

文章编号: 1006-3080(2025)06-0804-13

DOI: 10.14135/j.cnki.1006-3080.20250308002

基于双阶段双种群的约束多目标进化算法

王学武, 畅晨辉, 顾幸生

(华东理工大学能源化工过程智能制造教育部重点实验室, 上海 200237)

摘要:在处理约束多目标优化问题(Constrained Multi-Objective Optimization Problems, CMOPs)时, 平衡收敛性、可行性和多样性至关重要。现有的约束多目标进化算法(Constrained Multi-Objective Evolutionary Algorithms, CMOEAs)往往难以实现这种平衡, 在处理具有复杂可行域的优化问题上表现欠佳。本文提出了一种基于双阶段双种群的进化算法(CMOEA-DD)。在全局搜索阶段, Pop1 通过动态排名策略实现目标与约束之间的有效平衡, 向 Pareto 前沿收敛的同时保留了目标值良好的不可行个体, 进一步加快了收敛速度; Pop2 忽略约束, 旨在搜索整个空间, 通过共享后代信息引导 Pop1 穿越不可行区域; 在局部开发阶段, Pop2 逐步增加个体对约束条件的偏好程度, 同时通过提供多样性较好的可行个体, 改善 Pop1 的分布效果。在 4 个知名测试套件上进行的实验表明, CMOEA-DD 比 7 种代表性约束多目标进化算法更具竞争力。

关键词:约束多目标优化; 进化算法; 协同进化; 双种群; 双阶段

中图分类号: TP301

文献标志码: A

目前, 多目标进化算法(Multi-Objective Evolutionary Algorithms, MOEAs)大致可分为 3 类^[1]: 第 1 类是基于 Pareto 支配的算法, 代表算法为 NSGA-II^[2]; 第 2 类是基于分解的算法, 代表算法为 MOEA/D^[3]; 第 3 类是基于指标的算法, 代表算法为 IBEA^[4]。已有大量研究围绕无约束多目标优化问题(Multi-Objective Optimization Problems, MOPs)展开, Wang 等^[5]提出了一种基于统计信息的分布多目标进化算法, 算法在过渡阶段和局部优化阶段根据反馈的统计信息指导亲本的选择过程, 改善个体的分布性和收敛性。王学武等^[6]针对二维和三维的 MOPs, 提出了一种基于 Pareto 支配的两阶段多目标进化算法, 算法在全局搜索阶段依据临界子集的排序等级进行个体选择, 在局部调整阶段对种群中的个体进行微调, 进而改善个体的分布效果。约束多目标优化问题(Constrained Multi-Objective Optimization Problems, CMOPs)比 MOPs 更具挑战性, 因为约束条件会使解空间的很

大部分区域不可行, 或者将可行区域分割成狭窄的部分^[7]。经典算法往往难以实现约束与目标之间的平衡, 为解决这个难题, 研究人员从种群数量和进化过程中采用的阶段策略出发, 设计了多种约束多目标进化算法(Constrained Multi-Objective Evolutionary Algorithms, CMOEAs), 并将其划分为 3 类: 单种群双阶段算法、双种群单阶段算法和双种群双阶段算法。

在单种群双阶段算法中, 一个代表性算法是 PPS^[8], 此算法在推动阶段不考虑任何约束, 使种群能够快速收敛到无约束 Pareto 前沿(Unconstrained Pareto Front, UPF); 在拉动阶段, 算法采用改进的 epsilon 约束方法使种群从 UPF 逐渐收敛到约束 Pareto 前沿(Constrained Pareto Front, CPF)。另一个代表性算法是 ToP^[9], 它将进化过程分为两个阶段: 第 1 阶段, 通过将 CMOPs 转化为单目标优化问题来确定有前景的可行区域; 第 2 阶段, 应用现有的 CMOEAs 得到最

收稿日期: 2025-03-08

基金项目: 国家自然科学基金(62076095)

作者简介: 王学武(1972—), 男, 陕西合阳人, 博士, 副教授, 研究方向包括智能优化技术、工业机器人优化与调度、焊接专家系统、电弧增材制造、焊接机器人智能化技术。E-mail: wangxuewu@ecust.edu.cn

引用本文: 王学武, 畅晨辉, 顾幸生. 基于双阶段双种群的约束多目标进化算法[J]. 华东理工大学学报(自然科学版), 2025, 51(6): 804-816.

Citation: WANG Xuewu, CHANG Chenhui, GU Xingsheng. Constrained Multi-Objective Evolutionary Algorithm Based on Dual-Stage Dual-Population[J]. Journal of East China University of Science and Technology, 2025, 51(6): 804-816.

终的非支配个体。为了更好地平衡目标与约束, Tian等^[1]提出了一种名为CMOEA-MS的两阶段算法,它采用不同的适应度评估策略来调整目标和约束的优先级,并根据当前种群的进化状态切换进化阶段。

研究人员从协同进化机制在大规模优化领域取得的成功中获得启发,尝试开发协同进化算法来解决CMOPs^[10]。Li等^[11]提出了一种名为C-TAEA的双存档算法,以收敛性为导向的存档(CA)旨在向CPF收敛,以多样性为导向的存档(DA)则通过探索CA尚未覆盖的区域来维持多样性。Tian等^[12]提出了一种基于双种群的协同进化算法(CCMO),该算法通过相同的优化器分别进化两个种群,其中辅助种群借助种群间的弱合作策略来提升原始种群对CMOPs的求解效果。

近年来双种群和双阶段结合的策略引起了研究人员的极大关注。Ming等^[10]提出了一种具有双种群(mainPop和auxPop)和双阶段策略(探索阶段和开发阶段)的进化算法(DD-CMOEA):在探索阶段,mainPop和auxPop分别探索可行区域和不可行区域;在开发阶段,mainPop通过牵引策略帮助auxPop收敛到CPF。Zou等^[13]提出了一种名为CAEAD的双种群双阶段进化算法,算法将进化过程分为进化阶段和退化阶段,并根据次级种群的状态切换进化阶段。Xie等^[14]提出了一种结合双种群(开发种群和牵引种群)和两个阶段的进化算法(DPTPEA):在第1阶段,牵引种群通过忽略约束条件,拉动开发种群穿越不可行区域;在第2阶段,算法通过种群间的动态合作策略、动态环境选择策略和边界点方向采样策略,增强牵引种群的探索能力。袁志超等^[15]提出了一种双阶段双种群的进化算法(DPDSEA),算法在第2阶段,根据UPF和CPF的位置对CMOPs进行分类,当二者完全分离或者部分重合时,采用随机扰动策略促进主种群的收敛和分布。

为了进一步平衡收敛性、可行性和多样性,本文提出了一种基于双阶段双种群的进化算法(CMOEA-DD):在全局搜索阶段,Pop2忽略约束牵引Pop1跨越不可行区域,同时Pop1通过动态排名策略保留了目标值较好的个体,这些优势个体通过信息共享机制,提升了其他个体跨越不可行区域的概率;在局部开发阶段,Pop2从UPF向CPF收敛,收敛过程中探索到的可行个体有助于改善Pop1的分布效果。

1 基本概念

一般来说,最小化CMOPs可以表述如下^[16]:

$$\begin{aligned} & \text{Minimize } \mathbf{F}(x) = [f_1(x), f_2(x), \dots, f_M(x)]^T \\ & \text{subject } x \in \Omega \\ & g_j(x) \leq 0, j = 1, \dots, p \\ & h_j(x) = 0, j = p+1, \dots, q \end{aligned} \quad (1)$$

式中, x 是决策空间 Ω 中的决策变量, $\mathbf{F}(x)$ 是由相互冲突的目标函数组成的目标向量, $g_j(x)$ 和 $h_j(x)$ 分别是第 j 个不等式约束和等式约束。 x 相对某个约束条件的违反程度通常按如下方式计算^[17]:

$$\text{CV}_j(x) = \begin{cases} \max\{0, g_j(x)\}, j = 1, \dots, p \\ \max\{0, |h_j(x)| - \delta\}, j = p+1, \dots, q \end{cases} \quad (2)$$

