

Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer 的研究报告

摘要

摘要—本文对论文 “Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer” 中的 CMOEMT 算法进行复现。通过深入分析原论文算法原理，详细阐述了复现过程中的关键步骤，包括多任务框架构建、知识表达与迁移策略、算法流程实现以及参数设置等方面。经过复现实验与原论文实验结果对比，验证了复现算法的有效性和准确性。对复现研究进行总结与展望，为相关领域研究提供了有价值的参考。

关键词：约束多目标优化；进化迁移优化（ETO）；知识迁移；多任务处理。

1 引言

约束多目标优化问题（Constrained Multi-objective Optimization Problems, CMOPs）在现实世界的应用和科学研究中广泛存在，是多目标优化领域的重要研究方向。例如，城市公交线路的车辆调度、测试资源分配、车辆路径规划以及 Web 服务位置分配等实际应用都涉及 CMOPs。

随着该领域的发展，为全面反映真实世界的复杂场景，各种具有不同特征和挑战的 CMOP 基准测试集被设计出来，如 C-DTLZ、DC-DTLZ、MW、LIR-CMOP 和 DAS-CMOP 等。同时，大量的约束多目标优化进化算法（Constrained Multi-Objective Evolutionary Algorithms, CMOEAs）也应运而生以求解这些 CMOPs。然而，现有的 CMOEAs 通常采用分治的思想，借助额外的辅助问题（通常是约束忽略或约束松弛问题）来协助处理原始的 CMOP，包括基于多种群和多阶段的方法等。

进化迁移优化（Evolutionary Transfer Optimization, ETO）中的进化多任务优化（Evolutionary Multitasking Optimization, EMT）概念与在解决 CMOPs 时原始问题和辅助问题的并行处理思想相契合。EMT 旨在不同协作任务之间实现积极的经验共享和迁移，这与基于分治的 CMOEAs 有相似之处。因此，EMT 在处理 CMOPs 方面可能具有有效性和潜力。

基于上述考虑，原论文提出了一种新的用于解决 CMOPs 的多任务框架。选题依据如下：

首先，现实中大量 CMOPs 的存在需要更有效的求解方法，而现有的方法在应对某些复杂情况时存在一定局限性。其次，EMT 在多目标优化中的应用尚处于起步阶段，特别是在 CMOPs 方面的研究较少，有较大的探索空间。原论文研究的选题意义重大。在理论方面，新

的多任务框架和知识迁移策略为解决 CMOPs 提供了新思路和方法，有助于推动约束多目标优化领域的理论发展。在实际应用中，能够为上述车辆调度等实际问题提供更高效、精准的解决方案，从而提高资源利用效率、降低成本、优化决策等，具有显著的实际应用价值。

2 相关工作

在约束多目标优化领域，已有诸多研究工作致力于解决相关问题，这些工作大致可分为两类：一是传统的约束多目标优化进化算法 (CMOEAs)，其主要基于分治思想或采用增强的约束处理技术；二是进化多任务优化 (EMT) 在多目标优化 (MOPs) 中的应用探索，包括不同的模型构建与基准测试设计等。以下将对这些相关工作进行详细分类概括与描述。

2.1 传统约束多目标优化进化算法 (CMOEAs)

2.1.1 基于分治思想的 CMOEAs

在处理复杂的 CMOPs 时，许多研究采用了分治思想，将问题分解为多个子问题或阶段进行求解。

多种群策略通过维护多个种群，并为每个种群采用不同的约束处理技术 (CHTs) 来实现协作优化。例如，Tian 等提出的 CCMO 框架 [11]，其中一个主种群基于约束优先原则 (CDP) 进行进化，另一个辅助种群则基于帕累托优势 (PD) 进行操作。主种群和辅助种群在各自的进化过程中，通过不同的选择、交叉和变异操作，朝着各自的目标优化，并且在一定阶段进行信息交互，以期达到整体的优化效果。

Ming 等提出的双种群算法类似 [8]，通过设计带罚函数的约束松弛技术用于辅助种群，使得种群在不同的约束处理方式下协同进化。在该算法中，主种群注重在可行区域内寻找最优解，辅助种群则通过罚函数在一定程度上探索不可行区域，从而在两者之间实现信息共享和协同优化。

多阶段策略将进化过程划分为不同阶段，各阶段使用不同的 CHTs。例如，Tian 等的自动切换两阶段策略，第一阶段利用 CDP 探索可行区域和约束 Pareto 前沿 (CPF)，第二阶段使用 PD 跨越不可行区域，在不同阶段根据目标和约束的特点，有针对性地进行优化操作，以平衡目标优化和约束满足。

Fan 等的推挽搜索方法也类似 [2]，推阶段将种群推向无约束 Pareto 前沿 (UPF)，拉阶段将种群拉回可行区域和 CPF。这种推挽操作有助于在不同的优化方向上引导种群进化，避免陷入局部最优，提高算法的搜索能力和优化效果。

