

融入 Cubic-Sine 混合映射与 Centre 策略的多目标粒子群算法

骆怡¹, 刘衍民², 陈建杰³

1. 贵州大学 数学与统计学院, 贵阳 550025
2. 遵义师范学院 数学学院, 贵州 遵义 563006
3. 贵州民族大学 数据科学与信息工程学院, 贵阳 550025

摘要:目的 多目标粒子群算法以其操作简便和快速收敛的特点,在解决复杂的多目标优化问题中展现出巨大潜力。然而,如何在确保算法快速收敛的同时,有效保持解的多样性,一直是该领域面临的关键挑战。方法 为了攻克这一挑战,本研究开创性地提出了一种结合了 Cubic-Sine 混合映射与 Centre 策略的多目标粒子群优化算法(CSMOPSO),该算法以其独特的设计实现了性能的全方位显著提升。在 CSMOPSO 算法的初始化阶段,通过精心设计的混沌参数选择机制,算法能够智能地决定采用 Cubic 或 Sine 混沌映射来生成初始种群。这一策略不仅确保了种群的多样性和均匀分布,还为算法在广阔的解空间内展开高效搜索奠定了坚实基础,显著增强了其全局搜索能力。进一步地,CSMOPSO 算法还引入了锥域来存储非支配解,并借助 Centre 策略来优化外部存档的管理。在这些创新措施共同作用下,使得存档中的非支配解质量得到显著提升,从而平衡了算法的收敛性和多样性。结果 通过在一系列基准测试问题(如 ZDT 和 UF 系列)上对比 CSMOPSO 与其他多目标粒子群算法和多目标进化算法,实验结果表明:CSMOPSO 算法在多样性和收敛速度方面均展现出卓越的性能,并且在多数测试函数上的表现均优于其他算法。结论 CSMOPSO 算法成功地实现了收敛性与多样性的良好平衡,显著提升了其整体性能,在求解多目标优化问题上具有较强的竞争力。

关键词:多目标粒子群算法;Cubic-Sine 混合映射;锥域;Centre 策略

中图分类号:O643 文献标识码:A doi:10.16055/j.issn.1672-058X.2026.0003.017

A Multi-Objective Particle Swarm Algorithm Incorporating Cubic-Sine Hybrid Mapping with Centre Strategy

LUO Yi¹, LIU Yanmin², CHEN Jianjie³

1. School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China
2. School of Mathematics, Zunyi Normal University, Zunyi 563006, Guizhou, China
3. School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang 550025, China

Abstract: Objective The multi-objective particle swarm algorithm shows great potential in solving complex multi-objective optimization problems due to its simple operation and fast convergence. However, how to effectively maintain the diversity of solutions while ensuring the fast convergence of the algorithm has always been a key challenge in this field.

Methods To address this challenge, this study innovatively proposes a multi-objective particle swarm optimization algorithm that combines the Cubic-Sine hybrid mapping and the Centre strategy (CSMOPSO). With its unique design, this algorithm achieves a comprehensive and significant improvement in performance. In the initialization stage of the

收稿日期:2024-12-30 修回日期:2025-03-05 文章编号:1672-058X(2026)03-0144-15

基金项目:贵州省高等学校进化人工智能重点实验室(黔教技[2022]059)项目资助;贵州省高等学校多粒度智能计算创新团队(黔教技[2023]084号)项目资助;贵州省重点实验室项目“遵义市酱香白酒分子重点实验室”资助(黔科合平台人才-ZZSG[2024]014)。

作者简介:骆怡(2000—),女,贵州遵义人,硕士研究生,从事优化理论及智能算法研究。

通信作者:刘衍民(1978—),男,山东临沂人,博士生导师,教授,从事优化理论及智能算法研究。Email:yanmin7813@163.com。

引用格式:骆怡,刘衍民,陈建杰.融入 Cubic-Sine 混合映射与 Centre 策略的多目标粒子群算法[J].重庆工商大学学报(自然科学版),2026,43(3):144-158.

Luo Yi, Liu Yanmin, Chen Jianjie. A multi-objective particle swarm algorithm incorporating Cubic-Sine hybrid mapping with Centre strategy[J]. Journal of Chongqing Technology and Business University (Natural Science Edition), 2026, 43(3): 144-158.

CSMOPSO algorithm, through a carefully designed chaotic parameter selection mechanism, the algorithm can intelligently decide whether to use the Cubic or Sine chaotic mapping to generate the initial population. This strategy not only ensures the diversity and uniform distribution of the population but also lays a solid foundation for the algorithm to conduct an efficient search in a vast solution space, significantly enhancing its global search ability. Furthermore, the CSMOPSO algorithm introduces a cone domain to store non-dominated solutions and uses the Centre strategy to optimize the management of the external archive. Under the combined action of these innovative measures, the quality of non-dominated solutions in the archive is significantly improved, and thus the convergence and diversity of the algorithm are balanced. **Results** By comparing CSMOPSO with other multi-objective particle swarm algorithms and multi-objective evolutionary algorithms on a series of benchmark test problems (such as the ZDT and UF series), the experimental results show that the CSMOPSO algorithm exhibits excellent performance in terms of both diversity and convergence speed and outperforms other algorithms on most test functions. **Conclusion** The CSMOPSO algorithm successfully achieves a good balance between convergence and diversity, significantly improving its overall performance and demonstrating strong competitiveness in solving multi-objective optimization problems.