其中, δ 是容忍参数。

总的约束违反程度定义如下:

$$\varphi(x) = \sum_{j=1}^q \text{CV}_j(x) \quad (3)$$

如果不存在相互支配的解,那么这个解集就构成了一个Pareto最优集。所有Pareto最优集在目标空间中的映射构成了Pareto前沿(Pareto Front, PF)。具体地说,对于一个CMOP,把它的PF称为CPF,而其无约束的部分称为UPF。求解CMOPs的主要目标是找到一个具有良好收敛性和多样性的可行解集,以此来逼近真实的CPF^[14]。

2 基于双阶段双种群的约束多目标进化算法

2.1 算法框架

算法1给出了CMOEA-DD的伪代码,如表1所示。

首先对Pop1和Pop2进行随机初始化,随后进化过程分两个阶段展开:全局搜索阶段和局部开发阶段。下一代Pop1和Pop2分别由算法2(表2)和算法3(表3)产生。种群的进化状态通过目标值的变化来确定^[13]。首先计算Pop2中非支配个体数量 N_c ,再计算当前代和上一代的个体的目标值总和,如果这两个总和之间的差值 σ 小于预定阈值 κ (本文中 $\kappa = 1 \times 10^{-2}$),并且Pop2中的个体都为非支配个体,那么就认为Pop2进行了充分探索,种群已经收敛到了UPF,进化进入局部开发阶段。局部开发阶段采用SPEA2^[18]中提到的适应度函数进行适应度评估。Pop2通过epsilon约束方法从UPF收敛到CPF,Pop1则基于可行性继续向CPF收敛。算法4给出了stage和epsilon更新的伪代码,如表4所示。

2.2 全局搜索

为了在全局搜索阶段平衡Pop1的收敛性和可行性,本文设计了一种基于可行性排名和收敛性排

表 1 算法 1
Table 1 Algorithm 1

Algorithm 1: Procedure of CMOEA-DD

Input: N (population size);
Output: Pop1

- 1: Pop1, Pop2 \leftarrow Random Initialization(N);
- 2: stage = global search
- 3: While termination criterion not fulfilled do
- 4: If stage = global search
- 5: $M1 \leftarrow$ Matching Parent(2, Pop1, N),
 $M2 \leftarrow$ MatchingParent(2, Pop2, N);
- 6: $O1 \leftarrow$ OperateGA(Pop1, $M1$, N), $O2 \leftarrow$ OperateGA(Pop2, $M2$, N);
- 7: Pop1 \leftarrow Algorithm 2;
- 8: Pop2 \leftarrow Algorithm 3;
- 9: Else
- 10: $S1 \leftarrow$ Calculate Fitness(Pop1, N), $S2 \leftarrow$ Calculate Fitness(Pop2, N);
- 11: $M1 \leftarrow$ Matching Parent(2, N , $S1$), $M2 \leftarrow$ Matching Parent(2, N , $S2$);
- 12: $O1 \leftarrow$ OperateGA(Pop1, $M1$, N), $O2 \leftarrow$ OperateGA(Pop2, $M2$, N);
- 13: Pop1 \leftarrow Environment Selection(Pop1, $O1$, $O2$, N , 0);
- 14: Pop2 \leftarrow Environment Selection(Pop2, $O1$, $O2$, N , epsilon);
- 15: End If
- 16: $N_c \leftarrow$ Number of non-dominant solutions in Pop2;
- 17: $\sigma \leftarrow$ Calculate the objective function change of Pop2;
- 18: if $N_c < N$ and $\sigma < \kappa$ then
- 19: stage, epsilon \leftarrow Algorithm 4;
- 20: End If
- 21: End While
- 22: Return Pop1

名环境选择策略。首先, 根据约束支配原则(CDP)确定个体之间的支配关系, 随后产生非支配层。按照式(4)计算每个个体的拥挤距离。在确定种群的收敛性排名时, 不会直接比较个体的目标函数值, 个体首先会被划分到子区域中, 子区域的定义如下^[19]:

$$\psi_i = \{F(x) \in \mathbb{R}^m \mid \forall j: \langle F(x), \lambda_j \rangle \leq \langle F(x), \lambda_i \rangle\} \quad (4)$$

其中, $i, j \in \{1, 2, \dots, N\}$, λ_i, λ_j 分别是第 i, j 个权向量, $\langle F(x), \lambda_i \rangle$ 是 $F(x)$ 与 λ_i 之间的夹角。

通过计算每个子区域中个体的聚合函数值来评估个体的收敛性, 这有助于个体更快地收敛到 CPF。聚合函数值的计算公式如下^[3]:

$$g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i|\} \quad (5)$$

其中, z^* 是由每个目标的最小值指定的参考点, $f_i(x)$ 是第 i 个目标函数, z_i 是参考点 z^* 的第 i 个分量。

通过参数 α 来实现两个排名的动态调整, α 的计算公式如下:

$$\alpha = \frac{1}{1 + e^{-20t/T}} \quad (6)$$

其中, t 和 T 分别是当前进化代数和最大进化代数。 α 从 0.5 开始, 在全局搜索阶段结束时趋近于 1。在进化的早期, α 等于 0.5, 表明约束和目标同等重要。在整个进化过程中, Pop1 逐渐增强对约束的偏好, 然而 $1 - \alpha$ 仍然大于 0, 表明 Pop1 并没有完全忽略个体的目标值。

综合排名 Rank 由式(7)计算, RankC 表示种群的收敛性排名, RankF 表示种群的可行性排名。

$$\text{Rank} = \alpha \text{RankC} + (1 - \alpha) \text{RankF} \quad (7)$$

表 2 算法 2
Table 2 Algorithm 2

Algorithm 2: Environmental selection of Pop 1

Input: Pop1, $O1$, $O2$, N , M ;
Output: Pop1;

- 1: RankC \leftarrow Initialize convergence ranking;
- 2: RankF \leftarrow Initialize feasibility ranking;
- 3: $\Lambda \leftarrow$ Uniform Point(M , N);
- 4: $M \leftarrow$ Merge Population(Pop1, $O1$, $O2$);
- 5: $R \leftarrow$ Related Individuals(M , Λ);
- 6: $\{F_1, \dots, F_N\} \leftarrow$ Calculate dominance relationship based on CDP;
- 7: CrowdDis \leftarrow Calculate the crowding distance of individuals;
- 8: If $\text{NUM}(F_i) \leq N$
- 9: RankF \leftarrow sort($\{F_1, \dots, F_N\}$, CrowdDis);
- 10: Else
- 11: RankF \leftarrow Truncation Method(F_i , CrowdDis, $N - \text{Num}(F_1)$);
- 12: End If
- 12: $A \leftarrow$ Calculate the value of the aggregation function;
- 13: RankC \leftarrow sort(A , CrowdDis);
- 14: Pop1 \leftarrow serset(RankC, RankF);
- 15: Return Pop1

表 3 算法 3
Table 3 Algorithm 3

Algorithm 3: Environmental selection of Pop2

Input: Pop2, $O1$, $O2$, N ;
Output: Pop2;

- 1: OBJ \leftarrow Calculate the objective value of Pop2;
- 2: $\{F_1, \dots, F_N\} \leftarrow$ Calculate dominance relationship based on OBJ;
- 3: CrowdDis \leftarrow Calculate the crowding distance of individuals;
- 4: Pop $\leftarrow \emptyset$;
- 5: While $\text{NUM}(\text{Pop}) \leq N$
- 6: Pop \leftarrow select($\{F_1, \dots, F_N\}$);
- 7: End While
- 8: Pop \leftarrow delete(Pop, $\text{NUM}(\text{Pop}) - N$, CrowdDis);
- 9: Pop2 \leftarrow Pop;
- 10: Return Pop2

表4 算法4
Table 4 Algorithm 4

Algorithm 4: Update stage, epsilon

Input: $\varepsilon(k)$ (Current epsilon), μ (Threshold value),
 τ (Variation parameter);

