

基于信息共享的改进双归档高维多目标进化算法^{*}

丁炜超, 李佳宁, 顾春华, 刘佳豪, 董文波

(华东理工大学 信息科学与工程学院, 上海 200237)

通信作者: 董文波, wbdong@ecust.edu.cn



摘 要: 高维多目标优化问题 (many-objective optimization problem, MaOP) 广泛存在于科学研究和工程应用领域. 受高维目标冲突引起的非支配解集数量呈指数增加影响, 传统的多目标进化算法在求解 MaOP 时面临计算复杂度增加、解质量降低等困难. 为此, 提出一种基于信息共享的改进双归档高维多目标进化算法 (improved two-archive high-dimensional multi-objective evolutionary algorithm based on information sharing, Two-Arch/IS), 旨在利用双归档算法计算复杂度低、收敛及多样性独立优化等优势特性, 实现高维多目标优化问题的高效求解. 相较于传统的算法, 首先, Two-Arch/IS 基于空间划分的子种群互映更新策略实现档案库的维护, 进一步增强种群的多样性表现; 其次, 利用基于角度选择与转移密度估计的存档截断策略移除档案库中冗余解, 在进化过程中保持算法的选择压力; 最后, 在种群进化过程中引入边界解驱动的信息补偿机制, 增强收敛性存档和多样性存档间的信息交流, 实现种群个体间的优势互补. 将 Two-Arch/IS 与其他代表性的算法一同在 69 个具有 2–20 个目标的基准测试与真实世界问题上进行性能对比实验. 实验结果表明, Two-Arch/IS 算法在高维多目标优化问题上能够有效克服种群收敛性与多样性的冲突, 并在不同性能评价指标上均表现出明显优势.

关键词: 高维多目标优化问题; 双归档进化算法; 存档截断策略; 种群互映更新; 信息共享机制

中图法分类号: TP301

中文引用格式: 丁炜超, 李佳宁, 顾春华, 刘佳豪, 董文波. 基于信息共享的改进双归档高维多目标进化算法. 软件学报, 2026, 37(3): 1143–1169. <http://www.jos.org.cn/1000-9825/7496.htm>

英文引用格式: Ding WC, Li JN, Gu CH, Liu JH, Dong WB. Improved Two-archive High-dimensional Multi-objective Evolutionary Algorithm Based on Information Sharing. Ruan Jian Xue Bao/Journal of Software, 2026, 37(3): 1143–1169 (in Chinese). <http://www.jos.org.cn/1000-9825/7496.htm>

Improved Two-archive High-dimensional Multi-objective Evolutionary Algorithm Based on Information Sharing

DING Wei-Chao, LI Jia-Ning, GU Chun-Hua, LIU Jia-Hao, DONG Wen-Bo

(School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract: Multi-objective optimization problem (MaOP) is widely encountered in scientific research and engineering applications. Due to the exponential increase in the number of non-dominated solutions caused by objective conflicts, traditional multi-objective evolutionary algorithms face challenges such as increased computational complexity and degraded solution quality when solving MaOPs. To address these issues, this study proposes an improved two-archive high-dimensional multi-objective evolutionary algorithm based on information sharing, Two-Arch/IS, for the efficient solution of MaOP. The proposed algorithm leverages the inherent advantages of the two-archive framework, including low computational complexity and independent optimization of convergence and diversity. Distinct from traditional algorithms, archive maintenance in Two-Arch/IS is achieved through a subpopulation reflection and update strategy based on space partitioning, which enhances population diversity. Furthermore, an archive truncation strategy based on angle selection and shift-based

^{*} 基金项目: 国家自然科学基金 (62403201); 上海市基础研究特区计划 (22TQ1400100-16); 上海市自然科学基金 (24ZR1415200, 23ZR1414900, 22ZR1416500)

收稿时间: 2024-05-07; 修改时间: 2024-10-14, 2025-05-25; 采用时间: 2025-06-20; jos 在线出版时间: 2025-12-10

CNKI 网络首发时间: 2025-12-11

density estimation is adopted to eliminate redundant solutions from the archive, thus maintaining selection pressure during the evolutionary process. Finally, a boundary-solution-driven information compensation mechanism is introduced to facilitate information exchange between the convergence and diversity archives, enabling effective complementarity among individuals in the population. In this study, Two-Arch/IS is benchmarked against several representative algorithms on 69 widely used test instances and real-world problems with 2 to 20 objectives. Experimental results demonstrate that the proposed algorithm effectively addresses the conflict between convergence and diversity in high-dimensional many-objective optimization and exhibits superior performance across multiple evaluation metrics.

Key words: high-dimensional multi-objective optimization problem (MaOP); two-archive evolutionary algorithm; archive truncation strategy; population reflection update; information sharing mechanism

随着信息技术的快速发展,多目标优化问题的待优化目标数量不断增加.当优化目标从2-3个扩展到超过3个时,衍生出了最优化研究中新的研究方向:高维多目标优化问题(many-objective optimization problems, MaOP). MaOP在实际应用中,如无人机的概念设计^[1]、高速列车的轻量化和结构设计^[2]、轨道交通中吸能结构的优化设计^[3]等,都扮演着至关重要的角色.除上述系统工程应用之外, MaOP还被广泛应用于不同领域的专业性问题的,包括但不限于软件工程领域^[4]、能源领域^[5,6]、汽车领域^[7]等,因此长期以来广受学术界和工业界的高度关注.在求解2-3维度的多目标优化问题时,由于目标数量较少,算法的非支配排序策略能够有效区分个体间优劣,从而得到质量较好的解集,求解侧重点在于算法的收敛性和多样性.而在求解 MaOP时,个体间的优劣关系变得模糊,这增加了搜索的难度,导致算法收敛过程也较为缓慢甚至停滞.因此算法在求解 MaOP时,需要更加高效的搜索策略来搜索庞大的目标空间,不仅需要关注收敛性和多样性,搜索效率也尤为重要.

在求解高维多目标优化问题时,各个目标间存在的冲突导致某一目标的改善往往会削弱其他目标的表现^[8],因此解决此类问题的核心是找到一组折中解,即帕累托最优解.进化算法,以其自适应搜索、精英保留等特点,能够在单次运行中有效探索解空间,并获得一组高质量解,这使得进化算法在解决 MaOP时能发挥出色性能,从而涌现出了大量的多目标进化算法(multi-objective evolutionary algorithm, MOEA).然而,随着问题目标数量的增长,算法面临的主要挑战来自种群中非支配个体的比例急剧增大(甚至可能所有解都是非支配的),此时算法的环境选择机制将不能施予种群足够的选择压力.一方面,缺乏明确的优劣排序使算法难以推进种群向帕累托前沿移动,从而影响收敛性能;另一方面,过于注重多样性可能导致解集偏离帕累托前沿,由于收敛性和多样性之间的冲突加剧,算法难以在两者之间实现有效平衡.

针对收敛性和多样性冲突难以平衡的问题,现有研究人员分别从以下几个角度进行优化和缓解:(1)修改传统的帕累托支配关系,通过松弛严格的支配关系,扩大种群中个体支配区域,以增强种群向帕累托前沿(Pareto front, PF)方向移动的选择压力,如 MaOEA/d²^[9]、Ra-dominance^[10]、Angle-dominance^[11]等;(2)引入密度估计,通过度量个体分布情况、指导搜索方向等方式,增强算法的多样性表现,如 SDE^[12]、GREA^[13]等;(3)构建综合性指标,反映种群在不同维度的性能状态,指导种群收敛,如 NSGAIII-ICO^[14]、KnEA^[15]、Lp-norm^[16]等.上述方法的共同点是:算法在一次种群优化迭代过程中同时保持种群的多样性和收敛性.然而,随着目标变量维数的增加,不同目标间冲突加剧,种群中几乎所有的个体都是互不支配,这种“一次迭代同时优化”的机制,很难从根本上解决收敛性和多样性之间的冲突问题.

Two-Arch (two-archive) 算法是 Praditwong 等人^[17]提出的一种新的多目标进化算法.该算法首次将非支配解集分解为两个档案库,收敛性档案(convergence archive, CA)和多样性档案(diversity archive, DA),分别促进种群的收敛性和多样性,从而实现收敛性和多样性的独立优化.区别于以往的多目标进化算法, Two-Arch 算法的分离优化能够精准提升解集的质量,而不是在收敛性与多样性之间进行模糊的权衡,另外算法降低了收敛性和多样性之间的冲突与干扰.算法利用迭代过程中产生的非支配解集更新 CA 和 DA,根据特定选择原则,将其中表现出支配地位的个体加入 CA,并删除被支配的个体,以维护 CA 的收敛性;对于剩余个体,将其加入 DA.在档案库大小超出预设阈值时,根据 DA 中个体到 CA 的距离剔除 DA 中额外个体,从而维护 DA 的多样性. Two-Arch 最大的特点是通过维护两个档案库,使得算法在不增加计算复杂度的前提下实现种群收敛性和多样性的独立优化,缓解了种群的收敛性和多样性冲突,在处理 2-3 维多目标优化问题时凸显出较强的优势.然而,传统的 Two-Arch 算法框

架本质上是基于帕累托支配关系的算法, 因此当目标维数增大时, 算法很难有效维护种群收敛性和多样性平衡. 此外, Two-Arch 算法中缺少对 CA 多样性的维护, 因其特殊的档案库截断机制, 可能导致最终得到一个多样性严重不足的种群, 详见第 1.2 节.

为此, 本文提出了一种基于信息共享的改进双归档高维多目标进化算法 Two-Arch/IS (improved two-archive high-dimensional multi-objective evolutionary algorithm based on information sharing), 旨在借助双归档算法的独特优势实现 MaOP 的高效求解. 其主要思想是在进化过程中保持两个档案库, 以分别促进种群的收敛性和多样性. 针对高维目标空间广阔带来的多样性挑战, Two-Arch/IS 采取了基于子种群的档案库更新策略, 通过均匀划分档案库, 确保算法在不同方向上实现均衡搜索, 从而增强其多样性表现. 同时, Two-Arch/IS 提出了一种基于角度选择与转移密度估计的存档截断策略, 通过综合评价指标有效评估个体收敛性与多样性, 在维护种群质量的同时弥补了原有算法框架的缺陷. 此外, Two-Arch/IS 引入了一种信息补偿机制, 强化档案库间信息交流, 进而实现种群综合质量的提升. 本文提出的 Two-Arch/IS 算法与几种现有的流行高维多目标进化算法一同在测试集 DTLZ 和 ZDT 上进行了性能评估, 结果表明了 Two-Arch/IS 算法具有较明显的性能优势.

本文主要贡献如下.

(1) 设计一种基于空间划分的档案库更新策略, 利用权重向量划分目标空间, 基于划分后的子种群更新收敛性档案和多样性档案, 利用均匀分布的权重向量增强种群的多样性表现.

(2) 提出基于角度选择与转移密度估计的存档截断策略, 在保持搜索效率的同时借助综合指标转移密度估计移除种群中表现不佳的个体, 提高种群向帕累托前沿移动的选择压力.

(3) 引入基于边界解驱动的信息补偿机制, 通过在表现出不同特性的档案库间进行边界解的迁移, 实现收敛性档案和多样性档案间的优势互补.

(4) 提出一种改进的算法框架 Two-Arch/IS, 并在测试集 DTLZ 和 ZDT 上进行对比实验, 与现有的双归档改进算法及流行的高维多目标进化算法进行对比分析以全面验证 Two-Arch/IS 性能.

本文第 1 节介绍相关工作. 第 2 节详细介绍 Two-Arch/IS 的算法框架以及具体设计. 第 3 节展示 Two-Arch/IS 和一些现有算法的性能结果对比及原因分析. 第 4 节对本文工作进行总结和展望.

1 相关工作

1.1 高维多目标进化算法

高维多目标优化问题, 作为一类极具挑战性的多目标优化问题, 对传统的多目标进化算法提出了严峻考验. 其根源在于, 随着目标维数的递增, 算法面临多重困境. 首先, 目标维数的增加引起帕累托支配关系逐渐失效以及算法选择压力不足问题, 进而导致种群收敛性和多样性冲突难以平衡^[18], 直接影响算法的收敛速度和解的质量; 其次, 多样性不足是算法需要克服的另一大难题. 由于高维空间中维度的增加, 目标空间的广阔性使得有限的解稀疏地分布在高维空间中, 难以有效地描绘帕累托前沿的形状. 此外, 算法的性能对帕累托前沿的形状十分敏感, 对不同的 PF 需要采取不同的搜索策略和处理方式, 这使得多样性维护更加困难; 最后, 不可避免地, 随着目标数目的增长, 高维多目标进化算法的计算复杂度和空间复杂度急剧上升, 不仅直接影响算法的性能, 更可能导致算法在处理大规模问题时失去实用性.

近年来, 研究者从多个角度对高维多目标进化算法进行研究, 根据算法进化机制的不同, 现有高维多目标进化算法可大致划分为如下几类.

(1) 基于帕累托支配关系的 MOEA. 在这类算法框架中, 个体之间的优劣关系取决于其在种群中所处的支配层级, 同时利用拥挤度距离或密度函数维持种群的多样性. 随着目标数的不断增长, 种群中非支配个体的比例急剧上升, 由于种群规模的限制, 严重削弱了帕累托支配在选择过程中的作用, 导致算法难以筛选出有效个体. 因此在求解 MaOP 时, 为避免帕累托优势关系失效导致算法退化, 研究人员普遍采取松弛帕累托优势关系的策略, 如 Angle-dominance^[11], θ -dominance^[19], 以缓解该类算法的性能衰退. 然而, 部分现有的支配方法难以有效兼顾收敛性与多样性^[20], 并且大多方法需要额外设置参数^[21], 在极大程度上影响了算法性能, 因此设计更加有效的支配方法仍待

进一步研究与探索.

(2) 基于分解的 MOEA. 基于分解的方法通过聚合函数将多目标优化问题分解为多个单目标子问题, 使用进化算法以协作的方式对子问题进行优化, 最终得到一组近似的最优解集合, 近年来提出的代表性算法有 MOEA/D-ROD^[22]、IM-MOEA/D^[23]. 这类方法有效避免了算法的选择压力在高维下失效的难题, 并且子问题数量不会呈指数级增长, 但算法的性能在极大程度上依赖于权重向量在解空间中的分布情况. 对于具有不规则 PF 的问题, 生成有效划分 PF 的权重向量集是一个巨大的挑战. 此外, PF 的形状复杂多变, 传统的权重向量生成方法难以适用所有的情况, 因此主要的研究难点在于设计能够适应多种 PF 形态的权重向量集生成方法.

(3) 基于指标的 MOEA. 该类方法通过定义一系列指标衡量解集优劣, 基于指标推进演化过程, 有效应对高维空间中的搜索难题, 实现有效且均衡的搜索. 经典的评价指标包括超体积 (hypervolume, HV)、逆世代距离 (inverted generational distance, IGD) 等, 因其良好的理论支撑被广泛应用于多种算法中^[24,25]. 然而, 这些指标的计算复杂度通常较高, 在处理高维或大规模问题时, 算法耗时很长. 其次, 基于指标推进种群演化, 这意味着算法性能过度依赖于指标的设计. 尽管存在一些指标能够同时兼顾收敛性与多样性, 但在不同问题中的表现可能存在差异, 无法在所有场景下均有效. 此外, 为特定问题选取合适的指标也是一大难题.

针对 MaOP, Two-Arch 算法因其计算复杂度低、收敛性与多样性独立优化的优势, 为该问题提供了新的求解思路. 然而, Two-Arch 框架本质上属于基于帕累托支配关系的 MOEA, 在求解 MaOP 时, 常面临帕累托优势关系失效的困境, 因此造成算法选择压力不足. 为应对这一问题, 本文提出了一种改进的双归档进化算法, 旨在通过多种策略缓解该难题, 提高算法在高维优化问题中的综合性能.

1.2 Two-Arch 算法

Two-Arch 算法是一种低复杂度的多目标进化算法, 算法将非支配解集划分为两个档案库: 收敛性档案 CA 和多样性档案 DA, 分别促进算法的两个目标——收敛性和多样性.

算法流程图如后文图 1 所示, 基本框架类似于一般的 MOEA, 即种群的复制和迭代. 在繁殖过程, CA 和 DA 的并集将作为 MOEA 框架中的父种群, 而 Two-Arch 并未设置特殊的父种群选择机制. 在通过交叉和突变操作获得新一代个体后, Two-Arch 根据非支配解集对 CA 和 DA 进行更新. 由于 CA 和 DA 的目标不同, Two-Arch 为其制定了不同的选择原则. 针对获得的非支配解集, 支配 CA 或 DA 中任意个体的解将被加入 CA (具备支配地位的后代), 而未能支配两者的解则加入 DA (无支配地位的后代). 在种群迭代过程中, CA 和 DA 中的任何支配个体都将被移除. 在 Two-Arch 中, CA 和 DA 被视为整体解集, 总规模固定 (假设为 N), 但各自大小灵活调整. Two-Arch 根据 DA 中个体与 CA 的距离来移除 DA 中额外个体, 而 CA 中个体保持不变. 由于 CA 的大小不会超过上限 N , 当 CA 达到最大规模时, DA 将被清空, 此时加入 CA 的解均具备支配地位, 并至少替代一个被支配的 CA 个体.

Two-Arch 算法作为首个实现收敛性与多样性独立优化的算法, 在未增加额外算法复杂度的前提下, 展现出了其独特的优势. 实验数据充分证明, Two-Arch 算法取得了较为显著的研究成果. 然而, 随着目标维数的增加, Two-Arch 算法的优异性能逐渐下降, 这主要归因于两点: (1) 鉴于 Two-Arch 算法本质上是一种基于帕累托的多目标进化算法, 传统的帕累托占优是一种严格的支配关系, 因此在求解具有大量目标的多目标问题时, 种群中非支配解的比例急剧增加导致种群选择压力不足, 难以收敛至 PF, 算法的优化效果受到影响; (2) 档案更新机制尚存不足. 尽管 CA 的作用是促进收敛, 但由于缺乏对 CA 多样性的维护, 可能导致算法被迫停滞. 当 CA 中的所有解都在 PF 上并且 CA 的大小已经达到最大 (上限 N) 时, DA 将被丢弃. 此时如果新的非支配解不能支配档案库中的个体, 本应将其添加到 DA, 但此时 CA 和 DA 的总大小已经达到上限, 并且无法删除 CA 中的任何解, 因此算法将被迫停滞, 不再更新 CA 和 DA, 算法最终得到的结果是多样性严重缺失的 CA.

