (6.21e-4) -	3.6821e-2 (1.59e-3) -	3.0721e-2 (5.52e-4) =	5.8398e-2 (6.14e-3) -	3.3558e-2 (1.00e-3) -	3.0370e-2 (6.08e-4)
ZXH_CF7	1.3780e-1 (1.14e-1) -	1.3271e-1 (7.86e-2) -	9.0138e-2 (7.91e-2) -	1.0365e-1 (1.30e-1) =	5.2364e-1 (2.20e-1) -	8.0073e-2 (6.51e-2) -	4.4608e-2 (4.98e-2)
ZXH_CF8	3.2654e-2 (6.82e-4) =	3.2720e-2 (5.42e-4) =	5.3453e-2 (4.36e-3) -	3.1976e-2 (6.57e-4) +	1.7186e-1 (4.41e-2) -	4.1904e-2 (1.33e-3) -	3.2946e-2 (6.86e-4)
ZXH_CF9	2.6654e-2 (4.50e-4) =	2.6424e-2 (5.79e-4) =	6.6236e-2 (1.08e-2) -	2.6643e-2 (5.10e-4) =	4.9453e-1 (1.41e-1) -	2.8581e-2 (9.39e-4) -	2.6523e-2 (4.90e-4)
ZXH_CF10	1.1297e-1 (1.12e-1) -	9.7642e-2 (9.88e-2) =	1.0681e-1 (6.77e-2) -	1.1856e-1 (1.71e-1) -	1.2987e+0 (3.36e-1) -	9.8088e-2 (1.34e-1) -	4.5718e-2 (3.70e-2)
ZXH_CF11	2.7964e-2 (4.08e-4) =	2.8235e-2 (6.14e-4) =	3.5390e-2 (1.05e-3) -	2.8048e-2 (5.05e-4) =	4.3690e-1 (2.00e-1) -	2.8264e-2 (4.50e-4) =	2.8076e-2 (4.38e-4)
ZXH_CF12	9.4962e-2 (2.24e-1) =	3.2035e-2 (2.26e-2) =	4.5366e-2 (5.86e-2) -	3.8504e-2 (6.29e-2) -	4.6553e-1 (8.86e-2) -	2.6011e-2 (8.97e-4) -	2.4035e-2 (4.60e-4)
ZXH_CF13	1.3438e-1 (1.38e-1) =	3.3115e-2 (7.02e-2) +	1.4953e-1 (1.84e-1) -	1.7646e-1 (2.30e-1) =	1.1159e+0 (3.88e-1) -	7.8558e-2 (1.22e-1) -	5.3473e-2 (1.37e-1)
ZXH_CF14	2.5179e-3 (1.12e-4) -	2.3678e-3 (5.13e-5) +	2.8236e-3 (9.18e-5) -	2.4415e-3 (6.94e-5) =	9.4626e-2 (6.02e-2) -	2.9953e-3 (8.57e-5) -	2.4576e-3 (4.12e-5)
ZXH_CF15	1.0317e-2 (3.78e-2) -	6.3469e-2 (1.72e-1) =	1.4732e-2 (4.60e-2) -	1.8582e-2 (7.08e-2) -	1.3441e-1 (1.07e-1) -	2.9513e-3 (5.95e-5) -	2.8825e-3 (4.82e-5)
ZXH_CF16	2.6429e-3 (4.03e-5) +	2.6674e-3 (4.64e-5) =	1.4597e-2 (5.26e-3) -	2.6432e-3 (4.53e-5) +	2.3447e-2 (4.50e-2) -	2.6918e-3 (5.44e-5) =	2.6748e-3 (5.09e-5)
+/-/=	3/7/6	4/4/8	1/15/0	4/4/8	0/16/0	0/14/2	

Table 5. Comparison of HV index of DPSCEA and other algorithms on the ZXH_CF test set
表 5. DPSCEA 与其他算法在 ZXH_CF 测试集的 HV 指标比较