Keywords: multi-objective particle swarm algorithm; Cubic-Sine hybrid mapping; cone domain; Centre strategy

随着时代的发展和科技进步,优化问题构成了现实生活场景中一类极为广泛且复杂的挑战。它们在工程实践和科学研究等众多领域内频繁出现,包括路径规划^[1]、水资源与排污的协同管理^[2]、计算机视觉的智能巡检^[3],以及图像分割^[4]等。这些问题通常涉及多个需要同步优化的目标,这些目标经常相互冲突和相互制约,因此被称为多目标优化问题(MOP)。在应对MOP的过程中,最主要的问题是如何找到一组能够均匀分布在解空间并且相互之间没有支配关系的解集。许多研究人员花了很多时间研究这个问题,并提出了许多元启发式优化算法。这些算法包括遗传算法(GA)^[5]、差分进化算法(DE)^[6]、鸚鵡优化算法^[7]、非洲秃鹫优化算法^[8]、粒子群优化算法(PSO)^[9]等。其中,PSO作为一种极具代表性的群体智能算法,自1995年由Kennedy和Eberhart提出以来,便凭借其实现简便、收敛迅速的特点,在多目标优化问题的求解领域内得到了广泛的应用与推崇。

许多权威研究表明:PSO在应对MOP时表现出极其有利的优势。其中,多目标粒子群优化算法(MOPSO)^[10]是PSO的创新扩展,巧妙地模拟了鸟群的社会行为机制。它可以灵活应对复杂多变的优化挑战。在多目标优化领域,MOPSO已被证明是一种有效且可靠的解决方案,因为它操作简单、收敛迅速。然而,必须指出的是,大多数MOPSO算法在初始化种群时仍使用随机策略。这种随机性不仅可能阻碍种群向最优解的进化路径,还可能在搜索过程中引入不必要的随机误差,从而影响最终解的质量。为了克服这些局限性,研究者们不断探索并提出了多种经过改进的MOPSO算法。在混沌优化领域,尽管它擅长于单目标问题的求解,但研究者Khodadadi等^[11]在混沌优化领

域进行了勇敢的尝试。他们提出了多目标混沌博弈优化算法(MOCGO),以研究其在多目标优化中的潜在性能。MOCGO应用于八个复杂的工程设计问题,巧妙地结合了外部存档和领导选择机制。MOCGO在帕累托最优解的收敛性和覆盖率等方面均展现出了优于现有方法的表现。与此同时,Chu等^[12]针对地形与无人机性能约束下的路径规划问题,提出了改进的非线性动态惯性权重PSO算法。该算法通过精心设计的代价函数来规划路径,并在提升收敛速度和适应度方面取得了显著成效。然而,对于可飞路径的优化效果仍有待进一步提升。为此,他们进一步采用了速度自适应、混沌初始化以及改进后的logistic混沌映射等策略,从而显著增强了算法的整体性能。此外,Xu等^[13]也提出了将Circle映射与正弦余弦因子相结合的粒子群算法改进方案。他们借助Circle映射技术,实现了种群初始化的均匀性和多样性,这一做法不仅加快了算法的收敛速度,还提升了求解结果的精确度。另外,通过实施非线性递减的惯性权重以及正弦余弦因子策略,该算法在全局搜索与局部优化之间达到了更佳的平衡状态。

随着种群的不断迭代与进化,非支配解的数量会逐渐累积并显著增加。然而,作为存储精英解集的外部存档,其容量却存在固有的局限性。因此,制定一个高效且合理的存档管理策略变得至关重要,以便在持续涌现的新非支配解与已存储的解之间做出明智的选择与取舍。对于MOPSO算法而言,若收敛速度过快,则可能导致早熟现象,使算法陷入局部最优解而牺牲解的多样性;反之,若收敛速度过慢,则难以有效地逼近真实的帕累托前沿(PF)。为了提高存档中解的质量,研究者们对MOPSO算法进行了深入的探索与改进,旨在实现解的收敛性与多样性之间的最佳平衡。

一些研究者致力于将外部存档维护策略引入粒子群优化算法中,以期达到更好的效果。例如,Qu 等^[14]提出了一种创新的网格引导粒子群优化器,该算法利用网格技术识别有潜力的子区域,并生成多个子种群以保持解的多样性,同时采用外部存档维护策略来优化帕累托最优集,从而提高了搜索效率。针对 MOPSO 算法存在的过早收敛、收敛性和多样性不足等问题,Zhang 等^[15]提出了一种新颖的多目标优化算法,即综合指标与分层粒子群算法。该算法巧妙地结合了拐点距离和分布系数的外部存档策略,以及综合评价指标,有效地提升了算法的收敛性和解的多样性。此外,Chen 等^[16]引入两两竞争选优粒子,与全局领导者融合增强信息交互,提升全局探索力,结合全局提速与密度估计维护存档,平衡收敛性与多样性。与此同时,Liu 等^[17]也提出了一种创新的外部存档更新策略。首先将种群中的前半数优质解依次纳入存档,并据此重新构建存档的主导差分矩阵,以强化粒子的搜索方向,随后,剔除后半数相对较差的解,从而进一步提升了粒子的搜索效率。最后,单一的外部存档策略比较常见,所以 Zhou 等^[18]采用自适应网格与混合机制双重策略,确保非支配解均匀分布,防止种群退化,通过加权策略在存档中选优,增强多样性,提高粒子探索真实 Pareto 前沿概率。鉴于存档容量的有限性以及非支配解数量的不断增多,全面保留所有解将极大地增加计算负担。因此,实施有效的维护策略以筛选并剔除低质量的解变得尤为关键。

