

基于混合多策略约束处理的多目标进化算法

赖轩宇

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23日; 录用日期: 2026年3月18日; 发布日期: 2026年3月25日

摘 要

约束多目标优化问题(CMOPs)广泛存在于工程实践中, 如何在满足约束条件的同时获得分布良好的 Pareto 前沿是该领域的核心挑战。本文提出了一种基于混合多策略约束处理的多目标进化算法(Hybrid Multi-Strategy Constraint Handling Based Evolutionary Algorithm for Constrained Multi-Objective Optimization, HMSC-EC)。该算法采用三档案协同机制, 分别负责前向探索、多样性增强和可行性利用, 并结合自适应环境选择和基于进化代数的算子选择策略, 有效平衡了收敛性和多样性。实验结果表明, HMSC-EC 在多个基准测试问题上表现优异, 特别是在处理复杂约束时具有显著优势。

关键词

约束多目标优化, 进化算法, 多档案机制, 自适应策略, Pareto 前沿

Hybrid Multi-Strategy Constrained Evolutionary Algorithm for Multi-Objective Optimization

Xuanyu Lai

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

Received: February 23, 2026; accepted: March 18, 2026; published: March 25, 2026

Abstract

Constrained Multi-Objective Optimization Problems (CMOPs) are prevalent in engineering practice. A core challenge in this field is how to obtain a well-distributed Pareto front while satisfying all constraints. This paper proposes a Hybrid Multi-Strategy Constraint handling based Evolutionary algorithm for Constrained multi-objective optimization (HMSC-EC). This algorithm employs a three-archive collaborative mechanism, responsible for forward exploration, diversity enhancement, and feasibility exploitation, respectively. It integrates adaptive environmental selection and generation-

dependent operator selection strategies to effectively balance convergence and diversity. Experimental results demonstrate that HMSC-EC performs excellently on multiple benchmark problems, showing significant advantages particularly when dealing with complex constraints.

Keywords

Constrained Multi-Objective Optimization, Evolutionary Algorithm, Multi-Archive Mechanism, Adaptive Strategy, Pareto Front

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

1.1. 研究背景

多目标优化问题 MOPs (Multi-Objective Optimization Problems) 广泛存在于工程设计、资源调度、生产规划等实际应用领域。与单目标优化不同, MOP 需要同时优化多个相互冲突的目标函数, 这使得不存在单一的最优解, 而是存在一组权衡解, 即 Pareto 最优解集。这些解在目标空间形成的曲面称为 Pareto 前沿, PF (Pareto Front)。

在实际工程应用中, 优化问题往往不仅涉及多个目标, 还伴随着各种约束条件, 如资源限制、物理定律、安全规范等。这类问题被称为约束多目标优化问题 CMOPs (Constrained Multi-Objective Optimization Problems)。CMOPs 的数学形式可表示为:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_M(x))^T \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j = 1, 2, \dots, J \\ & h_k(x) = 0, \quad k = 1, 2, \dots, K \\ & x \in \Omega \end{aligned} \quad (1)$$

其中, $x = (x_1, x_2, \dots, x_n)^T$ 是 n 维决策变量向量, $\Omega \subseteq R^n$ 是决策空间, $F: \Omega \rightarrow R^M$ 是目标函数向量, M 是目标函数数量, $g_j(x)$ 和 $h_k(x)$ 分别表示 J 个不等式约束和 K 个等式约束。问题的可行域定义为满足所有约束条件的决策变量集合。

CMOPs 的求解难度远高于无约束 MOPs, 主要原因在于: 约束条件的存在使得可行域可能非常狭小、不连续或具有复杂的几何形状, 这给搜索带来了巨大挑战; 其次, 约束条件与目标函数之间可能存在复杂的相互作用, 使得算法需要在满足约束和优化目标之间进行权衡; 并且不同类型的约束(如线性/非线性、等式/不等式)需要不同的处理策略。

多目标进化算法, MOEAs (Multi-Objective Evolutionary Algorithms) 因其基于种群的搜索机制和对问题特性的低依赖性, 已成为求解 MOPs 的主流方法。经典的 MOEAs 如 NSGA-II、SPEA2、MOEA/D 等, 在无约束问题上取得了显著成功。然而, 将这些算法直接应用于 CMOPs 往往效果不佳, 因为它们缺乏有效的约束处理机制。因此, 设计专门针对 CMOPs 的约束多目标进化算法 CMOEAs (Constrained Multi-Objective Evolutionary Algorithms) 成为该领域的重要研究方向。

1.2. 研究动机

尽管近年来多目标进化算法在求解复杂约束多目标优化问题方面取得了显著进展, 但仍面临诸多挑

战。在无约束多目标优化问题中,算法需在收敛性与多样性之间实现平衡。而在 CMOPs 中,这一平衡的实现更为困难。一方面,约束条件对搜索空间形成限制,易使算法陷入局部可行域;另一方面,过度关注可行性可能导致种群多样性缺失,使算法过早收敛至局部最优。传统基于帕累托支配的选择机制在处理约束时,往往倾向于可行解,这可能致使算法在进化早期就放弃对不可行区域的探索,进而错过通向全局最优的路径。当问题的可行域相较于整个搜索空间极为狭小时,随机初始化的种群可能几乎全为不可行解。在此情形下,算法需先找到可行域,方可进行有效优化。然而,传统的约束处理方法通常假设种群中存在一定数量的可行解,这在可行域狭小的问题中并不适用。此外,当可行域不连续或呈现复杂几何形状时,算法需在多个可行区域间进行探索,这对算法的全局搜索能力提出了更高要求。现有的约束处理方法主要涵盖罚函数法、可行性规则、 ε -约束法等。罚函数法需要精心设计罚函数并调整罚因子,而这些参数往往依赖于问题特性,难以自动确定。可行性规则虽简便有效,但在可行解稀缺时可能导致搜索停滞。 ε -约束法通过动态调整约束容忍度来平衡可行性与收敛性,但其性能高度依赖于 ε 的衰减策略。更为关键的是,这些方法大多采用单一的约束处理策略,难以满足进化过程不同阶段的需求。在进化早期,算法需探索整个搜索空间以发现可行域;在进化中期,需在可行域边界进行探索以寻求更优解;在进化后期,则需在可行域内开展精细搜索以提高收敛精度。多档案机制通过维护多个子种群来达成不同的进化目标,已被证实是提升 CMOEAs 性能的有效途径。然而,现有的多档案方法存在以下不足:其一,档案间的协同机制不够完善,各档案往往独立进化,缺乏信息交流;其二,档案的更新策略较为固定,难以依据进化状态进行动态调整;其三,档案的资源分配通常是静态的,无法根据问题特性和进化阶段进行自适应调整。

基于上述挑战,本文旨在设计一种新型的 CMOEa,通过以下机制提升算法性能:采用三档案协同框架,分别承担前向探索、多样性维持和可行性利用的任务;引入自适应环境选择机制,依据种群状态动态切换选择策略;设计基于进化代数的算子选择机制,平衡全局搜索和局部开发;提出基于角度距离的多样性维持机制,有效处理复杂约束下的解分布问题。

1.3. 主要贡献

针对上述挑战,本研究提出了一种基于混合多策略约束处理的多目标进化算法, HMSC-EC (Hybrid Multi-Strategy Constraint Handling Based Evolutionary Algorithm for Constrained Multi-Objective Optimization)。

1.3.1. 三档案协同进化框架

本文提出了一种新颖的三档案协同机制,包括前向探索档案 FA (Forward exploration Archive)、多样性增强档案 DA (Diversity enhancement Archive)和可行性利用档案 FEA (Feasibility Exploitation Archive)。三个档案分工明确、相互协作。FA 专注于目标空间的探索,维护当前发现的非支配解集,采用基于 Pareto 支配的筛选和拥挤距离的截断策略,为算法提供收敛方向的指引。FA 不考虑约束条件,允许不可行解的存在,这使得算法能够探索可行域边界附近的潜在优质解。DA 负责维持种群的多样性,采用基于参考向量和角度距离的选择机制。通过在目标空间中均匀分布参考向量,DA 确保解集能够覆盖整个 Pareto 前沿。特别地,DA 采用约束 Pareto 支配关系,在保证多样性的同时逐步引导种群向可行域移动。FEA 作为算法的主种群,专注于可行解的开发和利用。FEA 采用约束 Pareto 支配进行筛选,优先保留可行解。当可行解充足时,FEA 使用自适应环境选择策略进行截断;当可行解不足时,则补充约束违反度最小的不可行解,以维持种群规模并保持向可行域的搜索压力。