Problem	APSEA	BiCo	CTAEA	CCMO	ToP	URCMO	DPSCEA
ZXH_CF1	8.3103e-1 (1.43e-3) +	8.3117e-1 (1.66e-3) +	8.3421e-1 (1.74e-3) +	8.3143e-1 (1.12e-3) +	7.6610e-1 (1.78e-2) -	8.1976e-1 (3.18e-3) -	8.2666e-1 (1.56e-3)
ZXH_CF2	4.9390e-1 (1.32e-1) -	4.5531e-1 (1.97e-1) -	4.8194e-1 (1.50e-1) -	4.6070e-1 (1.93e-1) -	2.9341e-1 (8.22e-2) -	5.3785e-1 (1.75e-3) -	5.4388e-1 (3.38e-3)
ZXH_CF3	5.1032e-1 (2.53e-3) +	5.1385e-1 (2.73e-3) +	5.1238e-1 (2.23e-3) +	5.1217e-1 (2.65e-3) +	2.5686e-1 (6.45e-2) -	4.7598e-1 (3.79e-3) -	5.0002e-1 (4.06e-3)
ZXH_CF4	2.7759e-1 (1.31e-1) =	3.3286e-1 (1.06e-1) =	3.4202e-1 (1.04e-1) -	2.9285e-1 (1.42e-1) =	1.2800e-3 (6.53e-3) -	3.2302e-1 (1.19e-1) -	3.8438e-1 (8.18e-2)
ZXH_CF5	2.7199e-1 (7.18e-2) =	2.8470e-1 (5.00e-2) =	2.6562e-1 (6.60e-2) -	2.5850e-1 (9.50e-2) =	1.6942e-1 (7.08e-2) -	2.9818e-1 (2.45e-3) -	3.0480e-1 (2.97e-3)
ZXH_CF6	2.2453e-1 (9.39e-4) =	2.2486e-1 (8.88e-4) =	2.2345e-1 (1.29e-3) -	2.2524e-1 (1.19e-3) =	1.8909e-1 (8.08e-3) -	2.1938e-1 (1.46e-3) -	2.2480e-1 (1.11e-3)
ZXH_CF7	2.2675e-1 (1.30e-1) -	2.2006e-1 (1.00e-1) -	2.7271e-1 (1.08e-1) -	2.7807e-1 (1.27e-1) =	2.1011e-2 (4.37e-2) -	2.8587e-1 (8.88e-2) -	3.3775e-1 (6.89e-2)
ZXH_CF8	2.3431e-1 (1.09e-3) +	2.3408e-1 (1.22e-3) +	2.1402e-1 (4.62e-3) -	2.3563e-1 (1.36e-3) +	1.0286e-1 (2.75e-2) -	2.2095e-1 (1.93e-3) -	2.3281e-1 (1.20e-3)
ZXH_CF9	2.7016e-1 (1.56e-3) +	2.7145e-1 (1.33e-3) +	2.3827e-1 (1.02e-2) -	2.7130e-1 (1.31e-3) +	9.2516e-2 (2.84e-2) -	2.5993e-1 (2.81e-3) -	2.6848e-1 (1.42e-3)
ZXH_CF10	2.1943e-1 (1.32e-1) -	2.3149e-1 (1.22e-1) =	2.1867e-1 (9.86e-2) -	2.4081e-1 (1.32e-1) =	1.8756e-6 (8.60e-6) -	2.4831e-1 (1.20e-1) -	3.0575e-1 (7.70e-2)
ZXH_CF11	4.2623e-1 (2.49e-3) +	4.2672e-1 (1.80e-3) +	3.8863e-1 (5.58e-3) -	4.2669e-1 (1.56e-3) +	2.2432e-1 (7.04e-2) -	4.2235e-1 (2.14e-3) -	4.2413e-1 (2.92e-3)
ZXH_CF12	6.7460e-1 (1.57e-1) =	7.3831e-1 (4.59e-2) =	7.0768e-1 (1.18e-1) -	7.2923e-1 (1.10e-1) -	3.2611e-1 (8.35e-2) -	7.4821e-1 (1.60e-3) -	7.5395e-1 (1.78e-3)
ZXH_CF13	1.9141e-1 (1.19e-1) -	2.8669e-1 (7.91e-2) +	1.9683e-1 (1.37e-1) -	1.9450e-1 (1.39e-1) =	2.6869e-3 (1.37e-2) -	2.4531e-1 (1.11e-1) -	2.8028e-1 (9.52e-2)
ZXH_CF14	5.4282e-1 (2.41e-4) -	5.4321e-1 (1.78e-4) -	5.4233e-1 (2.83e-4) -	5.4307e-1 (2.09e-4) -	4.1423e-1 (4.79e-2) -	5.4181e-1 (2.33e-4) -	5.4332e-1 (7.57e-5)
ZXH_CF15	6.5239e-1 (4.59e-2) =	6.0444e-1 (1.57e-1) -	6.4991e-1 (5.23e-2) -	6.4420e-1 (7.55e-2) -	4.8574e-1 (1.02e-1) -	6.6148e-1 (1.72e-4) -	6.6185e-1 (9.42e-5)
ZXH_CF16	7.7818e-1 (1.02e-4) +	7.7824e-1 (9.01e-5) +	7.6308e-1 (7.33e-3) -	7.7820e-1 (8.07e-5) +	7.6936e-1 (1.46e-2) -	7.7790e-1 (1.49e-4) -	7.7813e-1 (6.50e-5)
+/-/=	6/5/5	7/4/5	2/14/0	6/4/6	0/16/0	0/16/0	