2.1.2 基于增强 CHT 的 CMOEAs

为了更好地处理复杂约束和不可行区域，部分研究致力于设计增强的 CHT。

Ma 等提出的新适应度函数 [7]，通过同时考虑 PD 和 CDP，依据收敛性能来松弛约束，在适应度计算中综合考虑目标函数值、约束违反程度和收敛性相关指标，使得算法在处理约束时更加灵活和有效。

Jiao 等应用问题转换方法 [4] 将 CMOP 转换为动态 MOP，重新定义目标函数和约束条件，以适应约束处理需求。这种动态转换能够更好地应对约束的变化和复杂性，提高算法在

不同约束场景下的适应性。

Ma 和 Wang 设计的基于位移的罚函数 [6], 基于收敛和多样性松弛约束, 使得算法在处理复杂约束时能更好地平衡目标优化和约束满足。罚函数根据解的约束违反情况和多样性特征, 对解进行适当的惩罚或奖励, 引导种群朝着可行且优化的方向进化。

2.2 进化多任务优化 (EMT) 及其在多目标优化 (MOPs) 中的应用

2.2.1 EMT 模型与应用

EMT 模型 [3] 旨在通过多个任务之间的协同优化, 实现知识和经验的共享与迁移, 从而提高整体的优化效率。在多目标优化问题中, EMT 的应用方式多种多样。例如, Qiao 等提出的 EMCMO 算法 [9], 将解集视为知识的载体, 根据约束 Pareto 前沿 (CPF) 与无约束 Pareto 前沿 (UPF) 之间的关系来决定知识转移的策略。具体而言, 当 CPF 与 UPF 存在特定的重叠关系时, EMCMO 会有针对性地在不同任务间传递解, 试图利用源任务中的有用信息加速目标任务的优化进程。这种基于关系判断的知识转移方式, 为解决复杂的多目标优化问题提供了一种新的思路。

MTCMO 算法 [10] 则基于- 约束技术构建动态辅助任务。在该算法中, 辅助任务并非固定不变, 而是根据的动态变化进行调整。通过这种方式, MTCMO 能够更好地适应不同的约束强度和问题特征, 在求解原始 CMOP 时提供更灵活的辅助作用。例如, 在处理具有动态约束条件的多目标优化问题时, MTCMO 可以根据约束的变化实时调整辅助任务, 从而更有效地引导种群向最优解方向进化。

2.2.2 多任务多目标优化基准测试

为了更系统地研究多任务多目标优化算法的性能, 研究者们设计了一系列专门的基准测试集。这些基准测试问题通常包含多个具有不同特征的任务, 每个任务在目标函数、约束条件以及搜索空间等方面都存在差异。例如, Yuan 等提出的基准测试问题 [12], 其中的任务可能在目标函数的复杂度、约束的紧密度以及搜索空间的形状和范围等方面各不相同。这种多样性为评估多任务多目标优化算法提供了丰富的测试场景, 能够全面地检验算法在不同情况下的性能表现。

在这些基准测试的基础上, 许多新的算法不断涌现并进行性能测试。Chen 等提出的多目标多任务优化算法 [1] 利用转移秩来增加正迁移的概率, 同时借助 KNN 模型增强解的多样性。通过这种方式, 该算法在处理多个任务时, 能够更有效地在任务间传递有益信息, 避免陷入局部最优解, 从而提高整体优化性能。Liang 等的算法 [5] 则聚焦于子空间对齐和自适应差分进化, 通过改进任务间的正知识迁移效果, 进一步提升了多任务优化的效率。在面对复杂的多任务多目标优化问题时, 该算法能够更好地利用不同任务之间的关联和互补性, 加速种群向 Pareto 前沿逼近。

这些研究工作不仅丰富了多任务多目标优化领域的算法库, 也为进一步深入研究该领域的理论和应用奠定了坚实的基础。随着研究的不断深入, EMT 在多目标优化中的应用有望取得更多的突破, 为解决实际中的复杂多目标优化问题提供更有效的解决方案。