Output: stage, epsilon ;

```

1: If  $\varepsilon(k) > \mu$ 
2:   epsilon =  $(1 - \tau) * \varepsilon(k)$ ;
3:   stage = local exploitation ;
4: Else
5:   epsilon = max( $\varepsilon$ ) ;
6:   stage = global search ;
7: End If
8: Return stage, epsilon

```

2.3 局部开发

在局部开发阶段,CMOEA-DD的适应度计算方式和SPEA2^[18]中策略相同,计算公式如下:

$$\text{fit}(x) = \sum_{y \in S_x} |R_y| + \frac{1}{\text{dis}(x, x') + 2} \quad (8)$$

其中, $|R_y|$ 表示 S_x 中的个体 y 支配其他个体的数量; 式中右侧的第1项首先将支配个体 x 的所有个体加入到集合 S_x 中, 之后统计集合 S_x 中的个体支配其他个体的数量; 式中右侧的第2项首先计算了个体 x 与种群中其他个体 x' 之间的距离, 之后对距离值进行排序, 将 x 与距离值排名为 $|\sqrt{2n}|$ 的邻居之间距离的倒数作为 x 的拥挤距离。个体的适应度值越小, 表示其在收敛性、可行性和多样性这几个指标上的表现越好, 适应度小于1的个体不被其他个体支配。

2.4 算法复杂度分析

多目标优化问题的问题数目为 M , 种群大小为 N , CMOEA-DD的计算复杂度主要体现在算法的两个阶段。在全局搜索阶段, 非支配排序、二元竞赛、产生后代的计算复杂度分别为 $O(N^2)$ 、 $O(N)$ 、 $O(MN)$ 。Pop2的环境选择由非支配排序和计算拥挤距离决定, 相应的计算复杂度分别为 $O(N^2)$ 、 $O(N \lg N)$ 。Pop1的环境选择涉及到了子区域划分个体、非支配排序、计算拥挤距离、计算收敛性排名、计算可行性排名, 相应的计算复杂度分别为 $O(MN^2)$ 、 $O(N^2)$ 、 $O(N \lg N)$ 、 $O(MN^2)$ 、 $O(N^2 \lg N)$ 。所以, 全局搜索阶段的计算复杂度为 $O(N^2 \lg N)$ 。局部开发涉及个体适应度的计算、二元竞赛、产生后代和环境选择, 相应的计算复杂度分别为 $O(MN)$ 、 $O(N)$ 、 $O(MN)$ 、 $O(N^2 \lg N)$, 所以局部开发阶段的计算复杂度为 $O(N^2 \lg N)$ 。结合两个阶段, 算法的总的计算复杂度为 $O(N^2 \lg N)$ 。

3 仿真测试与分析

3.1 算法参数设置与测试套件

所有实验均在 PlatEMOv4.6 平台进行^[20]。将3种经典算法(NSGA-II^[2]、C-MOEA/D^[21]、NSGA-III^[21])以及4种最近提出的算法(PPS^[8]、ToP^[9]、CMOEA-MS^[11]、CAEAD^[13])与CMOEA-DD进行比较。种群数量设置为100, 最大评估次数设置为100000, 并且所有算法在测试问题上独立运行30次。

CMOEA-DD、CAEAD、CMOEA-MS、NSGA-II、NSGA-III和C-MOEA/D采用模拟二元交叉和多项式变异产生后代, PPS和ToP采用差分进化和多项式变异产生后代, 交叉概率设为1, 变异的概率设为 $1/d$ (d 是决策变量的个数), 交叉和变异的分布指数设置为20, 差分进化中的参数CR和F分别设为1和0.5。C-MOEA/D的权向量个数 T 设置为 $N/10$ 。PPS的其他参数设置为: 控制代数 $T_c = 0.8 \times T_{\max}$, 控制可行区域与不可行区域搜索的偏好参数 $\bar{\omega} = 0.95$, 控制约束松弛度降低速度的参数: $\tau = 0.1$ 、 $cp = 2$ 。ToP的阶段切换条件为: 可行解比例 $P_f > 1/3$ 或可行解在聚合目标维度上的分布跨度 $\delta < 0.2$ 。CMOEA-DD和CAEAD控制进化阶段的参数设置为: $\mu = 5 \times 10^{-5}$ 、 $\tau = 0.05$ 。CMOEA-MS设置进化阶段的参数 λ 为0.5。

3.2 测试套件

本文选择4个基准测试套件来验证CMOEA-DD的有效性, 测试套件分别为LIR-CMOP^[22]、MW^[7]、DASC MOP^[23]以及DC_DTLZ^[11]。LIR-CMOP包含14个测试问题, 其中LIR-CMOP13和LIR-CMOP14是三目标问题, 其余问题是双目标问题: 该套件的特点是存在非常大的不可行区域, 其CPF由若干不连续的片段组成。MW同样包含14个测试问题, 其中MW4、MW8和MW14是三目标问题, 其余的问题都是双目标问题: 该套件包含各种问题特征, 例如非线性约束以及不同几何形状的CPF^[7]。DASC MOP测试套件同时呈现多样性困难、收敛困难和可行性困难, 这使算法很难找到非支配可行解, 其中DASC MOP7、DASC MOP8和DASC MOP9是三目标问题, 其余的问题都为双目标问题。DC_DTLZ测试套件中的测试问题均为三目标问题。

3.3 性能指标

采用IGD^[24]和HV^[25]来评估CMOEA-DD的性能。IGD的定义如下:

$$\text{IGD}(P, Q) = \frac{\sum_{X \in P} \text{dis}(X, Q)}{|P|} \quad (9)$$

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 的最小欧式距离。IGD 值越小, 算法的综合性能越好。

HV 的定义如下:

$$\text{HV} = \varrho \left(\bigcup_{i=1}^{|\mathcal{S}|} v_i \right) \quad (10)$$

其中, ϱ 为勒贝格测度, $|\mathcal{S}|$ 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, 算法的综合性能越好。

对 IGD 和 HV 的结果进行 90% 置信水平的秩和比检验, 结果见表 5~表 10 所示: 其中符号“+”、“-”、“ \approx ”表示相应的算法优于、劣于或者与 CMOEA-DD 相当; 算法获得的最优值在每个表中加粗显示, NAN 表明算法没有找到可行的解决方案。

3.4 测试结果分析

表 5 和表 6 分别示出了 8 种算法在 MW 测试套件上得到的 IGD 和 HV。在所有测试问题上, CMOEA-DD 获得的 IGD 和 HV 明显优于 NSGA-II、ToP 以及 CAEAD, 并且在 10 个问题上展现出了强大

的竞争力。PPS 仅在 MW11 上获得了最好的 HV; NSGA-III 和 CMOEA-MS 在 MW4 上获得了更好的 IGD 和 HV, 同时 NSGA-III 也在 MW14 上获得了最好的 HV; C-MOEA/D 在 MW4 上取得了最好的 IGD 和 HV, 同时也在 MW12 上取得了最好的 HV。

MW5 的 CPF 由不连续的片段组成, 这给许多现有的 CMOEAs 带来了挑战。图 1 示出了 8 种算法在 MW5 上的种群分布。ToP 被一个很大的不可行区域阻挡, 最终未能找到 CPF。NSGA-II、CAEAD 以及 PPS 仅找到了 CPF 的一部分, 出现了明显的局部收敛。NSGA-III 和 CMOEA-MS 只找到了 CPF 的一部分。C-MOEA/D 最终得到的种群展现出了良好的多样性, 但是收敛性较差, 只有 CMOEA-DD 得到了分布良好的种群。

表 7 和表 8 列出了 8 种算法在 LIR-CMOP 测试套件上获得的 IGD 和 HV。在所有测试问题上, CMOEA-DD 获得的 IGD 和 HV 明显优于 NSGA-II、C-MOEA/D、NSGA-III 以及 ToP。CMOEA-MS 在处理 LIR-CMOP13 和 LIR-CMOP14 时获得的 IGD 和 HV 优于 CMOEA-DD。PPS、CAEAD 以及 CMOEA-DD 在 LIR-CMOP 测试套件中的不同测试问题上各展所长。PPS 在 LIR-CMOP1、LIR-CMOP5 和 LIR-CMOP6 时获得了最好的 IGD 和 HV; CAEAD 在处