综合 LIRC MOP 和 ZXH-CF 两个测试集的实验结果, DPSCEA 算法在约束多目标优化问题上表现出显著的综合优势。在 IGD 指标方面, DPSCEA 在大多数测试问题中取得了最优或次优值, 例如在 LIRC MOP9-LIRC MOP14 和 ZXH_CF5、ZXH_CF7、ZXH_CF10、ZXH_CF12、ZXH_CF13 等高复杂度问

题上,其 IGD 值大幅低于 APSEA、BiCo、CTAEA、CCMO、ToP 和 URCMO 等对比算法。这表明 DPSCEA 在处理高维约束、多模态问题和非线性 Pareto 前沿时,具有更强的收敛性和鲁棒性。在相对简单的 LIRCMOP1-LIRCMOP4 和部分 ZXH-CF 低约束问题上,各算法性能差距较小,DPSCEA 仍能稳定保持与最优算法相当或更优的结果,展示出其良好的适应性。在 HV 指标上,DPSCEA 同样表现出色,在绝大多数问题中获得了最大的超体积值,特别是在 ZXH_CF4、ZXH_CF7、ZXH_CF12、ZXH_CF14、ZXH_CF15 等高约束复杂问题上,其 HV 值显著高于对比算法,解集覆盖范围更广、多样性更强。这验证了 DPSCEA 在严格约束下有效维持种群多样性,避免前沿收缩。

在 30 个测试问题中,DPSCEA 对 ToP 和 URCMO 实现了全面显著优胜,对 CTAEA 胜出比例超过 90%,对其他算法也表现出明显优势。其优异性能源于动态参数自适应与约束增强机制的整合,提升了约束处理效率和全局搜索能力。实验结果证明 DPSCEA 在约束多目标进化优化领域的可行性和先进性。

3.3.3. 种群分类性能分析

为了进一步验证所提出的重叠关系分类和自适应策略的有效性,本文设计了两种算法变体:DPSCEA-H, DPSCEA-L。DPSCEA-H 固定为高重叠策略的变体,DPSCEA-L 固定为低重叠策略的变体。如表 6 所示,DPSCEA 在性能指标上显著优于其算法变体,这充分说明了本文所提出的重叠关系分类和自适应策略的有效性。

Table 6. Comparison of DPSCEA and its variants on the LIRCMOP and ZXH_CF test sets

表 6. DPSCEA 与其变体在 LIRCMOP 和 ZXH_CF 测试集的指标比较

算法	DPSCEA-H	DPSCEA-L	DPSCEA
LIRCMOP (IGD) (+/-/=)	2/9/3	1/8/5	—
LIRCMOP(HV) (+/-/=)	3/7/4	2/8/5	—
ZXH-CF (IGD) (+/-/=)	3/11/2	2/9/5	—
ZXH-CF (HV) (+/-/=)	2/11/3	2/9/4	—

4. 总结

本文围绕解决约束多目标优化问题(CMOPs),提出了一种新的自适应双种群演化算法(DPSCEA),DPSCEA 通过动态分类 CPF 与 UPF 的重叠关系,实现自适应演化策略和参数调整。该算法采用双种群框架,在学习阶段利用 GA 和 DE 混合生成子代,评估种群多样性和关系类型;在演化阶段根据高/中/低重叠场景自适应进行策略切换,提升目标信息利用效率,并通过精英转移机制引导种群演化。

实验在 LIR-CMOP 和 ZXH-CF 测试集上进行,与六种主流算法比较,结果表明 DPSCEA 在 IGD 和 HV 指标上整体领先,尤其在高维约束和多模态问题中表现出更强的收敛性、多样性和鲁棒性。Wilcoxon 秩和检验确认其显著优势,胜出比例高达 90%。这验证了自适应框架在平衡约束与目标方面的有效性,为复杂优化问题提供了新思路。