三个档案通过共享后代种群实现信息交流,并共同参与交配池的构建,形成了一个有机的协同进化系统。这种设计使得算法能够同时兼顾收敛性、多样性和可行性,有效应对 CMOPs 的多重挑战。

1.3.2. 自适应环境选择机制

本文提出了一种基于种群状态的自适应环境选择机制。该机制通过监测 FEA 中非支配解的比例 ρ 来评估当前种群的进化状态，并据此动态选择合适的环境选择策略：当 $\rho < 0.3$ 时，表明种群收敛性不足，此时采用 SPEA2 的适应度分配与截断策略，以增强收敛压力；当 $\rho > 0.7$ 时，表明种群多样性可能不足，此时采用改进的 NSGA-III 策略，借助参考点机制提升多样性；当 $0.3 \leq \rho \leq 0.7$ 时，表明种群处于平衡状态，此时采用 NSGA-II 的拥挤距离策略，维持收敛性与多样性的平衡。

这种自适应机制使得算法能够根据进化过程的实际需求动态调整选择压力，避免了单一策略的局限性。与传统的固定策略相比，自适应环境选择能够更好地适应不同问题特性和不同进化阶段的需求。

1.3.3. 基于进化代数的算子选择策略

本文设计了一种基于进化进度的算子选择策略。差分进化，DE (Differential Evolution) 算子和遗传算法 GA (Genetic Algorithm) 算子各有优势：DE 算子具有较强的全局搜索能力，适合在进化早期进行大范围探索；GA 算子具有较好的局部开发能力，适合在进化后期进行精细搜索。

本文通过正弦函数动态调整 DE 算子的使用概率：

$$P_{DE} = 0.5 + 0.3 \cdot \sin\left(\pi \cdot \frac{FE}{\max FE}\right) \quad (2)$$

其中 FE 是当前函数评价次数， $\max FE$ 是最大评价次数。该策略具有以下特点：在进化初期 ($FE/\max FE \approx 0.25$)， P_{DE} 达到峰值 0.8，充分利用 DE 的全局搜索能力；在进化中期 ($FE/\max FE \approx 0.5$)， P_{DE} 回落到 0.5，平衡探索和开发；在进化后期 ($FE/\max FE \approx 0.75$)， P_{DE} 降至最低 0.2，主要依靠 GA 进行局部精细搜索。

这种动态调整策略使得算法能够在不同进化阶段采用最合适的搜索算子，提高了算法的整体性能。

1.3.4. 基于角度距离的多样性维持机制

针对约束条件下解分布不均的挑战，本文在多样性档案(DA)中引入了角度距离选择机制。该机制通过在目标空间预设一组均匀的参考向量，将复杂的分布控制转化为个体与向量间的夹角计算。在更新档案时，算法会为每个参考向量匹配一个角度距离最小且约束违反度最低的解，从而确保解集能够沿着预设的方向均匀排布。

与传统的拥挤距离相比，这种做法的逻辑优势在于它能够通过参考向量直接引导解的搜索方向，使分布控制更加直观。同时，由于角度反映的是个体间的相对位置关系，它对目标数值的大小范围并不敏感。这意味着即便不同目标之间的数值量级相差很大，算法也无需依赖复杂的归一化处理就能保持稳定的分布效果，有效避免了因目标量级失衡导致的筛选偏差。

而本文提出的自适应角度阈值机制：

$$h = \bar{h} \cdot \left(1 + \frac{N_{feas}}{N_s}\right) \quad (3)$$

能够根据可行解数量 N_{feas} 动态调整阈值 h 。当可行解较少时，阈值较小，选择压力较大，促使算法快速向可行域移动；当可行解充足时，阈值增大，选择压力减小，允许算法在更大范围内探索，从而维持多样性。

HMSC-EC 通过三档案协同、自适应环境选择、基于进化代数的算子选择和角度距离机制的有机结合，构建了一个灵活、高效的约束多目标优化框架。实验结果表明，HMSC-EC 在多个基准测试问题上表现优异，特别是在处理复杂约束和狭小可行域问题时具有显著优势。

2. 相关工作

2.1. 多目标进化算法

自 20 世纪 80 年代多目标进化算法(MOEAs)被提出以来, 该领域已逐渐发展出多种成熟的选择机制以应对复杂的优化需求。早期研究主要侧重于基于 Pareto 支配的方法, 其中 NSGA-II 作为代表性算法, 利用快速非支配排序将种群分层, 并结合无参数的拥挤距离机制维持解的多样性[1]。尽管该算法在处理双目标和三目标问题时表现出优异的鲁棒性, 但其单纯依赖距离的机制在复杂前沿形状上仍存在一定局限。为了进一步强化对复杂边界形态的刻画能力, SPEA2 采用了截然不同的适应度分配逻辑, 通过计算每个个体的强度值并结合 k 近邻密度估计来分配适应度, 同时采用严格的截断策略维持外部档案[2]。这种机制在保持边界解方面具有优势, 但计算复杂度较高。与此同时, 差分进化(DE)等强大的连续空间搜索算子被引入, 其基于向量差分的变异机制极大地提升了算法的局部非线性开采能力[3]。然而, 基于 Pareto 支配的方法在处理多目标($M > 3$)问题时面临严重的“选择压力不足”问题。随着目标数量的增加, 种群中大部分个体瞬间变为非支配解, 导致算法难以区分解的优劣, 收敛速度显著下降, 这一现象被称为“维数灾难”。为弥补支配机制在高维空间中的局限性, 基于分解的方法 MOEA/D 被提出[4]。该类算法通过权重向量将多目标问题转化为一系列单目标子问题进行协同求解, 利用切比雪夫等聚合函数引导搜索。这种机制将多样性的维持转化为对权重向量的分布控制, 不仅简化了多样性管理, 还通过相邻子问题间的信息共享大幅提升了搜索效率。但 MOEA/D 的性能高度依赖于权重向量的设置, 均匀分布的权重在具有复杂 Pareto 前沿形状的问题上可能产生资源错配。与此同时, 基于指示器的方法通过引入超体积(HV)或反转世代距离(IGD)等性能指标直接指导演化过程。此类算法在理论上与 Pareto 支配严格兼容, 能够同时兼顾解集的收敛性与均匀性。然而, 超体积的计算复杂度随目标数量呈指数级增长, 限制了其在超多目标优化中的广泛应用。针对高维目标空间中选择压力匮乏的挑战, 基于参考点的方法应运而生。NSGA-III 及其衍生框架摒弃了单纯依靠拥挤距离的做法, 转而利用预设的结构化参考点引导解的分布, 通过计算个体与参考点的关联性来确保种群在超平面上的覆盖度[5]。后续研究进一步探索了自适应参考点调整策略, 以期在非规则 Pareto 前沿上实现更优的拟合。Tian 等人构建了集成海量主流演化算法、基准测试集及性能评价指标的开源平台 PlatEMO, 为多目标进化计算领域的算法复现与性能公平对比提供了标准化的基准环境[6]。

2.2. 约束处理技术

作为约束多目标进化算法(CMOEAs)的核心逻辑, 约束处理技术决定了算法在跨越不可行域时, 于可行性与目标最优性之间取得平衡的能力。在众多传统演化策略中, 罚函数方法通过将约束违反度(CV)整合至目标函数中, 以代数惩罚降低不可行解的竞争力。静态罚函数实现简单, 但高度依赖罚因子的设定: 因子过小难以形成有效约束, 过大则极易导致位于可行域边界的优质个体因轻微违约而被淘汰, 进而引发早熟收敛。为了克服对参数的依赖, 可行性规则提供了一种直观的比较准则: 在可行解与不可行解的比对中, 赋予可行解绝对的生存优先级。这种无参数设计提升了易用性, 但也带来了潜在风险。一方面, 严苛的判别界限可能导致搜索轨迹的不连续; 另一方面, 在进化初期可行解极其稀缺时, 该规则易导致种群迅速向少数局部可行解聚拢。此外, 完全忽略不可行解的目标信息, 使算法失去了利用“桥梁个体”跨越搜索障碍的机会。相比之下, ϵ -约束方法引入了一个动态调整的约束容忍度阈值 $\epsilon(t)$ 。随着进化的进行, 该阈值逐渐减小并最终收敛到 0。这种方法在进化早期允许一定程度的违约以扩大搜索空间, 在后期逐步收紧网络引导种群向可行域移动。然而, 衰减速率的设置深刻依赖于具体优化问题的约束景观特性。针对收敛稳定性的需求, 随机排序方法以固定的概率仅考虑目标函数值进行比较, 通过引入随机性来自动平衡可行域探测与目标寻优。多目标化方法则将总体约束违反度作为额外的目标函数, 将 CMOP