综上所述,粒子群算法尽管在多样性和收敛性方面已取得了一定进展,但如何在两者之间实现有效平衡仍然是当前算法面临的一大挑战。鉴于此,本文创新提出 CSMOPSO 算法,该算法巧妙地融合了 Cubic-Sine 混合映射与 Centre 策略。在算法初始阶段,采用 Cubic-Sine 混合映射技术初始化种群,显著提升多样性及全局搜索能力,为后续优化奠定坚实基础。其次,为有效管理非支配解,在外部存档中构建锥域结构,确保所有非支配解得到妥善保存,为算法迭代提供丰富解集资源。最后,面对存档容量限制,当达到存储上限时,采用 Centre 策略优化存档,计算并移除密集区域中心粒子,使其他粒子加速收敛至帕累托前沿,有效减少冗余解,进一步增强种群收敛性。通过这一系列改进与创新,本研究成功实现算法收敛性与多样性的平衡。

1 基础工作

1.1 多目标优化问题

多目标优化问题指的是在特定的约束条件下,需要同时改进多个目标函数以达到最优状态的问题,从

形式上讲,它可以被表述为

$$\min f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \quad (1)$$

$$\text{Subject to } g(\mathbf{x}) \leq 0, h(\mathbf{x}) = 0, \mathbf{x} \in \Omega \quad (2)$$

其中, $f(\mathbf{x})$ 表示 m 维的目标向量; $f_i(\mathbf{x})$ 是第 i 个目标函数; $\mathbf{x} = (x_1, x_2, \dots, x_n)$ 是决策变量向量; $g(\mathbf{x})$ 和 $h(\mathbf{x})$ 分别是不等式和等式约束; Ω 是可行解空间。

1.2 粒子群算法

粒子群算法(PSO)是1995年由 Kennedy 和 Eberhart 提出的群智能优化算法,模仿鸟群协作觅食。在 PSO 中,粒子代表鸟,位置为决策变量,速度由个体最优和全局最优决定,体现协作觅食。第 i 个粒子的速度和位置的更新公式如下:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pBest_i(t) - X_i(t)) + c_2 r_2 (gBest_i(t) - X_i(t)) \quad (3)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (4)$$

其中, ω 是惯性权重系数, c_1 和 c_2 是学习因子, r_1 和 r_2 是 $[0, 1]$ 内均匀生成的随机数。

速度更新式(3)由三部分组成, $\omega V_i(t)$ 是对先前速度的记忆项,它体现了上一次速度对当前速度方向和大小的延续性影响。其次, $c_1 r_1 (pBest_i(t) - X_i(t))$ 是该粒子当前位置和该粒子的历史最优位置之间的一个矢量,是自我认知项;最后 $c_2 r_2 (gBest_i(t) - X_i(t))$ 是社会认知项,它反映了粒子之间通过共享信息和协同工作来学习的能力,这种能力促使粒子在全局范围内更有效地探索。

2 算法设计

MOPSO 算法在追求性能时,面临的核心挑战在于如何巧妙地平衡主要种群与外部存档中的收敛性与多样性。相较于传统的 MOPSO,CSMOPSO 算法在两大关键环节上实现了显著改进:种群初始化策略的优化以及外部存档管理的创新。接下来,将深入解析这一提议的 CSMOPSO 算法,其操作流程如图 1 所示,该算法精妙地由三大核心模块构建而成。首先,为了从源头上提升初始种群的质量,CSMOPSO 算法独特地引入了 Cubic-Sine 混合映射技术,用于种群的初始化过程。这一创新举措不仅极大地丰富了种群的多样性,还有效增强了其全局搜索能力,为后续的优化过程奠定了坚实的基础。其次,为了实现对粒子信息的高效存储与即时更新,CSMOPSO 算法在外部存档中开创性地构建了锥域结构,专门用于储存非支配解,为算法的后续迭代提供了宝贵的解集资源。最后,针对外部存档可能溢出的难题,CSMOPSO 算法精心设计了 Centre 策略。这一策略能够在保留收敛性与多样性俱佳的粒子的同时,精准剔除密集区域的冗余粒子,从而推动种群更加迅速、准确地逼近真实的 PF(Pareto 前沿)。

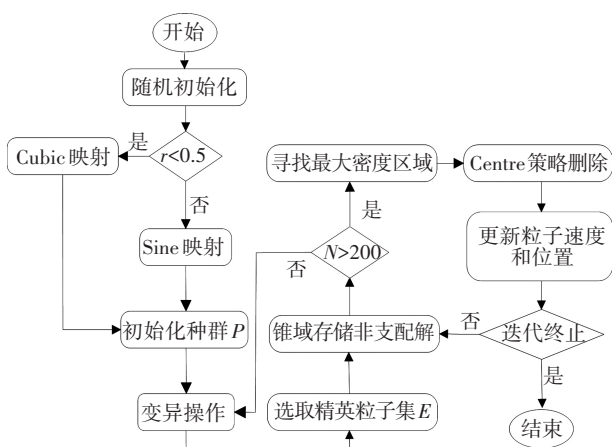


图 1 CSMOPSO 算法流程示意图

Fig. 1 Flowchart of CSMOPSO algorithm

2.1 Cubic-Sine 混合映射

先进的初始化技术在优化中至关重要,它能有效提高找到全局最优解的可能性,减少成本,并提升解的整体质量。然而,目前大多数处理 MOPSO 问题的算法都依赖随机生成的初始种群,虽然这样做能够增加搜索的广泛性,但也可能引入随机误差,从而影响收敛的精确度和速度。因此,开发更加先进的初始化技术变得尤为重要,这些技术应能够更精确地定位潜在的最优解区域,从而加快收敛进程,并降低误差带来的不利影响。