1.3 Two-Arch 变体算法

鉴于 Two-Arch 算法在解决多目标优化问题时所表现出的潜力, 近年来, 研究人员从档案更新机制、种群搜索策略等角度对其进行了优化改进, 并将其应用到不同类型的多目标优化问题中, 包括: 高维多目标优化问题、约束多目标优化问题 (constrained multi-objective optimization problem, CMOP) 和多模态多目标优化问题 (multi-

modal multi-objective optimization problem, MMMOP), 具体描述如下。

针对高维多目标优化问题, Wang 等人^[26]提出了具有代表性的双存档算法 Two_Arch2. 该算法为 CA 和 DA 分别设计了收敛性和多样性的更新策略, 进一步强调其作用, 其中集成了多个优势关系, 而不是单一的帕累托优势关系, 有效避免了高维目标环境下难以收敛的问题. 另外, 设计了基于 L_p 范数的相似性度量方案, 有效评估个体间的相似性, 维护了种群的多样性. Ye 等人^[27]在 Two-Arch 框架基础上设计了 3 种不同的搜索方式以增强不同阶段的收敛性和多样性. Dai 等人^[28]提出了一种基于多搜索策略的双存档进化算法 TwoArchM, 第 1 种搜索策略在档案库中选择收敛解以促进收敛性, 第 2 种搜索策略平衡算法的局部开发和全局探索能力. Cai 等人^[29]提出了基于聚合框架的 TwoArchA, 设计了改进的切比-切夫函数作为收敛度量在 CA 中发挥重要作用, 同时利用解与权重向量的垂直距离保持 DA 的多样性.

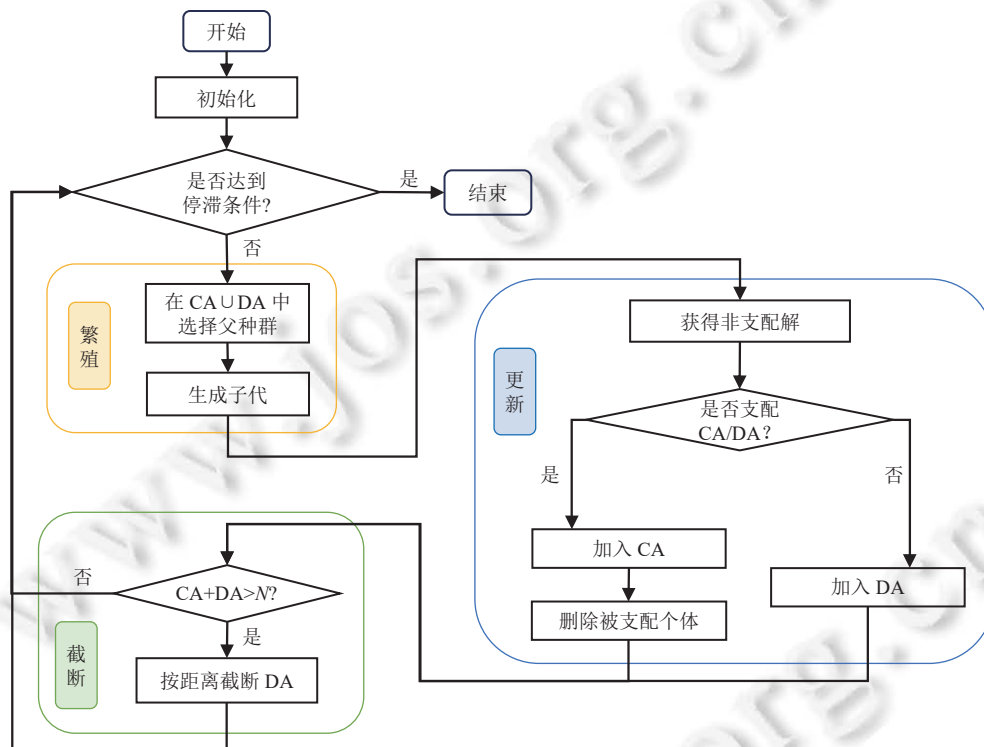


图 1 Two-Arch 流程图

针对约束多目标优化问题, 进化计算领域的核心挑战在于如何应对约束条件的复杂性, 同时兼顾算法的收敛性与多样性^[30]. 双归档机制可以通过 CA 维持收敛性和可行性驱动, DA 则不受约束限制, 充分探索目标空间以提升多样性^[31,32]. 对于多模态多目标优化问题, 由于决策空间中多个位置可能映射到目标空间的同一点, 因此难以获取多种形态的全部最优解. 此类问题中, 双归档算法利用 CA 强化目标空间的收敛性, 同时收集多个子空间的帕累托解, DA 在增强目标空间和决策空间多样性的同时, 有效引导种群避开局部最优区域^[33,34].

本文的目的是设计一种针对 MaOP 的高效算法, 能够有效平衡种群收敛性与多样性, 并确保在不同目标维度和不同类型的帕累托前沿上均表现出色. 尽管已有一些高维双归档算法, 如 Two_Arch2^[26]和 TwoArchA^[29], 但其仅在 DTLZ1–DTLZ4 问题上展现出优越性, 而在面临具有退化帕累托前沿的问题时性能大幅下降, 主要原因是未能充分考虑到不规则 PF 问题的复杂性. 此外, 算法 TwoArchM^[28]仅在有限的目标维度上进行了测试, 缺乏对各个目标维度上性能的充分验证. 因此, 亟需设计一种更具鲁棒性的算法框架, 以应对复杂多样的 MaOP 优化场景, 并在不同目标维度上进行充分验证.

2 基于信息共享的改进双归档高维多目标进化算法 Two-Arch/IS

2.1 算法框架

为了能够更有效地求解 MaOP, Two-Arch/IS 采取了 Two-Arch 的基本思想, 维护两个档案库分别促进收敛性和多样性. 在此基础上, Two-Arch/IS 进一步引入了繁殖搜索策略, 以增强算法的搜索能力; 设计了不同的环境选择过程, 以更有效地选择高质量解; 此外, 算法还加强了档案库之间的个体信息交流, 从而提升整体性能. Two-Arch/IS 算法的流程图如图 2 所示.

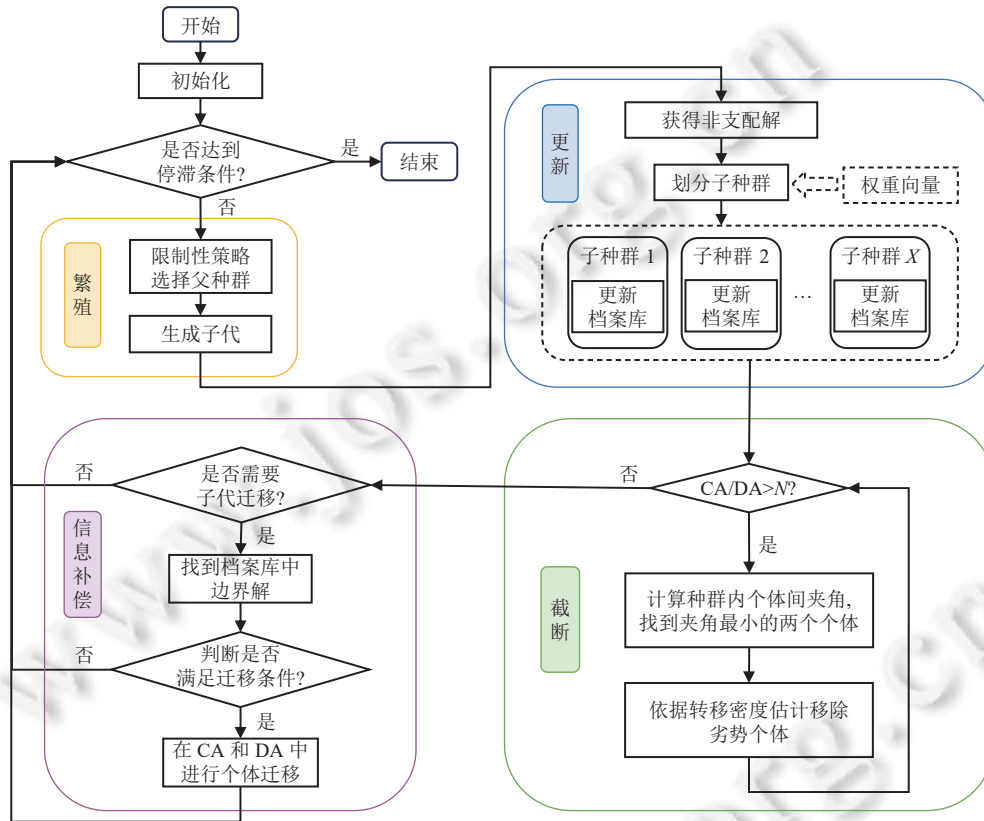


图 2 Two-Arch/IS 框架流程图

在种群进化过程中, Two-Arch/IS 设计了特别的种群繁殖策略, 其中特别设置了一个限制性的参数 δ , 用于指导以不同概率在档案库中选择性地挑选父种群, 以此产生高质量的子代个体. 通过交叉和突变操作产生新的子代后, Two-Arch/IS 依据产生的非支配解集和算法初始化得到的一组权重向量对档案库进行更新, 通过权重向量划分种群, 在各个子种群内部根据规则独立更新档案库, 更新完成后合并子种群. 此时 Two-Arch/IS 检查档案库大小是否超出预设上限 N , 若超出, 则移除档案库中额外的劣势解. Two-Arch/IS 基于角度选择与转移密度估计的方法从档案库中选择综合质量差的个体并删除, 以确保两个档案库大小均不超出限制. 最后, Two-Arch/IS 在特定情况下实施档案库间的信息补偿, 通过选择档案库中的边界解, 根据一系列条件判断是否交换个体, 以此为 CA 增加多样性信息, 为 DA 增加收敛性信息. 为了更清晰地展示算法迭代过程, 算法 1 给出了 Two-Arch/IS 框架的伪代码.

算法 1. Two-Arch/IS 算法框架.

输入: 种群大小 N , 目标数量 M , 限制性交配参数 δ , 最大迭代次数 MG ;

输出: 收敛性档案 CA.

```

1. 初始化:  $[P, CA, DA, W] = \text{Initialize}(N, M)$ ;
2.  $t = 0$ ;
3. while  $t < MG$  do
4.   根据参数  $\delta$  选择交配父种群:  $MP = \text{Restrictive\_Mating\_Selection}(CA, DA, \delta)$ ; //算法 2
5.   使用交叉变异方法生成子代:  $Q = \text{Offspring\_Generation}(MP)$ ;
6.   档案库更新:  $[CA, DA] = \text{Update\_Archive}(Q, CA, DA, W)$ ; //算法 3
7.   档案库截断:  $[CA, DA] = \text{Delete\_Redundant\_Individuals}(CA, DA, N)$ ; //算法 4
8.   子代迁移:  $[CA, DA] = \text{Individual\_Migration}(CA, DA, MG, t)$ ; //算法 5
9.    $t = t + 1$ ;
10. end while
11. return CA

```

2.2 限制性交配选择策略

Two-Arch/IS 的繁殖过程基于限制性的交配选择策略进行, 通过在特定情况下选择收敛性和多样性表现卓越的个体作为父代种群, 从而产生性能表现更好的子代种群, 进而提高算法在目标空间中的搜索能力. 繁殖过程首先以限制性的选择方法确定交配种群, 随后遵循 Two-Arch 算法中定义的交叉和变异方法, 生成新一代种群. 限制性交配选择父代的过程涵盖两种情况: 一种是在整个种群中随机选择父代种群, 另一种则是分别从两个档案库中选取表现突出的个体作为父代种群. 随机选择的方式确保子代具有较强的随机性和多样性, 能够广泛探索目标空间; 而特定父代的选择策略通过收敛性和多样性突出的父代, 保证子代个体具备高质量特性, 增强了算法的选择压力, 进而提升种群向 PF 收敛的能力.

Two-Arch/IS 预先定义了一个参数 δ , 以控制算法随机选择父代的概率. 通过实验验证, 当 δ 的取值为 0.7 时, 算法展现出较好的性能 (见第 3.2 节). 下面将详细阐述 Two-Arch/IS 算法选择亲本的具体过程.

首先, Two-Arch/IS 根据轮盘赌方法确定当前过程选择父代种群的方式. 随机选择父代种群时, 算法在 CA 和 DA 的并集中进行选择. 对于限制性的选择策略, 算法首先从 CA 中随机选择两个个体. 为筛选具有更优收敛潜力的亲本 1, 采用目标空间中的欧几里得距离作为评估标准. 具体而言, 计算当前个体与 CA 中所有其他个体的距离之和, 选择距离和最小的解作为亲本. 依据在于, 在连续凸帕累托前沿场景下, 距离之和最小的解等价于收敛最优解, 而在非凸、离散等复杂 PF 场景中, 该解表现为目标空间的几何中心解, 反映当前种群可达的最均衡权衡状态. 距离求和计算公式如下, x 代表当前计算个体, P 表示当前种群集合, $dist$ 为个体 x 和 y 间的欧几里得距离.

$$d(x, P) = \sum_{y \in P} dist(x, y) \quad (1)$$

同样地, 在 DA 中进行类似操作以选择更具多样性潜力的亲本 2, 依赖欧几里得距离进行多样性的评估. 算法计算当前个体到 DA 中所有其他个体的距离之和, 选择距离之和最大的个体, 即为目标空间的边界表征解, 能够引导种群向未探索区域扩张, 并通过定位种群分布极值点给出更好的多样性贡献.

为便于理解 Two-Arch/IS 的父种群选择方法, 算法 2 给出了限制性交配选择策略的具体过程.

算法 2. 父种群选择 Restrictive_Mating_Selection.

输入: 收敛性档案 CA, 多样性档案 DA, 限制性交配参数 δ ;

输出: 父代种群 $MP = \{q_1, q_2\}$.

```

1. if  $rand < \delta$  then
2.   在 CA 和 DA 中随机选择父种群  $q_1$  和  $q_2$ ;

```

3. else

4. 在 CA 中随机选择个体 x_1 和 x_2 ;
5. 在 x_1 和 x_2 中选择收敛性更好的个体作为 q_1 ;
6. 在 DA 中随机选择个体 x_3 和 x_4 ;
7. 在 x_3 和 x_4 中选择多样性更好的个体作为 q_2 ;

8. end if

9. return $MP = \{q_1, q_2\}$

2.3 基于空间划分的子种群互映更新策略

Two-Arch/IS 通过一组均匀分布的权重向量划分目标空间, 在划分后的各个子种群中执行档案库的更新操作. 利用基于分解的框架, 使得种群能够更均匀地收敛至 PF, 在确保收敛性的同时, 增强算法的多样性表现.

在 Two-Arch/IS 中, 根据目标数量 M 的不同, 算法会生成相应大小的权重向量集, 以实现对目标空间的划分. 权重向量集的生成方式参照了文献 [35] 中的方法. 具体地, 对于不同的目标数量 M , 确定每个目标维度上的采样个数 H , 其中采样步长为 $1/H$. 该方法对 M 维空间进行采样, 最终产生 C_{H+M-1}^{M-1} 个均匀分布在目标空间中的权重向量.

算法 3. 档案库更新 Update_Archive.

输入: 收敛性档案 CA, 多样性档案 DA, 新的子代解 Q , 权重向量集 W ;

输出: 收敛性档案 CA, 多样性档案 DA.

1. $R = Q$ 中的非支配解集;
 2. X = 权重向量的数量;
 3. 划分 CA: $[CA_1, \dots, CA_X] = \text{Divide_Individuals}(CA, W)$; //公式 (2)
 4. 划分 DA: $[DA_1, \dots, DA_X] = \text{Divide_Individuals}(DA, W)$;
 5. 划分 R : $[R_1, \dots, R_X] = \text{Divide_Individuals}(R, W)$;
 6. **for** $i = 1$ **to** X **do**
 7. 根据 CA_i 和 DA_i 删除 R_i 中的支配解;
 8. 基于 R_i 更新 CA_i 和 DA_i ;
 9. **end for**
 10. $CA = \{CA_1, \dots, CA_X\}$;
 11. $DA = \{DA_1, \dots, DA_X\}$;
 12. **return** CA, DA
-

在种群经过繁殖过程产生新一代个体后, 对当前的 CA、DA 以及产生的非支配解集执行子种群划分操作. 通过遍历权重向量集, 计算档案库中所有个体与当前权重向量的欧几里得距离, 根据距离最小原则, 选择对应的个体加入当前权重向量所对应的子种群. 具体计算方式如公式 (2), S_i 表示划分后的子种群, k 表示当前为第 k 轮划分, x_i 表示种群中个体, P 为待划分种群, w_j 表示第 j 个权重向量. 完成一次权重向量集的遍历后, 每个子种群中便包含至少一个个体. 对于剩余的个体, 继续遍历权重向量集进行划分, 直至所有个体都被划分至子种群. 这种划分方式的优势在于, 除了最后一次遍历外, 每次遍历结束后, 各子种群中的个体数量均保持一致, 从而确保子种群的均匀性. 这种均匀方式不仅有助于提升算法效率, 也在后续的档案库更新过程中有效促进整体种群的多样性表现.

$$S_i = \bigcup \{x_i^{(k)} | x_i^{(k)} = \arg \min_{x \in P} \|F(x) - w_j\|_2\} \quad (2)$$

本文也探索了另一种子种群划分方法, 其基本思想是将个体分配给与其欧几里得距离最近的权重向量所对应的子种群. 计算方式如公式 (3), 对于种群 P 中当前个体 x , 计算其与各权重向量 w_j 间的欧氏距离, 并将其分配至

对应子种群 S_i 中. 这种划分方式仅需一次遍历种群即可完成, 虽然时间复杂度相对较低, 但在种群进化过程中我们发现子种群大小存在较大差异, 甚至部分子种群为空, 这种非均衡状态极有可能导致算法过早地陷入局部最优解, 进而削弱种群的多样性.

$$S_i = \left\{ x \in P \mid i = \arg \min_j \|F(x) - w_j\|_2 \right\} \quad (3)$$

图3以二维目标空间为例, 直观地展示了两种种群划分方法得到的不同结果. 其中, 图3(a)是本文采取的划分方法, 在首次遍历权重向量集后, 个体{b, c, e}被分配至相应的子种群中, 完成第2次遍历后, 子种群划分结束, 得到的是3个大小均为2的均衡子种群. 而图3(b)则展示了另一种划分方法, 该方法只需一次遍历种群, 但得到的结果是3个大小分别为3、0和3的子种群, 其中第2个子种群为空, 这种不均衡的划分在后续的档案库更新过程中严重阻碍了种群多样性的维持.