转化为无约束 MOP。尽管可以直接复用成熟的 MOEAs，但这增加了目标维度，容易引发选择压力不足和原始目标之间的错综冲突。近年来，随着工程应用复杂度的剧增，研究者在上述基础技术之上，提出了一系列针对复杂约束景观的高级系统化演化架构。在多阶段与推拉搜索机制方面，Fan 等人提出了推拉搜索(PPS)框架[7]，通过在前期完全无视约束进行“推”搜索，并在后期利用自适应 ϵ -机制“拉”回可行域，有效跨越了巨大的不可行障碍；Ming 等人设计了一种简单的双阶段进化算法(C-TSEA) [8]，在第一阶段全力搜索可行域边界并在第二阶段激活多目标协同优化；针对具有海量异构约束的问题，Sun 等人提出了一种多阶段渐进式约束注入算法(C3M) [9]，避免了由多重硬约束碰撞导致的种群全军覆没。在协同进化与双种群机制方面，Tian 等人提出了基于协同进化框架的 CCMO 算法[10]，通过并行的受限主种群与无约束辅助种群之间的超级后代池共享机制，极大提升了跳出局部可行岛屿的能力；Liu 等人则提出了双向协同进化框架(BiCo) [11]，从可行侧的精细推延与不可行侧的持续高压进行双向施压；此外，Liu 等人还提出了一种基于约束 - 帕累托支配与多样性增强策略的进化算法[12]，以精细控制前沿断裂边缘的种群分布。针对多子种群架构带来的算力分配挑战，Tian 等人设计了一种面向多群体 CMOPs 的自适应种群规模调控机制[13]，根据进化状态动态缩减或扩张子种群，显著提升了资源利用率。此外，随着对高维认知边界的拓展，Ming 等人将进化多任务学习与跨域知识迁移(CMT)引入该领域[14]，通过构建辅助任务并实施隐式拓扑迁移，实现了从不可行域提取进化知识的降维打击。最后，为了填补理论基准与真实工业界域之间的断层，Qiao 等人设计了一套具备多峰态、高度非线性特征的可扩展高维约束基准测试集及配套搜索算法[15]，为检验现代 CMOEAs 在高维约束空间中的真实破壁能力提供了极其重要的理论评测标准。

3. 算法思想

3.1. 算法框架

HMSC-EC 的核心思想是通过三个功能互补的档案协同进化，在满足约束条件的同时实现收敛性和多样性的平衡。算法的伪代码如图 1 所示：

```

Algorithm 1: HMSC-EC
Input: 种群规模  $N$ , 最大评估次数  $\max FE$ , 环境选择参数  $e_1, e_2$ 
Output: 可行性利用档案 FEA
1  $N_s \leftarrow \lfloor N/3 \rfloor$ ; // 每个档案的大小
2  $\text{Population} \leftarrow$  初始化种群;
3  $\text{FA} \leftarrow \emptyset$ ; // 前向探索档案
4  $\text{DA} \leftarrow \emptyset$ ; // 多样性增强档案
5  $\text{FEA} \leftarrow \text{Population}$ ; // 可行性利用档案
6  $\text{Offspring} \leftarrow \text{Population}$ ;
7  $z_{\min} \leftarrow \min(\text{Population.objs}) - 10^{-6}$ ;
8 while 未达到终止条件 do
    // 更新参考点
    9  $z_{\min} \leftarrow \min(z_{\min}, \min(\text{Offspring.objs}) - 10^{-6})$ ;
    // 更新三个档案
    10  $\text{FA} \leftarrow$  更新前向探索档案 ( $\text{FA}, \text{Offspring}, z_{\min}, N_s, e_1$ );
    11  $\text{DA} \leftarrow$  更新多样性增强档案 ( $\text{DA}, \text{Offspring}, z_{\min}, N_s$ );
    12  $\text{FEA} \leftarrow$  更新可行性利用档案 ( $\text{FEA}, \text{Offspring}, N, e_2$ );
    // 构建交配池
    13  $\text{Pop}_1 \leftarrow \text{FA}$ ;
    14  $\text{Pop}_2 \leftarrow \text{DA}$ ;
    15  $\text{Pop}_3 \leftarrow$  从 FEA 中随机选择  $\lfloor N/3 \rfloor$  个个体;
    // 自适应算子选择
    16  $\text{progress} \leftarrow FE / \max FE$ ;
    17  $p_{DE} \leftarrow 0.5 + 0.3 \times \sin(\pi \times \text{progress})$ ;
    18 if  $\text{rand}() < p_{DE}$  then
    19 |  $\text{Offspring} \leftarrow \text{DE}(\text{Pop}_1), \text{DE}(\text{Pop}_2), \text{DE}(\text{Pop}_3)$ ;
    20 else
    21 |  $\text{Offspring} \leftarrow \text{GA}(\text{Pop}_1), \text{GA}(\text{Pop}_2), \text{GA}(\text{Pop}_3)$ ;
    // 自适应环境选择策略
    22  $\text{NonDom} \leftarrow$  计算 FEA 中的非支配解;
    23  $\text{ratio} \leftarrow \sum \text{NonDom} / |\text{FEA}|$ ;
    24 if  $\text{ratio} < 0.3$  then
    25 |  $e_1 \leftarrow 1; e_2 \leftarrow 1$ ; // 使用 SPEA2
    26 else if  $\text{ratio} > 0.7$  then
    27 |  $e_1 \leftarrow 3; e_2 \leftarrow 3$ ; // 使用改进 NSGA-III
    28 else
    29 |  $e_1 \leftarrow 2; e_2 \leftarrow 2$ ; // 使用 NSGA-II
30 return FEA;
  
```

Figure 1. Pseudocode for the HMSC-EC Algorithm

图 1. HMSC-EC 算法伪代码

3.2. 三档案协同机制

3.2.1. 前向探索档案(FA)

FA 的设计目标是探索目标空间, 发现潜在的优质解方向。FA 采用无约束的 Pareto 支配关系, 允许不可行解的存在, 这使得算法能够探索可行域边界附近的区域。

FA 的更新过程如图 2 所示:

Algorithm 2: 更新前向探索档案

Input: FA, Offspring, $z_{\min}$, N_s , Add
Output: 更新后的 FA

```

1 FA  $\leftarrow$  FA  $\cup$  Offspring;
2 NonDom  $\leftarrow$  通过帕累托支配计算非支配解;
3 FA  $\leftarrow$  FA[NonDom];
4 if  $|FA| \leq N_s$  then
5   通过随机重复填充至  $N_s$  个个体;
6 else
7   PopObj  $\leftarrow$  (FA.objs -  $z_{\min}$ ) / ( $z_{\max} - z_{\min} + 10^{-6}$ );
8   CrowdDis  $\leftarrow$  计算拥挤距离 (PopObj);
9   按拥挤距离降序排列, 选择前  $N_s$  个个体;
10 return FA;
```

Figure 2. Pseudocode for FA file update

图 2. FA 档案更新伪代码

3.2.2. 多样性增强档案(DA)

DA 的设计目标是维持种群的多样性, 确保解能够均匀分布在 Pareto 前沿上。DA 采用基于参考向量和角度距离的选择机制, 这种机制在处理复杂 Pareto 前沿形状时比传统的拥挤距离更有效。

DA 的更新过程如图 3 所示:

Algorithm 3: 更新多样性增强档案

Input: DA, Offspring, $z_{\min}$, N_s
Output: 更新后的 DA

```

1 S  $\leftarrow$  DA  $\cup$  Offspring;
2 DA  $\leftarrow$   $\emptyset$ ;
3 NonDom  $\leftarrow$  通过约束帕累托支配计算非支配解;
4 S  $\leftarrow$  S[NonDom];
5 CV  $\leftarrow$   $\sum \max(0, S.cons)$ ; // 约束违反值
6 // 计算角度距离
7 W  $\leftarrow$  生成  $N_s$  个均匀分布参考向量;
8  $\theta_{W,W} \leftarrow \arccos(1 - \text{余弦距离}(W, W))$ ;
9  $h \leftarrow \text{mean}(\min(\theta_{W,W}))$ ;
10 // 自适应调整 h
11  $n_f \leftarrow \sum (CV \leq 0)$ ;
12 if  $n_f > 0$  then
13    $h \leftarrow h \times (1 + n_f / N_s)$ ;
14 归一化目标函数值;
15  $\theta_{S,W} \leftarrow \sin(\arccos(1 - \text{余弦距离}(W, S.objs)))$ ;
16 for  $i = 1$  to  $N_s$  do
17   list  $\leftarrow$  找到与第  $i$  个参考向量角度距离  $\leq h$  的解;
18   if  $|list| = 0$  then
19     选择角度距离最小的解加入 list;
20   根据约束违反值和角度距离选择最优解加入 DA;
21 return DA;
```

Figure 3. Pseudocode for DA file update

图 3. DA 档案更新伪代码

3.2.3. 可行性利用档案(FEA)

FEA 作为算法的主种群, 专注于可行解的开发和利用。FEA 采用约束 Pareto 支配进行筛选, 优先保留可行解, 并使用自适应环境选择策略进行截断, 具体档案更新过程如图 4 所示。

Algorithm 4: 更新可行性利用档案

Input: FEA, Offspring, N , Add
Output: 更新后的 FEA

```

1 FEA  $\leftarrow$  FEA  $\cup$  Offspring;
2 NonDom  $\leftarrow$  通过约束帕累托支配计算非支配解;
3 FEA  $\leftarrow$  FEA[NonDom];
4 CV  $\leftarrow \sum \max(0, \text{FEA.cons})$ ;
5 feasible  $\leftarrow (CV \leq 0)$ ;
6 if  $\sum \text{feasible} \geq N$  then
7   FEA  $\leftarrow$  FEA[feasible];
8   if  $|\text{FEA}| > N$  then
9     switch Add do
10      case 1 do
11        | 通过 SPEA2 截断函数删除多余个体;
12      case 2 do
13        | 通过 NSGA-II 拥挤距离选择前  $N$  个个体;
14      case 3 do
15        | 使用改进 NSGA-III 参考向量选择  $N$  个个体;
16 else
17   按约束违反值升序排列不可行解;
18   FEA  $\leftarrow$  可行解  $\cup$  前  $(N - \sum \text{feasible})$  个不可行解;
19 return FEA;
```

Figure 4. Pseudocode for FEA file update**图 4.** FEA 档案更新伪代码

3.3. 自适应策略

3.3.1. 基于进化代数的算子选择

进化算子的选择对算法性能有重要影响。不同的算子适用于不同的搜索阶段：差分进化算子具有较强的全局搜索能力，适合在进化早期进行大范围探索；遗传算法算子具有较好的局部开发能力，适合在进化后期进行精细搜索。

DE 算子使用 DE/rand/1/bin 策略，对于个体 $\mathbf{x}_i$ ，生成后代的过程为：随机选择三个互不相同的个体 $\mathbf{x}_{r1}, \mathbf{x}_{r2}, \mathbf{x}_{r3}$ ，生成变异向量： $\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3})$ ，进行二项交叉：对于每个维度 j ，

$$u_{i,j} = \begin{cases} v_{i,j}, & \text{if } r \text{ and } () < \text{CR or } j = j_{\text{rand}} \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (4)$$

其中 F 是缩放因子，CR 是交叉概率， j_{rand} 是随机选择的维度。

GA 算子使用模拟二进制交叉(SBX)和多项式变异(PM)：选择两个父代 x_1, x_2 ，然后进行 SBX 交叉：对于每个维度 j ，

$$\begin{aligned} c_{1,j} &= 0.5 \left[(1 + \beta_j) x_{1,j} + (1 - \beta_j) x_{2,j} \right] \\ c_{2,j} &= 0.5 \left[(1 - \beta_j) x_{1,j} + (1 + \beta_j) x_{2,j} \right] \end{aligned} \quad (5)$$

其中 β_j 由分布指数 η_c 控制，进行 PM 变异：对于每个维度 j ，其中 δ_j 由分布指数 η_m 控制。

$$x'_j = x_j + \delta_j \cdot (x_j^{\text{upper}} - x_j^{\text{lower}}) \quad (6)$$

本文提出基于正弦函数的基于进化代数的算子选择策略：

$$P_{\text{DE}}(t) = 0.5 + 0.3 \cdot \sin\left(\pi \cdot \frac{\text{FE}}{\max \text{FE}}\right) \quad (7)$$

其中 FE 是当前函数评价次数， $\max \text{FE}$ 是最大评价次数，该策略具有以下特点。

初期探索 ($\text{FE}/\max \text{FE} \in [0, 0.25]$)： P_{DE} 从 0.5 增加到 0.8，DE 算子占主导，充分利用其全局搜索能力，快速探索搜索空间。

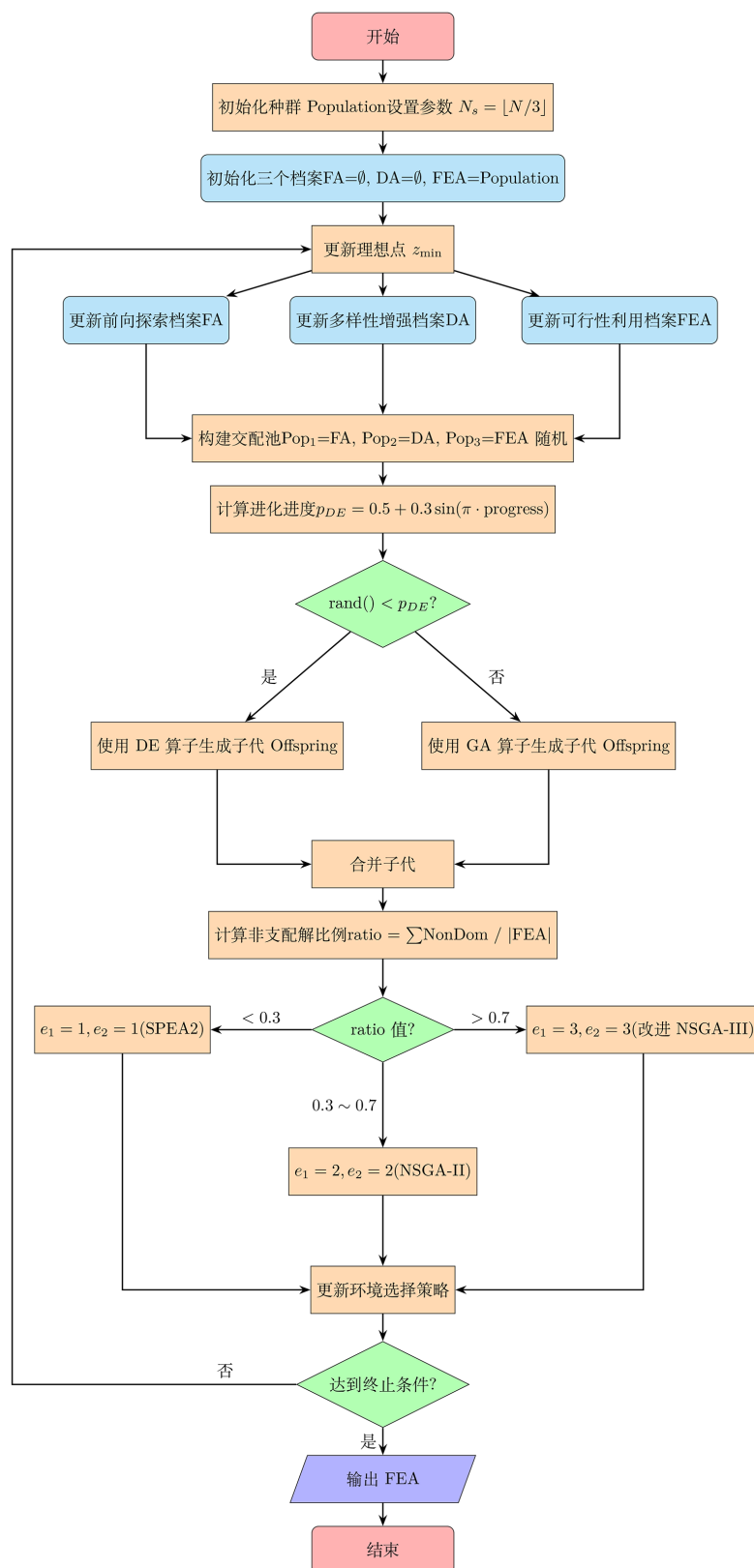


Figure 5. Flowchart of the HMSC-EC algorithm
图 5. HMSC-EC 算法流程图

中期平衡($FE/\max FE \in [0.25, 0.75]$): P_{DE} 先从 0.8 降至 0.5, 再从 0.5 降至 0.2, DE 和 GA 的使用逐渐平衡, 然后 GA 逐渐占主导, 实现从探索到开发的平滑过渡。