为了解决 MOPSO 算法初始种群随机性带来的问题,本文提出了一种结合 Cubic 混沌映射式(5)与 Sine 混沌映射式(6)的策略,旨在增强解的多样性和全局搜索能力。具体而言,首先生成一个在 0~1 之间均匀分布的混沌序列,并遍历这个序列的每个值。根据设定的混沌参数阈值 0.5,本文选择性地对这些值应用 Cubic 映射或 Sine 映射。

紧接着,本文将这些经过混沌映射且范围调整后的数值加以整合,塑造出初始种群中的各个个体,进而打造出一个优质的初始化种群。通过巧妙地交替运用 Cubic 混沌映射式(5)与 Sine 混沌映射式(6),本文开创了一种融合两者复杂动力学特征的新型混沌序列生成方法。此方法赋予种群在较少的迭代次数内就能探寻到更高品质解的能力,不仅有效节省了计算资源,还显著提升了算法的整体性能与效率。本文所采用的 Cubic-Sine 混沌映射函数具体表述如下:

$$\text{Cubic}(x_{n+1}) = 4x_n^3 + 3x_n \quad (5)$$

$$\text{Sine}(x_{n+1}) = r \cdot \sin(\pi \cdot x_n) \quad (6)$$

其中,Sine 函数本身不是混沌的,但可以通过某些变形或与其他函数组合来产生混沌行为。这里使用了一个简单的变形,其中 r 是一个缩放因子, $r=0.9$ 。

本文通过利用式(7)描述生成的混沌序列从而确定每一个粒子的位置

$$p(i, j) = \text{lb}(j) + \text{chaos_var} * (\text{ub}(j) - \text{lb}(j)) \quad (7)$$

式(7)采用一个混沌映射函数(如 Cubic 或 Sine 函数)对初始值为[0, 1]区间内随机数的混沌变量 chaos_var 进行多次迭代,随后将这个混沌变量通过线性映射转换到特定的搜索空间内,以此来确定初始化种群中每个粒子的位置。

2.2 Centre 策略维护外部存档

在 MOPSO 算法中,精选并保留优秀粒子对于获取既收敛又多样化的最优解具有举足轻重的作用。外部存档作为搜索过程中非支配解的记录库,为粒子的选择提供了宝贵的备选资源,但其容量却受限于种群规模。当非支配解的数量超出存档容量时,维护存档以确保其不超标的任务便显得尤为重要,其核心在于精心挑选那些兼具收敛性和多样性的解,一旦超出容量限制,则需审慎剔除部分解。为此,本研究创新性地提出了锥域划分策略以更新存档,并配套采用 Centre 策略进行存档的维护。该策略巧妙地将非支配解分配到各个锥域之中,当外部存档的容量超出预设范围时,便依据 Centre 策略从密度最高的区域中剔除中心粒子,从而确保存档中始终留存着最优质、最具代表性的非支配解。接下来将详尽阐述外部存档的更新与维护机制,以保障存档内容的持续优化与精练。

传统上, MOPSO 算法采用自适应网格策略来保存外部存档中的非支配解,但每当粒子超出网格边界时,便需重新计算网格,这一过程需要对非支配解频繁地进行网格划分与更新,从而导致了较高的计算复杂度,尤其在处理规模庞大的优化问题时,这一问题更为凸显。为了突破这一限制,CSMOPSO 算法引入了一项新颖策略:借助锥域来对外部存档进行划分。具体来说,该策略以原点为基准点,将外部存档分割成多个等面积的锥形区域。紧接着,算法会精确计算每个非支配解与 x 轴上单位向量之间的夹角,并依据夹角的具体数值,将这些解有序地分配到对应的锥形区域内进行存储,如图 2 所直观展示。这一创新方法确保了所有非支配解都能一次性地被高效且准确地收纳进外部存档之中。

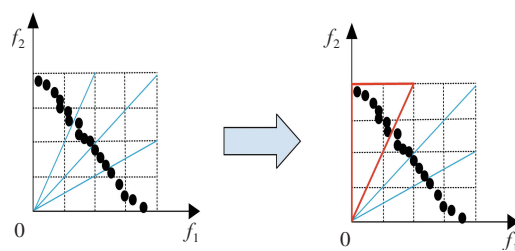


图 2 非支配解的存储

Fig. 2 Storage of non-dominated solutions

当外部存档的容量超过设定的最大值时,算法会

启用 Centre 策略来进行有效的存档管理。具体执行流程如下:首先,在图 2 所示的场景中,算法会精准识别出外部存档中容纳非支配解数量最多的锥形区域,这一步骤的结果清晰地呈现在图 3 中;随后,算法利用欧式距离式(8)来计算该锥形区域内每个非支配解与其他所有非支配解之间的具体距离,并将这些距离值妥善保存;紧接着,算法会计算这些距离的平均数值,并找出平均距离最小的非支配解,即该锥形区域内最为中心的非支配解,然后将其从存档中移除,如图 3 所示。这一过程会循环进行,直至外部存档的容量重新符合限制要求。此时,外部存档中的粒子已被视为精英粒子集,算法将停止进一步的删除动作。

$$d_{ij} = \text{sqrt} \sum_{k=1}^n (x_{ik} - x_{jk})^2 \quad (8)$$

其中, d_{ij} 代表第 i 个粒子与第 j 个粒子的欧式距离, x_{ik} 和 x_{jk} 是第 i 个粒子和第 j 个粒子在 k 维上的坐标, n 是空间维度。