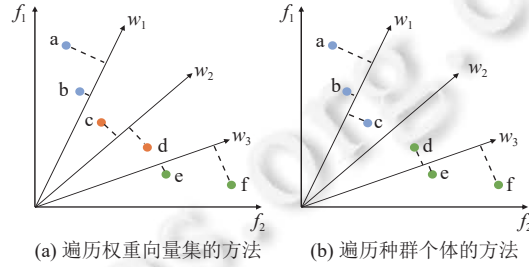


图3 种群划分示意图 (以2目标为例)

在子种群划分完成后, Two-Arch/IS 利用 R_i 更新 CA_i 和 DA_i , 更新规则主要参照 Two-Arch 算法的更新方式. 具体地, 首先移除 R_i 中被 CA_i 和 DA_i 支配的个体, 其次将非支配解集 R_i 中的个体与 CA_i 和 DA_i 中的个体进行比较, 如果存在个体可以支配 CA_i 或 DA_i 中的个体, 则将其加入 CA_i , 并移除 CA_i 中被支配的个体, 否则将其加入 DA_i . 当所有子种群更新完成后, 算法合并 CA_i 作为新的 CA , 合并 DA_i 作为新的 DA . 算法3展示了 Two-Arch/IS 基于空间划分的子种群互映更新策略实现档案库维护的过程.

2.4 基于角度选择与转移密度估计的截断策略

在档案库更新完成后, Two-Arch/IS 采用基于角度选择与转移密度估计的方法移除档案库中超过种群数量限制的冗余解. 在档案库维护过程中, Two-Arch/IS 为 CA 和 DA 设置了独立的阈值 N , 避免 Two-Arch 可能导致的多样性丧失问题. 该策略通过识别目标空间中夹角最小的一组个体, 即搜索方向最为相似的个体, 并剔除其一, 从而最大限度地保留了种群个体的多样性. 文献[12]中提出的 SDE (shift-based density estimation) 策略能够兼顾收敛性和多样性, 有效评估个体的综合性能, 因此截断过程中采取 SDE 作为核心评估指标, 确保种群在保持收敛性的同时, 尽可能维持多样性. 该策略能够有效维护种群收敛性和多样性的平衡, 同时有效降低了在高维问题下的计算成本.

假设档案库中的理想点 (ideal point) 为 $Z^{\min} = (z_1^{\min}, z_2^{\min}, \dots, z_M^{\min})$, 极点 (nadir point) 为 $Z^{\max} = (z_1^{\max}, z_2^{\max}, \dots, z_M^{\max})$. 为便于进行后续的分析, 首先将个体转移至归一化之后的目标空间, 档案库中的个体 X_i 的目标向量可以表示为:

$$F'(X_i) = (f'_1(X_i), f'_2(X_i), \dots, f'_M(X_i)) \quad (4)$$

$$f'_m(X_i) = \frac{f_m(X_i) - z_m^{\min}}{z_m^{\max} - z_m^{\min}}, \quad m = 1, 2, \dots, M \quad (5)$$

其中, $f'_m(X_i)$ 表示个体 X_i 在第 m 个目标上的值. 随后, 计算档案库中任意两个个体 X_i 和 X_j 在归一化目标空间中的夹角, 具体计算方式为:

$$\theta_{i,j} = \arccos \left| \frac{F'(X_i) \cdot F'(X_j)}{\|F'(X_i)\| \cdot \|F'(X_j)\|} \right| \quad (6)$$

在得出档案库中所有个体在归一化目标空间中两两之间的夹角后, 反复执行以下步骤: 在每次迭代中, 首先识别出夹角 θ 最小的两个个体 X_i 和 X_j , 随后, 依据文献 [12] 中所述的 SDE 策略及第 k 近邻密度估计器 (the k -th nearest neighbor density estimator)^[36], 估算这两个个体在归一化目标空间中的转移密度值 $SDE(X_i)$ 和 $SDE(X_j)$. 基于转移密度估计值, 删除档案库中性能较差的个体. 过程持续进行, 直到档案库的大小不再超过预设上限 N . 具体计算方式如下.

SDE 策略在估计个体 X_i 的密度时, 会根据种群 P 中其他个体相对于 X_i 的收敛性表现转移位置, 表示如下:

$$SDE(X_i, P) = D(dist(X_i, p'_1), dist(X_i, p'_2), \dots, dist(X_i, p'_{N-1})) \quad (7)$$

$$f_m(p'_j) = \begin{cases} f_m(X_i), & \exists f_m(X_i) < f_m(p'_j) \\ f_m(p'_j), & \text{otherwise} \end{cases}, \quad m = 1, 2, \dots, M \quad (8)$$

其中, N 表示 P 的大小, p'_i 表示转移之后的个体 p_i ($p_i \in P$ 并且 $p_i \neq X_i$), 转移方式按照公式 (8) 定义, $dist(X_i, p'_i)$ 表示个体 X_i 和 p'_i 间的相似程度, 此处使用欧几里得距离作为衡量标准. 函数 D 计算目标个体与种群中其他个体的相似度, 通过引入的 k 近邻密度估计器实现, 具体表示为:

$$D(X) = \frac{1}{\varphi^K(X) + 2} \quad (9)$$

其中, $\varphi^K(X)$ 是个体 X_i 与其他个体转移目标向量距离的集合 X 中的第 K 小值, 通常 $K = \sqrt{|N|}$.

为了更明确地展示档案库截断的计算过程, 精准描述具体操作, 我们在算法 4 中详细阐述了其计算步骤.

算法 4. 档案库截断 Delete_Redundant_Individuals.

输入: 收敛性档案 CA, 多样性档案 DA, 种群大小 N ;

输出: 收敛性档案 CA, 多样性档案 DA.

1. 在归一化空间中计算 CA 中任意两个个体间夹角;
 2. **while** $|CA| > N$ **do**
 3. 找到 CA 中拥有最小夹角 $\theta_{i,j}$ 的个体 x_i 和 x_j ; //公式 (6)
 4. 计算 $SDE(x_i)$ 和 $SDE(x_j)$; //公式 (7)
 5. 删除 SDE 值更大的个体;
 6. **end while**
 7. 在归一化空间中计算 DA 中任意两个个体间夹角;
 8. **while** $|DA| > N$ **do**
 9. 找到 DA 中拥有最小夹角 $\theta_{i,j}$ 的个体 x_i 和 x_j ; //公式 (6)
 10. 计算 $SDE(x_i)$ 和 $SDE(x_j)$; //公式 (7)
 11. 删除 SDE 值更大的个体;
 12. **end while**
 13. **return** CA, DA
-

为了展示 Two-Arch/IS 截断过程的有效性, 以二维目标空间为例, 图 4 对比了 Two-Arch/IS 和 SPEA-II+SDE^[12] 的个体选择过程. 图 4(a) 中, Two-Arch/IS 首先筛选出夹角最小的两个个体 b、c, 接着利用公式 (7)–公式 (9) 计算个体 b 和 c 的转移密度估计值, 剔除拥有更大 SDE 值的个体 c, 从而保留个体 {a, b, d, e} 进入下一次迭代. 图 4(b) 中, SPEA-II+SDE 则根据每个个体的转移密度估计值进行筛选, 此时个体 d 相较于其他个体在收敛性上并无明显优势, 因此 SPEA-II+SDE 为其分配较高的密度值进而剔除, 选择个体 {a, b, c, e} 进入下一次迭代.

值得强调的是, 个体 a、e 在维护种群多样性方面发挥着关键作用, 因为它们位于相对稀疏的区域, 为种群提供了更广泛的搜索空间. 相比之下, 个体 b、c 的搜索方向极为相似, 如果从中剔除其一, 则能够最大化种群的搜索潜力. 因此结合角度选择与转移密度估计的方法能够引导种群从多个不同方向逼近 PF, 有效维护了种群多样性.

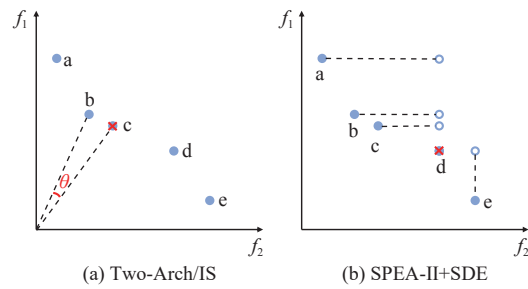


图4 Two-Arch/IS 和 SPEA-II+SDE 的截断方式 (以 2 目标为例)

2.5 基于边界解驱动的信息补偿机制

至此, 算法中各部分目的明确, 但为进一步促进档案库间信息交流, 实现收敛性表现突出个体和多样性表现突出个体之间的优势互补, 从而增强种群的综合质量, Two-Arch/IS 设计了一种基于边界解驱动的信息补偿机制. 当档案库中的边界解达到预设的条件时, 收敛性边界解将被引入 DA, 同样地, 多样性边界解将被引入 CA, 通过互相引导进化, 从而提高整体解集的质量.

在种群进化过程中, CA 的首要任务是确保足够的收敛性. 过早地进行个体迁移, 尤其是在种群进化前期, 可能会对 CA 造成干扰, 从而导致收敛性不足, 减缓种群向 PF 移动的速度, 降低算法整体效率. 因此, Two-Arch/IS 并未选择在每一代都执行档案库间的信息补偿操作, 而是选择在种群进化后期基本收敛时, 每隔 10 代执行一次档案库间的个体迁移, 具体的操作如下.

针对档案库中的最小边界解, 标记 CA 中某一目标上目标值最小的个体为 x_1 , DA 中目标值最小的个体为 x_2 . $f_m(x)$ 计算个体 x 在 m 目标维度上的目标值, 若当前目标维度上满足 $f_m(x_1) < f_m(x_2)$, 则执行个体迁移操作, 将 x_1 加入 DA, 并将其从 CA 移除. 同样地, 对于最大边界解, 标记 CA 中某一目标上目标值最大的个体为 y_1 , DA 中目标值最大的个体为 y_2 . 若当前目标维度上满足 $f_m(y_1) < f_m(y_2)$, 执行相应操作, 将 y_2 加入 CA, 并将其从 DA 中移除.

为直观地呈现 Two-Arch/IS 的信息补偿机制, 我们以伪代码的形式在算法 5 中给出了具体的计算过程.

算法 5. 子代迁移 Individual_Migration.

输入: 收敛性档案 CA, 多样性档案 DA, 最大迭代次数 MG , 当前迭代次数 t ;

输出: 收敛性档案 CA, 多样性档案 DA.

```

1. if ( $t/MG > 0.5$ ) && ( $\text{mod}(t, 10) == 0$ ) then
2.   if ( $\text{size}(\text{CA}) > 0$ ) && ( $\text{size}(\text{DA}) > 0$ ) then
3.     for  $m = 1$  to  $M$  do
4.       在  $m$  目标上找到 CA 中目标值最小的个体  $x_1$ ;
5.       在  $m$  目标上找到 DA 中目标值最小的个体  $x_2$ ;
6.       if  $f_m(x_1) < f_m(x_2)$  then
7.         将  $x_1$  移动到 DA;
8.       end if
9.       在  $m$  目标上找到 CA 中目标值最大的个体  $y_1$ ;
10.      在  $m$  目标上找到 DA 中目标值最大的个体  $y_2$ ;
11.      if  $f_m(y_1) < f_m(y_2)$  then
12.        将  $y_2$  移动到 CA;
13.      end if
14.    end for

```

15. end if
16. end if
17. return CA, DA

表 1 展示了 2 目标问题上档案库间个体迁移情况的示例. 针对档案库中的最小边界解, 在目标 1 上 $f_1(x_1) < f_1(x_2)$, 这意味着 x_1 在目标 1 上具有更优的表现, 因此将其移动到 DA, 以其在收敛性方面的优势引导 DA 进化. 而在目标 2 上 $f_2(x_3) > f_2(x_4)$, 因此不做任何处理. 针对最大边界解, 在目标 1 上 $f_1(y_1) > f_1(y_2)$, 不做任何处理. 在目标 2 上 $f_2(y_3) < f_2(y_4)$, 因此将 y_4 移动到 CA, 利用具有更大边界值的个体引导 CA 进化. 通过一系列的个体迁移操作, 旨在优化档案库间的信息交流, 确保 CA 和 DA 分别发挥其优势的同时, 能够互相引导进化.

表 1 2 目标问题上的 Two-Arch/IS 中个体迁移情况示例

处理内容	最小边界解		最大边界解	
	目标1	目标2	目标1	目标2
收敛性档案	$\min f_1(x_1) = 0.10$	$\min f_2(x_3) = 0.13$	$\max f_1(y_1) = 0.90$	$\max f_2(y_3) = 0.87$
多样性档案	$\min f_1(x_2) = 0.12$	$\min f_2(x_4) = 0.11$	$\max f_1(y_2) = 0.89$	$\max f_2(y_4) = 0.92$
操作	将 x_1 移动到DA	不做处理	不做处理	将 y_4 移动到CA

2.6 复杂度分析

在 Two-Arch/IS 中, 算法的复杂度主要来自档案库的更新、截断操作以及档案库间的信息补偿. 对于具有 M 个目标的多目标优化问题, 算法设置种群规模为 N , 子种群的大小为 N_s . 档案库更新过程中, 首先需要获取子代中非支配解, 需要的时间复杂度为 $O(MN^2)$, 随后划分档案库和非支配解集需要的时间复杂度为 $O(MN^2)$, 在子种群内部局部更新档案库的时间复杂度参考 Two-Arch 算法, 为 $O(MN_s^2)$. 由于在任何条件下满足 $N_s < N$, 因此档案库更新过程的时间复杂度可表示为 $O(MN^2)$. 截断过程首先计算种群中两两个体间夹角并找到夹角最小的两个个体, 需要的复杂度为 $O(MN^2)$, 随后通过转移密度估计值删除两个个体中综合性能较差的个体, 计算两个个体的转移密度估计值相比于 $O(MN^2)$ 可忽略. 由于截断过程需要不断执行上述操作, 直至档案库大小满足条件, 因此截断过程需要的时间复杂度可估计为 $O(MN^3)$. 为进行档案库间的信息补偿, 获取各个目标上的边界解需消耗 $O(MN)$ 的时间复杂度, 随后进行边界解目标值的比较以及个体交换, 其复杂度可忽略. 综上分析, Two-Arch/IS 算法的复杂度主要来自档案库的截断过程, 表示为 $O(MN^3)$.

3 实验结果与分析

在本节, 我们首先验证了 Two-Arch/IS 在高维问题上的卓越性能, 通过对比几种流行的 Two-Arch 变体算法以及现有高维多目标进化算法, 全面深入地评估其在 MaOP 上的表现. 紧接着, 在低维问题上进一步证明了 Two-Arch/IS 的有效性. 随后, 为验证 Two-Arch/IS 各部分的有效性, 本文针对 Two-Arch/IS 提出的几种策略进行消融实验. 在此基础上, 通过对比几种高维多目标进化算法, 验证 Two-Arch/IS 的收敛速度优势; 最后, 对比多种不同类型的算法衡量 Two-Arch/IS 的计算复杂度. 实验在具有 3.10 GHz Intel(R) Core(TM) i5-11300H 处理器和 Windows 10 64 bit 操作系统的设备上进行.

3.1 实验设置

(1) 测试问题

实验中采用了多样化的测试问题, 具体包括多目标优化的基准测试问题集 DTLZ 和 ZDT, 以及面向组合优化的多目标背包问题 (multi-objective knapsack problem, MOKP) 和多线段距离最小化问题 (multi-line distance minimization problem, MLDMP).

DTLZ^[37]测试问题集是 MaOP 领域经典的测试实例, 包含了具有不同 PF 的多种测试问题, 包括平面 PF (DTLZ1)、

凹曲面 PF (DTLZ2-4)、退化 PF (DTLZ5-6) 以及断开 PF (DTLZ7), 能够全面评估多目标进化算法的性能. 另外, 其重要特点是目标数量的可扩展性, 通过调整问题的目标数量, 从而同时检测算法在高维和低维目标下的性能表现. 实验在不同的测试问题上进行, 设置目标数量包括 {2, 3, 4, 6, 8, 10, 15, 20}, 总的测试实例数目为 56.

ZDT^[38]测试问题集是经典的 2 目标测试实例, 其不同测试问题的 PF 具有不同的难度特征, 包括基本的曲面 PF (ZDT1-2)、不连续 PF (ZDT3)、混合 PF (ZDT4) 以及复杂的非均匀分布 PF (ZDT6), 这种多样性使得 ZDT 测试问题集能够很好地检测算法在低维目标上逼近真实 PF 的能力. 实验在 ZDT1-4 以及 ZDT6 问题上进行了测试, 总测试实例数目为 5.

多目标背包问题 (MOKP)^[39]和多线段距离最小化问题 (MLDMP)^[40]是两种常见的组合优化问题. MOKP 是一种涉及多个背包的优化问题, 目标是同时最大化多个存在冲突的目标函数 (收益、重量等), 在给定的容量约束下, 选择物品的最优组合. MLDMP 旨在最小化多个线段之间的距离, 通常应用于空间布局和路径规划等领域. 该问题的特性在于寻找多个解的过程中需要平衡不同线段之间的距离最小化. 这两类问题具有多样性和挑战性, 有助于测试算法在不同情况下的表现. 实验设置目标数量为 {4, 6, 8, 10}, 总的测试实例数量为 8.

(2) 对比算法

为测试 Two-Arch/IS 算法的有效性, 本文选取了 14 种有代表性的多目标进化算法作为对比算法以检验 Two-Arch/IS 性能, 具体包括经典的 Two-Arch^[17]算法, 代表性的 Two-Arch 改进算法 Two_Arch2^[26]、C-TAEA^[41]、TriMOEATAR^[33]、TwoArchM^[28]和 TwoArchA^[29], 优异的高维多目标进化算法 NSGA-III^[42]、IBEA^[43]、KnEA^[15]、MOEA/D^[44]、SMS-EMOA^[45]、HEA^[46]、TSNSGA-II^[47], 以及面向低维问题的基于网格的进化算法 GREATER^[13]. 选择上述算法的原因主要有以下几方面.