表 5 8 种算法在 MW 测试套件上获得的 IGD

Table 5 IGD obtained from eight algorithms on MW test suite

Function	IGD							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
MW1	0.0147	0.0226	0.0115	0.0040	NAN	0.0022	0.1500	0.0019
MW2	0.0300	0.0185	0.0239	0.1620	0.1700	0.0354	0.0892	0.0168
MW3	0.0094	0.0362	0.0380	0.0063	0.5120	0.0054	0.0058	0.0054
MW4	0.0550	0.0415	0.0421	0.0584	NAN	0.0440	0.0926	0.0496
MW5	0.2890	0.0265	0.1780	0.3560	NAN	0.0263	0.3590	0.0027
MW6	0.0302	0.0187	0.0441	0.5720	0.5930	0.0211	0.4130	0.0138
MW7	0.0498	0.0047	0.0558	0.0057	0.0873	0.0292	0.0055	0.0051
MW8	0.0612	0.0527	0.1750	0.1280	0.6620	0.0518	0.1230	0.0491
MW9	0.0815	0.0317	0.0681	0.0168	0.1150	0.1920	0.4660	0.0048
MW10	0.2110	0.0876	0.1380	0.3850	0.3410	0.0357	0.3810	0.0184
MW11	0.3080	0.1800	0.4760	0.0074	0.6470	0.0102	0.0064	0.0063
MW12	0.0056	0.0052	0.0697	0.0106	0.9450	0.0077	0.4270	0.0050
MW13	0.4820	0.1280	0.2800	0.4470	0.7230	0.1100	0.2670	0.0524
MW14	0.1250	0.2120	0.1130	0.1400	0.4130	0.1340	0.1960	0.1070
+/-/=	0/14/0	2/9/3	1/12/1	0/14/0	0/13/1	0/10/4	0/14/0	

表 6 8 种算法在 MW 测试套件上获得的 HV
Table 6 HV obtained from eight algorithms on MW test suite

Function	HV							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
MW1	0.4770	0.4700	0.4790	0.4850	0	0.4890	0.3540	0.4890
MW2	0.5400	0.5570	0.5480	0.3820	0.3790	0.5340	0.4590	0.5600
MW3	0.5410	0.5260	0.5250	0.5430	0.2170	0.5430	0.5420	0.5430
MW4	0.8240	0.8410	0.8410	0.8080	0	0.8370	0.7690	0.8280
MW5	0.2190	0.3160	0.2570	0.2080	0	0.3090	0.1960	0.3230
MW6	0.2910	0.3050	0.2860	0.0968	0.1220	0.3010	0.1300	0.3110
MW7	0.3950	0.4110	0.3940	0.4120	0.3560	0.4080	0.4110	0.4120
MW8	0.4960	0.5270	0.4470	0.3610	0.1300	0.5190	0.3790	0.5310
MW9	0.3480	0.3740	0.3510	0.3760	0.2720	0.2760	0.1230	0.3970
MW10	0.3210	0.3890	0.3570	0.2310	0.2460	0.4200	0.2320	0.4350
MW11	0.3690	0.4010	0.3270	0.4480	0.2830	0.4450	0.4470	0.4470
MW12	0.6040	0.6050	0.5520	0.5960	0.0071	0.6010	0.2510	0.6040
MW13	0.3610	0.4330	0.3830	0.2780	0.2240	0.4300	0.3320	0.4540
MW14	0.4520	0.4440	0.4680	0.4480	0.3350	0.4630	0.4320	0.4660
+/-/=	0/11/3	2/9/3	2/11/1	1/11/2	0/13/1	1/10/3	0/13/1	

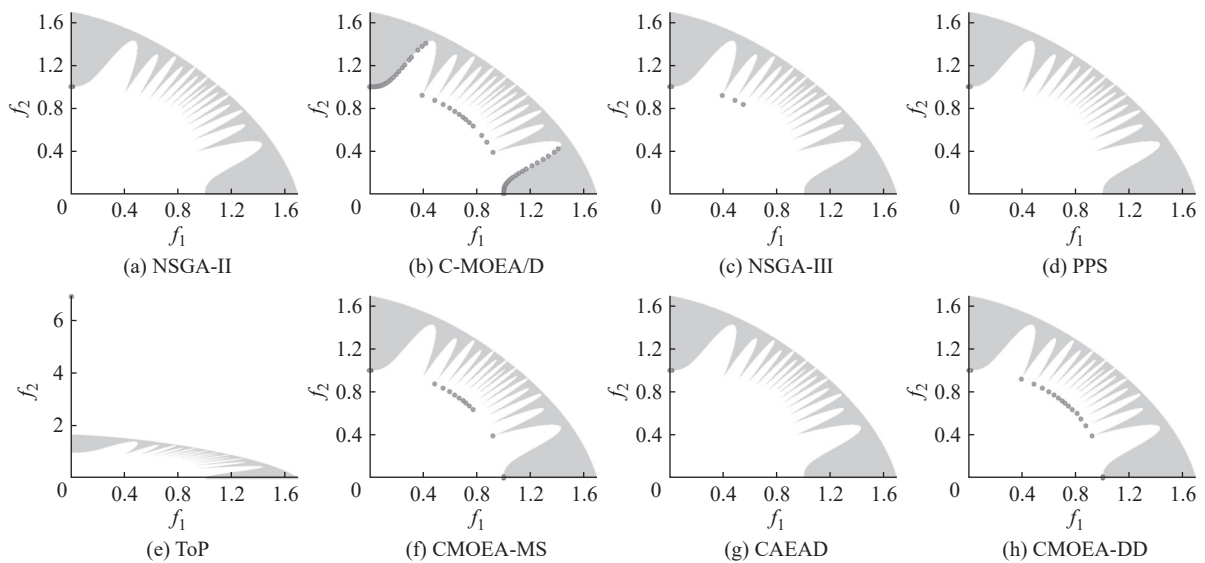


图 1 8 种算法在 MW5 上的种群分布

Fig. 1 Population distribution of eight algorithms on MW5

理 LIR-CMOP3、LIR-CMOP4、LIR-CMOP7 和 LIR-CMOP8 时, 获得了最好的 IGD 和 HV; CMOEA-DD 在处理 LIR-CMOP9、LIR-CMOP10、LIR-CMOP11 和 LIR-CMOP12 时, 获得了最好的 IGD 和 HV。从秩和检验的结果来看, CMOEA-DD 在这个测试套件所展现出的优势更为明显。

图 2 示出了 8 种算法在 LIR-CMOP10 问题上的

种群分布。NSGA-II、C-MOEA/D 和 NSGA-III 种群中的大多数个体被不可行区域阻挡, 最终停留在了该区域的外侧。ToP 种群中的部分个体跨越了不可行区域, 找到了一部分 CPF。PPS 在推动阶段成功找到了 UPF。然而, 在拉动阶段, 该算法向已经探索过的可行区域收敛。CMOEA-MS 和 CAEAD 种群中的所有个体都成功穿越了不可行区域, 但是只找到了

表 7 8 种算法在 LIR-CMOP 测试套件上获得的 IGD
Table 7 IGD obtained from eight algorithms on LIR-CMOP test suite