图 3 外部存档的维护

Fig. 3 Maintenance of external archive

2.3 算法的核心执行步骤

算法的核心执行步骤如下:

- (1) 配置 CSMOPSO 算法的必要参数。
- (2) 使用 Cubic-Sine 混合映射来初始化种群。
- (3) 对粒子实施变异处理,并评估其适应度值。
- (4) 根据非支配排序原则,挑选出精英粒子集合 E 。
- (5) 锥域划分策略存储非支配解以及 Centre 策略维护存档。
- (6) 选取 $gBest$ 和 $pBest$ 。
- (7) 通过式(3)和式(4)更新粒子的速度和位置。
- (8) 若当前的迭代轮次尚未触及预设的最大值,则返回到步骤(2)继续执行迭代;反之,则终止迭代并输出最终优化解集合。

3 仿真实验与结果分析

3.1 测试问题

为了全方位地评估 CSMOPSO 算法的性能表现,本文采纳了 ZDT^[19] 和 UF^[20] 这两个基准测试套件中的共计 15 个测试问题。这些测试问题涵盖了 ZDT 基准套件的 5 个双目标测试案例,以及 UF 基准套件的 7 个双目标测试案例和 3 个三目标测试案例。这些测试问题

的特征丰富多样,其 Pareto 前沿形状复杂多变,涵盖了凹凸性、多模态性和不规则性等特征,为验证算法的可靠性和效率提供了理想的测试平台。在决策变量数量的配置上,对于双目标测试问题,本文主要选取了 30 个决策变量作为标准配置,但针对 ZDT4 和 ZDT6 这两个特例,则灵活地将决策变量数量调整为 10 个。而对于三目标测试问题,则统一采用了 30 个决策变量的配置。这样的设置安排有助于本文深入且全面地探究 CSMOPSO 算法在处理各类不同特性及维度的多目标优化问题时的综合表现。

为了进行对比实验,本文选取了 4 个先进的 MOPSO (包括 NMPSO^[21]、MPSOD^[22]、MOPSO^[10] 和 MOPSOCD^[23]) 和 4 个先进的 MOEA (包括 SPEAR^[24]、NSGAIII^[25]、IDBEA^[26] 和 DGEA^[27]), 为每个测试问题都详细设定了实验参数,如总体粒子数 (N)、目标数 (M)、决策空间维度数 (D) 和最大评估数 (FES),并将这些详细参数列在表 1 中。在接下来的部分,本文将深入阐述实验的具体实施步骤及所取得的成果。为确保算法性能对比的公正性,所有对比算法的相关参数设置均严格遵循其原始参考文献中的规定,各算法的核心参数配置如表 1 所示。

表 1 15 个测试问题的参数设置

Table 1 Parameter settings of 15 test problems

Problem	N	M	D	FES
ZDT1~ZDT3	200	2	30	10 000
ZDT4 and ZDT6	200	2	10	10 000
UF1~UF7	200	2	30	10 000
UF8~UF10	200	3	30	10 000

每个算法的总体规模均设定为 200,且函数评估的最大次数限制在 10 000 次。对于每个测试问题,每种算法都单独运行了 30 次,随后记录下这 30 次运行中得到的 IGD 和 HV 值。这些算法的结果是在配备英特尔(R)核心(TM)i7-7700 CPU@60 GHz 处理器的 Windows 10 系统上,使用 MATLAB R2023b 软件以 60 GHz 的频率计算得出的,PlatEMO 平台为比较中的每一种算法都提供了源代码支持。

3.2 性能评估标准

本文采用了两个性能指标来评估性能,分别是反世代距离 (IGD)^[28] 和超体积 (HV)^[29]。

(1) IGD 作为一项综合性的性能指标,旨在精确度量算法所得 Pareto 最优解集与真实 Pareto 前沿 (PF) 之间的接近程度,进而对算法的收敛性能与多样性表现做出有效评判。具体而言,IGD 值越小,即意味着算法

在收敛性和多样性上的展现越来越出色,IGD 的计算公式如下:

$$IGD(S, |P^*|) = (1/|P^*|) \sum_{x^* \in P^*} \text{dist}(S, x^*) \tag{9}$$

其中, $\text{dist}(S, x^*)$ 为 P^* 中解 x^* 与 S 之间的最小欧式距离, S 是算法获得的帕累托最优解集; P^* 是均匀分布在真实帕累托前沿上的解集。