3 本文方法

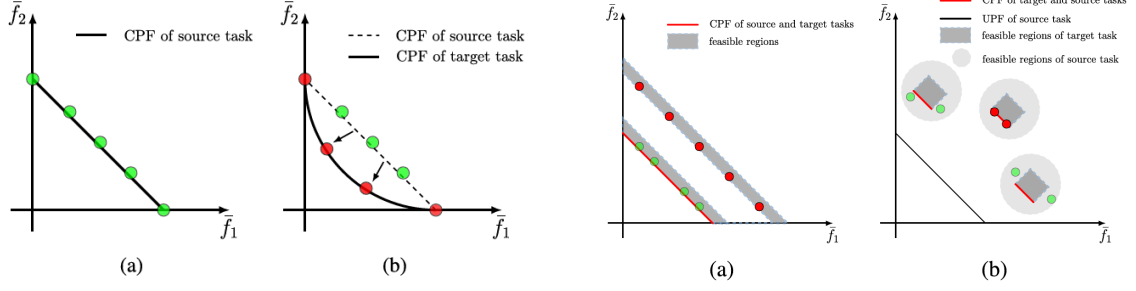


图 1. 多任务处理和知识迁移在解决多目标优化问题 (MOPs) 中的作用。(a) 源任务的处理。(b) 在源自源任务的转移知识辅助下对目标任务的处理。

图 2. 在解决复杂约束多目标优化问题 (CMOPs) 中多任务处理和知识迁移的潜在益处。可行域是 (a) 内外层结构和 (b) 不连续片段。

3.1 动机与关键问题

进化多任务优化 (EMT) 作为一种新兴的进化优化范式, 已在多种问题的求解中展现出潜力。然而, 其在约束多目标优化问题 (CMOPs) 中的应用却相对匮乏。为了说明多任务处理和知识迁移在解决 CMOPs 中的优势, 考虑一个包含源任务和目标任务 (即两个 CMOPs) 的示例, 如图 1 所示。

在处理源任务 (图 1(a)) 时, 若能获得一组均匀分布在约束 Pareto 前沿 (CPF) 上的可行非支配解 (绿色点), 当利用这些解来协助解决目标任务 (图 1(b)) 时, 会带来两方面好处。其一, 源任务中获得的部分可行非支配解可能对目标任务同样可行且非支配, 从而可直接被目标任务继承; 其二, 源任务的解相较于随机生成的解更接近目标任务的 CPF (如图 1(b) 中的弧线所示), 能够加速目标任务向其 CPF 的收敛。

在面对更复杂的 CMOPs 时, 如可行域呈现内外层结构 (图 2(a)) 或不连续片段 (图 2(b)) 的情况, 知识转移同样具有重要意义。在图 2(a) 中, 若源任务的解已到达 CPF 而目标任务的解尚未到达, 且两者 CPF 相同或相似 (常见于 DC - DTLZ 和 MW 问题), 源任务的解可帮助目标任务跳出外层可行区域。在图 2(b) 中, 若源任务的可行域包含目标任务的可行域, 源任务的可行非支配解可能到达目标任务的可行域, 从而有助于检测目标任务中未被发现的可行区域 (常见于约束松弛的辅助问题)。

不同的辅助问题适用于不同类型的 CMOPs。约束忽略的辅助问题适用于 CPF 是无约束 Pareto 前沿 (UPF) 的一部分或与之相同的情况 (如图 2(a)), 而约束松弛的辅助问题则更适用于 CPF 与 UPF 不重叠的情况 (如图 2(b))。因此, 合理选择辅助问题对于提高整体框架性能至关重要, 这也凸显了构建有效多任务框架时源任务和目标任务协同合作的重要性。

综上, 为成功将 EMT 应用于单任务 CMOPs, 需解决三个关键问题: 任务的建立与合作、不同任务间知识的表达以及任务间的知识转移内容。

3.2 提出的三任务框架

为解决上述任务建立与合作的问题, 提出一种三任务框架, 如图 3 所示。基于此框架, 一个 CMOP 由三个任务 T_1 、 T_2 和 T_3 共同求解。

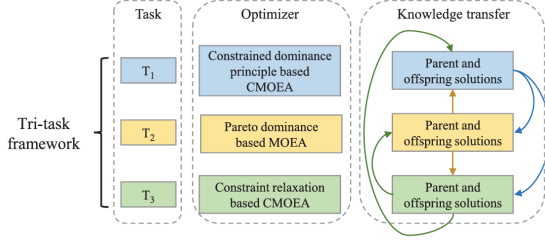


图 3. 所提出的三任务多任务模型的总体框架。

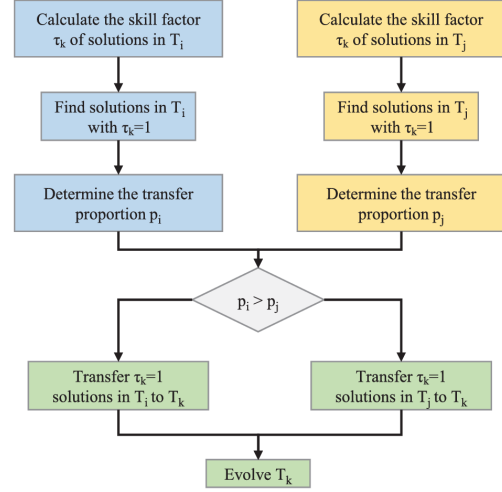


图 4. 所提出的知识转移模型的总体框架。

任务 T_1 采用约束优先原则 (CDP) 作为约束处理技术 (CHT)，并使用基于 CDP 的约束多目标进化算法 (CMOEA) 作为优化器。该任务适用于可行域连续且较大的 CMOPs，因为其对可行非支配解的偏好能够确保在无不可行障碍的情况下逼近 CPF。然而，当可行域不连续或较小时，仅使用 T_1 容易使种群陷入局部可行区域。