Function	IGD							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
LIR-CMOP1	0.3000	0.2830	0.3020	0.0355	0.3190	0.3270	0.0692	0.0869
LIR-CMOP2	0.2520	0.2350	0.2630	0.0290	0.2840	0.2730	0.0248	0.0608
LIR-CMOP3	0.3150	0.2740	0.3210	0.0816	0.3420	0.3230	0.0326	0.0712
LIR-CMOP4	0.2820	0.2460	0.2920	0.0820	0.3240	0.3030	0.0292	0.0593
LIR-CMOP5	1.2200	1.2200	1.2300	0.0077	1.2100	0.4800	0.5400	0.0231
LIR-CMOP6	1.3500	1.3500	1.3500	0.0077	1.3200	0.4810	0.3580	0.1010
LIR-CMOP7	0.6120	1.3800	0.6350	0.1250	1.5300	0.1320	0.0332	0.1280
LIR-CMOP8	1.1200	1.6300	1.2800	0.1800	1.5400	0.2100	0.0395	0.1450
LIR-CMOP9	0.9900	0.8690	0.9970	0.3780	0.6230	0.7970	0.4470	0.2370
LIR-CMOP10	0.8580	0.8100	0.9080	0.2860	0.4570	0.4740	0.2130	0.0247
LIR-CMOP11	0.6920	0.8380	0.8400	0.2640	0.4450	0.5490	0.1380	0.0131
LIR-CMOP12	0.8120	0.6320	0.8160	0.2340	0.3150	0.3350	0.1810	0.0769
LIR-CMOP13	1.3300	1.3000	1.3200	0.1390	1.3200	0.0928	0.2450	0.1050
LIR-CMOP14	1.2800	1.2600	1.2700	0.1260	1.2700	0.0950	0.1510	0.1040
+/-/=	0/14/0	0/14/0	0/14/0	4/7/3	0/14/0	2/11/1	6/8/0	

表 8 8 种算法在 LIR-CMOP 测试套件上获得的 HV
Table 8 HV obtained from eight algorithms on LIR-CMOP test suite

Function	HV							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
LIR-CMOP1	0.1130	0.1180	0.1140	0.2220	0.1070	0.1090	0.2010	0.1970
LIR-CMOP2	0.2350	0.2380	0.2270	0.3480	0.2200	0.2280	0.3470	0.3290
LIR-CMOP3	0.1000	0.1090	0.0978	0.1750	0.0913	0.0995	0.1910	0.1770
LIR-CMOP4	0.1990	0.2110	0.1910	0.2810	0.1760	0.1900	0.3030	0.2890
LIR-CMOP5	0.0000	0.0000	0.0000	0.2910	0.0000	0.1210	0.1450	0.2810
LIR-CMOP6	0.0000	0.0000	0.0000	0.1970	0.0019	0.0896	0.1180	0.1720
LIR-CMOP7	0.1680	0.0477	0.1610	0.2460	0.0228	0.2450	0.2830	0.2450
LIR-CMOP8	0.0826	0.0073	0.0591	0.2360	0.0218	0.2310	0.2810	0.2460
LIR-CMOP9	0.1050	0.1610	0.1140	0.4410	0.2700	0.2240	0.3790	0.4840
LIR-CMOP10	0.1030	0.1500	0.0866	0.5630	0.4420	0.4280	0.5980	0.6910
LIR-CMOP11	0.2670	0.2180	0.1970	0.5240	0.3980	0.3890	0.6090	0.6870
LIR-CMOP12	0.2470	0.3590	0.2620	0.5110	0.4620	0.4610	0.5240	0.5830
LIR-CMOP13	0.0001	0.0004	0.0003	0.4970	0.0066	0.5560	0.4310	0.5360
LIR-CMOP14	0.0004	0.0010	0.0007	0.5190	0.0070	0.5560	0.4860	0.5440
+/-/=	0/14/0	0/14/0	0/14/0	4/7/3	0/14/0	2/11/1	5/8/1	

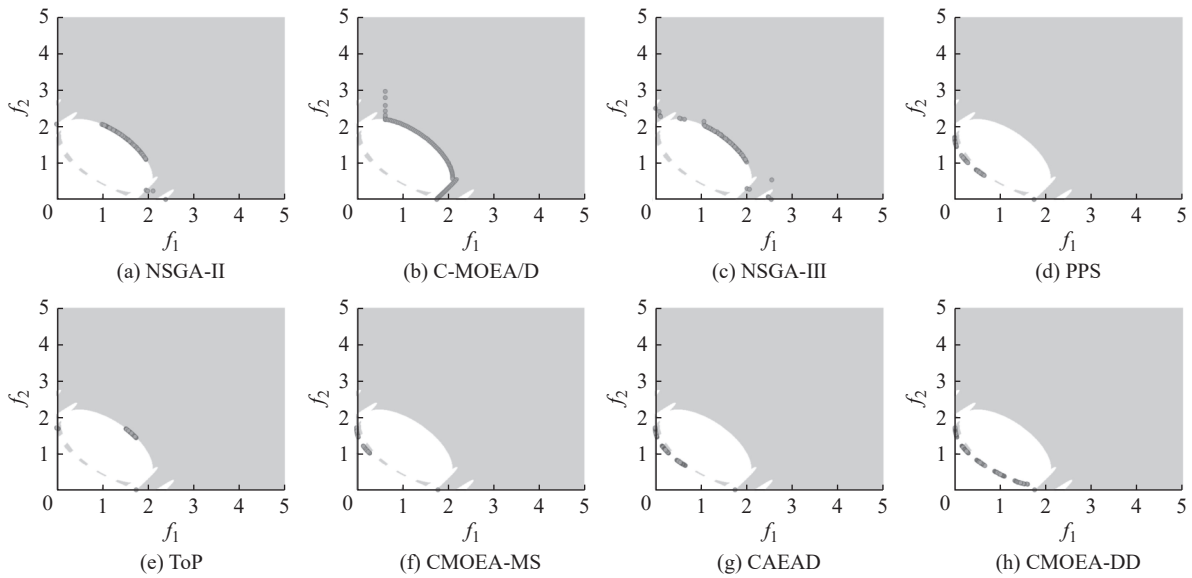


图 2 8 种算法在 LIR-CMOP10 上的种群分布

Fig. 2 Population distribution of eight algorithms on LIR-CMOP10

部分 CPF, 因为算法在两个阶段都没有充分考虑目标函数, 导致种群在后期对解空间的探索有限, 只有 CMOEA-DD 得到了分布良好的种群。

8 种算法在 LIR-CMOP 测试问题上 HV 的变化曲线如图 3 所示。总之, 经典算法在处理这类具有较大不可行区域的 CMOPs 时会面临重大挑战, 这些算法均在早期收敛, 摆脱局部最优的能力较弱。从

LIR-CMOP10 和 LIR-CMOP12 这两个问题的收敛曲线中可以看出, CMOEA-DD 获得了最优的 HV, 且算法收敛速度并没有因此变慢, 其次是 CAEAD、CMOEA-MS 和 PPS, 这表明双阶段双种群算法在解决复杂可行域问题时展现出了一定的优势。进一步对 LIR-CMOP3 和 LIR-CMOP6 的 HV 变化曲线进行分析可知, CAEAD、CMOEA-MS 和 CMOEA-DD 解