(2) HV 指标也是一种综合性的评估工具,它用于衡量算法所得解集的性能。这个指标是通过比较算法得到的解集 PS 与一个在目标空间中均匀分布的样本点集合所围成的体积,来估计算法解集的收敛性和多样性。具体来说, HV 值越大,意味着算法的解集不仅更接近全局最优(即收敛性好),而且解集内的解之间差异较大(即多样性好)。在计算 HV 时,通常会引入一个参考点 $Z' = (Z'_1, Z'_2, \dots, Z'_m)$, 则 HV 的计算公式为

$$HV(S) = \delta(\cup_{x \in S} [f_1(x), z'_1] \times \dots \times [f_m(x), z'_m]) \tag{10}$$

式(10)中, S 为算法得到的帕累托解集, δ 为勒贝格测度。实验参数设置如表 2 所示。

表 2 CSMOPSO 等比较算法的参数设置
Table 2 Parameter settings of CSMOPSO and compared algorithms

算法名称	参数设置
MOPSO	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], \text{div} = 10$
NMPSO	$\omega \in [0.1, 0.5], c_1, c_2, c_3 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
MOPSOCD	$\omega \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5]$
MPSOD	$\omega \in [0.1, 0.9], c_1, c_2, c_3 \in [1.5, 2.5], p_c = 0.9, p_m = 1/n, F = 0.5, CR = 0.5, \eta_m = 20, \eta_c = 20$
DEGA	$R = 10$
SPEAR	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
NSGAIH	$p_m = 1/n, p_c = 1.0, \eta_m = 20, \eta_c = 20$
IDBEA	$p_m = 1/D, p_c = 1.0$
CSMOPSO	$\omega \in [0.4, 0.9], c_1 = c_2 = 2, p = 1/N$

3.3 实验结果与数据分析

根据对表 3 和表 4 数据的深入分析,可以明确得出以下结论:本文所提出的 CSMOPSO 算法在综合性能方

面显著优于其他 8 个对比算法。具体而言,表 3 详细展示了 CSMOPSO 与 8 种基于 PSO 的多目标优化算法(包括 NMPSO、MPSOD、MOPSO、MOPSOCD、SPEAR、NSGAIH、IDBEA 和 DGEA)在 15 个测试问题上的 IGD 指标表现,其中平均值以“Mean”标记,标准偏差则以“Std”表示。在这些数据中,CSMOPSO 算法取得的 IGD 最佳结果被特别加粗,凸显了其卓越的性能。尤为值得一提的是,在表 3 的总结行中,CSMOPSO 算法在 15 个测试问题中夺得了 10 个 IGD 最佳值,这充分证明了其在多数测试场景下均能保持最优的 IGD 指标,表现极为亮眼。详细来看,在 ZDT 基准套件包含的 4 个测试问题(ZDT1、ZDT2、ZDT3 和 ZDT4)上,CSMOPSO 算法的表现全面超越了 8 种对比算法。而在 UF 系列基准套件中,对于 6 个测试问题(UF3、UF4、UF7、UF8、UF9 和 UF10),CSMOPSO 算法的性能同样优于 8 种对比算法。对于 UF 系列中的其他测试问题,CSMOPSO 算法的表现同样可圈可点:在 UF2 和 UF5 上,其性能仅次于 SPEAR 和 NSGAIH;在 UF6 上,它紧随 NMPSO 和 NSGAIH 之后,位居第 3;而在 UF1 上,它排在第 4 位,仅落后于 NMPSO、SPEAR 和 NSGAIH。

表 4 详细呈现了 CSMOPSO 与其他 8 种对比算法在 HV 指标上的对比结果。与 IGD 指标的对比结果相类似,CSMOPSO 在 ZDT 和 UF 这两大基准测试套件上均进行了 30 次独立的运行测试,涵盖了全部的 15 个基准问题。在其中的 10 个测试问题上,CSMOPSO 的表现尤为出色,均取得了最佳成绩,并且在所有对比算法中脱颖而出,展现了其卓越的性能。特别是在收敛性方面,CSMOPSO 更是展现出了相较于其他对比算法的明显优势。总体来看,CSMOPSO 收获了 10 个最佳的 HV 值,而其他的对比算法,如 NMPSO、MPSOD、MOPSO、MOPSOCD、SPEAR、NSGAIH、IDBEA 和 DGEA,则分别仅获得了 1、0、0、0、1、2、0、1 个最佳 HV 值。表 4 中列出的最佳 HV 值数据与 IGD 指标所得的结果高度一致,进一步有力地验证了 CSMOPSO 算法在多数测试问题上均能够取得最佳的 HV 值。从实验数据的反馈来看,CSMOPSO 算法与所挑选的对比算法相比,依然展现出了强大的竞争力。

表 3 CSMOPSO 算法与 8 个其他算法在 15 个测试问题上进行 IGD 性能指标对比的结果

Table 3 Comparison of IGD performance indicators between the CSMOPSO algorithm and eight other algorithms on 15 test problems

测试问题	IGD	NMPSO	MPSOD	MOPSO	MOPSOCD	SPEAR	NSGAIH	IDBEA	DGEA	CSMOPSO
ZDT1	Mean	0.6591e-2	1.1019e-1	1.5565e+0	1.8388e-2	1.8738e-1	7.5401e-2	5.3413e-1	1.0414e+0	1.0982e-2
	Std	(1.24e-2)	(4.76e-2)	(7.16e-2)	(4.75e-2)	(0.39e-2)	(1.26e-2)	(6.76e-2)	(0.42e-1)	(0.21e-4)

续表(表3)