1) Two-Arch 是一个经典的算法框架, 在进化过程中将解集分为两个档案库以分别维护种群收敛性和多样性, 缓解高维下收敛性和多样性之间的冲突, 未引入额外参数. Two-Arch/IS 在此基础上优化改进, 对比 Two-Arch 以检验 Two-Arch/IS 性能的提升.

2) Two_Arch2 是典型的 Two-Arch 变体算法, 为档案库设计了不同的选择原则, 并集成多种优势关系, 缓解了算法难以收敛的问题, 而 Two-Arch/IS 在档案库维护和更新上重新设计, 两种算法改进思想较为类似.

3) C-TAEA 在 Two-Arch 算法基础上引入了协同进化的概念, 在环境选择过程中利用存档间的交互协作, 进一步促进存档收敛性和多样性, 亦未引入额外参数, 算法中部分思想与 Two-Arch/IS 中的信息补偿机制较为相似.

4) TriMOEATAR 能够处理具有多个冲突目标的优化问题, 在 Two-Arch 框架中融入了一组参考向量集用于增强种群多样性, 该思想与 Two-Arch/IS 中的子种群互映更新策略有一定相似度.

5) TwoArchM 分离优化收敛性与多样性, 基于多搜索策略平衡局部开发和全局搜索能力, 促进种群收敛; 设计了新的基于范数的多样性保持机制维持种群多样性, 未引入额外参数.

6) TwoArchA 在 Two-Arch 方法中引入了聚合框架, 并为 CA 和 DA 重新设计了环境选择策略, 强化其在收敛性和多样性方面的作用.

7) NSGA-III 是高维多目标进化算法领域的经典算法, 核心在于使用一组参考集 (可以根据目标空间自动生成) 引导种群向 PF 进化, 并在环境选择中采取了高效的非支配排序方法, 未引入额外参数.

8) IBEA 是一种代表性的基于指标的高维多目标进化算法, 基于二元指标定义优化目标, 并在环境选择过程中以该指标进行度量, 具备较强的可扩展性, 并且不需要引入额外的多样性保持机制.

9) KnEA 是一种基于帕累托关系改进的多目标进化算法, 主要思想是通过挖掘邻域内的 knee points, 在保证多样性的同时强调种群的收敛性.

10) MOEA/D 是一种基于分解的多目标进化算法, 能够将多目标问题分解为一组单目标子问题, 通过优化子问题的解逼近整个 PF, 具有较好的性能和灵活性.

11) SMS-EMOA 基于非支配排序评判个体等级, 利用 HV 指标作为选择标准丢弃对多样性贡献较差的个体, 以此指导进化过程, 重点关注的分布和收敛性, 未引入额外参数.

12) HEA 是一种基于超支配关系的多目标进化算法,能够有效度量个体收敛性,缓解了进化过程中收敛性和多样性难以平衡的问题,同时能够在进化过程中动态调整算法的选择压力。

13) TSNSGA-II 是一种平衡收敛性和多样性的两阶段进化算法,不同进化阶段分别促进收敛性与多样性,以此有效求解高维多目标优化问题,而且未引入额外参数。

14) GREa 提出了一种基于网格的解空间搜索方式,在保持种群广泛分布的同时,增强向 PF 的选择压力,在低维问题上具有较好的性能。

(3) 性能指标

实验结果的评估使用 IGD (inverted generational distance)^[48] 指标,IGD 计算了真实 PF 到算法获得解集的平均距离,可同时反映出解集的收敛性和多样性。具体计算公式如下:

$$IGD(PF^*, PF) = \frac{\sum_{p \in PF^*} d(p, PF^*)}{|PF|} \quad (10)$$

其中, PF^* 是算法运行得到的解集, PF 是在真实的帕累托前沿上均匀分布的目标向量集合, $d(p, PF^*)$ 表示个体 p 到 PF^* 的最短距离。IGD 值越小,意味算法获得的解集与真实帕累托前沿越接近,证明其综合性能更好。

为保证实验结果的可靠性,在每个测试实例中,算法独立执行 30 次以获得 IGD 均值和方差作为最终评估结果。具体结果在第 3.3 节中展示并说明。

(4) 参数设置

为确保实验的公平性,各算法在同个测试实例上设定了统一的种群规模和评估次数。针对目标维数不超过 10 的测试实例,设置种群规模为 100,评估次数 30000;对于 15 目标的测试实例,设置种群规模为 200,评估次数 80000;对于 20 目标的测试实例,设置种群规模为 200,评估次数 90000。

各算法的特有参数设置也尤为关键,本文针对 Two-Arch/IS 中的限制性交配概率参数 δ 进行了敏感性分析,确定参数 δ 的最佳取值为 0.7。针对其他对比算法,则根据其对应文献中的参考建议值来设置其特有参数,如: Two_Arch2 算法中,设置 Lp 范数的参数 p 为 $1/m$; TriMOEATAR 算法中,设置检测峰值的精度大小 eps_peak 为 0.01,在收敛性档案中选择父节点的概率 p_con 为 0.5; TwoArchA 中,邻域大小 T 和概率 δ 分别固定为 20、0.9; IBEA 算法中,设置适应度比例因子 $kappa$ 为 0.05; KnEA 算法中,设置种群中 knee points 的比例为 0.5; MOEA/D 算法中,权重向量生成方式的参数 $type$ 固定为 1,代表线性权重向量生成; GrEA 算法中,针对不同目标维数,设置每个目标上网格划分数量 div 的范围在 4~50 之间; HEA 算法中,平衡收敛性和多样性的参数 $maxT$ 设置为 0.05。

3.2 参数 δ 的敏感性分析

在本节,我们讨论了参数 δ 对 Two-Arch/IS 算法性能的影响。 δ 作为种群繁殖过程中的关键参数,决定种群是否进行限制性交配的概率,其取值范围可在 0~1 之间灵活调整。较小的 δ 值会导致限制性交配的次数的增加,无法保证子代个体的随机性,从而影响算法的多样性和全局搜索能力。较大的 δ 值则减少限制性交配次数,可能造成搜索策略对算法性能的促进作用减弱,使得算法难以找到高质量的解。因此,在设定 δ 值时,需要权衡其对算法性能的影响,以达到最优的算法表现。为了探索不同 δ 值的 Two-Arch/IS 算法在多维度、复杂特征问题上的表现,选取了 5、10、15 目标的 DTLZ1 和 DTLZ4 问题以及 2 目标的 ZDT1 和 ZDT6 问题进行 IGD 收敛实验,其中 δ 的取值包括 {0.3, 0.5, 0.7, 0.9}, 实验结果如图 5 所示。

观察图 5 中结果,当 δ 取值为 0.7 时,算法在大多数情况下展现了最快的收敛速度,并且最终的 IGD 值表现较为优秀。在图 5(c) 中,尽管 δ 取值为 0.9 的算法在种群进化后期超越了 δ 取值为 0.7 的算法,但两者 IGD 值差异不大,且 δ 取值为 0.7 的算法保持了更快的收敛速度。在图 5(d) 中,初始阶段 δ 取值为 0.5 的算法收敛最快,但其性能表现较不稳定,而 δ 取值为 0.7 的算法则展现了稳步下降的趋势。在图 5(f) 中,尽管 δ 取值为 0.3 的算法在种群进化后期 IGD 值下降幅度较大,但由于实际情况下算法的评估次数有限, δ 取值为 0.7 的算法在有限时间内展现出了更为优越的 IGD 表现。因此,在后续的实验中,为了兼顾收敛速度和优化质量,将 δ 的取值固定为 0.7。

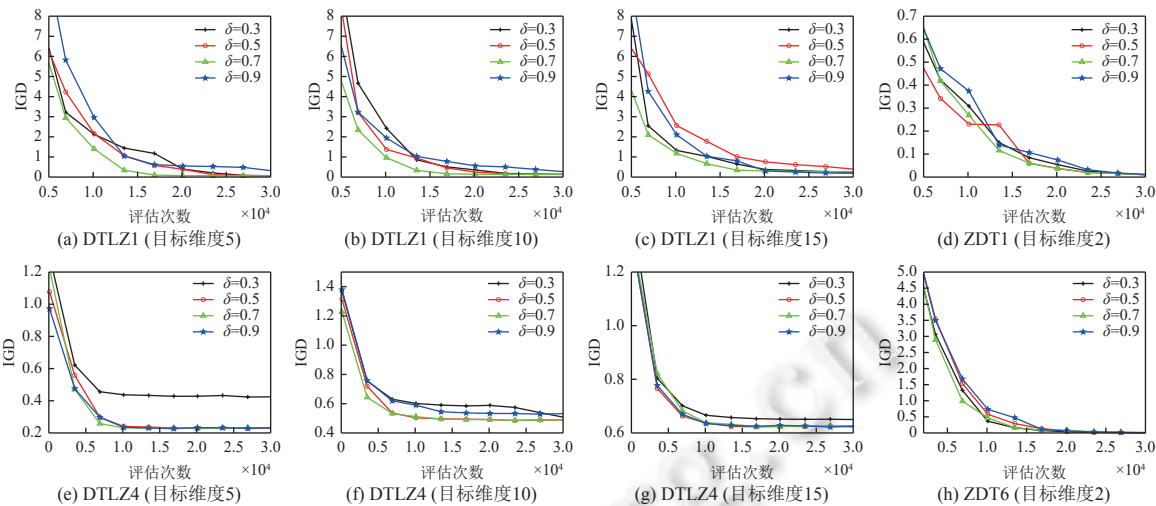


图 5 具有不同 δ 值的 Two-Arch/IS 在 DTLZ 和 ZDT 问题上的 IGD 收敛曲线图

3.3 高维目标问题上的对比实验结果与分析

在本节中, 为验证 Two-Arch/IS 在高维多目标优化问题上的卓越性能, 比较了 Two-Arch/IS 和其他 13 种算法在各个测试实例上的 IGD 值差异, 采取 IGD 均值和方差作为评判结果. 对比实验包括与不同类型的 Two-Arch 算法在 4–20 目标的 DTLZ 测试问题上的比较, 与流行高维多目标进化算法在 4–20 目标的 DTLZ 测试问题上的比较, 以及在 4–10 维度的实际测试问题 MOKP 和 MLDMP 上的比较. 对于每个测试实例, 表现最好的算法以粗体标记, 括号中是方差结果. 为更准确地检测不同算法间的性能差异, 本文采用统计学意义上的 Wilcoxon Rank-Sum 检验. 在显著水平为 0.05 的情况下, “+”表示当前算法在性能上显著优于 Two-Arch/IS, “=”表示当前算法与 Two-Arch/IS 无明显优劣差异, “-”则表示当前算法在性能上相对于 Two-Arch/IS 存在劣势.

(1) 与 Two-Arch 变体算法在 DTLZ 测试问题上的比较

为了比较 Two-Arch/IS 在同类型算法下的竞争力, 选择了 6 种 Two-Arch 框架算法作为对比算法, 包括 Two-Arch、Two_Arch2、C-TAEA、TriMOEATAR、TwoArchA 以及 TwoArchM, 在 4–20 目标的 DTLZ1–7 问题上进行了 IGD 结果的比较, 实验结果如表 2 所示.

表 2 Two-Arch/IS 和不同类型的 Two-Arch 算法在高维 DTLZ 问题上的 IGD 结果

测试问题	目标数目	Two-Arch/IS	Two-Arch	Two_Arch2	C-TAEA	TriMOEATAR	TwoArchM	TwoArchA
DTLZ1	4	4.9339E-2 (1.96E-2)	1.7324E-1 (5.87E-2) -	5.5068E-2 (6.59E-3) -	5.5721E-2 (9.17E-4) -	4.9809E-2 (6.05E-3) -	1.2479E-1 (7.17E-2) -	1.5854E-1 (5.26E-1) -
	6	9.7828E-2 (3.44E-2)	3.1041E-1 (1.41E-1) -	1.0736E-1 (6.07E-2) -	2.0210E-1 (1.42E-1) -	1.0587E-1 (4.66E-2) -	2.9439E-1 (9.17E-2) -	1.0891E-1 (2.48E-1) -
	8	1.5939E-1 (7.08E-2)	6.5512E-1 (3.58E-1) -	1.3646E-1 (3.96E-2) =	3.8914E-1 (2.03E-1) -	1.4549E-1 (6.32E-2) =	5.2729E-1 (2.73E-1) -	2.1162E-1 (7.57E-1) -
	10	1.8356E-1 (7.37E-2)	9.3650E-1 (7.05E-1) -	1.7862E-1 (5.56E-2) =	8.1429E-1 (7.48E-1) -	2.2862E-1 (1.15E-1) -	6.0306E-1 (2.54E-1) -	3.3234E-1 (4.07E-1) -
	15	1.3163E-1 (2.73E-3)	3.4162E-1 (4.72E-2) -	1.4750E-1 (5.64E-3) -	3.4068E-1 (1.62E-1) -	1.8806E-1 (3.80E-2) -	5.5114E-1 (2.99E-1) -	2.0997E-1 (6.72E-1) -
	20	1.7919E-1 (7.45E-3)	3.8241E-1 (6.49E-2) -	2.1859E-1 (1.00E-2) -	3.7069E-1 (1.13E-1) -	2.0544E-1 (7.22E-3) -	1.0049E+1 (2.10E+1) -	2.1127E-1 (2.33E-1) -
DTLZ2	4	1.5525E-1 (3.11E-3)	2.3508E-1 (2.01E-2) -	1.3458E-1 (2.02E-3) +	1.6747E-1 (3.29E-3) -	1.3850E-1 (2.09E-4) +	1.3871E-1 (1.66E-3) +	1.5397E-1 (4.98E-3) =
	6	2.9212E-1 (2.87E-3)	3.8606E-1 (2.20E-2) -	2.8304E-1 (4.56E-3) +	3.3347E-1 (2.96E-3) -	2.7929E-1 (5.92E-4) +	2.8085E-1 (1.28E-3) +	2.8197E-1 (1.52E-3) +

表 2 Two-Arch/IS 和不同类型的 Two-Arch 算法在高维 DTLZ 问题上的 IGD 结果 (续)

测试问题	目标数目	Two-Arch/IS	Two-Arch	Two_Arch2	C-TAEA	TriMOEATAR	TwoArchM	TwoArchA
DTLZ2	8	3.9197E-1 (2.74E-3)	5.2125E-1 (2.93E-2) -	4.0623E-1 (4.83E-3) -	4.3577E-1 (6.78E-3) -	3.8357E-1 (6.97E-4) +	3.8057E-1 (1.59E-3) +	3.9894E-1 (5.66E-3) -
	10	4.7686E-1 (2.90E-3)	6.6423E-1 (4.06E-2) -	4.9208E-1 (6.56E-3) -	5.7275E-1 (4.50E-3) -	5.1608E-1 (3.53E-3) -	5.1379E-1 (4.38E-2) -	5.4042E-1 (4.65E-3) -
	15	5.4409E-1 (4.84E-3)	8.2740E-1 (3.49E-2) -	5.9202E-1 (5.70E-3) -	6.3952E-1 (2.49E-3) -	6.6315E-1 (3.59E-3) -	5.4458E-1 (1.42E-3) =	6.2912E-1 (8.50E-3) -
	20	6.2367E-1 (2.29E-3)	9.5963E-1 (3.02E-2) -	8.0332E-1 (6.63E-3) -	7.6521E-1 (3.55E-3) -	7.3567E-1 (3.07E-3) -	6.2401E-1 (6.07E-3) =	8.1567E-1 (4.52E-3) -
DTLZ3	4	4.8838E+0 (2.05E+0)	2.0920E+0 (1.43E+0) +	1.4226E+0 (1.19E+0) +	4.7195E+0 (2.14E+0) =	1.3868E+0 (7.83E-1) +	1.0245E+0 (7.23E-1) +	4.6880E+0 (1.87E+1) +
	6	4.4899E+0 (1.70E+0)	1.2354E+1 (4.76E+0) -	2.6970E+0 (1.84E+0) +	9.3593E+0 (6.29E+0) -	5.1102E+0 (2.20E+0) =	4.2738E+0 (2.34E+0) =	6.9498E+0 (6.97E+0) -
	8	4.5326E+0 (1.95E+0)	2.7480E+1 (8.49E+0) -	4.6125E+0 (1.74E+0) -	1.8487E+1 (9.79E+0) -	7.0781E+0 (2.50E+0) -	1.2865E+1 (5.57E+0) -	5.8237E+0 (1.23E+1) -
	10	5.7788E+0 (1.67E+0)	3.5272E+1 (1.11E+1) -	9.9040E+0 (4.17E+0) -	2.8174E+1 (1.01E+1) -	8.2047E+0 (2.75E+0) -	2.3020E+1 (2.17E+1) -	5.1378E+0 (1.48E+1) +
	15	1.5993E+0 (8.97E-1)	1.4042E+1 (6.47E+0) -	6.8737E+0 (4.18E+0) -	1.9758E+1 (1.27E+1) -	3.4872E+0 (1.42E+0) -	1.9801E+2 (7.38E+1) -	1.1152E+1 (1.74E+1) -
	20	1.2400E+0 (1.13E+0)	1.7800E+1 (7.30E+0) -	2.0461E+1 (1.02E+1) -	6.0088E+0 (2.69E+0) -	2.8924E+0 (1.00E+0) -	2.5155E+2 (1.16E+2) -	2.2450E+1 (9.83E+0) -
DTLZ4	4	1.8505E-1 (9.13E-2)	3.7697E-1 (1.39E-1) -	1.3501E-1 (2.19E-3) +	1.6681E-1 (3.88E-3) +	1.8376E-1 (1.11E-1) =	2.1614E-1 (1.37E-1) -	2.1775E-1 (1.34E-1) -
	6	3.2476E-1 (5.94E-2)	5.7929E-1 (1.17E-1) -	2.8777E-1 (4.75E-3) +	3.3092E-1 (3.32E-3) =	2.9723E-1 (2.68E-2) =	2.9026E-1 (3.55E-2) +	2.9859E-1 (7.25E-2) +
	8	4.1051E-1 (3.24E-2)	7.3008E-1 (1.21E-1) -	4.1431E-1 (5.56E-3) -	4.2757E-1 (4.66E-3) =	4.0921E-1 (1.93E-2) =	3.8080E-1 (1.61E-3) +	4.4106E-1 (3.73E-2) -
	10	4.9340E-1 (1.69E-2)	7.9086E-1 (8.29E-2) -	4.9152E-1 (6.70E-3) =	5.6421E-1 (4.01E-3) -	5.3762E-1 (9.47E-3) -	5.1511E-1 (2.62E-2) -	5.1245E-1 (3.92E-3) -
	15	5.6024E-1 (2.17E-3)	8.8176E-1 (5.37E-2) -	5.7441E-1 (4.36E-3) -	6.2956E-1 (5.64E-3) -	7.0307E-1 (2.30E-3) -	5.6563E-1 (9.48E-4) =	6.4545E-1 (1.47E-2) -
	20	6.1600E-1 (1.17E-3)	1.1044E+0 (4.48E-2) -	7.3714E-1 (6.87E-3) -	7.6289E-1 (2.01E-3) -	7.4930E-1 (2.26E-3) -	6.2008E-1 (1.03E-3) -	6.9606E-1 (1.28E-2) -
DTLZ5	4	3.6911E-2 (6.06E-3)	7.1100E-2 (1.32E-2) -	4.4714E-2 (4.67E-3) -	6.6650E-2 (1.20E-2) -	4.5610E-2 (7.30E-3) -	8.8868E-2 (1.69E-2) -	5.5790E-2 (1.06E-2) -
	6	8.7485E-2 (1.50E-2)	8.7812E-2 (2.03E-2) =	1.1159E-1 (1.87E-2) -	1.4483E-1 (4.61E-2) -	1.3845E-1 (6.56E-2) -	1.6663E-1 (3.77E-2) -	9.6713E-2 (1.21E-2) -
	8	1.1496E-1 (1.89E-2)	1.1649E-1 (2.94E-2) =	1.7261E-1 (3.38E-2) -	1.8186E-1 (5.01E-2) -	2.4139E-1 (6.66E-2) -	3.1275E-1 (5.23E-2) -	1.0946E-1 (2.09E-2) +
	10	1.3084E-1 (1.93E-2)	1.3479E-1 (4.19E-2) =	2.3506E-1 (4.65E-2) -	2.7374E-1 (7.93E-2) -	4.8692E-1 (7.16E-2) -	3.3988E-1 (7.38E-2) -	1.3139E-1 (1.84E-2) =
	15	1.4586E-1 (2.37E-2)	1.5312E-1 (3.83E-2) -	3.4937E-1 (1.10E-1) -	3.1887E-1 (1.07E-1) -	5.9429E-1 (2.73E-2) -	5.6431E-1 (1.21E-1) -	1.6282E-1 (1.24E-2) -
	20	1.4888E-1 (4.07E-2)	1.9812E-1 (3.32E-2) -	5.5643E-1 (1.06E-1) -	3.5739E-1 (8.27E-2) -	6.0261E-1 (2.91E-2) -	8.1851E-1 (7.54E-2) -	1.7683E-1 (1.60E-2) -
DTLZ6	4	5.7666E-2 (1.87E-2)	4.1950E-1 (2.44E-1) -	5.5910E-2 (4.74E-3) =	2.2642E-1 (1.81E-1) -	4.7171E-2 (1.05E-2) +	1.7017E-1 (3.94E-2) -	1.2714E-1 (2.24E-1) -
	6	1.4606E-1 (4.07E-2)	8.7755E-1 (6.00E-1) -	1.9906E-1 (6.11E-2) -	6.5559E-1 (3.57E-1) -	2.2789E-1 (5.35E-2) -	3.2013E-1 (5.62E-2) -	2.8997E-1 (2.25E-1) -
	8	2.4125E-1 (1.09E-1)	1.9573E+0 (8.21E-1) -	3.2116E-1 (9.65E-2) -	8.6451E-1 (4.12E-1) -	2.4875E-1 (7.49E-2) =	8.4913E-1 (2.81E-1) -	3.8178E-1 (4.94E-1) -
	10	2.5337E-1 (7.45E-2)	2.7224E+0 (8.31E-1) -	6.9154E-1 (2.84E-1) -	2.6388E+0 (9.54E-1) -	5.1617E-1 (2.00E-1) -	9.6385E-1 (3.24E-1) -	2.9570E-1 (8.09E-1) -