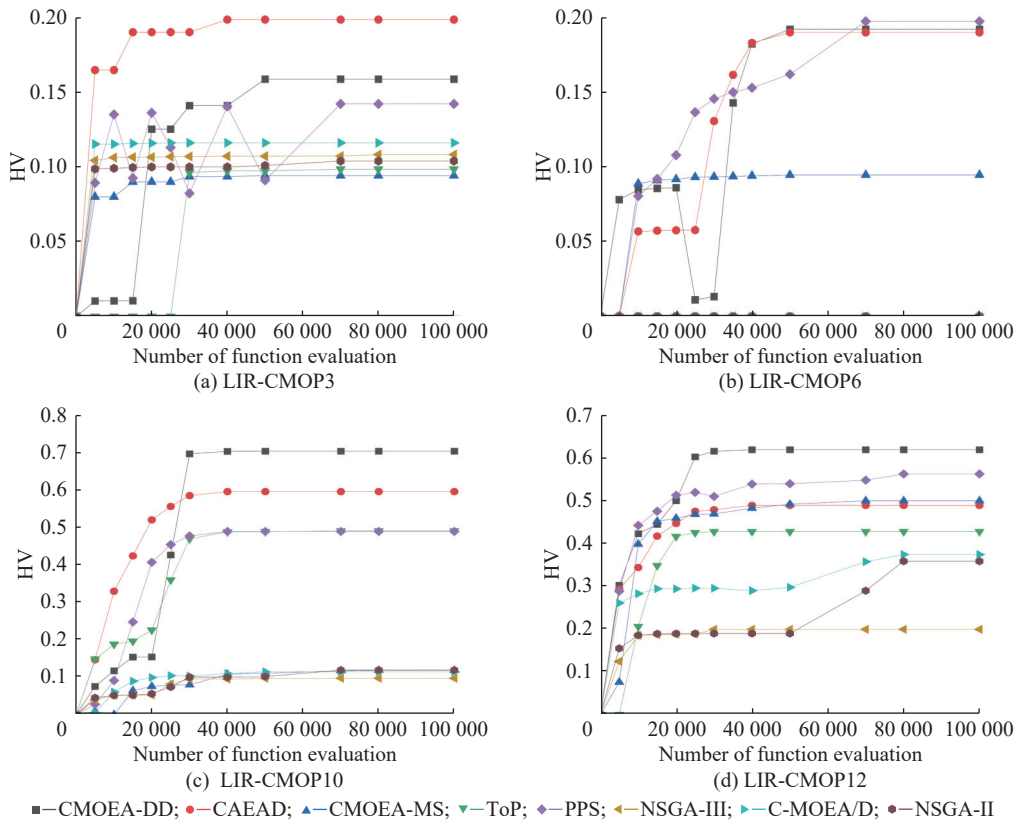


图 3 8 种算法在 LIR-CMOP 测试问题上 HV 的变化曲线

Fig. 3 Variation curves of HV for eight algorithms in LIR-CMOP testing problem

决测试问题 LIR-CMOP3, CAEAD、PPS和 CMOEA-DD 解决测试问题 LIR-CMOP6 时, 进化虽然会在一段时间内陷入停滞状态, 但是之后会进入有效的搜索状态, 这表明通过进化阶段的切换(使用不同的进化策略或者改变目标和约束的偏好程度)可以增加

种群的探索力度。

表 9 和表 10 分别示出了 8 种算法在 DASCMP 和 DC_DTLZ 测试套件上获得的 IGD 和 HV。CMOEA-DD 在所有测试问题上的 IGD 和 HV 显著优于 NSGA-II 以及 ToP; NSGA-III 在 DC1_DTLZ1 上获得了最好

表 9 8 种算法在 DASCMP 和 DC_DTLZ 测试套件上获得的 IGD

Table 9 IGD obtained from eight algorithms on DASCMP and DC_DTLZ test suites

Function	IGD							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
DASCMP1	0.7410	0.6970	0.7490	0.0879	0.7660	0.7310	0.0039	0.3090
DASCMP2	0.2800	0.2420	0.2730	0.0052	0.6160	0.2590	0.0047	0.1470
DASCMP3	0.3620	0.3680	0.3490	0.2740	0.6930	0.3440	0.0200	0.3560
DASCMP4	0.1200	0.0707	0.2950	0.2210	NAN	0.0548	0.0182	0.0024
DASCMP5	0.3020	0.0109	0.4050	0.1580	NAN	0.1100	0.0300	0.0030
DASCMP6	0.5260	0.4760	0.5100	0.3460	NAN	0.3150	0.1110	0.0345
DASCMP7	0.0512	0.0432	0.0410	0.2120	NAN	0.0366	0.1760	0.0353
DASCMP8	0.0660	0.0834	0.0710	0.1750	NAN	0.0396	0.1330	0.0412
DASCMP9	0.3820	0.3670	0.4710	0.2050	0.6810	0.3580	0.0425	0.0428
DC1_DTLZ1	0.0147	0.0183	0.0137	0.0320	0.0345	0.0194	0.0805	0.0147
DC1_DTLZ3	0.0433	0.0451	0.0425	0.4290	0.8420	0.0766	1.7700	0.0370
DC2_DTLZ1	NAN	0.1160	0.1350	0.0475	NAN	0.0207	0.0630	0.0239
DC2_DTLZ3	NAN	0.5610	NAN	0.4140	NAN	0.3930	0.4530	0.2150
DC3_DTLZ1	0.1450	0.1070	0.1570	0.3880	2.4300	0.0348	1.0700	0.0083
DC3_DTLZ3	1.6300	1.1900	1.4700	2.5200	7.4300	1.2500	2.8000	0.1640
+/-/=	0/14/1	0/14/1	1/14/0	1/11/3	0/15/0	2/9/4	3/11/1	

表 10 8 种算法在 DASCMP 和 DC_DTLZ 测试套件上获得的 HV

Table 10 HV obtained from eight algorithms on DASCMP and DC_DTLZ test suites

Function	HV							
	NSGA-II	C-MOEA/D	NSGA-III	PPS	ToP	CMOEA-MS	CAEAD	CMOEA-DD
DASCMP1	0.0064	0.0127	0.0066	0.1900	0.0091	0.0125	0.2120	0.1330
DASCMP2	0.2460	0.2540	0.2460	0.3550	0.0884	0.2550	0.3550	0.2990
DASCMP3	0.2090	0.2030	0.2110	0.2300	0.0386	0.2130	0.3110	0.2050
DASCMP4	0.1530	0.1840	0.0883	0.1400	0	0.1870	0.1930	0.2030
DASCMP5	0.1800	0.3470	0.1410	0.2690	0	0.2870	0.3360	0.3510
DASCMP6	0.0585	0.1230	0.0652	0.1730	0	0.1640	0.2780	0.3050
DASCMP7	0.2820	0.2860	0.2840	0.2080	0	0.2860	0.2350	0.2850
DASCMP8	0.1980	0.1960	0.2010	0.1580	0	0.2080	0.1780	0.2050
DASCMP9	0.1250	0.1230	0.1100	0.1640	0.0641	0.1290	0.2040	0.2040
DC1_DTLZ1	0.6130	0.6270	0.6310	0.5730	0.5630	0.6050	0.4710	0.6170
DC1_DTLZ3	0.4620	0.4730	0.4700	0.2940	0.1800	0.4580	0.0783	0.4660
DC2_DTLZ1	0	0.5990	0.5500	0.7560	0	0.8400	0.7250	0.8300
DC2_DTLZ3	0	0.0079	0	0.1710	0	0.1960	0.1590	0.3790
DC3_DTLZ1	0.1810	0.2840	0.1610	0.2040	0.0138	0.4090	0.0303	0.5250
DC3_DTLZ3	0.0000	0.0000	0.0000	0.0306	0.0000	0.0000	0.0785	0.2600
+/-/=	0/15/0	3/11/1	2/11/2	2/11/2	0/15/0	3/9/3	3/11/1	

的 IGD 和 HV; C-MOEA/D 仅在 DC1_DTLZ3 上获得了最好的 HV; PPS 在 DASCMP2 上获得了最好的 HV; CMOEA-MS 在 DASCMP8 和 DC2_DTLZ1 上获得了最好的 IGD 和 HV, 在 DASCMP7 上获得了最好的 HV; CAEAD 在 DASCMP1 和 DASCMP3 上获得了最好的 IGD 和 HV, 在 DASCMP2 上获得了最好的 IGD。

图 4 示出了 8 种算法在 DASCMP5 上的种群分布。ToP 未能找到可行区域, NSGA-II、NSGA-III 和 PPS 陷入非最优的可行区域。CMOEA-MS 在第 2 阶段对目标的优先级设置较低, 使得算法容易陷入局部最优。虽然 C-MOEA/D 找到了大部分 CPF, 但其收敛性和多样性较差。CAEAD 和 CMOEA-DD 都找到了所有 CPF, 但 CMOEA-DD 获得的种群分布

更加均匀。图 5 示出了 8 种算法在 DC3_DTLZ1 上的种群分布。PPS 和 ToP 仅找到了 3 个 CPF 中的部分片段, 虽然其他算法成功识别出了 CPF 的 3 个部分, 但是它们最终的种群分布并不均匀, 只有 CMOEA-DD 得到了分布良好的种群。