测试问题	IGD	NMPSO	MPSOD	MOPSO	MOPSOCD	SPEAR	NSGAIII	IDBEA	DGEA	CSMOPSO
ZDT2	Mean	5.2888e-2	1.5883e-1	0.9762e+0	1.8469e-1	4.0225e-1	1.5143e-1	1.9297e+0	8.9422e-1	9.1540e-3
	Std	(8.66e-2)	(1.21e-1)	(1.92e-1)	(0.12e-1)	(1.44e-1)	(0.75e-2)	(0.05e-1)	(0.72e-1)	(1.63e-4)
ZDT3	Mean	8.9765e-2	1.9330e-1	1.1707e+0	5.9712e-2	1.4730e-1	7.1677e-2	0.1796e-1	1.0047e+0	9.5466e-3
	Std	(0.13e-2)	(5.18e-2)	(7.42e-2)	(6.89e-2)	(1.55e-2)	(1.12e-2)	(4.30e-2)	(0.33e-1)	(1.18e-3)
ZDT4	Mean	1.6833e+1	0.8328e+1	1.0041e+1	0.0862e+1	0.1863e+0	1.9940e+0	0.9102e+1	7.1153e+0	8.4111e-1
	Std	(5.84e+0)	(6.69e+0)	(4.46e+0)	(7.82e+0)	(8.18e-1)	(6.98e-1)	(6.77e+0)	(4.48e+0)	(0.72e-1)
ZDT6	Mean	0.3625e-3	1.8129e-2	5.6768e+0	0.4751e-3	1.0891e+0	1.1587e+0	4.4576e+0	1.9199e-3	6.9346e-3
	Std	(0.71e-4)	(1.08e-2)	(0.76e-1)	(1.71e-3)	(0.17e-1)	(0.26e-1)	(0.25e-1)	(5.47e-4)	(5.33e-5)
UF1	Mean	1.3150e-1	0.7711e-1	4.7270e-1	6.6327e-1	1.3989e-1	1.2135e-1	0.9116e-1	6.9740e-1	1.4986e-1
	Std	(0.05e-2)	(4.04e-2)	(7.77e-2)	(1.42e-1)	(0.99e-2)	(0.41e-2)	(6.94e-2)	(1.51e-1)	(1.44e-2)
UF2	Mean	8.4032e-2	1.1061e-1	1.0558e-1	1.4117e-1	7.4025e-2	7.6498e-2	1.6487e-1	1.5944e-1	8.1908e-2
	Std	(5.98e-3)	(1.04e-2)	(1.33e-2)	(1.20e-2)	(7.31e-3)	(5.35e-3)	(1.59e-2)	(0.19e-2)	(5.76e-3)
UF3	Mean	0.6546e-1	5.0314e-1	4.5879e-1	0.8186e-1	4.3222e-1	4.6068e-1	6.0126e-1	5.5970e-1	0.4765e-1
	Std	(6.78e-2)	(1.46e-2)	(0.04e-2)	(7.00e-2)	(1.81e-2)	(1.25e-2)	(0.17e-2)	(0.77e-2)	(0.66e-2)
UF4	Mean	6.4796e-2	9.8518e-2	1.8177e-1	7.9631e-2	8.5873e-2	8.8665e-2	1.0694e-1	1.2274e-1	6.4718e-2
	Std	(7.73e-3)	(4.12e-3)	(0.38e-3)	(9.46e-3)	(0.11e-3)	(0.70e-3)	(0.05e-3)	(9.36e-3)	(5.08e-3)
UF5	Mean	1.7750e+0	0.8033e+0	0.6121e+0	0.8611e+0	1.1804e+0	1.0966e+0	0.9743e+0	0.1098e+0	1.7327e+0
	Std	(4.65e-1)	(0.08e-1)	(0.11e-1)	(4.87e-1)	(1.90e-1)	(0.55e-1)	(0.68e-1)	(6.28e-1)	(1.27e-1)
UF6	Mean	6.7252e-1	1.3391e+0	0.0588e+0	0.5811e+0	7.0148e-1	5.8570e-1	1.9746e+0	0.5491e+0	6.9710e-1
	Std	(1.31e-1)	(0.21e-1)	(0.63e-1)	(7.95e-1)	(9.36e-2)	(6.90e-2)	(0.29e-1)	(7.16e-1)	(6.79e-2)
UF7	Mean	0.2506e-1	0.4803e-1	6.1652e-1	6.0049e-1	1.8261e-1	1.7630e-1	4.4433e-1	7.1200e-1	1.2483e-1
	Std	(1.60e-1)	(7.04e-2)	(8.08e-2)	(1.56e-1)	(5.50e-2)	(6.96e-2)	(9.92e-2)	(1.43e-1)	(1.31e-2)
UF8	Mean	4.6217e-1	5.5728e-1	4.3218e-1	7.6255e-1	0.2164e-1	0.7550e-1	0.5303e-1	6.8705e-1	0.4655e-1
	Std	(7.47e-2)	(5.17e-2)	(0.72e-2)	(1.58e-1)	(1.87e-2)	(4.92e-2)	(0.67e-2)	(1.63e-1)	(0.70e-2)
UF9	Mean	4.5019e-1	6.6221e-1	5.4509e-1	9.0611e-1	4.7212e-1	4.9274e-1	5.4234e-1	7.5556e-1	1.5911e-1
	Std	(8.07e-2)	(4.02e-2)	(0.44e-2)	(1.65e-1)	(6.51e-2)	(4.07e-2)	(5.29e-2)	(8.87e-2)	(1.80e-2)
UF10	Mean	1.4837e+0	4.1440e+0	1.4074e+0	4.8471e+0	1.9856e+0	1.6520e+0	0.9044e+0	4.6000e+0	6.7638e-1
	Std	(0.94e-1)	(0.48e-1)	(0.18e-1)	(9.05e-1)	(0.68e-1)	(4.59e-1)	(5.78e-1)	(9.54e-1)	(0.23e-1)
Best/All		0/15	0/15	0/15	0/15	1/15	3/15	0/15	1/15	10/15