表 2 Two-Arch/IS 和不同类型的 Two-Arch 算法在高维 DTLZ 问题上的 IGD 结果 (续)

测试问题	目标数目	Two-Arch/IS	Two-Arch	Two_Arch2	C-TAEA	TriMOEATAR	TwoArchM	TwoArchA
DTLZ6	15	3.1577E-1 (1.18E-1)	2.4017E+0 (9.89E-1) -	7.1754E-1 (4.51E-2) -	1.4377E+0 (6.65E-1) -	5.9438E-1 (7.82E-2) -	2.5042E+0 (4.47E-1) -	5.5365E-1 (7.39E-1) -
	20	3.4380E-1 (8.85E-2)	3.4643E+0 (9.11E-1) -	7.3121E-1 (3.63E-2) -	1.1282E+0 (5.85E-1) -	6.0164E-1 (2.65E-2) -	3.6909E+0 (4.63E-1) -	2.7440E+0 (4.45E-1) -
DTLZ7	4	2.8423E-1 (1.62E-1)	4.1166E-1 (8.39E-2) -	2.6806E-1 (2.06E-1) =	2.5328E-1 (1.19E-1) =	1.8205E-1 (1.07E-2) +	1.8730E-1 (5.97E-2) =	3.4208E-1 (4.82E-1) -
	6	4.8491E-1 (9.02E-2)	1.3473E+0 (4.07E-1) -	4.2941E-1 (7.38E-2) +	5.8028E-1 (3.66E-2) -	7.9714E-1 (6.09E-2) -	5.1981E-1 (7.96E-2) -	7.5987E-1 (1.08E+0) -
	8	8.2499E-1 (1.24E-1)	3.0244E+0 (6.60E-1) -	6.9199E-1 (1.17E-2) +	1.1269E+0 (3.64E-1) -	2.4717E+0 (4.56E-1) -	9.7448E-1 (1.55E-1) -	3.9448E+0 (4.76E-1) -
	10	9.9898E-1 (4.65E-2)	7.1970E+0 (2.11E+0) -	9.9219E-1 (3.11E-2) =	2.7645E+0 (1.15E+0) -	3.3916E+0 (3.19E-1) -	4.2119E+0 (7.43E-1) -	4.5873E+0 (1.37E+0) -
	15	1.4489E+0 (1.86E-2)	2.8875E+1 (7.04E+0) -	1.7074E+0 (9.30E-2) -	9.4468E+0 (1.04E+0) -	7.0799E+0 (3.69E-1) -	8.0030E+0 (7.55E-1) -	8.4441E+0 (1.76E+0) -
	20	1.8475E+0 (1.75E-2)	5.8841E+1 (4.62E+0) -	2.7388E+0 (7.34E-1) -	1.5784E+1 (2.00E-1) -	1.1099E+1 (6.99E-1) -	1.2806E+1 (9.26E-1) -	4.4271E+0 (2.10E+0) -
	+/-/-	-	1/3/38	8/6/28	1/4/37	6/6/30	6/5/31	5/2/35

从表 2 中观察到, Two-Arch/IS、Two-Arch、Two_Arch2、C-TAEA、TriMOEATAR、TwoArchM 和 TwoArchA 在这些测试实例上表现出最佳 IGD 均值的次数分别为 24、0、10、0、3、3 和 2, 其中 Two-Arch/IS 的表现优于其他算法。

此外, 根据 Wilcoxon 秩和检验结果来看, Two-Arch/IS 相对于 Two-Arch、Two_Arch2、C-TAEA、TriMOEATAR、TwoArchM 和 TwoArchA 算法的净胜得分 (Two-Arch/IS 结果表示为“+”的次数减去表示为“-”的次数) 分别为 37、20、36、24、25 和 30。因此, Two-Arch/IS 相较于其他类型的 Two-Arch 算法能够在高维测试问题上表现出更为突出的优势, 同时证明了 Two-Arch/IS 在同类型算法下的竞争力。在 DTLZ 的 42 个测试实例中, Two-Arch/IS 在 DTLZ5 和 DTLZ6 问题上 10 次表现出了最佳 IGD 值, 而在 DTLZ4 和 DTLZ7 问题上表现欠佳。DTLZ5 和 DTLZ6 均具有退化 PF, DTLZ5 的帕累托前沿连续且均匀, DTLZ6 问题则引入了更多的局部最优解, Two-Arch/IS 中均匀分布的参考向量能够很好地划分目标空间, 同时尽可能维持收敛性和多样性的平衡, 使得算法能够在复杂问题中有效搜索解空间, 在此类问题上表现出色。而 DTLZ4 和 DTLZ7 的 PF 表现出非均匀分布、不连通的特性, 因此 Two-Arch/IS 中的参考向量难以均匀分割 PF, 算法性能表现出了局限性。

从上述实验结果来看, 相较于现有的 Two-Arch 变体算法, Two-Arch/IS 算法展现出了其独特的竞争力, 表现出较好的 IGD 优势。但在部分情况下, IGD 值并不能完全显示解集的优劣, 因此为了进一步展示 Two-Arch/IS 算法的效果, 图 6 分别为 7 种算法在 10 目标的 DTLZ1 问题上获得的种群的平行坐标图。从图 6 可以观察到, 在 DTLZ1(10) 测试实例上, Two-Arch 算法得到的种群在 1-3 目标上分布欠佳; Two_Arch2、C-TAEA TwoArchM 和 TwoArchA 算法在不同目标间表现不够均衡, 并在个别目标上表现出了极大值, 其中 C-TAEA 的综合表现最为逊色。TwoArchM 和 TwoArchA 的收敛性表现略有不足, 种群中个体的目标值较大, 其中 TwoArchM 综合表现更差; Two-Arch/IS 和 TriMOEATAR 算法得到的种群则展现出了良好的收敛性和多样性, 各目标维度上表现相对均衡, 但 Two-Arch/IS 在各个目标上的收敛性均表现更为出色, 是综合表现最好的算法。

(2) 与流行高维多目标进化算法在 DTLZ 测试问题上的比较

为探究 Two-Arch/IS 在处理高维多目标优化问题时的性能表现, 选择 7 种不同类型的流行高维多目标进化算法, 包括 NSGA-III、KnEA、IBEA、HEA、TSNSGA-II、MOEA/D 以及 SMS-EMOA, 通过比较算法在处理不同类型问题时的表现, 深入比较 Two-Arch/IS 的优缺点。实验在 4-20 目标的 DTLZ1-7 问题上进行, 实验结果如表 3 所示。

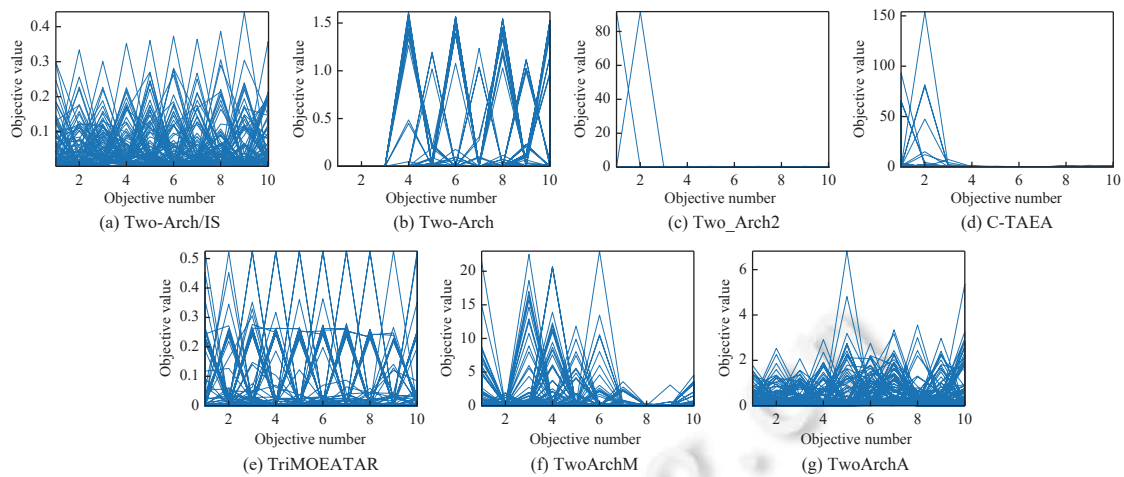


图 6 Two-Arch/IS 和不同类型的 Two-Arch 算法在 10 目标 DTLZ1 问题上的种群平行坐标图

表 3 Two-Arch/IS 和其他高维多目标进化算法在高维 DTLZ 问题上的 IGD 结果

测试问题	目标数目	Two-Arch/IS	NSGA-III	KnEA	IBEA	HEA	TSNSGA-II	MOEA/D	SMS-EMOA
DTLZ1	4	4.9339E-2 (1.96E-2)	4.9557E-2 (3.70E-3)	1.1673E-1 (5.15E-2)	2.3432E-1 (2.90E-2)	4.8849E-2 (6.53E-4)	5.1405E-2 (5.94E-3)	6.2153E-2 (5.15E-2)	1.9058E-1 (2.22E-1)
	6	9.7828E-2 (3.44E-2)	1.2540E-1 (5.94E-2)	2.0587E-1 (9.59E-2)	2.6687E-1 (2.47E-2)	9.2448E-2 (2.49E-2)	9.9527E-2 (4.78E-3)	9.8877E-2 (5.52E-2)	2.2403E+0 (2.22E+0)
	8	1.5939E-1 (7.08E-2)	2.1276E-1 (1.45E-1)	5.4815E+0 (3.95E+0)	2.9228E-1 (3.08E-2)	1.1615E-1 (3.33E-2)	1.9086E-1 (5.12E-2)	1.2975E-1 (4.87E-2)	9.6447E+0 (8.93E+0)
	10	1.8356E-1 (7.37E-2)	4.8366E-1 (3.57E-1)	8.2897E+0 (6.08E+0)	3.2506E-1 (3.00E-2)	3.4630E-1 (2.95E-1)	2.6442E-1 (6.81E-2)	1.2462E-1 (2.08E-2)	1.0657E+1 (8.06E+0)
	15	1.3163E-1 (2.73E-3)	1.8638E-1 (2.08E-2)	6.3392E+0 (8.47E+0)	3.1762E-1 (3.61E-2)	1.9248E-1 (2.75E-2)	3.3986E-1 (2.28E-2)	1.7276E-1 (1.84E-2)	3.5431E+0 (2.50E+0)
	20	1.7919E-1 (7.45E-3)	3.5380E-1 (9.04E-2)	1.3528E+1 (1.07E+1)	4.0626E-1 (1.69E-2)	2.0535E-1 (6.37E-3)	6.8319E-1 (3.20E-1)	2.8527E-1 (1.67E-1)	4.3477E+0 (3.34E+0)
DTLZ2	4	1.5525E-1 (3.11E-3)	1.4032E-1 (1.15E-5)	1.5067E-1 (4.34E-3)	1.5805E-1 (1.68E-3)	1.4032E-1 (8.85E-5)	1.4065E-1 (1.66E-4)	1.4036E-1 (2.12E-5)	1.6290E-1 (4.05E-3)
	6	2.9212E-1 (2.87E-3)	2.7946E-1 (1.35E-4)	3.1294E-1 (1.28E-2)	2.9499E-1 (3.98E-3)	2.8007E-1 (5.42E-4)	2.8235E-1 (9.45E-4)	2.7836E-1 (4.69E-4)	3.0449E-1 (4.52E-3)
	8	3.9197E-1 (2.74E-3)	4.2221E-1 (6.55E-2)	4.1931E-1 (1.03E-2)	4.0198E-1 (4.84E-3)	3.8930E-1 (1.14E-3)	3.9494E-1 (2.79E-3)	3.8472E-1 (4.57E-4)	4.7517E-1 (9.28E-2)
	10	4.7686E-1 (2.90E-3)	5.6781E-1 (5.50E-2)	5.1911E-1 (1.37E-2)	5.0820E-1 (5.01E-3)	5.3775E-1 (2.74E-3)	5.6162E-1 (1.31E-2)	5.0086E-1 (6.30E-3)	7.4192E-1 (4.48E-2)
	15	5.4409E-1 (4.84E-3)	6.4502E-1 (1.76E-2)	6.6806E-1 (1.65E-1)	5.7838E-1 (6.64E-3)	7.1403E-1 (5.65E-3)	1.5624E+0 (4.09E-1)	1.1118E+0 (8.20E-2)	9.3585E-1 (5.25E-2)
	20	6.2367E-1 (2.29E-3)	1.0125E+0 (3.49E-2)	7.5457E-1 (1.58E-1)	6.7514E-1 (1.80E-2)	7.6454E-1 (2.69E-3)	2.0314E+0 (6.41E-1)	8.3060E-1 (1.24E-2)	1.0440E+0 (2.85E-2)
DTLZ3	4	4.8838E+0 (2.05E+0)	1.1456E+0 (1.30E+0)	8.2960E-1 (8.46E-1)	6.1068E-1 (1.91E-1)	2.5000E-1 (2.62E-1)	4.4704E-1 (4.68E-1)	5.1470E+0 (5.45E+0)	4.0480E+0 (2.90E+0)
	6	4.4899E+0 (1.70E+0)	5.3930E+0 (2.78E+0)	1.2224E+1 (4.95E+0)	7.6390E-1 (3.66E-1)	5.5485E-1 (3.72E-1)	2.4777E+0 (1.34E+0)	3.7847E+0 (4.55E+0)	6.9844E+1 (3.16E+1)
	8	4.5326E+0 (1.95E+0)	7.4970E+0 (4.56E+0)	1.5586E+2 (4.82E+1)	7.8764E-1 (2.41E-1)	7.9166E-1 (4.63E-1)	4.8627E+0 (2.98E+0)	4.4043E+0 (3.25E+0)	1.1608E+2 (3.74E+1)
	10	5.7788E+0 (1.67E+0)	1.3058E+1 (6.28E+0)	3.5237E+2 (9.49E+1)	9.8908E-1 (3.75E-1)	1.0207E+0 (7.30E-1)	1.1255E+1 (7.65E+0)	1.8070E+0 (9.99E-1)	1.0782E+2 (3.65E+1)
	15	1.5993E+0 (8.97E-1)	5.5582E+0 (3.38E+0)	7.2278E+2 (1.17E+2)	8.1705E-1 (2.07E-2)	9.4543E-1 (9.45E-2)	3.6017E+2 (2.85E+2)	2.3254E+0 (1.49E+0)	8.4731E+1 (1.69E+1)

表3 Two-Arch/IS 和其他高维多目标进化算法在高维 DTLZ 问题上的 IGD 结果 (续)