3.5 策略有效性分析

为了直观地展示所提出策略的有效性, 本文使用 CMOEA-DD 对 MW5、LIR-CMOP5、LIR-CMOP11 和 DASCMP4 4 个问题展开测试, 结果如图 6 所示。MW5 的 CPF 是 UPF 的一部分, Pop2 帮助 Pop1 跨越不可行区域, 并为 Pop1 提供了具有良好多样性的可行非支配个体, Pop2 的作用主要体现在全局搜索阶段的早期。LIR-CMOP5 的 CPF 与 UPF 相同, Pop2 在进化早期就收敛到了 CPF(UPF), 并在进化

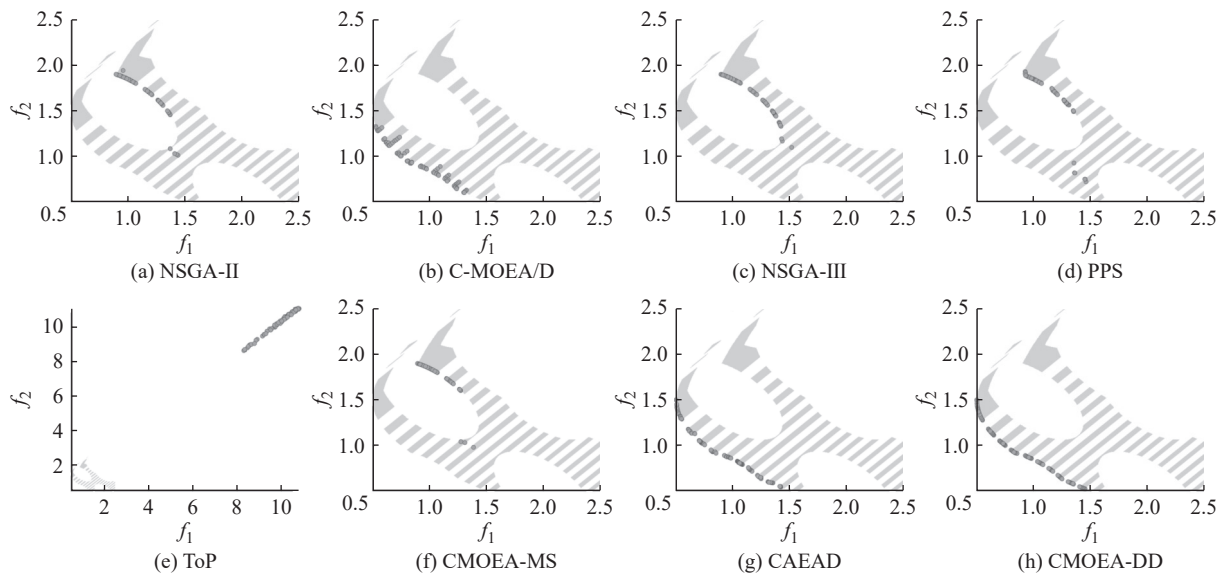


图 4 8 种算法在 DASCMP5 上的种群分布

Fig. 4 Population distribution of eight algorithms on DASCMP5

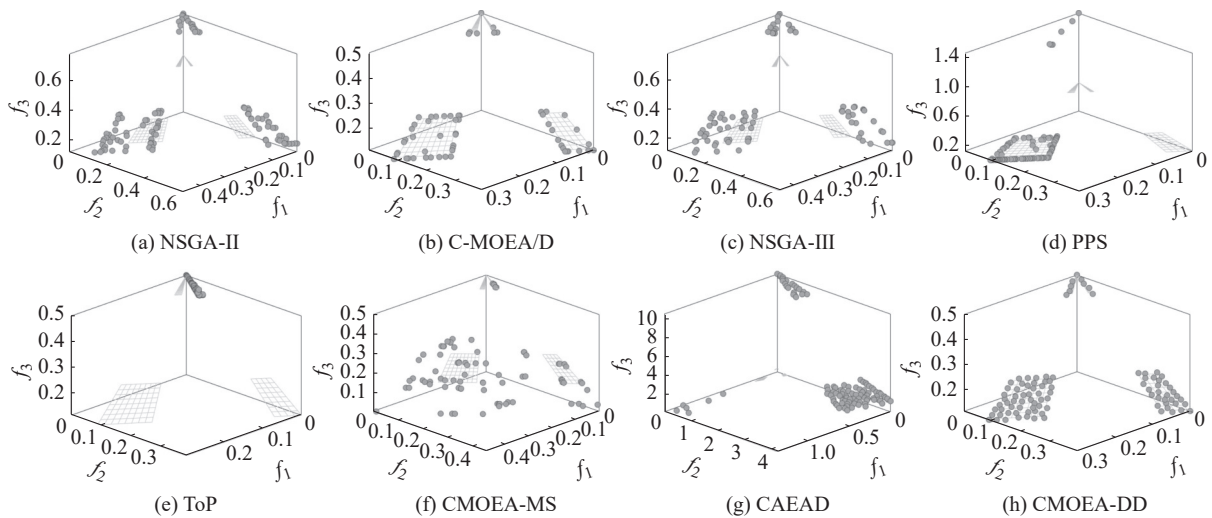


图 5 8 种算法在 DC3_DTLZ1 上的种群分布

Fig. 5 Population distribution of eight algorithms on DC3_DTLZ1

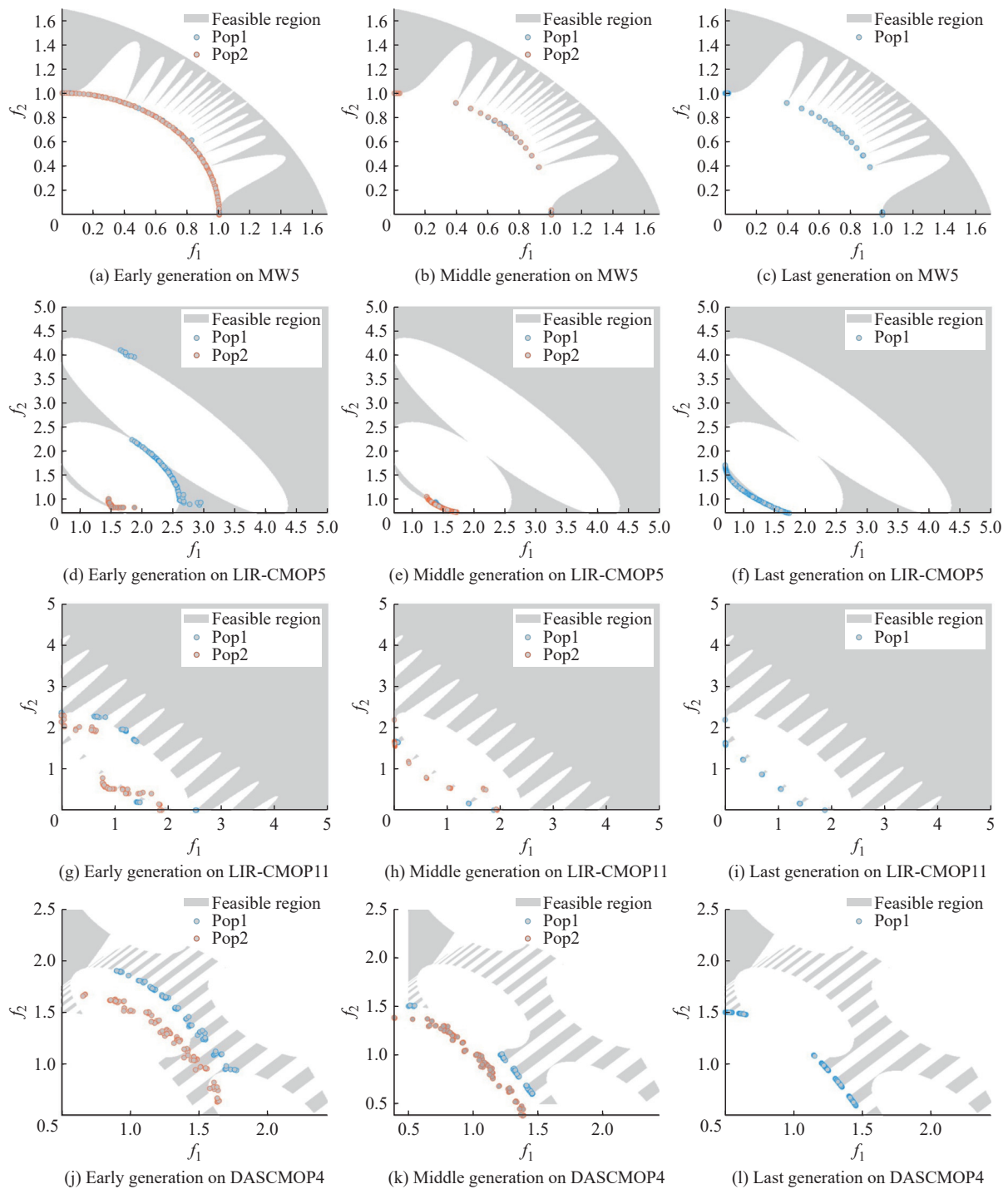


图 6 CMOEA-DD 在不同进化阶段的种群分布

Fig. 6 Population distribution of CMOEA-DD at different evolutionary stages

的中后期通过探索 CPF 的其他区域,协助 Pop1 找到完整的 CPF。对于 LIR-CMOP11, Pop1 通过两个排名的策略保留了一些具有良好目标值的个体,这些个体探索了目标空间中的未知区域,降低了 Pop1 陷入局部最优的可能性。在进化中期, Pop2 探索了一些 Pop1 未覆盖到的 CPF 片段,使得 Pop1 能够在进化的后期成功找到完整的 CPF。在 DASCMP4 问题上, Pop2 主要通过牵引作用帮助 Pop1 向 CPF 收敛。

4 结束语

本文提出了一种基于双阶段(全局搜索和局部开发)双种群(Pop1 和 Pop2)的进化算法 CMOEA-DD。CMOEA-DD 的两个种群被赋予了不同的任务,以并行方式展开搜索,这种方式使种群可以探索更广阔目标空间。双阶段的设计赋予了 Pop2 更为灵活的探索特性。在实验部分,本文使用 4 个基准测试套

件将CMOEA-DD与7种代表性CMOEA进行了比较,之后通过种群的演化过程分析了CMOEA-DD所采用策略的有效性。

参考文献:

- [1] TIAN Y, ZHANG Y J, SU Y S, *et al.* Balancing objective optimization and constraint satisfaction in constrained evolutionary multi-objective optimization[J]. *IEEE Transactions on Cybernetics*, 2021, 52(9): 9559-9572.
- [2] DEB K, PRATAP A, AGARWAL S, *et al.* A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [3] ZHANG Q F, LI H. MOEA/D: A multi-objective evolutionary algorithm based on decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [4] ZITZLER E, KÜNZL S. Indicator-based selection in multi-objective search[C]//International Conference on Parallel Problem Solving from Nature. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 832-842.
- [5] 王学武, 谢祖洪, 周昕, 等. 基于统计信息反馈的分步多目标优化 [J]. 华东理工大学学报 (自然科学版), 2022, 48(5): 665-676.
- [6] 王学武, 高进, 陈三燕, 等. 基于 Pareto 支配的两阶段多目标优化算法 [J]. 华东理工大学学报 (自然科学版), 2022, 48(6): 806-815.
- [7] MA Z W, WANG Y. Evolutionary constrained multi-objective optimization: Test suite construction and performance comparisons[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(6): 972-986.
- [8] 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.
- [9] LIU Z Z, WANG Y. Handling constrained multi-objective optimization problems with constraints in both the decision and objective spaces[J]. *IEEE Transactions on Evolutionary Computation*, 2019, 23(5): 870-884.
- [10] MING M J, WANG R, ISHIBUCHI H, *et al.* A novel dual-stage dual-population evolutionary algorithm for constrained multi-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2021, 26(5): 1129-1143.
- [11] LI K, CHEN R Z, FU G T, *et al.* Two-archive evolutionary algorithm for constrained multi-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 23(2): 303-315.
- [12] TIAN Y, ZHANG T, XIAO J H, *et al.* A coevolutionary framework for constrained multi-objective optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2020, 25(1): 102-116.
- [13] ZOU J, SUN R Q, YANG S X, *et al.* A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems[J]. *Information Sciences*, 2021, 579: 89-102.
- [14] XIE S M, LI K S, WANG W X, *et al.* A tractive population assisted dual-population and two-phase evolutionary algorithm for constrained multi-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2023, 29(1): 31-45.
- [15] 袁志超, 杨磊, 田井林, 等. 面向复杂约束多目标优化问题的双种群双阶段进化算法 [J]. 计算机应用, 2025, 45(8): 2656-2665.
- [16] DEB K. Multi-objective optimisation using evolutionary algorithms: An introduction[C]//Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing. London: Springer London, 2011: 3-34.