任务 T_2 采用帕累托优势 (PD) 来忽略约束，并基于 PD 的多目标进化算法 (MOEA) 进行优化。此任务适用于 CPF 与 UPF 相同或部分相同的 CMOPs，通过搜索 UPF 来获取 CPF。但对于 CPF 不完全靠近 UPF (部分重叠或不重叠) 的 CMOPs 并不适用，且单独使用 T_2 不能保证最终得到可行解集。

任务 T_3 采用约束松弛技术作为 CHT，并运用基于约束松弛的 CMOEA 作为优化器。由于存在约束松弛，该任务能够在松弛的可行区域中保留一些不可行解，克服了 CDP 和 PD 在处理可行域小且不连续、CPF 与 UPF 部分重叠或不重叠的 CMOPs 时的不足。不过，单独使用 T_3 同样不能确保最终得到可行解集，因为它放松了约束并保留了不可行解。

这三个任务之间的合作能够带来诸多潜在益处。例如， T_2 获得的可行非支配解可加速 T_1 向 CPF 的收敛；在约束松弛下， T_3 能有效搜索可行区域，其得到的可行解可能帮助 T_1 检测遗漏的可行区域；若 CMOP 收敛困难 (由距离函数或多模态等原因导致)， T_1 和 T_3 得到的可行或半可行解可能对 T_2 非支配，从而帮助其收敛到 UPF； T_1 的可行非支配解有助于定位可行区域，补充 T_3 可能遗漏的部分；同理， T_2 的半可行非支配解也能帮助 T_3 定位其遗漏的可行区域。

综上所述，所提出的三任务框架实现了三个任务之间的积极协作，共同应对不同类型的 CMOPs。

3.3 知识表达与转移

为解决不同任务间知识表达的问题，提出一种针对三任务框架的知识表达方法。与现有研究不同，本方法将父代和子代解均视为知识。这是因为进化算法中后代的产生具有高度随机性，单纯转移后代集如同转移随机种群，缺乏指导意义。

具体而言，设计技能因子 τ 来评估一个解对任务的适用性。对于解 x ，其关于任务 T_i ($i = 1, 2, 3$) 的技能因子 τ_i 定义如下：- 对于任务 T_1 ，若 $\phi(x) = 0$ 且 $n_f(x) = 1$ ，则 $\tau_1 = 1$ ，

表示解 x 可行且非支配, 适合 T_1 ; 否则 $\tau_1 = 0$ 。- 对于任务 T_2 , 若 $n_f(x) = 1$, 则 $\tau_2 = 1$, 意味着解 x 非支配, 适合 T_2 ; 否则 $\tau_2 = 0$ 。- 对于任务 T_3 , 若 $\phi(x) \leq \varepsilon$ 且 $n_f(x) = 1$, 则 $\tau_3 = 1$, 即任何满足约束松弛且非支配的解可能对 T_3 有帮助; 否则 $\tau_3 = 0$ 。这里 n_f 表示由快速非支配排序确定的支配等级数量, 且每个解的 n_f 是在所有任务的解中共同确定的。通过这种方式, 只有在所有三个任务中均为可行非支配的解才适合 T_1 , 所有任务中非支配的解适合 T_2 , 满足约束松弛的解适合 T_3 , 从而使技能因子值能够准确代表解所携带的知识。

基于上述技能因子, 提出一种新的知识转移模型, 如图 4 所示。当进化第 k 个任务 T_k 时, 计算其他两个任务 T_i 和 T_j 中解的技能因子 τ_k 值, 然后确定转移比例 p_t , 其计算公式为 $p_t = \frac{\sum_{x \in T_k} \tau_k(x)}{|T_k|}$, 其中 t 表示第 t 个源任务, k 表示第 k 个目标任务。转移比例较大的源任务被视为更适合的源任务 (辅助问题), 仅将该源任务中 $\tau_k = 1$ 的解转移到目标任务。这样做的目的是避免负转移并降低时间复杂度。关于时间复杂度的详细分析在后续部分给出, 同时在实验部分也对 CPU 运行时间进行了详细测试。

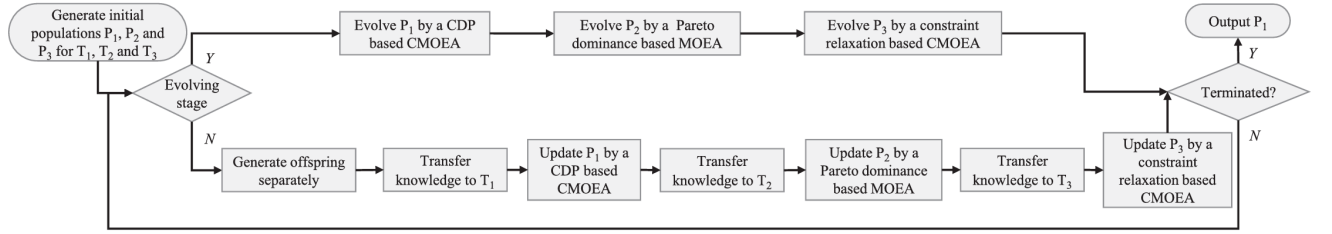


图 5. 所提出的约束多目标进化多任务 (CMOEMT) 算法的总体框架

3.4 提出的 CMOEMT 算法

CMOEMT 算法是一种创新的约束多目标进化多任务算法, 旨在高效解决约束多目标优化问题 (CMOPs)。它基于进化多任务 (EMT) 的思想, 通过构建多个相关任务并在任务之间进行知识转移, 利用不同任务的特点和优势来协同优化主任务, 以更有效地搜索到约束 Pareto 前沿 (CPF)。算法流程如图 5 所示