后期开发($FE/\max FE \in [0.75, 1.0]$): P_{DE} 从 0.2 回升到 0.5, 主要依靠 GA 进行局部精细搜索, 同时保留一定的 DE 算子使用以避免过早收敛。

这种动态调整策略使得算法能够在不同进化阶段采用最合适的搜索算子, 提高了算法的整体性能。相比固定概率或线性调整, 正弦函数提供了更平滑的过渡和更灵活的控制。

3.3.2. 自适应环境选择

环境选择策略决定了哪些个体能够生存到下一代, 直接影响算法的收敛性和多样性。不同的选择策略各有优劣, 适用于不同的种群状态。

种群状态评估: 使用 FEA 中非支配解的比例 ρ 来评估种群状态:

$$\rho = \frac{|ND(FEA)|}{|FEA|} \quad (8)$$

其中 $ND(FEA)$ 是 FEA 中的非支配解集。 ρ 反映了种群的进化状态, 当 ρ 较低时: 表明种群中大部分解被其他解支配, 收敛性不足, 需要增强选择压力; 当 ρ 较高时: 表明种群中大部分解互不支配, 可能存在多样性不足或收敛到局部最优的风险; 当 ρ 适中时: 表明种群处于平衡状态, 收敛性和多样性都较好。

因此我们根据种群的进化状态制定不同的策略选择规则: 当 $\rho < 0.3$ 时, 使用 SPEA2 策略, 原因是 SPEA2 通过适应度分配机制, 对被支配的解施加较大的选择压力, 并且适应度计算考虑了个体的强度值和密度信息, 其截断策略优先删除适应度较差的个体, 加速收敛。适用于种群收敛性不足, 需要快速向 Pareto 前沿逼近的情况; 当 $\rho > 0.7$ 时, 使用改进的 NSGA-III 策略: 原因是 NSGA-III 使用参考点机制引导解的分布, 通过将个体关联到参考点, 确保解在目标空间中均匀分布, 并且优先保留关联个体较少的参考点的个体, 增强多样性, 适用场于种群多样性不足或过早收敛, 需要扩大搜索范围的情况; 当 $0.3 \leq \rho \leq 0.7$ 时, 使用 NSGA-II 策略: 原因是 NSGA-II 使用拥挤距离维持多样性, 在非支配排序的基础上, 优先保留拥挤距离大的个体, 同时平衡了收敛性和多样性, 适合种群处于良好状态时使用。

3.4. HMSC-EC 算法流程图

完整的 HMSC-EC 算法流程图如图 5 所示。

4. 实验研究

4.1. 测试问题

为了全面评估 HMSC-EC 的性能, 本文选择了两个广泛使用的约束多目标优化基准测试集, 共包含 23 个测试问题: LIRCMOP 测试集、DASCMOP 测试集。这些测试问题涵盖了双目标和三目标优化场景, 决策变量维度从 6 维到 30 维不等。四个测试集覆盖了约束多目标优化的多种典型场景, 能够全面评估算法在不同难度和特征问题上的性能。

4.2. 实验指标

评估 CMOEAs 的性能需要同时考虑收敛性、多样性和可行性。本文采用以下两个广泛使用的性能指标: HV 超体积指标衡量解集支配的目标空间体积、IGD+指标衡量算法获得的解集与真实 Pareto 前沿的距离。HV 值越大, 表示算法性能越好。HV 是唯一严格单调的 Pareto 兼容指标, 即如果解集 P_1 支配解集 P_2 , 则必有 $HV(P_1) > HV(P_2)$ 。IGD+同时反映了收敛性和多样性, 如果解集未收敛到 Pareto 前沿, 距离

会较大；如果解集分布不均匀，某些参考点的距离会较大。

4.3. 实验结果

HMSC-EC 在 2 个约束多目标优化基准测试集，23 个测试函数上与 6 个当前主流的约束多目标算法进行了对比测试。以下表 1、表 2 分别为算法在 LIRCMOP 的 IGD+/HV 测试结果。

Table 1. Comparison of IGD+ indicators between HMSC-EC and other mainstream algorithms on the LIRCMOP
表 1. HMSC-EC 与其他主流算法在 LIRCMOP 测试集的 IGD+指标比较

Problem	MTCMO	CTSEA	CCMO	PPS	APSEA	CMOEAD	HMSC_EC
LIRCMOP1	1.2234e-1 (1.72e-2) -	2.3833e-1 (4.53e-2) -	2.1376e-1 (4.90e-2) -	2.0028e-2 (1.46e-2) =	2.3379e-1 (2.66e-2) -	2.3901e-1 (2.16e-2) -	1.7901e-2 (2.24e-3)
LIRCMOP2	7.3626e-2 (1.04e-2) -	1.6189e-1 (3.48e-2) -	1.3771e-1 (3.48e-2) -	1.7603e-2 (1.57e-2) +	1.5352e-1 (2.27e-2) -	1.4449e-1 (1.90e-2) -	2.1949e-2 (5.53e-3)
LIRCMOP3	1.2892e-1 (2.38e-2) -	2.5905e-1 (2.92e-2) -	2.2752e-1 (3.59e-2) -	5.7638e-2 (5.56e-2) -	2.3801e-1 (2.55e-2) -	2.3549e-1 (3.35e-2) -	1.2763e-2 (6.83e-3)
LIRCMOP4	8.8764e-2 (1.56e-2) -	1.8743e-1 (2.09e-2) -	1.6631e-1 (3.38e-2) -	3.7233e-2 (4.63e-2) =	1.7498e-1 (2.60e-2) -	1.6024e-1 (3.08e-2) -	1.8082e-2 (5.54e-3)
LIRCMOP5	7.8993e-1 (4.92e-1) -	2.3667e-1 (4.24e-2) -	2.1399e-1 (5.77e-2) -	6.7852e-3 (1.31e-3) +	9.8598e-1 (4.22e-1) -	1.3273e+0 (3.40e-1) -	1.3538e-2 (2.54e-3)
LIRCMOP6	1.0058e+0 (4.89e-1) -	2.6945e-1 (3.95e-2) -	2.4187e-1 (5.83e-2) -	5.0649e-2 (2.45e-1) -	1.0787e+0 (4.51e-1) -	1.3466e+0 (8.21e-4) -	1.2556e-2 (2.24e-3)
LIRCMOP7	1.0279e-1 (2.10e-2) -	1.1032e-1 (2.23e-2) -	1.0475e-1 (2.07e-2) -	1.0684e-1 (1.55e-2) -	1.0872e-1 (3.10e-2) -	1.6315e+0 (2.86e-1) -	1.3645e-2 (1.18e-2)
LIRCMOP8	1.5388e-1 (4.55e-2) -	1.7658e-1 (3.63e-2) -	1.3722e-1 (4.31e-2) -	1.4626e-1 (5.78e-2) -	1.4737e-1 (4.47e-2) -	1.6830e+0 (6.73e-4) -	1.2501e-2 (8.33e-3)
LIRCMOP9	7.4804e-1 (1.76e-1) -	5.6428e-1 (1.57e-1) -	3.6223e-1 (1.22e-1) -	2.1566e-1 (9.10e-2) -	7.7297e-1 (1.75e-1) -	8.3836e-1 (1.49e-1) -	6.9216e-2 (6.15e-3)
LIRCMOP10	6.5208e-1 (2.62e-1) -	3.4193e-1 (1.53e-1) -	1.3202e-1 (5.15e-2) -	2.0417e-1 (1.09e-1) -	7.1621e-1 (2.25e-1) -	7.7901e-1 (1.31e-1) -	1.7593e-2 (3.13e-3)
LIRCMOP11	4.4831e-1 (2.02e-1) -	2.2311e-1 (1.24e-1) -	5.0008e-2 (2.23e-2) -	2.6267e-1 (1.63e-1) -	4.5791e-1 (2.15e-1) -	8.2702e-1 (1.14e-1) -	7.6341e-3 (3.22e-3)
LIRCMOP12	3.8573e-1 (1.25e-1) -	3.3622e-1 (1.08e-1) -	1.5573e-1 (8.08e-2) -	1.3348e-1 (6.31e-2) -	3.4180e-1 (2.18e-1) -	5.5413e-1 (1.68e-1) -	1.3852e-2 (6.54e-3)
LIRCMOP13	1.3145e+0 (1.75e-3) -	4.3960e-2 (1.65e-3) -	4.4801e-2 (1.27e-3) -	1.6097e-1 (3.15e-1) -	1.3142e+0 (2.13e-3) -	1.3044e+0 (5.26e-4) -	4.0808e-2 (5.16e-4)
LIRCMOP14	1.2706e+0 (1.58e-3) -	4.5624e-2 (9.17e-4) -	4.6375e-2 (1.42e-3) -	6.2063e-2 (3.34e-3) -	1.2704e+0 (1.45e-3) -	1.2603e+0 (4.83e-4) -	4.1416e-2 (3.16e-4)
+/-/=	0/14/0	0/14/0	0/14/0	2/10/2	0/14/0	0/14/0	