测试问题	目标数目	Two-Arch/IS	NSGA-III	KnEA	IBEA	HEA	TSNSGA-II	MOEA/D	SMS-EMOA
DTLZ3	20	1.2400E+0 (1.13E+0)	1.3177E+1 (8.32E+0) -	7.0634E+2 (1.46E+2) -	1.0566E+0 (4.02E-2) +	1.0200E+0 (1.24E-1) +	5.7914E+2 (4.63E+2) -	3.2063E+0 (2.19E+0) -	9.7090E+1 (1.88E+1) -
	4	1.8505E-1 (9.13E-2)	2.3604E-1 (1.49E-1) -	1.5116E-1 (5.38E-3) +	1.9547E-1 (1.51E-1) =	4.4026E-1 (2.34E-1) -	1.4117E-1 (3.26E-4) +	4.4896E-1 (2.62E-1) -	2.8439E-1 (1.49E-1) -
	6	3.2476E-1 (5.94E-2)	4.0888E-1 (1.07E-1) =	3.0292E-1 (5.46E-3) +	3.6168E-1 (9.83E-2) =	5.0194E-1 (1.49E-1) -	2.8915E-1 (3.12E-3) +	6.1389E-1 (1.70E-1) -	4.5825E-1 (1.04E-1) -
DTLZ4	8	4.1051E-1 (3.24E-2)	4.9653E-1 (8.88E-2) =	4.0816E-1 (5.25E-3) +	4.5910E-1 (6.45E-2) -	5.5569E-1 (1.24E-1) -	4.0290E-1 (8.25E-3) +	7.5323E-1 (1.76E-1) -	5.2703E-1 (7.78E-2) -
	10	4.9340E-1 (1.69E-2)	5.9139E-1 (5.80E-2) -	5.1116E-1 (6.05E-3) -	5.3693E-1 (4.58E-2) -	6.9389E-1 (1.03E-1) -	5.7143E-1 (1.84E-2) -	8.4894E-1 (1.20E-1) -	6.1417E-1 (5.65E-2) -
	15	5.6024E-1 (2.17E-3)	6.4324E-1 (1.69E-2) -	5.6668E-1 (4.11E-3) -	5.8213E-1 (3.97E-3) -	8.6827E-1 (8.80E-2) -	1.8669E+0 (4.50E-1) -	1.0643E+0 (1.22E-1) -	8.5864E-1 (1.50E-1) -
	20	6.1600E-1 (1.17E-3)	9.7055E-1 (3.53E-2) -	6.2212E-1 (2.76E-3) -	6.7551E-1 (1.04E-2) -	8.5697E-1 (5.75E-2) -	1.8520E+0 (4.47E-1) -	1.1450E+0 (1.60E-1) -	1.3938E+0 (1.41E-1) -
	4	3.6911E-2 (6.06E-3)	5.3657E-2 (8.68E-3) -	1.7661E-1 (8.42E-2) -	4.6158E-2 (4.77E-3) -	6.9234E-2 (6.41E-3) -	9.2373E-2 (1.62E-2) -	3.6030E-2 (2.01E-4) =	3.6174E-2 (8.38E-3) =
DTLZ5	6	8.7485E-2 (1.50E-2)	1.2939E-1 (4.50E-2) -	2.3360E-1 (5.59E-2) -	7.5383E-2 (1.44E-2) +	2.0267E-1 (2.70E-2) -	3.0272E-1 (1.55E-1) -	3.4729E-2 (2.01E-3) +	6.3997E-2 (1.29E-2) +
	8	1.1496E-1 (1.89E-2)	2.6392E-1 (8.64E-2) -	2.8287E-1 (6.90E-2) -	1.3056E-1 (4.92E-2) =	2.5294E-1 (6.03E-2) -	3.2348E-1 (1.52E-1) -	5.2388E-2 (6.02E-3) +	9.4036E-2 (3.33E-2) +
	10	1.3084E-1 (1.93E-2)	2.3372E-1 (7.78E-2) -	3.9692E-1 (1.19E-1) -	1.9989E-1 (6.10E-2) -	2.1871E-1 (5.79E-2) -	3.5588E-1 (1.72E-1) -	7.3741E-2 (1.17E-4) +	1.9587E-1 (4.60E-2) -
	15	1.4586E-1 (2.37E-2)	2.4525E-1 (3.61E-2) -	4.9301E-1 (1.84E-1) -	2.4533E-1 (7.14E-2) -	3.2466E-1 (5.53E-2) -	4.8497E-1 (1.50E-1) -	3.2008E-1 (1.68E-1) -	2.4171E-1 (5.08E-2) -
	20	1.4888E-1 (4.07E-2)	1.1795E+0 (5.69E-1) -	7.8404E-1 (3.18E-1) -	2.7405E-1 (6.62E-2) -	3.5808E-1 (5.63E-2) -	4.7694E-1 (1.80E-1) -	2.3750E-1 (6.34E-3) -	2.5476E-1 (8.38E-2) -
	4	5.7666E-2 (1.87E-2)	1.0200E-1 (3.90E-2) -	2.0850E-1 (1.08E-1) -	5.4058E-2 (6.24E-3) =	1.4377E-1 (4.01E-2) -	2.2188E-1 (1.13E-1) -	3.5837E-2 (7.54E-4) +	6.0860E-1 (3.43E-1) -
DTLZ6	6	1.4606E-1 (4.07E-2)	9.8979E-1 (6.35E-1) -	6.9817E-1 (2.68E-1) -	1.5718E-1 (6.84E-2) =	2.7116E-1 (1.17E-1) -	5.6079E-1 (3.28E-1) -	2.1399E-1 (6.25E-1) -	9.0398E-1 (3.94E-1) -
	8	2.4125E-1 (1.09E-1)	1.9782E+0 (9.57E-1) -	1.5879E+0 (5.36E-1) -	2.6959E-1 (9.62E-2) =	4.1867E-1 (1.15E-1) -	8.4134E-1 (4.96E-1) -	2.4353E-1 (4.51E-1) =	1.6069E+0 (4.31E-1) -
	10	2.5337E-1 (7.45E-2)	2.4700E+0 (9.62E-1) -	1.6960E+0 (6.04E-1) -	3.6169E-1 (6.21E-2) -	8.6637E-1 (6.52E-1) -	6.1502E-1 (3.30E-1) -	9.8692E-2 (1.11E-1) +	1.9697E+0 (3.33E-1) -
	15	3.1577E-1 (1.18E-1)	9.5553E-1 (4.80E-1) -	1.9657E+0 (4.08E-1) -	3.4408E-1 (4.89E-2) -	6.1859E-1 (6.70E-2) -	4.1840E+0 (1.49E+0) -	3.7859E-1 (1.99E-1) -	2.7613E+0 (5.05E-1) -
	20	3.4380E-1 (8.85E-2)	5.0266E+0 (1.49E+0) -	1.7892E+0 (4.24E-1) -	3.8950E-1 (5.41E-2) -	6.2069E-1 (1.20E-1) -	3.7691E+0 (1.80E+0) -	2.3879E-1 (1.84E-5) +	3.6442E+0 (7.39E-1) -
	4	2.8423E-1 (1.62E-1)	2.4341E-1 (6.70E-2) =	1.8077E-1 (9.27E-2) =	2.9288E-1 (1.52E-1) -	4.7072E-1 (2.63E-1) -	2.4657E-1 (7.43E-2) +	4.4063E-1 (1.83E-1) -	5.3515E-1 (4.39E-1) -
DTLZ7	6	4.8491E-1 (9.02E-2)	5.2044E-1 (2.15E-2) -	4.1047E-1 (1.13E-2) +	6.0894E-1 (1.79E-1) -	1.8803E+0 (9.14E-1) -	4.8930E-1 (9.74E-3) =	1.2888E+0 (2.75E-1) -	2.1103E+0 (6.09E-1) -
	8	8.2499E-1 (1.24E-1)	9.9353E-1 (8.44E-2) -	7.6044E-1 (2.47E-2) =	1.0877E+0 (3.82E-1) -	4.3103E+0 (8.01E-1) -	1.4236E+0 (1.52E-1) -	1.6565E+0 (2.37E-1) -	4.0956E+0 (3.60E-1) -
	10	9.9898E-1 (4.65E-2)	2.5060E+0 (5.02E-1) -	1.2056E+0 (5.96E-2) -	1.8369E+0 (6.34E-1) -	3.8659E+0 (1.19E+0) -	3.1519E+0 (1.18E-1) -	2.0840E+0 (2.95E-1) -	5.6648E+0 (2.78E-1) -
	15	1.4489E+0 (1.86E-2)	8.0893E+0 (9.24E-1) -	2.4294E+0 (2.73E-1) -	3.9662E+0 (1.36E+0) -	1.1266E+1 (7.20E-1) -	9.1522E+0 (8.70E-1) -	6.1500E+0 (8.49E-1) -	1.3520E+1 (1.34E+0) -
	20	1.8475E+0 (1.75E-2)	1.4771E+1 (5.47E-1) -	4.2138E+0 (7.29E-1) -	1.1461E+1 (1.53E+0) -	1.4789E+1 (1.04E+0) -	1.3918E+1 (8.52E-1) -	3.4546E+0 (4.08E-1) -	8.0875E+1 (1.64E+1) -
+/-/-		-	3/4/35	6/3/33	7/7/28	11/1/30	8/2/32	12/4/26	3/2/37

从表 3 可以看出, Two-Arch/IS、NSGA-III、KnEA、IBEA、HEA、TSNSGA-II、MOEA/D 和 SMS-EMOA 算法各自获得最佳 IGD 均值的次数分别为 16, 1, 3, 3, 6, 3, 10, 0, 其中 Two-Arch/IS 算法排名第 1, 表现出优异的 IGD 性能. 此外, 根据 Wilcoxon 秩和检验结果, 在 42 个测试实例上, Two-Arch/IS 相对于其他算法 NSGA-III、KnEA、IBEA、HEA、TSNSGA-II、MOEA/D 和 SMS-EMOA 的净胜得分分别为 32、27、21、19、24、14、34. 因此, 在求解高维多目标优化问题时, Two-Arch/IS 相较于该领域的经典算法, 展现出了较大的 IGD 优势, 充分证明了其优越性. 值得注意的是, 随着目标维度的增大, Two-Arch/IS 的表现也在提升, 在大多 10–20 维度问题上, Two-Arch/IS 获得了最佳 IGD, 但在 DTLZ3 问题中, Two-Arch/IS 表现存在不足. Two-Arch/IS 的双归档和种群划分策略能够有效管理种群的收敛性与多样性, 避免早期收敛和局部最优. 随着目标维度的增加, 算法通过适应性强的信息补偿机制在复杂的目标空间中维持收敛性与多样性的平衡, 保证解的质量. 在非凸和不可微的 DTLZ3 问题上, IBEA 和 HEA 算法的表现都超越了 Two-Arch/IS. IBEA 算法具有独特的指标设置, 使其在处理难以收敛的问题时能够表现出色. HEA 算法能够动态调整算法选择压力, 同时平衡收敛性和多样性, 因此在具有多个局部最优的 DTLZ3 问题上表现良好. Two-Arch/IS 在利用权重向量划分目标空间时, 由于复杂的帕累托前沿形状, 导致划分无法准确地反映目标空间的实际结构, 从而影响算法的性能. 此外, 由于 DTLZ3 目标函数之间的复杂交互, SDE 评价指标可能无法准确评估解的质量, 从而影响种群的收敛性与多样性. 尽管如此, 在难以收敛的 DTLZ3 问题上, Two-Arch/IS 仅次于 IBEA 和 HEA, 排名第 3.

从上述实验结果来看, 在高维多目标优化问题上, Two-Arch/IS 算法相较于其他不同类别的算法都能够表现出相对较好的综合性能. 此外, 根据表 2 和表 3 的实验数据, 在大多数情况下, 随着目标维度的递增, Two-Arch/IS 算法的 IGD 优势愈发突出, 尤其在目标数超过 10 的情况下, 其性能表现具有绝对的优势. 为进一步展示种群中个体在不同目标维度上的分布情况, 图 7 给出了 8 种高维多目标进化算法在 10 目标的 DTLZ1 问题上获得的种群的平行坐标图. 从图 7 中可以观察到, NSGA-III、TSNSGA-II 和 SMS-EMOA 算法在各目标上表现不够均衡, 存在明显的极大值, 其中 SMS-EMOA 算法表现更为逊色; KnEA 算法得到的种群个体在各目标维度上的目标值普遍较大, 显示出较差的收敛性; IBEA 算法产生的有效解数量相对较少, 并且在 4、6 和 10 目标上分布性欠佳; HEA 算法得到的种群在部分目标上出现了极大值; MOEA/D 算法在不同目标维度上的表现存在差异, 分布性不足; 相比之下, Two-Arch/IS 算法展现出了最好的收敛性和多样性, 其种群个体在各个目标上的分布较为均衡, 综合表现较为出色.

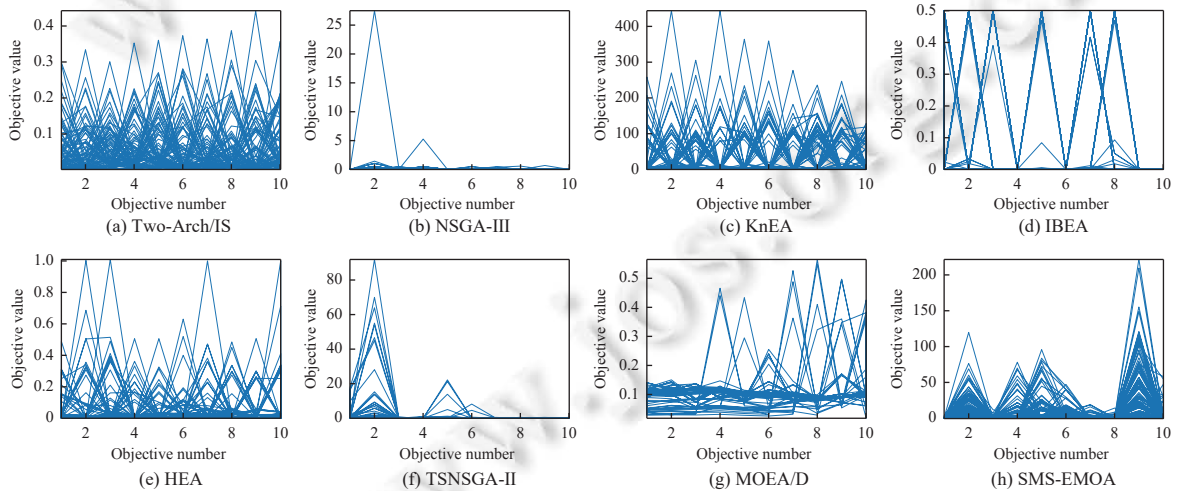


图 7 Two-Arch/IS 和对比高维多目标进化算法在 10 目标 DTLZ1 问题上的种群平行坐标图

(3) 在组合优化问题 MOKP 和 MLDMP 上的比较

为了测试 Two-Arch/IS 算法在实际组合优化问题中的表现, 选择了 4 种代表性的算法进行对比, 包括 Two_Arch2、IBEA、MOEA/D 以及 NSGA-III, 在 4–10 目标的 MOKP 和 MLDMP 问题上进行 IGD 结果比较, 实

验结果如表 4 所示.

表 4 Two-Arch/IS 和其他代表性算法在实际测试问题 MOKP 和 MLDMP 上的 IGD 结果

测试问题	目标数目	Two-Arch/IS	Two_Arch2	IBEA	MOEA/D	NSGA-III
MOKP	4	1.6502E+4 (8.38E+1)	1.6639E+4 (5.46E+1) –	1.6837E+4 (6.44E+1) –	1.6829E+4 (5.69E+1) –	1.6692E+4 (3.12E+1) –
	6	1.9283E+4 (8.77E+1)	1.9049E+4 (1.36E+2) +	1.9650E+4 (8.82E+1) –	1.9722E+4 (9.37E+1) –	1.9469E+4 (1.02E+2) –
	8	2.1088E+4 (1.11E+2)	2.1440E+4 (9.52E+1) –	2.1918E+4 (6.26E+1) –	2.2048E+4 (8.81E+1) –	2.1595E+4 (8.42E+1) –
	10	2.3323E+4 (1.19E+2)	2.2685E+4 (1.17E+2) +	2.3783E+4 (1.17E+2) –	2.4019E+4 (9.98E+1) –	2.3426E+4 (9.91E+1) =
MLDMP	4	9.3524E-2 (8.33E-4)	7.3231E-1 (1.57E-1) –	4.9898E-1 (5.44E-2) –	1.4598E-1 (2.15E-3) –	2.4325E-1 (3.48E-2) –
	6	1.3437E-1 (2.09E-3)	1.0586E+0 (1.44E-1) –	7.7357E-1 (1.71E-2) –	2.1658E-1 (3.68E-3) –	5.5142E-1 (1.14E-1) –
	8	1.7132E-1 (2.18E-3)	1.0759E+0 (1.35E-1) –	8.2829E-1 (7.61E-3) –	3.9780E-1 (3.00E-3) –	5.6489E-1 (1.13E-1) –
	10	2.0188E-1 (3.50E-3)	1.1266E+0 (1.48E-1) –	8.6806E-1 (1.07E-2) –	7.5015E-1 (2.64E-2) –	9.1564E-1 (1.77E-1) –
+/-/-		—	2/0/6	0/0/8	0/0/8	0/1/7

根据表 4 中的结果, 可以观察到, Two-Arch/IS 拥有最好的综合表现, 无论是与单个算法的对比, 还是在 IGD 最佳值出现的次数上, Two-Arch/IS 均位居第 1. 在 MOKP 问题中, 目标之间冲突性较强, 尽管 Two-Arch/IS 在算法设计上尽可能缓解收敛性和多样性的冲突, 但由于背包容量限制导致解空间本身较紧凑, 因此各算法在性能上的差距不大, 均能找到较好的折中解. MLDMP 具有相对较强的几何结构性和更复杂的解空间, Two-Arch/IS 的优势更加明显, 得益于其能够更好地处理几何结构性问题和优化目标之间的关系, 充分发挥了双档案机制和信息交流的优势, 从而超越了其他算法.

3.4 低维目标问题上的对比实验结果与分析

上述实验结果表明, Two-Arch/IS 算法在求解高维多目标优化问题时, 在 IGD 值表现、收敛性以及多样性方面均展现出了很好的优越性. 进一步地, 为了检验其在低维目标问题上的性能, 选取 4 种算法在 ZDT 测试问题及低维 DTLZ 测试问题上进行性能对比, 以证明其具有较好的适应性.