- [17] CAI Z X, WANG Y. A multi-objective optimization-based evolutionary algorithm for constrained optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 658-675.
- [18] ZITZLER E, LAUMANN S M, THIELE L. SPEA2: Improving the strength pareto evolutionary algorithm[J]. *TIK Report*, 2001, 103: 004284029.
- [19] LI K, DEB K, ZHANG Q F, *et al.* An evolutionary many-objective optimization algorithm based on dominance and decomposition[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 19(5): 694-716.
- [20] 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.
- [21] JAIN H, DEB K. 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*, 2013, 18(4): 602-622.
- [22] FAN Z, LI W J, CAI X Y, *et al.* An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions[J]. *Soft Computing*, 2019, 23: 12491-12510.
- [23] FAN Z, LI W J, CAI X Y, *et al.* Difficulty adjustable and scalable constrained multi-objective test problem toolkit[J]. *Evolutionary Computation*, 2020, 28(3): 339-378.
- [24] BOSMAN P A N, THIERENS D. The balance between proximity and diversity in multi-objective evolutionary algorithms[J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 174-188.
- [25] WHILE L, HINGSTON P, BARONE L, *et al.* A faster algorithm for calculating hypervolume[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(1): 29-38.

Constrained Multi-Objective Evolutionary Algorithm Based on Dual-Stage Dual-Population

WANG Xuewu, CHANG Chenhui, GU Xingsheng

(Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education,
East China University of Science and Technology, Shanghai 200237, China)

Abstract: When dealing with constrained multi-objective optimization problems (CMOPs), it is crucial to balance convergence, feasibility, and diversity. Existing constrained multi-objective evolutionary algorithms (CMOEA) often struggle to achieve such a balance, resulting in poor performance of the algorithms in handling optimization problems with complex feasible regions. To address this long-standing challenge, this paper proposes a novel evolutionary algorithm based on a two-stage and two-population collaborative optimization framework, termed CMOEA-DD. In the global exploration stage, the first population (Pop1) achieves an effective balance between objectives and constraints through a dynamic ranking strategy. Specifically, while guiding the population to stably converge towards the true Pareto front, this strategy deliberately retains infeasible individuals with excellent objective values, as such individuals can provide valuable directional information for subsequent feasible region exploration and further enhance the algorithm's convergence speed. Meanwhile, the second population (Pop2) ignores constraint conditions entirely and aims to conduct comprehensive and extensive exploration of the entire solution space, thereby avoiding the risk of being trapped in local feasible regions. Critically, Pop2 guides Pop1 to safely traverse potential infeasible regions by sharing high-quality offspring information, which helps Pop1 escape local optima and discover more promising feasible sub-regions. In the local development stage, Pop2 gradually increases the preference degree of individuals for the constraint conditions. At the same time, it improves the distribution effect of Pop1 by providing feasible individuals with good diversity. A large number of experiments conducted on four well-known test suites demonstrate that CMOEA-DD is more competitive than seven representative CMOEAs.

Key words: constrained multi-objective optimization; evolutionary algorithm; coevolution; dual-population; dual-stage

(责任编辑: 李娟)