任务构建: 该算法首先构建了三个不同的任务, 分别记为 T_1 、 T_2 和 T_3 。每个任务都针对不同类型的 CMOPs 或采用不同的约束处理技术。例如, T_1 可能侧重于处理可行域较大且连续的 CMOPs, 采用基于约束优先原则 (CDP) 的约束多目标进化算法 (CMOEA); T_2 适用于 CPF 与无约束 Pareto 前沿 (UPF) 部分相同或接近的情况, 运用基于帕累托优势 (PD) 的多目标进化算法 (MOEA); T_3 则针对可行域小且不连续、CPF 与 UPF 部分重叠或不重叠的 CMOPs, 采用基于约束松弛的 CMOEA。

初始化阶段: 在初始化时, 需要确定种群大小等相关参数, 并分别为三个任务生成初始种群 P_1 、 P_2 和 P_3 。同时, 设置当前迭代次数 $g = 0$ 。

进化与转移阶段: 在主循环中, 当 $g < \eta \times G_{max}$ (其中 η 为预设的比例系数, G_{max} 为最大迭代次数) 时, 进入进化阶段。在这个阶段, 三个任务各自独立地进行进化操作, 根据其特定的算法和策略对种群进行更新, 使种群逐渐向各自对应的目标前沿逼近。当 $g \geq \eta \times G_{max}$ 时, 进入转移阶段。首先分别独立生成三个任务的后代集, 然后进行知识转移操作。通过计算所有解的技能因子, 评估每个解对特定任务的适用性。对于每个任务, 根据其他任务中解的技能因子, 确定更合适的辅助任务, 并将辅助任务中满足特定条件的解转移到目标任务以

更新种群。例如，对于 T_1 ，若在 T_2 或 T_3 中存在技能因子较高且对 T_1 可能有帮助的解，则将这些解转移到 T_1 中。

终止与输出: 当迭代次数 $g = G_{max}$ 时，算法终止，并输出 P_1 作为最终解集。因为在整个进化过程中， P_1 是针对原始的约束多目标优化问题进行优化和更新的，所以最终输出的 P_1 中的解是在考虑了约束条件和多任务协同优化后的较优解。

与传统的基于 EMT 的方法和多种群框架相比，CMOEMT 算法的知识转移机制更加精细和有效。它不是简单地将整个解集进行转移，而是将个体解视为知识，通过技能因子的评估，仅转移对目标任务有用的解，避免了不必要的转移和计算开销，提高了知识转移的效率和准确性。通过构建三个不同的任务，CMOEMT 算法能够适应不同类型的 CMOPs。在进化过程中，三个任务可以相互协作和补充，充分发挥各自的优势，共同推动种群向约束 Pareto 前沿收敛。这种多任务的协作方式使得算法在处理复杂的约束多目标优化问题时具有更强的适应性和鲁棒性。在传统的多种群框架中，下一代种群选择通常只使用后代解集，可能会丢弃有用的父代解。而 CMOEMT 算法中，父代解可能通过技能因子被转移到下一代，若父代解对相应任务有用，则可被保留，从而更好地利用了种群中的信息，提高了算法的搜索效率。

综上所述，CMOEMT 算法通过独特的任务构建、精细的知识转移和有效的种群更新策略，为约束多目标优化问题的解决提供了一种高效、灵活和鲁棒的方法，在多目标优化领域具有重要的理论和实际应用价值。

4 复现细节

4.1 与已有开源代码对比

在着手复现 “Constrained Multiobjective Optimization via Multitasking and Knowledge Transfer” 中的 CMOEMT 算法时，本文对相关的开源代码资源进行了全面搜索与深入研究。幸运的是，找到了该论文的开源代码，这为本文的复现工作提供了重要的基础和参考。然而，在仔细分析过程中，也发现了一些可以进一步改进和优化的方面，以下将详细阐述本文的复现工作。

4.2 代码实现过程

考虑到算法的复杂性和计算需求，本文选择 MATLAB 作为复现的编程语言。MATLAB 提供了丰富的数学函数库和高效的矩阵运算能力，非常适合处理多目标优化问题中的各种计算任务，如目标函数评估、约束处理、解的选择与排序等。在开发环境方面，本文使用 MATLAB 自带的集成开发环境 (IDE)，其提供了代码编辑、调试和性能分析等功能，方便本文进行算法的开发和优化。