Table 2. Comparison of HV indicators between HMSC-EC and other mainstream algorithms on the LIRCMOP
表 2. HMSC-EC 与其他主流算法在 LIRCMOP 测试集的 HV 指标比较

Problem	MTCMO	CTSEA	CCMO	PPS	APSEA	CMOEAD	HMSC_EC
LIRCMOP1	1.6522e-1 (1.03e-2) -	1.1666e-1 (1.91e-2) -	1.2568e-1 (1.81e-2) -	2.2698e-1 (1.03e-2) =	1.1871e-1 (1.16e-2) -	1.1641e-1 (8.77e-3) -	2.2610e-1 (2.41e-3)
LIRCMOP2	2.9050e-1 (1.01e-2) -	2.2988e-1 (2.35e-2) -	2.4724e-1 (2.46e-2) -	3.4922e-1 (1.66e-2) +	2.3513e-1 (1.55e-2) -	2.3952e-1 (1.22e-2) -	3.4586e-1 (4.38e-3)
LIRCMOP3	1.4691e-1 (1.16e-2) -	9.9869e-2 (1.21e-2) -	1.0892e-1 (1.46e-2) -	1.7971e-1 (2.42e-2) -	1.0600e-1 (9.69e-3) -	1.0688e-1 (1.20e-2) -	1.9859e-1 (3.90e-3)

续表

LIRCMOP4	2.5001e-1 (1.09e-2) -	1.9113e-1 (1.24e-2) -	2.0395e-1 (2.05e-2) -	2.9012e-1 (3.39e-2) =	1.9830e-1 (1.48e-2) -	2.0734e-1 (1.82e-2) -	3.0406e-1 (4.00e-3)
LIRCMOP5	6.4425e-2 (7.54e-2) -	1.4990e-1 (1.82e-2) -	1.6093e-1 (2.57e-2) -	2.9029e-1 (1.19e-3) +	3.5166e-2 (6.51e-2) -	0.0000e+0 (0.00e+0) -	2.8544e-1 (1.99e-3)
LIRCMOP6	3.1952e-2 (4.63e-2) -	1.0538e-1 (1.14e-2) -	1.1594e-1 (1.42e-2) -	1.8989e-1 (3.59e-2) -	2.3993e-2 (4.09e-2) -	0.0000e+0 (0.00e+0) -	1.9286e-1 (1.08e-3)
LIRCMOP7	2.4778e-1 (8.61e-3) -	2.4449e-1 (8.91e-3) -	2.4659e-1 (8.46e-3) -	2.4580e-1 (6.47e-3) -	2.4503e-1 (1.22e-2) -	8.0543e-3 (4.41e-2) -	2.8992e-1 (6.08e-3)
LIRCMOP8	2.3713e-1 (1.09e-2) -	2.3003e-1 (9.16e-3) -	2.3926e-1 (1.25e-2) -	2.3847e-1 (2.06e-2) -	2.3663e-1 (1.27e-2) -	0.0000e+0 (0.00e+0) -	2.9034e-1 (4.65e-3)
LIRCMOP9	1.9772e-1 (6.54e-2) -	2.9649e-1 (6.57e-2) -	3.6349e-1 (5.96e-2) -	4.3829e-1 (5.40e-2) -	1.9303e-1 (6.99e-2) -	1.4354e-1 (4.48e-2) -	5.3326e-1 (4.85e-3)
LIRCMOP10	2.2463e-1 (1.83e-1) -	4.1282e-1 (1.44e-1) -	6.2008e-1 (3.43e-2) -	5.7207e-1 (7.97e-2) -	1.7024e-1 (1.66e-1) -	1.3241e-1 (6.55e-2) -	6.9808e-1 (1.75e-3)
LIRCMOP11	3.8349e-1 (1.29e-1) -	5.2381e-1 (1.09e-1) -	6.6399e-1 (1.14e-2) -	4.8639e-1 (1.22e-1) -	3.8204e-1 (1.35e-1) -	2.1029e-1 (3.25e-2) -	6.8936e-1 (2.22e-3)
LIRCMOP12	4.3728e-1 (5.85e-2) -	4.6210e-1 (4.58e-2) -	5.3007e-1 (4.93e-2) -	5.3711e-1 (4.37e-2) -	4.3908e-1 (1.05e-1) -	3.3724e-1 (6.25e-2) -	6.1395e-1 (3.28e-3)
LIRCMOP13	9.2627e-5 (9.84e-5) -	5.5546e-1 (1.55e-3) -	5.5455e-1 (1.36e-3) -	4.7683e-1 (1.30e-1) -	1.3611e-4 (1.24e-4) -	4.3473e-4 (2.00e-5) -	5.5696e-1 (5.43e-4)
LIRCMOP14	4.9333e-4 (3.19e-4) -	5.5489e-1 (1.03e-3) -	5.5434e-1 (1.27e-3) -	5.3043e-1 (5.04e-3) -	5.2776e-4 (3.12e-4) -	9.7758e-4 (2.29e-5) -	5.5735e-1 (3.71e-4)
+/-/=	0/14/0	0/14/0	0/14/0	2/10/2	0/14/0	0/14/0	

算法运行过程中，我们分别截取了算法运行的前、中、后期的种群分布图，其中图 6 为 HMSC-EC 在 LIRCMOP6 问题上前期的种群分布图，可以看出在进化初期，FA 可以无视约束，快速逼近到 pareto 前沿，为种群提供潜在的优质解的方向；图 7 则为 HMSC-EC 中期的种群分布图，此时在 FA 的带领下，种群整体已经到达了 pareto 前沿，但此时种群缺乏多样性，需要进一步探索；图 8 为种群最终的收敛结果，可以看出，无论是种群的收敛性，亦或是多样性，都已经达到了非常不错的效果。

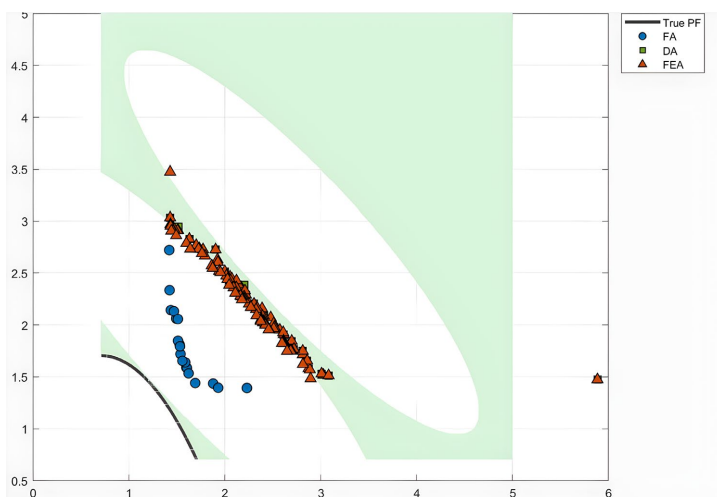


Figure 6. Initial population distribution of HMSC-EC on the LIRCMOP6 problem
图 6. HMSC-EC 在 LIRCMOP6 问题的前期种群分布图