未来工作可尝试将 DPSCEA 算法应用于超多目标优化问题或动态约束场景,并研究其他高级演化算子的策略,更进一步提升该算法在实际应用中的性能。

参考文献

- [1] 俞凯乐,廖家俊,毛嘉莉,等.多约束条件下钢铁物流车货匹配的多目标优化[J].计算机应用,2025,45(8):2477-

- 2483.
- [2] Liang, J., Ban, X., Yu, K., Qu, B., Qiao, K., Yue, C., *et al.* (2023) A Survey on Evolutionary Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **27**, 201-221. <https://doi.org/10.1109/tevc.2022.3155533>
 - [3] Kramer, O. (2010) A Review of Constraint-Handling Techniques for Evolution Strategies. *Applied Computational Intelligence and Soft Computing*, **2010**, Article ID: 185063. <https://doi.org/10.1155/2010/185063>
 - [4] Jiao, L., Luo, J., Shang, R. and Liu, F. (2014) A Modified Objective Function Method with Feasible-Guiding Strategy to Solve Constrained Multi-Objective Optimization Problems. *Applied Soft Computing*, **14**, 363-380. <https://doi.org/10.1016/j.asoc.2013.10.008>
 - [5] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**, 182-197. <https://doi.org/10.1109/4235.996017>
 - [6] Xie, S., Li, K., Wang, W., Wang, H., Peng, C. and Jalil, H. (2025) A Tractive Population-Assisted Dual-Population and Two-Phase Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **29**, 31-45. <https://doi.org/10.1109/tevc.2023.3345470>
 - [7] Liang, J., Qiao, K., Yu, K., Qu, B., Yue, C., Guo, W., *et al.* (2023) Utilizing the Relationship between Unconstrained and Constrained Pareto Fronts for Constrained Multiobjective Optimization. *IEEE Transactions on Cybernetics*, **53**, 3873-3886. <https://doi.org/10.1109/teyb.2022.3163759>
 - [8] Deb, K. and Jain, H. (2014) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEE Transactions on Evolutionary Computation*, **18**, 577-601. <https://doi.org/10.1109/tevc.2013.2281535>
 - [9] Tian, Y., Wang, R., Zhang, Y. and Zhang, X. (2025) Adaptive Population Sizing for Multi-Population Based Constrained Multi-Objective Optimization. *Neurocomputing*, **621**, Article ID: 129296. <https://doi.org/10.1016/j.neucom.2024.129296>
 - [10] Liu, Z., Wang, B. and Tang, K. (2022) Handling Constrained Multiobjective Optimization Problems via Bidirectional Coevolution. *IEEE Transactions on Cybernetics*, **52**, 10163-10176. <https://doi.org/10.1109/teyb.2021.3056176>
 - [11] Tian, Y., Zhang, T., Xiao, J., Zhang, X. and Jin, Y. (2021) A Coevolutionary Framework for Constrained Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, **25**, 102-116. <https://doi.org/10.1109/tevc.2020.3004012>
 - [12] Li, K., Chen, R., Fu, G. and Yao, X. (2019) Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 303-315. <https://doi.org/10.1109/tevc.2018.2855411>
 - [13] Liu, Z. and Wang, Y. (2019) Handling Constrained Multiobjective Optimization Problems with Constraints in Both the Decision and Objective Spaces. *IEEE Transactions on Evolutionary Computation*, **23**, 870-884. <https://doi.org/10.1109/tevc.2019.2894743>
 - [14] Tian, Y., Cheng, R., Zhang, X. and Jin, Y. (2017) Platemo: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Computational Intelligence Magazine*, **12**, 73-87. <https://doi.org/10.1109/mci.2017.2742868>
 - [15] Fan, Z., Li, W., Cai, X., Huang, H., Fang, Y., You, Y., *et al.* (2019) An Improved Epsilon Constraint-Handling Method in MOEA/D for CMOPs with Large Infeasible Regions. *Soft Computing*, **23**, 12491-12510. <https://doi.org/10.1007/s00500-019-03794-x>
 - [16] Zhou, Y., Xiang, Y. and He, X. (2021) Constrained Multiobjective Optimization: Test Problem Construction and Performance Evaluations. *IEEE Transactions on Evolutionary Computation*, **25**, 172-186. <https://doi.org/10.1109/tevc.2020.3011829>
 - [17] Bosman, P.A.N. and Thierens, D. (2003) The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, **7**, 174-188. <https://doi.org/10.1109/tevc.2003.810761>
 - [18] Zitzler, E. and Thiele, L. (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, **3**, 257-271. <https://doi.org/10.1109/4235.797969>