表 5 列出了 4 种算法在 5 个 ZDT 测试实例和 14 个 DTLZ 测试实例上获得的 IGD 均值与方差, Two-Arch/IS、KnEA、IBEA、GrEA 分别在 10、3、2、4 个测试实例上表现出最佳 IGD 均值. 根据 Wilcoxon 秩和检验结果来看, Two-Arch/IS 在整体表现上优于其他 3 种算法 KnEA、IBEA 和 GrEA, 在 4 种算法中整体排名第 1. 在 ZDT4 和 DTLZ7 上, Two-Arch/IS 的 IGD 值远低于其他算法, 这是由于其双档案机制能有效平衡收敛性和多样性, 因此能够解决存在一定挑战的具有局部最优或离散帕累托前沿的问题. 然而, 在 DTLZ3 问题上, Two-Arch/IS 在 4 种算法中表现最差, IGD 均值和方差均较高, 这是由于 DTLZ3 问题具有极强的欺骗性, 并包含大量局部帕累托前沿, 使得算法难以区分全局解, 导致最终获得较差的结果. 此外, KnEA 算法的核心优势在于利用 knee points 驱动以平衡收敛与多样性, 但其在简单或离散问题上表现一般, 且稳定性不足. IBEA 算法依赖指标推进种群进化, 但在部分情况下可能过度强调收敛, 导致分布性不足. GrEA 算法在低维问题下能够通过网格划分增强向帕累托前沿的选择压力, 但其难以区分局部与全局最优.

表 5 Two-Arch/IS 和其他算法在低维问题上获得的 IGD 结果

测试问题	目标数目	Two-Arch/IS	KnEA	IBEA	GrEA
ZDT1	2	5.4132E-3 (4.62E-4)	1.2249E-1 (6.82E-2) –	4.1205E-3 (5.78E-5) +	7.2383E-3 (7.21E-4) –
ZDT2	2	8.2739E-3 (2.20E-3)	9.8937E-2 (2.06E-2) –	8.2097E-3 (6.77E-4) =	7.8461E-3 (1.57E-4) =
ZDT3	2	1.0245E-2 (8.91E-3)	1.0162E-2 (5.16E-3) +	1.9056E-2 (8.01E-3) –	1.3732E-2 (7.97E-4) –
ZDT4	2	5.4707E-3 (9.14E-4)	2.6062E-1 (6.93E-2) –	9.4421E-2 (1.02E-1) –	2.7327E-1 (1.26E-1) –
ZDT6	2	3.7037E-3 (2.60E-4)	8.0300E-3 (1.82E-3) –	4.4460E-3 (1.03E-4) –	5.8863E-3 (1.77E-4) –
DTLZ1	2	3.9648E-3 (2.85E-3)	4.5012E-3 (3.30E-3) –	7.3505E-2 (1.14E-2) –	1.2699E-2 (2.38E-2) –
	3	4.8587E-2 (8.33E-2)	5.4960E-2 (3.22E-2) –	1.6587E-1 (3.28E-2) –	7.1659E-2 (7.17E-2) –
DTLZ2	2	1.3080E-2 (3.51E-3)	4.4838E-2 (1.56E-2) –	1.6584E-2 (1.60E-3) –	1.0570E-2 (7.48E-5) +
	3	7.5617E-2 (2.74E-3)	6.6327E-2 (3.05E-3) +	8.0821E-2 (2.48E-3) –	6.4057E-2 (6.70E-4) +

表 5 Two-Arch/IS 和其他算法在低维问题上获得的 IGD 结果 (续)

测试问题	目标数目	Two-Arch/IS	KnEA	IBEA	GrEA
DTLZ3	2	4.0221E+0 (2.74E+0)	1.5605E-1 (3.57E-1)+	3.7296E-1 (1.50E-1) +	1.4562E-1 (3.18E-1) +
	3	4.8346E+0 (3.12E+0)	1.6175E-1 (1.95E-1) +	4.8071E-1 (7.62E-3) +	2.6765E-1 (3.38E-1) +
DTLZ4	2	1.0370E-2 (2.40E-3)	1.8314E-1 (3.14E-1) -	3.3046E-1 (3.66E-1) -	8.3635E-2 (2.23E-1) =
	3	9.0088E-2 (8.53E-2)	9.4265E-2 (1.61E-1) -	8.0132E-2 (2.92E-3) +	1.4634E-1 (1.81E-1) -
DTLZ5	2	1.0414E-2 (2.44E-3)	4.5989E-2 (1.35E-2) -	1.6464E-2 (1.36E-3) -	1.0558E-2 (1.59E-4) =
	3	1.1542E-2 (2.10E-3)	9.3796E-3 (1.62E-3) +	1.6437E-2 (1.38E-3) -	2.1783E-2 (1.36E-3) -
DTLZ6	2	9.3458E-3 (1.02E-3)	7.4819E-2 (1.54E-2) -	2.9108E-2 (2.96E-3) -	1.0710E-2 (4.12E-5) -
	3	9.4169E-3 (1.09E-3)	1.2975E-2 (8.02E-3) =	2.6852E-2 (3.20E-3) -	2.2272E-2 (1.79E-4) -
DTLZ7	2	6.4720E-3 (7.98E-4)	3.2909E-2 (7.83E-2) -	4.9855E-2 (1.34E-1) -	4.1663E-2 (7.71E-2) -
	3	6.0493E-2 (3.98E-3)	9.4944E-2 (8.81E-2) -	1.4384E-1 (1.30E-1) -	1.1471E-1 (9.16E-2) -
+/-/-		-	5/1/13	4/1/14	4/3/12

为进一步展示种群中个体的分布情况, 选择和本文方法较为类似的 Two_Arch2 算法作为对比, 图 8 给出了 Two-Arch/IS 和 Two_Arch2 算法在 2 目标和 3 目标 DTLZ1 测试问题上的种群个体分布图. 从图 8(a) 和图 8(b) 可以观察到, 在 2 目标的 DTLZ1 问题上, Two-Arch/IS 算法展现出了更好的多样性, 个体均匀覆盖在帕累托前沿面上, 而 Two_Arch2 算法则存在局部区域未收敛. 从图 8(c) 和图 8(d) 可以观察到, 在 3 目标的 DTLZ1 问题上, Two-Arch/IS 依旧展现出了较好的多样性, 而 Two_Arch2 算法多样性较差.

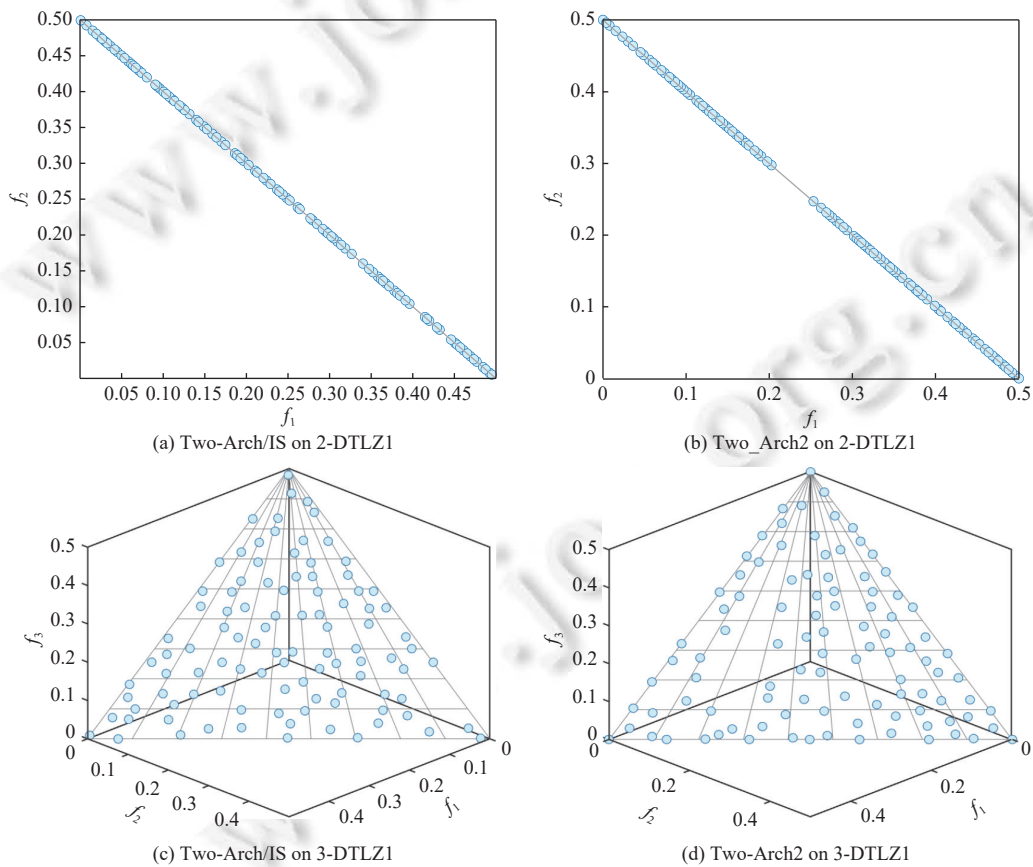


图 8 Two-Arch/IS 和 Two_Arch2 在 2 目标和 3 目标 DTLZ1 问题上的种群个体分布图

3.5 消融实验分析

为了验证 Two-Arch/IS 各组成部分的有效性, 我们基于指标 $GD^{[49]}$ 和 $IGD^{[48]}$ 进行了不同侧重的度量, 在 10 目标的 DTLZ3 问题上进行了测试, 绘制 GD 、 IGD 收敛曲线图, 如图 9 所示. 图中 5 条曲线分别代表以下 5 种算法: (1) Two-Arch/IS: 即本文提出的完整算法; (2) Nosub (对应引言中贡献 1): 去除基于空间划分的子种群更新策略的 Two-Arch/IS 变种算法, 即在该算法中, 移除了权重向量划分种群的操作, 在整个档案库中进行更新; (3) Nochange (对应引言中贡献 3): 去除基于边界解驱动的信息补偿机制的 Two-Arch/IS 变种算法, 该算法省略了截断过程之后的子代迁移机制; (4) Alt-divide (对应引言中贡献 1): 算法划分子种群的方法使用第 2.3 节提到的另一种复杂度较低的方法; (5) Delete (对应引言中贡献 2): 使用原 Two-Arch 算法截断方式的 Two-Arch/IS 变种算法, 即 CA 和 DA 总大小保持为 N , 仅对 DA 截断. 图 9(a) 展示了 5 种算法在 10 目标的 DTLZ3 问题上的 GD 收敛曲线, GD 值用于衡量算法生成的解集与真实 PF 之间的距离, GD 值越小表示算法收敛性越好. 图 9(b) 展示了 5 种算法在相同问题上的 IGD 收敛曲线, IGD 是综合性的评价指标.

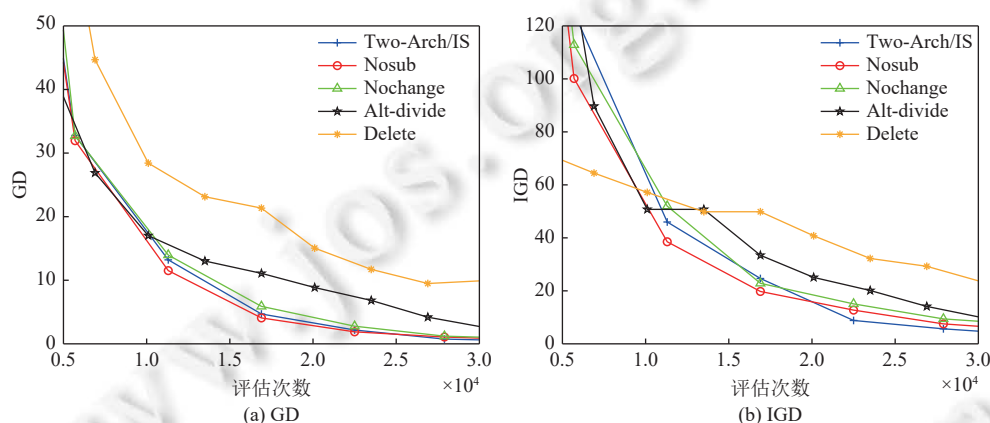


图 9 5 种算法在 10 目标 DTLZ3 问题上的 GD 和 IGD 收敛曲线

当评估次数介于 20000–25000 之间时, Nosub 在 GD 指标上的收敛性表现优于 Two-Arch/IS, 但 IGD 综合表现劣于 Two-Arch/IS, 这表明 Nosub 的多样性表现严重劣于 Two-Arch/IS, 由此可以推断, 基于空间划分的子种群更新策略在维护种群多样性方面发挥了关键作用. 除此之外, 通过对比最终收敛得到的 GD 值, 我们发现 Two-Arch/IS 的收敛性优于 Nosub, 这意味着子种群更新策略对算法收敛性也产生了积极影响.

另一方面, 从 Nochange 和 Two-Arch/IS 的 GD 值对比曲线, 我们可以清晰地发现, Two-Arch/IS 在收敛性上表现出了较强的优势, 这证明了 Two-Arch/IS 引入的基于边界解驱动的信息补偿机制对种群收敛性的促进作用. 通过进一步对比观察 IGD 曲线, 我们发现在种群进化后期阶段, 信息补偿机制发挥的作用尤为突出, 在加速种群收敛速度的同时, 也有效增强了算法的综合性能, 这充分证明了档案库之间信息交流的有效性.

对比 Alt-divide 和 Two-Arch/IS 算法, 观察到 Alt-divide 算法在 GD 和 IGD 指标上的表现均较差, 显示出算法在收敛性与多样性方面均存在不足. 这表明不合理的子种群划分方法无法有效促进算法的性能. 通过观察 Delete 与 Two-Arch/IS 算法, 在种群进化前期, Delete 算法的 IGD 值表现出最佳, 这是因为 Delete 算法仅对 DA 截断, 能够更快地收敛到局部最优解, 从而在早期表现出更好的 IGD 值. 但从整体来看, Delete 算法在 GD 和 IGD 指标上的表现均较差, 表明 Two-Arch/IS 算法的截断机制展现出更强的综合效果.

3.6 收敛性实验分析

为直观地考察算法在高维问题下的收敛速度, 图 10 展示了各算法在 10 目标的 DTLZ2 及 10 目标的 DTLZ6 测试实例上的 IGD 收敛曲线.

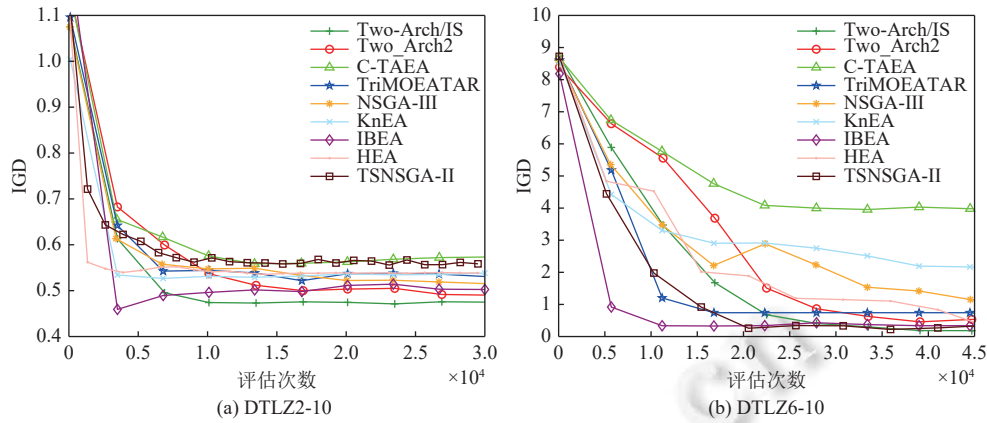


图 10 9 种算法在不同问题上获得的 IGD 收敛曲线

从图 10(a) 可以看出, 随着评估次数的增长, 9 种算法的 IGD 值曲线表现较为相似, 呈快速下降直至某阈值后趋于稳定. 在初始进化阶段, HEA 算法收敛速度最快, 其次是 TSNSGA-II 算法, 但两种算法最终的 IGD 表现较差, 表明算法可能在后期陷入局部最优. IBEA 算法也具有较快的收敛速度, 但后期 IGD 值表现甚至变差, 原因可能是种群多样性不足导致的搜索范围受限, 因此无法逃离局部最优. TriMOEATAR 算法在种群进化中期表现出明显的波动, 表明算法稳定性仍需提高. Two-Arch/IS 算法在初始阶段表现出较为快速的下降趋势, 并在评估次数约为 0.75×10^4 时, IGD 值达到 9 种算法中最优, 且后续趋于平稳, 显示出良好的全局搜索能力和持续优化能力.

从图 10(b) 可以看出, 9 种算法的 IGD 值整体呈下降趋势, 其中部分算法存在小幅波动. 相对而言, IBEA 算法因其独特的指标设置, IGD 值下降最快, 收敛速度表现较为突出; 其次收敛速度较快的算法是 TSNSGA-II 和 TriMOEATAR, 其后是 Two-Arch/IS 以及 HEA 算法; NSGA-III 算法的 IGD 值曲线呈现出快速下降, 随后上升再下降的波动, 最终 IGD 值表现并不突出; 而 C-TAEA 和 KnEA 算法的 IGD 值曲线下下降缓慢, 并且最终处于一个较大的数值, 表现较差. 在评估次数超过 3×10^4 次之后, Two-Arch/IS 算法的 IGD 值仍在小幅度下降, 最终获得最低的 IGD 值. 从整体的表现看来, Two-Arch/IS 算法虽然没有最快的收敛速度, 但最终的 IGD 值表现优异, 并且随着迭代次数不断增加, IGD 值曲线并未呈现增长的趋势, 展现出较强的稳定性.

3.7 算法运行时间分析

为更全面地衡量 Two-Arch/IS 的计算复杂度, 比较 12 种算法在 5 目标和 10 目标下的平均运行时间 (见图 11).