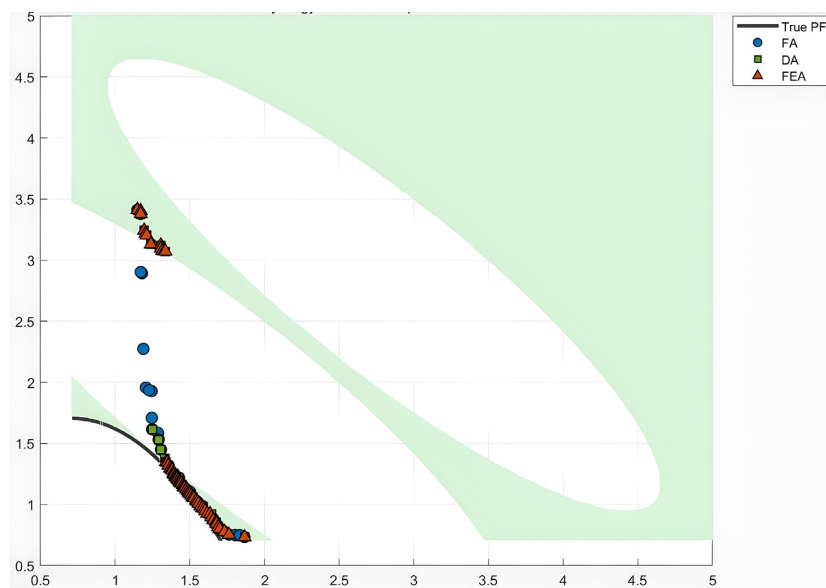


Figure 7. Mid-term population distribution of HMSC-EC on the LIRCMOP6 problem
图 7. HMSC-EC 在 LIRCMOP6 问题的中期种群分布图

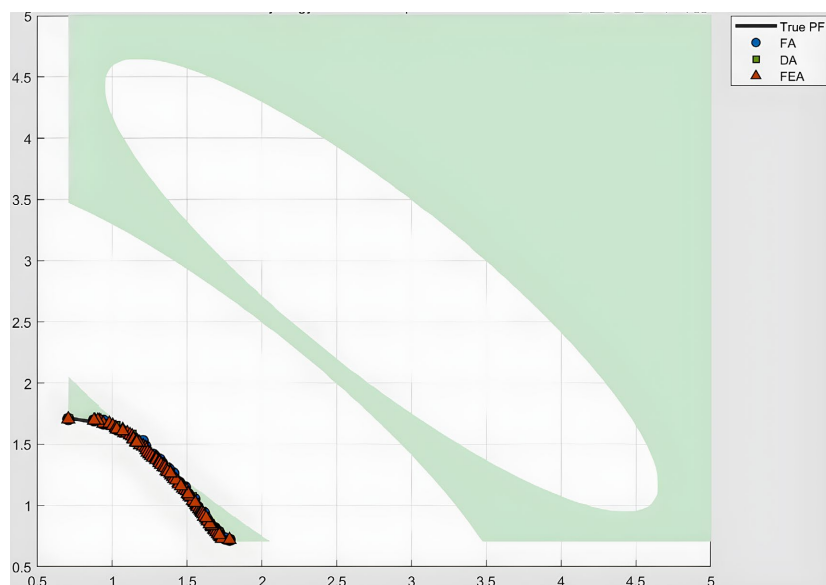


Figure 8. Late-stage population distribution of HMSC-EC on the LIRCMOP6 problem
图 8. HMSC-EC 在 LIRCMOP6 问题的后期种群分布图

算法在 DASCMP 测试集上基于 IGD+/HV 的测试结果如表 3、表 4 所示。

Table 3. Comparison of IGD+ indicators between HMSC-EC and other mainstream algorithms on the DASCMP
表 3. HMSC-EC 与其他主流算法在 DASCMP 测试集的 IGD+ 指标比较

Problem	MTCMO	CTSEA	CCMO	PPS	APSEA	CMOEA	HMSC_EC
DASCMP1	6.7003e-1 (7.23e-2) -	7.1293e-1 (5.05e-2) -	7.1701e-1 (3.22e-2) -	9.9098e-2 (1.72e-1) =	7.2739e-1 (2.97e-2) -	7.1209e-1 (3.41e-2) -	3.2064e-3 (2.94e-4)
DASCMP2	1.3396e-1 (9.68e-3) -	1.3991e-1 (1.00e-2) -	1.3931e-1 (8.48e-3) -	4.0488e-3 (2.01e-4) +	1.4829e-1 (8.97e-3) -	1.4650e-1 (1.15e-2) -	4.5950e-3 (1.58e-4)

续表

DASCMOP3	1.7029e-1 (2.44e-2) -	1.8395e-1 (1.91e-2) -	1.9250e-1 (1.79e-2) -	1.3094e-1 (7.88e-2) -	1.8339e-1 (2.54e-2) -	1.8843e-1 (1.21e-2) -	6.4348e-3 (6.60e-4)
DASCMOP4	8.3463e-4 (5.26e-4) +	1.1018e-3 (7.91e-4) +	1.1811e-3 (8.03e-4) +	2.4785e-1 (2.36e-1) -	1.0324e-2 (5.02e-2) -	2.0977e-1 (9.95e-2) -	1.6817e-3 (3.99e-4)
DASCMOP5	2.5841e-3 (4.48e-3) +	2.0984e-3 (1.87e-4) +	1.8512e-3 (7.28e-5) +	7.4091e-2 (1.86e-1) -	1.6246e-2 (7.73e-2) -	2.1334e-1 (2.40e-1) -	3.1734e-3 (4.45e-4)
DASCMOP6	1.7908e-1 (1.80e-1) -	1.7681e-2 (2.02e-2) =	1.2825e-2 (1.39e-2) =	3.4351e-1 (3.02e-1) -	3.0605e-1 (1.79e-1) -	5.4267e-1 (2.27e-1) -	5.8849e-3 (5.06e-4)
DASCMOP7	2.3097e-2 (8.11e-4) +	2.3922e-2 (9.24e-4) +	2.3452e-2 (7.76e-4) +	3.4100e-1 (2.87e-1) -	2.3142e-2 (6.37e-4) +	4.2182e-2 (1.89e-2) -	2.7762e-2 (1.23e-3)
DASCMOP8	1.8311e-2 (7.70e-4) +	1.8923e-2 (7.99e-4) +	1.8618e-2 (1.10e-3) +	2.0406e-1 (2.31e-1) -	1.8520e-2 (8.89e-4) +	1.0119e-1 (1.45e-1) -	2.2646e-2 (9.91e-4)
DASCMOP9	2.1965e-1 (4.38e-2) -	2.3988e-1 (3.56e-2) -	2.4646e-1 (3.44e-2) -	1.3416e-1 (8.15e-2) -	2.4071e-1 (3.74e-2) -	3.2548e-1 (1.09e-1) -	2.3795e-2 (9.75e-4)
+/-/=	4/5/0	4/4/1	4/4/1	1/7/1	2/7/0	0/9/0	

Table 4. Comparison of HV indicators between HMSC-EC and other mainstream algorithms on the DASCMOP