为了表示种群中的个体，本文设计了一个结构体来存储个体的目标函数值、约束条件值以及其他相关信息。例如，在处理 CMOP 问题时，一个个体的结构体可能包含 ‘objs’ (目标函数值数组)、‘cons’ (约束条件值数组) 等字段，方便在算法运行过程中对个体信息进行访问和操作。对于种群，本文使用结构体数组来表示，每个元素对应一个个体结构体。这样的设计使得本文能够方便地对种群进行整体操作，如选择、交叉、变异和环境选择等。

目标函数与约束条件评估函数：根据具体的测试问题，准确实现了目标函数和约束条件

的评估函数。这些函数接受个体的决策变量作为输入，计算相应的目标函数值和约束条件值。在实现过程中，严格遵循测试问题的数学定义，确保计算结果的准确性。选择操作函数：实现了多种选择操作函数，如锦标赛选择（‘TournamentSelection’）用于选择父代个体进行繁殖。在锦标赛选择函数中，根据给定的竞赛规模，从种群中随机选择个体进行比较，选择具有较好适应度（根据任务和算法阶段确定适应度计算方式）的个体进入交配池。交叉与变异操作函数：针对不同的任务和问题需求，实现了合适的交叉和变异操作函数。例如，在遗传算法（GA）相关任务中，使用模拟二进制交叉（‘OperatorGAhalf’）和多项式变异（‘OperatorDE’）操作来生成子代个体。这些操作函数按照相应的数学公式和算法规则，对父代个体的决策变量进行操作，以引入新的遗传信息，促进种群的进化。环境选择函数：分别为三个任务实现了环境选择函数（‘EnvironmentalSelectionT1’、‘EnvironmentalSelectionT2’、‘EnvironmentalSelectionT3’）。在这些函数中，根据任务的特点和当前种群的状态，选择合适的个体进入下一代种群。例如，在 T_1 的环境选择中，优先保留可行且非支配的解；在 T_3 的环境选择中，考虑解在约束松弛条件下的性能。

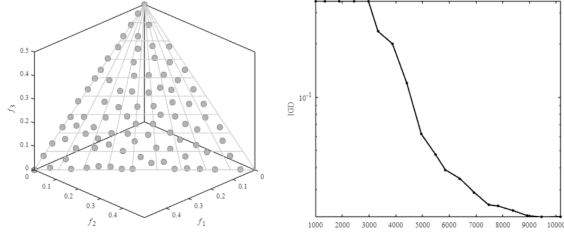
4.3 代码实现过程复现过程中的挑战与解决方法

CMOEMT 算法涉及多个任务的协同进化和复杂的知识转移策略，这使得算法的逻辑结构较为复杂。在实现过程中，本文通过绘制详细的算法流程图，将整个算法流程分解为多个子流程，逐步实现每个部分，并不断验证其正确性。例如，在实现知识转移过程时，详细分析了从计算技能因子到确定源任务和转移解的每一个步骤，确保每个环节的逻辑正确无误。对于算法中涉及的多种约束处理技术和多目标优化操作，本文深入研究相关的数学原理和算法规则，参考其他类似算法的实现经验，确保在实现过程中准确遵循算法设计。同时，通过编写大量的注释来解释代码的功能和逻辑，提高代码的可读性和可维护性，以便在后续调试和优化过程中能够快速理解和修改代码。

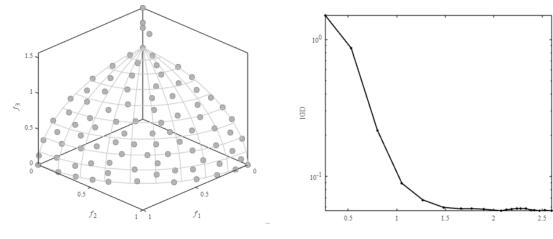
在处理大规模测试问题时，算法的计算效率成为一个关键问题。为了提高算法的性能，本文对代码中的关键计算部分进行了优化。例如，在计算解之间的支配关系和距离时，采用了更高效的数据结构和算法。使用基于树结构的数据结构来存储和查询解的支配关系，能够快速判断解之间的支配情况，减少了比较次数。在计算距离时，利用矩阵运算的优化技巧，如利用 MATLAB 的向量化计算能力，减少了循环计算的开销，显著提高了计算速度。另外，本文对算法中的参数进行了细致的调整和优化。通过大量的实验，观察不同参数值对算法性能的影响，找到在不同测试问题上性能表现较好的参数组合。例如，在调整约束松弛因子时，根据问题的约束强度和可行区域的特点，选择合适的松弛程度，使算法能够在保证解的可行性的同时，有效地探索可行区域。

通过以上复现细节的阐述，可以看出本文在复现 CMOEMT 算法过程中，深入理解算法原理，精心设计代码结构，合理选择实验设置和参数，并积极应对各种挑战，确保了复现工作的准确性和有效性，为进一步研究和应用该算法奠定了坚实的基础。