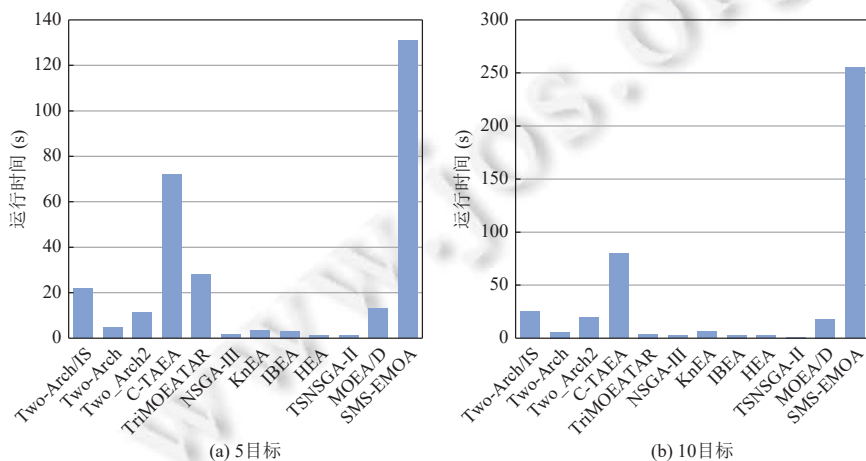


图 11 12 种算法在 5 目标和 10 目标测试问题下的平均运行时间

从图 11 结果可以看出, SMS-EMOA 算法因其特殊的指标设计导致运行时间复杂度较高, C-TAEA 算法次之, Two-Arch/IS、Two_Arch2、TriMOEATAR、MOEA/D 算法的表现相似, 而其余算法表现出更低的计算复杂度. Two-Arch/IS 相较于 Two-Arch 需要更多的运行时间, 这是因为 Two-Arch/IS 引入了多种改进机制提升算法性能, 虽然带来一定的计算负担, 但根据前文的实验结果及分析, Two-Arch/IS 相较于 Two-Arch 实现了显著的性能提升. 尽管 Two-Arch/IS 在运行时间上并不占优势, 但与其他算法的差距处于合理范围内, 并未显示出明显劣势.

4 结 论

本文提出了一种面向 MaOP 的双归档算法 Two-Arch/IS, 能够有效缓解 MaOP 中收敛性和多样性冲突导致种群难以收敛的困境. 为了应对算法在探索高维目标空间时面临的挑战, Two-Arch/IS 设计了一种限制性的繁殖策略, 通过选择亲本产生表现更佳的子代个体, 更有效地搜索解空间. 为了保证种群更有效地逼近 PF, Two-Arch/IS 重新设计了档案库的环境选择机制, 增加一组权重向量用于划分目标空间, 进而通过子种群更新两个档案库, 增强了种群的多样性表现, 避免种群陷入局部最优解, 同时借助角度选择与转移密度估计的方法移除档案库中劣势解, 在有效增强种群选择压力的同时极大程度上维护了种群多样性. 为进一步加强档案库间信息交流, Two-Arch/IS 设计了基于边界解的个体迁移机制, 通过边界解引导档案库的进化方向, 实现了 CA 和 DA 的优势互补. 通过在不同类型的测试问题上进行性能测试, Two-Arch/IS 相比于其他代表性算法在 MaOP 上表现出较为明显的优势, 同时在低维上也展现出良好的有效性. 未来的工作可以聚焦于探索更高效的个体迁移方法, 如根据当前档案库状态实现自适应的个体迁移, 通过动态调整个体迁移策略及相关参数, 从而更精准的指导搜索过程, 增强算法的适应性与综合性能.

References

- [1] Champasak P, Panagant N, Pholdee N, Bureerat S, Yildiz AR. Self-adaptive many-objective meta-heuristic based on decomposition for many-objective conceptual design of a fixed wing unmanned aerial vehicle. *Aerospace Science and Technology*, 2020, 100: 105783. [doi: 10.1016/j.ast.2020.105783]
- [2] Yao SG, Tian HQ, Xu P. Lightning design of carbody structure for heavy haul gondola. *Journal of Traffic and Transportation Engineering*, 2011, 11(1): 31–35, 57 (in Chinese with English abstract). [doi: 10.19818/j.cnki.1671-1637.2011.01.006]
- [3] Xie SC, Zhou H. Multi-objective optimisation of a vehicle energy absorption structure based on surrogate model. *Journal of Central South University*, 2014, 21(6): 2539–2546. [doi: 10.1007/s11771-014-2209-8]
- [4] Xiang Y, Yang XW, Zhou YR, Zheng ZB, Li MQ, Huang H. Going deeper with optimal software products selection using many-objective optimization and satisfiability solvers. *Empirical Software Engineering*, 2020, 25(1): 591–626. [doi: 10.1007/s10664-019-09761-2]
- [5] Cai XJ, Zhang JJ, Ning ZH, Cui ZH, Chen JJ. A many-objective multistage optimization-based fuzzy decision-making model for coal production prediction. *IEEE Trans. on Fuzzy Systems*, 2021, 29(12): 3665–3675. [doi: 10.1109/TFUZZ.2021.3089230]
- [6] Li YH, Liu ZJ, Li J. A multi-objective optimization algorithm of switched reluctance generator based on fuzzy logic NSGA-III. *Computer Engineering & Science*, 2021, 43(3): 534–541 (in Chinese with English abstract). [doi: 10.3969/j.issn.1007-130X.2021.03.020]
- [7] Liu QQ, Jin YC, Heiderich M, Rodemann T, Yu G. An adaptive reference vector-guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems. *IEEE Trans. on Cybernetics*, 2022, 52(5): 2698–2711. [doi: 10.1109/TCYB.2020.3020630]
- [8] Yang Z, Dong L, Cai XJ. Toward cooperative multi-agent video streaming perception. *SCIENTIA SINICA Informationis*, 2023, 53(1): 46–65 (in Chinese with English abstract). [doi: 10.1360/SSI-2021-0179]
- [9] Xie CW, Guo H, Wei W, Jiang L. MaOEA/d²: Many-objective evolutionary algorithm based on double distances. *Ruan Jian Xue Bao/Journal of Software*, 2023, 34(4): 1523–1542 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6702.htm> [doi: 10.13328/j.cnki.jos.006702]
- [10] Zou J, Yang QT, Yang SX, Zheng JH. Ra-dominance: A new dominance relationship for preference-based evolutionary multiobjective optimization. *Applied Soft Computing*, 2020, 90: 106192. [doi: 10.1016/j.asoc.2020.106192]
- [11] Liu Y, Zhu NB, Li KL, Li MQ, Zheng JH, Li KQ. An angle dominance criterion for evolutionary many-objective optimization. *Information Sciences*, 2020, 509: 376–399. [doi: 10.1016/j.ins.2018.12.078]
- [12] Li MQ, Yang SX, Liu XH. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2014, 18(3): 348–365. [doi: 10.1109/TEVC.2013.2262178]

- [13] Yang SX, Li MQ, Liu XH, Zheng JH. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2013, 17(5): 721–736. [doi: [10.1109/TEVC.2012.2227145](https://doi.org/10.1109/TEVC.2012.2227145)]
- [14] Cai XJ, Hu ZM, Zhang ZX, Wang Q, Cui ZH, Zhang WS. Multi-UAV coordinated path planning based on many-objective optimization. *SCIENTIA SINICA Informationis*, 2021, 51(6): 985–996 (in Chinese with English abstract). [doi: [10.1360/SSI-2020-0218](https://doi.org/10.1360/SSI-2020-0218)]
- [15] Zhang XY, Tian Y, Jin YC. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2015, 19(6): 761–776. [doi: [10.1109/TEVC.2014.2378512](https://doi.org/10.1109/TEVC.2014.2378512)]
- [16] Li BD, Tang K, Li JL, Yao X. Stochastic ranking algorithm for many-objective optimization based on multiple indicators. *IEEE Trans. on Evolutionary Computation*, 2016, 20(6): 924–938. [doi: [10.1109/TEVC.2016.2549267](https://doi.org/10.1109/TEVC.2016.2549267)]
- [17] Praditwong K, Yao X. A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm. In: *Proc. of the 2007 Int'l Conf. on Computational Intelligence and Security*. Guangzhou: Springer, 2007. 95–104. [doi: [10.1007/978-3-540-74377-4_11](https://doi.org/10.1007/978-3-540-74377-4_11)]
- [18] Pan LQ, Lin JQ, Wang HD, He C, Tan KC, Jin YC. Computationally expensive high-dimensional multiobjective optimization via surrogate-assisted reformulation and decomposition. *IEEE Trans. on Evolutionary Computation*, 2025, 29(4): 921–935. [doi: [10.1109/TEVC.2024.3380327](https://doi.org/10.1109/TEVC.2024.3380327)]
- [19] Yuan Y, Xu H, Wang B, Yao X. A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2016, 20(1): 16–37. [doi: [10.1109/TEVC.2015.2420112](https://doi.org/10.1109/TEVC.2015.2420112)]
- [20] Tian Y, Cheng R, Zhang XY, Su YS, Jin YC. A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2019, 23(2): 331–345. [doi: [10.1109/TEVC.2018.2866854](https://doi.org/10.1109/TEVC.2018.2866854)]
- [21] Zhu CW, Xu LH, Goodman ED. Generalization of Pareto-optimality for many-objective evolutionary optimization. *IEEE Trans. on Evolutionary Computation*, 2016, 20(2): 299–315. [doi: [10.1109/TEVC.2015.2457245](https://doi.org/10.1109/TEVC.2015.2457245)]
- [22] Lu X, Tan YY, Zheng W, Meng LL. A decomposition method based on random objective division for MOEA/D in many-objective optimization. *IEEE Access*, 2020, 8: 103550–103564. [doi: [10.1109/ACCESS.2020.2999417](https://doi.org/10.1109/ACCESS.2020.2999417)]
- [23] Farias LRC, Araújo AFR. IM-MOEA/D: An inverse modeling multi-objective evolutionary algorithm based on decomposition. In: *Proc. of the 2021 IEEE Int'l Conf. on Systems, Man, and Cybernetics (SMC)*. Melbourne: IEEE, 2021. 462–467. [doi: [10.1109/SMC52423.2021.9658650](https://doi.org/10.1109/SMC52423.2021.9658650)]
- [24] Sun YA, Yen GG, Yi Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. on Evolutionary Computation*, 2019, 23(2): 173–187. [doi: [10.1109/TEVC.2018.2791283](https://doi.org/10.1109/TEVC.2018.2791283)]
- [25] Dutta S, Raju MSS, Mallipeddi R, Das KN. Adaptive mating selection based on weighted indicator for multi/many-objective evolutionary algorithm. *Applied Soft Computing*, 2023, 139: 110223. [doi: [10.1016/j.asoc.2023.110223](https://doi.org/10.1016/j.asoc.2023.110223)]
- [26] Wang HD, Jiao LC, Yao X. Two_Arch2: An improved two-archive algorithm for many-objective optimization. *IEEE Trans. on Evolutionary Computation*, 2015, 19(4): 524–541. [doi: [10.1109/TEVC.2014.2350987](https://doi.org/10.1109/TEVC.2014.2350987)]
- [27] Ye TY, Wang H, Zeng T, Omran MGH, Wang F, Cui ZH, Zhao J. An improved two-archive artificial bee colony algorithm for many-objective optimization. *Expert Systems with Applications*, 2024, 236: 121281. [doi: [10.1016/j.eswa.2023.121281](https://doi.org/10.1016/j.eswa.2023.121281)]
- [28] Dai C. Two-archive evolutionary algorithm based on multi-search strategy for many-objective optimization. *IEEE Access*, 2019, 7: 79277–79286. [doi: [10.1109/ACCESS.2019.2917899](https://doi.org/10.1109/ACCESS.2019.2917899)]
- [29] Cai L, Qu SR, Cheng GJ. Two-archive method for aggregation-based many-objective optimization. *Information Sciences*, 2018, 422: 305–317. [doi: [10.1016/j.ins.2017.08.078](https://doi.org/10.1016/j.ins.2017.08.078)]
- [30] Liu ZQ, Sun N, Wu YM, Yang T, Liang X, Fang YC. Multi-objective trajectory planning for 5-DOF underactuated tower cranes with state constraints. *SCIENTIA SINICA Informationis*, 2022, 52(3): 521–538 (in Chinese with English abstract). [doi: [10.1360/SSI-2021-0106](https://doi.org/10.1360/SSI-2021-0106)]
- [31] Li W, Gong WY, Ming F, Wang L. Constrained multi-objective evolutionary algorithm with an improved two-archive strategy. *Knowledge-based Systems*, 2022, 246: 108732. [doi: [10.1016/j.knosys.2022.108732](https://doi.org/10.1016/j.knosys.2022.108732)]
- [32] Xia MM, Dong MG. A novel two-archive evolutionary algorithm for constrained multi-objective optimization with small feasible regions. *Knowledge-based Systems*, 2022, 237: 107693. [doi: [10.1016/j.knosys.2021.107693](https://doi.org/10.1016/j.knosys.2021.107693)]
- [33] Liu YP, Yen GG, Gong DW. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. *IEEE Trans. on Evolutionary Computation*, 2019, 23(4): 660–674. [doi: [10.1109/TEVC.2018.2879406](https://doi.org/10.1109/TEVC.2018.2879406)]
- [34] Li ZP, Zou J, Yang SX, Zheng JH. A two-archive algorithm with decomposition and fitness allocation for multi-modal multi-objective optimization. *Information Sciences*, 2021, 574: 413–430. [doi: [10.1016/j.ins.2021.05.075](https://doi.org/10.1016/j.ins.2021.05.075)]
- [35] Das I, Dennis JE. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 1998, 8(3): 631–657. [doi: [10.1137/S1052623496307510](https://doi.org/10.1137/S1052623496307510)]
- [36] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*. Athens: Int'l Center for Numerical

- Methods in Engineering, 2001. 95–100.
- [37] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. In: Proc. of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002. 825–830. [doi: 10.1109/CEC.2002.1007032]
 - [38] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2000, 8(2): 173–195. [doi: 10.1162/106365600568202]
 - [39] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation*, 1999, 3(4): 257–271. [doi: 10.1109/4235.797969]
 - [40] Li MQ, Grosan C, Yang SX, Liu XH, Yao X. Multiline distance minimization: A visualized many-objective test problem suite. *IEEE Trans. on Evolutionary Computation*, 2018, 22(1): 61–78. [doi: 10.1109/TEVC.2017.2655451]
 - [41] Li K, Chen RZ, Fu GT, Yao X. Two-archive evolutionary algorithm for constrained multiobjective optimization. *IEEE Trans. on Evolutionary Computation*, 2019, 23(2): 303–315. [doi: 10.1109/TEVC.2018.2855411]
 - [42] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. on Evolutionary Computation*, 2014, 18(4): 577–601. [doi: 10.1109/TEVC.2013.2281535]
 - [43] Zitzler E, Künzli S. Indicator-based selection in multiobjective search. In: Proc. of the 8th Int'l Conf. on Parallel Problem Solving from Nature-PPSN VIII. Birmingham: Springer, 2004. 832–842. [doi: 10.1007/978-3-540-30217-9_84]
 - [44] Zhang QF, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation*, 2007, 11(6): 712–731. [doi: 10.1109/TEVC.2007.892759]
 - [45] Emmerich M, Beume N, Naujoks B. An EMO algorithm using the hypervolume measure as selection criterion. In: Proc. of the 3rd Int'l Conf. on Evolutionary Multi-criterion Optimization. Guanajuato: Springer, 2005. 62–76. [doi: 10.1007/978-3-540-31880-4_5]
 - [46] Liu Z, Han F, Ling QH, Han H, Jiang J. A many-objective optimization evolutionary algorithm based on hyper-dominance degree. *Swarm and Evolutionary Computation*, 2023, 83: 101411. [doi: 10.1016/j.swevo.2023.101411]
 - [47] Ming F, Gong WY, Wang L. A two-stage evolutionary algorithm with balanced convergence and diversity for many-objective optimization. *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, 2022, 52(10): 6222–6234. [doi: 10.1109/TSMC.2022.3143657]
 - [48] Ishibuchi H, Masuda H, Tanigaki Y, Nojima Y. Modified distance calculation in generational distance and inverted generational distance. In: Proc. of the 8th Int'l Conf. on Evolutionary Multi-criterion Optimization. Guimarães: Springer, 2015. 110–125. [doi: 10.1007/978-3-319-15892-1_8]
 - [49] van Veldhuizen DA. Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations [Ph.D. Thesis]. Wright Patterson Air Force Base: Air Force Institute of Technology, 1999.

附中文参考文献

- [2] 姚曙光, 田红旗, 许平. 重载敞车车体结构轻量化设计. *交通运输工程学报*, 2011, 11(1): 31–35, 57. [doi: 10.19818/j.cnki.1671-1637.2011.01.006]
- [6] 李艺辉, 刘作军, 李洁. 基于模糊逻辑 NSGA-III 的开关磁阻发电机多目标优化算法. *计算机工程与科学*, 2021, 43(3): 534–541. [doi: 10.3969/j.issn.1007-130X.2021.03.020]
- [8] 杨铮, 董亮, 蔡新军. 面向多设备协同场景的实时视频流分析系统. *中国科学: 信息科学*, 2023, 53(1): 46–65. [doi: 10.1360/SSI-2021-0179]
- [9] 谢承旺, 郭华, 韦伟, 姜磊. MaOEA/d²: 一种基于双距离构造的高维多目标进化算法. *软件学报*, 2023, 34(4): 1523–1542. <http://www.jos.org.cn/1000-9825/6702.htm> [doi: 10.13328/j.cnki.jos.006702]
- [14] 蔡星娟, 胡钊鸣, 张志霞, 王茜, 崔志华, 张文生. 基于高维多目标优化的多无人机协同航迹规划. *中国科学: 信息科学*, 2021, 51(6): 985–996. [doi: 10.1360/SSI-2020-0218]
- [30] 刘卓清, 孙宁, 吴易鸣, 杨桐, 梁潇, 方勇纯. 考虑状态约束的五自由度塔式吊车多目标最优轨迹规划. *中国科学: 信息科学*, 2022, 52(3): 521–538. [doi: 10.1360/SSI-2021-0106]

作者简介

丁炜超, 博士, 副教授, 主要研究领域为群体智能与演化计算, 多目标优化算法, 模式识别.

李佳宁, 硕士生, 主要研究领域为群体智能与演化计算.

顾春华, 博士, 教授, 博士生导师, 主要研究领域为云计算, 物联网, 信息安全.

刘佳豪, 硕士生, 主要研究领域为多目标优化算法.

董文波, 博士, 讲师, CCF 专业会员, 主要研究领域为多视图机器学习, 深度高斯过程, 多目标优化.