表 4. HMSC-EC 与其他主流算法在 DASCMOP 测试集的 HV 指标比较

Problem	MTCMO	CTSEA	CCMO	PPS	APSEA	CMOEAD	HMSC_EC
DASCMOP1	6.7003e-1 (7.23e-2) -	7.1293e-1 (5.05e-2) -	7.1701e-1 (3.22e-2) -	9.9098e-2 (1.72e-1) =	7.2739e-1 (2.97e-2) -	7.1209e-1 (3.41e-2) -	3.2064e-3 (2.94e-4)
DASCMOP2	1.3396e-1 (9.68e-3) -	1.3991e-1 (1.00e-2) -	1.3931e-1 (8.48e-3) -	4.0488e-3 (2.01e-4) +	1.4829e-1 (8.97e-3) -	1.4650e-1 (1.15e-2) -	4.5950e-3 (1.58e-4)
DASCMOP3	1.7029e-1 (2.44e-2) -	1.8395e-1 (1.91e-2) -	1.9250e-1 (1.79e-2) -	1.3094e-1 (7.88e-2) -	1.8339e-1 (2.54e-2) -	1.8843e-1 (1.21e-2) -	6.4348e-3 (6.60e-4)
DASCMOP4	8.3463e-4 (5.26e-4) +	1.1018e-3 (7.91e-4) +	1.1811e-3 (8.03e-4) +	2.4785e-1 (2.36e-1) -	1.0324e-2 (5.02e-2) -	2.0977e-1 (9.95e-2) -	1.6817e-3 (3.99e-4)
DASCMOP5	2.5841e-3 (4.48e-3) +	2.0984e-3 (1.87e-4) +	1.8512e-3 (7.28e-5) +	7.4091e-2 (1.86e-1) -	1.6246e-2 (7.73e-2) -	2.1334e-1 (2.40e-1) -	3.1734e-3 (4.45e-4)
DASCMOP6	1.7908e-1 (1.80e-1) -	1.7681e-2 (2.02e-2) =	1.2825e-2 (1.39e-2) =	3.4351e-1 (3.02e-1) -	3.0605e-1 (1.79e-1) -	5.4267e-1 (2.27e-1) -	5.8849e-3 (5.06e-4)
DASCMOP7	2.3097e-2 (8.11e-4) +	2.3922e-2 (9.24e-4) +	2.3452e-2 (7.76e-4) +	3.4100e-1 (2.87e-1) -	2.3142e-2 (6.37e-4) +	4.2182e-2 (1.89e-2) -	2.7762e-2 (1.23e-3)
DASCMOP8	1.8311e-2 (7.70e-4) +	1.8923e-2 (7.99e-4) +	1.8618e-2 (1.10e-3) +	2.0406e-1 (2.31e-1) -	1.8520e-2 (8.89e-4) +	1.0119e-1 (1.45e-1) -	2.2646e-2 (9.91e-4)
DASCMOP9	2.1965e-1 (4.38e-2) -	2.3988e-1 (3.56e-2) -	2.4646e-1 (3.44e-2) -	1.3416e-1 (8.15e-2) -	2.4071e-1 (3.74e-2) -	3.2548e-1 (1.09e-1) -	2.3795e-2 (9.75e-4)
+/-/=	4/5/0	4/4/1	4/4/1	1/7/1	2/7/0	0/9/0	

三档案协同机制是 HMSC-EC 的核心创新。FA、DA 和 FEA 分工明确，相互补充：FA 通过无约束探索发现潜在的优质解方向，避免算法过早陷入局部可行域，而 DA 通过角度距离机制维持解的多样性，确保解能够均匀分布在 Pareto 前沿上，FEA 专注于可行解的开发，通过自适应环境选择平衡收敛性和多样性。实验结果表明，HMSC-EC 在 LIRCMOP 测试集这种具有大不可行域问题上表现尤为出色，这正是三档案协同机制的优势所在。相比之下，单档案或双档案的对比算法在这类问题上往往难以找到可行域或收敛到局部最优。

两层自适应机制：算子选择根据进化进度动态调整 DE 和 GA 算子的使用概率，在进化早期充分探索，在进化后期精细开发，自适应环境选择根据种群状态动态切换 SPEA2/NSGA-II/NSGA-III 策略，适应不同的进化阶段需求，这种多层次的自适应机制使得 HMSC-EC 能够适应不同问题特性和不同进化阶段，表现出良好的鲁棒性。实验结果显示，HMSC-EC 在三个不同特性的测试集上都取得了优异的性能，证明了其广泛的适用性。

综合 HMSC-EC 在 LIRCMOP、DASCMOP、DOC 这三个基准测试集上的 HV 与 IGD+这两个指标的结果，HMSC-EC 在绝大多数的测试函数上取得了绝对的优势，即无论是解集支配的目标空间体积亦或是算法获得的解集与真实 Pareto 前沿的距离，HMSC-EC 算法都取得较大的优势，充分证明了 HMSC-EC 算法对于种群的多样性与收敛性都有相较于其他主流算法更强的优势，验证了 HMSC-EC 在严格约束下能够有效维持种群对于真实 Pareto 前沿的探索能力与种群个体间的差异性。

5. 结论

本文针对约束多目标优化问题，提出了一种基于混合多策略约束处理的多目标进化算法 HMSC-EC。算法的主要贡献总结如下：

1. 三档案协同进化框架：提出了一种新颖的三档案协同机制，包括前向探索档案 FA、多样性增强档案 DA 和可行性利用档案 FEA。三个档案分工明确、相互协作，有效平衡了收敛性、多样性和可行性。FA 通过无约束探索发现潜在优质解，DA 通过角度距离机制维持多样性，FEA 专注于可行解的开发。
2. 多层次自适应机制：引入了基于进化代数的算子选择和自适应环境选择两层机制。基于进化代数的算子选择通过正弦函数动态调整 DE 和 GA 算子的使用概率，在不同进化阶段采用最合适的搜索算子。自适应环境选择根据非支配解比例动态切换 SPEA2/NSGA-II/NSGA-III 策略，适应不同的进化需求。
3. 基于角度距离的多样性维持机制：在 DA 中引入了基于角度距离的选择机制，相比传统的拥挤距离，角度距离能够更精确地控制解的分布方向，对目标值的尺度不敏感。自适应角度阈值机制能够根据可行解数量动态调整选择压力，在维持多样性的同时逐步引导种群向可行域移动。

参考文献

- [1] 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>
- [2] Zitzler, E., Laumanns, M. and Thiele, L. (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK Report.
- [3] Das, S. and Suganthan, P.N. (2010) Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation*, **15**, 4-31. <https://doi.org/10.1109/tevc.2010.2059031>
- [4] Zhang, Q.F. and Li, H. (2007) MOEA/D: A Multi-Objective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, **11**, 712-731. <https://doi.org/10.1109/tevc.2007.892759>
- [5] Deb, K. and Jain, H. (2013) An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated 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>
- [6] 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>
- [7] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., et al. (2019) Push and Pull Search for Solving Constrained Multi-Objective Optimization Problems. *Swarm and Evolutionary Computation*, **44**, 665-679. <https://doi.org/10.1016/j.swevo.2018.08.017>
- [8] Ming, F., Gong, W., Zhen, H., Li, S., Wang, L. and Liao, Z. (2021) A Simple Two-Stage Evolutionary Algorithm for Constrained Multi-Objective Optimization. *Knowledge-Based Systems*, **228**, Article 107263. <https://doi.org/10.1016/j.knsys.2021.107263>

-
- [9] Sun, R., Zou, J., Liu, Y., Yang, S. and Zheng, J. (2022) A Multistage Algorithm for Solving Multiobjective Optimization Problems with Multiconstraints. *IEEE Transactions on Evolutionary Computation*, **27**, 1207-1219. <https://doi.org/10.1109/tevc.2022.3224600>
 - [10] 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>
 - [11] Liu, Z.Z., Wang, B.C. and Tang, K. (2021) Handling Constrained Multiobjective Optimization Problems via Bidirectional Coevolution. *IEEE Transactions on Cybernetics*, **52**, 10163-10176. <https://doi.org/10.1109/tevc.2021.3056176>
 - [12] Liu, Z., Han, F., Ling, Q., Han, H. and Jiang, J. (2025) Constraint-Pareto Dominance and Diversity Enhancement Strategy-Based Evolutionary Algorithm for Solving Constrained Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, **29**, 2771-2784. <https://doi.org/10.1109/tevc.2024.3525153>
 - [13] Tian, Y., Wang, R., Zhang, Y. and Zhang, X. (2025) Adaptive Population Sizing for Multi-Population Based Constrained Multi-Objective Optimization. *Neurocomputing*, **621**, Article 129296. <https://doi.org/10.1016/j.neucom.2024.129296>
 - [14] Ming, F., Gong, W., Wang, L. and Gao, L. (2022) Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer. *IEEE Transactions on Evolutionary Computation*, **28**, 77-89. <https://doi.org/10.1109/tevc.2022.3230822>
 - [15] Qiao, K., Liang, J., Yu, K., Yue, C., Lin, H., Zhang, D., *et al.* (2023) Evolutionary Constrained Multiobjective Optimization: Scalable High-Dimensional Constraint Benchmarks and Algorithm. *IEEE Transactions on Evolutionary Computation*, **28**, 965-979. <https://doi.org/10.1109/tevc.2023.3281666>