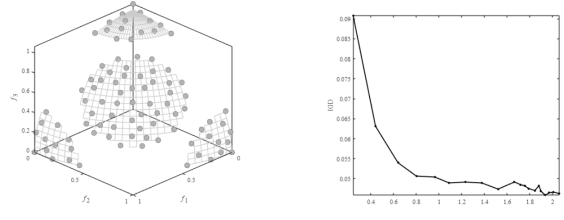
5 实验结果分析



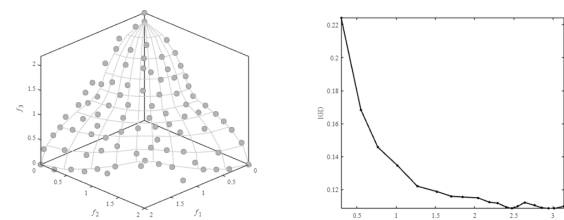
C1 DTLZ1



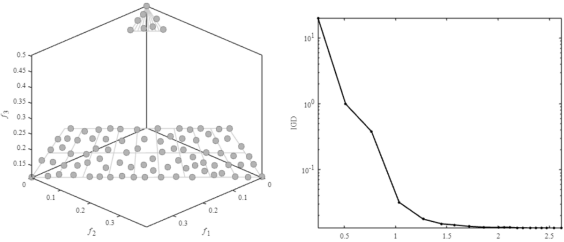
C1 DTLZ3



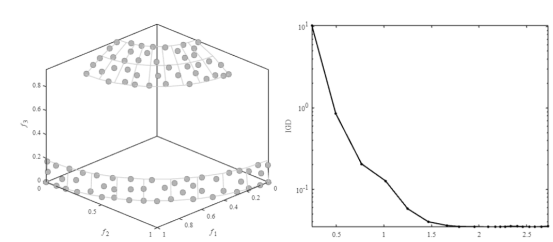
C2 DTLZ2



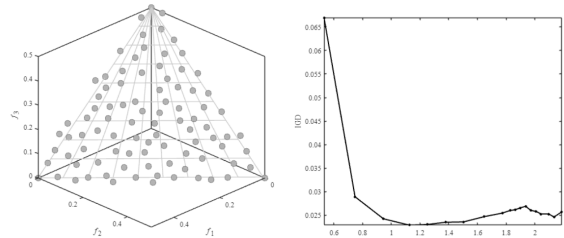
C3 DTLZ4



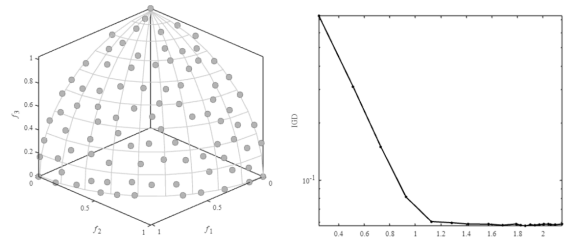
DC1 DTLZ1



DC1 DTLZ3



DC2 DTLZ1



DC2 DTLZ3

本次复现实验采用了与原论文相同的测试问题集、算法与问题参数设置。具体而言，本文使用了包括 DTLZ (CDTLZ 和 DC - DTLZ) 在内的多个具有不同特征和挑战的 CMOP 基准测试问题集。算法方面，本文复现了原论文中的 CMOEMT 算法。

在各个三维目标空间图中，本文可以看到算法生成的解点（以灰色点表示）在目标空间中的分布情况。例如，在 C1 DTLZ2、C1 DTLZ3 等问题中，解点呈现出一定的规律性，沿着目标空间的特定方向分布。对于一些问题，解点能够较好地覆盖可行区域，而在另一些问题中，解点可能在某些区域较为密集，而在其他区域较为稀疏。这反映了问题本身的复杂性和算法在探索目标空间时的表现。通过观察解点与理想帕累托前沿 (PF) 的接近程度，可以初步判断算法的收敛性。在某些问题中，解点能够较为接近理论上的最优前沿，表明算法在收敛方面有较好的表现。然而，在一些具有复杂约束或较大不可行区域的问题中，解点可能距离最优前沿较远，显示出算法在收敛上存在一定的挑战。

IGD（反向世代距离）折线图展示了随着迭代次数或运行时间的增加，IGD 值的变化情况。在大多数问题中，本文可以看到 IGD 值呈现出下降的趋势，这表明算法在不断地优化解的质量，使其更接近真实的帕累托前沿。例如，在 DC1 DTLZ3 等问题中，IGD 值在初始阶段下降较为迅速，这可能是由于算法在早期能够快速探索到较优的解区域。随着迭代的进行，IGD 值的下降速度逐渐减缓，反映了算法在后期收敛速度变慢。观察 IGD 折线图的最终值，可以比较不同问题下算法的性能。在一些较为简单的问题中，最终的 IGD 值能够达到较低的水平，表明算法能够有效地找到接近帕累托前沿的解。然而，在一些复杂的约束多目标优化问题中，最终的 IGD 值可能相对较高，说明算法在这些问题上仍有改进的空间。

从三维目标空间图和解点的分布来看，算法在一些规则的约束帕累托前沿（CPF）问题上表现出较好的收敛性。这可能是因为这些问题的结构相对简单，算法能够较为容易地找到最优方向进行搜索。然而，在具有不规则 CPF 或较大不可行区域的问题中，算法的收敛性受到一定的影响。这是由于算法在搜索过程中需要花费更多的时间和资源来避开不可行区域，从而导致收敛速度变慢。

在目标空间图中，解点的分布情况反映了算法在保持解的多样性方面的表现。在一些问题中，解点能够较为均匀地分布在可行区域内，这表明算法在探索目标空间时能够保持一定的多样性，避免过早地收敛到局部最优解。但是，在某些问题中，解点的分布可能出现不均匀的情况，这可能会导致算法错过一些潜在的最优解区域，影响算法的整体性能。

通过与原论文中的实验结果进行对比，本文发现复现实验在一些问题上得到了相似的结果。例如，在 IGD 值的变化趋势和最终值方面，对于一些较为简单的问题，复现结果与原论文结果较为接近，这表明本文的复现实验在一定程度上成功地再现了原算法的性能。然而，在一些复杂问题上，复现结果可能与原论文存在一定的差异。这可能是由于实验环境、参数设置的细微差别，或者是原论文中未详细说明的一些算法实现细节所导致。这些差异为本文进一步优化算法和深入研究提供了方向。

通过对复现实验结果的详细描述和分析，本文可以看到 CMOEMT 算法在处理约束多目标优化问题时具有一定的优势，但在面对复杂的问题结构和约束条件时，仍存在改进的空间。未来的研究可以集中在优化算法的搜索策略，提高算法在复杂问题上的收敛性和多样性，以及进一步探究算法参数对性能的影响等方面。

6 总结与展望

本研究复现了 CMOEMT 算法，深入剖析其原理，细致阐述复现步骤，通过实验验证了有效性。复现过程中，对比开源代码改进了结构与效率，利用 MATLAB 实现算法功能。实验表明，算法在处理约束多目标优化问题上有一定优势，但仍存不足。

当前不足主要体现在算法性能方面，大规模问题计算资源消耗大、收敛性和解质量有待提升；知识转移策略适应性不足，对解信息挖掘不够深入；参数设置不够智能，对不同问题适应性有限。

未来研究方向包括：优化算法流程，提高计算效率，增强在复杂问题上的收敛性；改进知识转移策略，使其更智能、自适应，拓展知识表达维度；实现参数自适应调整，建立参数与问题特征关系模型；拓展应用领域，融合其他先进算法或技术，提升算法实用性与解决复杂问题的能力。

参考文献

- [1] Hongyan Chen, Hai-Lin Liu, Fangqing Gu, and Kay Chen Tan. A multiobjective multitask optimization algorithm using transfer rank. *IEEE Transactions on Evolutionary Computation*, 27(2):237–250, 2023.
- [2] Zhun Fan, Wenji Li, Xinye Cai, Hui Li, Caimin Wei, Qingfu Zhang, Kalyanmoy Deb, and Erik Goodman. Push and pull search for solving constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, 44:665–679, 2019.
- [3] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2016.
- [4] Ruwang Jiao, Sanyou Zeng, Changhe Li, Shengxiang Yang, and Yew-Soon Ong. Handling constrained many-objective optimization problems via problem transformation. *IEEE Transactions on Cybernetics*, 51(10):4834–4847, 2021.
- [5] Zhengping Liang, Hao Dong, Cheng Liu, Weiqi Liang, and Zexuan Zhu. Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution. *IEEE Transactions on Cybernetics*, 52(4):2096–2109, 2022.
- [6] Zhongwei Ma and Yong Wang. Shift-based penalty for evolutionary constrained multiobjective optimization and its application. *IEEE Transactions on Cybernetics*, 53(1):18–30, 2023.
- [7] Zhongwei Ma, Yong Wang, and Wu Song. A new fitness function with two rankings for evolutionary constrained multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(8):5005–5016, 2021.
- [8] Mengjun Ming, Anupam Trivedi, Rui Wang, Dipti Srinivasan, and Tao Zhang. A dual-population-based evolutionary algorithm for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(4):739–753, 2021.
- [9] Kangjia Qiao, Kunjie Yu, Boyang Qu, Jing Liang, Hui Song, and Caitong Yue. An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 26(2):263–277, 2022.
- [10] Kangjia Qiao, Kunjie Yu, Boyang Qu, Jing Liang, Hui Song, Caitong Yue, Hongyu Lin, and Kay Chen Tan. Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 27(3):642–656, 2023.
- [11] Ye Tian, Tao Zhang, Jianhua Xiao, Xingyi Zhang, and Yaochu Jin. A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 25(1):102–116, 2021.

- [12] Jiawei Yuan, Hai-Lin Liu, Yew-Soon Ong, and Zhaoshui He. Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 26(2):379–391, 2022.