表 4 CSMOPSO 算法与 8 个其他算法在 15 个测试问题上进行 HV 性能指标对比的结果

Table 4 Comparison of HV performance indicators between the CSMOPSO algorithm and eight other algorithms on 15 test problems

测试问题	HV	NMPSO	MPSOD	MOPSO	MOPSOCD	SPEAR	NSGAIII	IDBEA	DGEA	CSMOPSO
ZDT1	Mean	6.9313e-1	5.6363e-1	0.0000e+0	7.0165e-1	4.8444e-1	6.2197e-1	1.5141e-1	0.8977e-2	7.1202e-1
	Std	(1.36e-2)	(6.28e-2)	(0.00e+0)	(5.91e-2)	(0.37e-2)	(1.58e-2)	(4.23e-2)	(9.29e-2)	(0.08e-4)
ZDT2	Mean	4.0387e-1	0.6487e-1	0.0000e+0	0.9566e-1	7.6911e-2	0.5518e-1	0.0000e+0	0.6247e-2	4.3996e-1
	Std	(7.77e-2)	(1.09e-1)	(0.00e+0)	(1.52e-1)	(5.15e-2)	(0.44e-2)	(0.00e+0)	(6.54e-2)	(0.43e-4)
ZDT3	Mean	5.7208e-1	4.6608e-1	9.8143e-4	5.6321e-1	5.0588e-1	5.5383e-1	0.7828e-1	0.2812e-2	5.9588e-1
	Std	(8.03e-3)	(5.86e-2)	(0.65e-3)	(4.85e-2)	(0.20e-2)	(7.57e-3)	(0.10e-2)	(4.89e-2)	(4.50e-4)
ZDT4	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.7219e-4	0.8178e-4	0.0000e+0	0.0000e+0	1.4349e-1
	Std	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(1.49e-3)	(0.09e-3)	(0.00e+0)	(0.00e+0)	(1.18e-1)

续表(表4)

测试问题	HV	NMPSO	MPSOD	MOPSO	MOPSOCD	SPEAR	NSGAIII	IDBEA	DGEA	CSMOPSO
ZDT6	Mean	0.8970e-1	0.7434e-1	0.0000e+0	0.8860e-1	5.6598e-4	0.0000e+0	0.0000e+0	0.9007e-1	0.8497e-1
	Std	(0.35e-4)	(1.01e-2)	(0.00e+0)	(1.54e-3)	(0.29e-3)	(0.00e+0)	(0.00e+0)	(4.98e-4)	(1.42e-4)
UF1	Mean	5.2052e-1	0.4682e-1	1.8933e-1	6.7014e-2	5.1016e-1	5.3501e-1	0.3408e-1	6.8660e-2	4.8904e-1
	Std	(4.04e-2)	(4.31e-2)	(5.81e-2)	(8.32e-2)	(0.75e-2)	(0.23e-2)	(6.08e-2)	(6.80e-2)	(0.17e-2)
UF2	Mean	6.1827e-1	5.8129e-1	6.0963e-1	5.4022e-1	6.2624e-1	6.2587e-1	5.0183e-1	5.2712e-1	6.2672e-1
	Std	(7.30e-3)	(1.20e-2)	(7.68e-3)	(1.58e-2)	(7.21e-3)	(5.34e-3)	(1.90e-2)	(0.11e-2)	(5.90e-3)
UF3	Mean	0.7921e-1	1.6942e-1	1.8169e-1	0.4899e-1	0.2374e-1	1.9516e-1	1.0083e-1	1.2519e-1	0.0312e-1
	Std	(5.71e-2)	(1.37e-2)	(0.35e-2)	(5.60e-2)	(1.50e-2)	(9.57e-3)	(1.46e-2)	(0.20e-2)	(0.23e-2)
UF4	Mean	0.5927e-1	0.0863e-1	0.1189e-1	0.3204e-1	0.2409e-1	0.2292e-1	0.9526e-1	0.7698e-1	0.5420e-1
	Std	(1.09e-2)	(5.80e-3)	(0.76e-3)	(1.22e-2)	(0.86e-3)	(4.71e-3)	(4.20e-3)	(1.11e-2)	(6.79e-3)
UF5	Mean	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	9.4987e-5	0.0000e+0	0.0000e+0	0.0000e+0	0.000e+0
	Std	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(5.20e-4)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)
UF6	Mean	1.6966e-2	0.0000e+0	0.0000e+0	0.0000e+0	8.5578e-3	0.4425e-2	0.0000e+0	0.0000e+0	0.610e-3
	Std	(0.34e-2)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(1.59e-2)	(0.34e-2)	(0.00e+0)	(0.00e+0)	(6.23e-3)
UF7	Mean	0.5422e-1	0.6274e-1	0.9023e-2	4.0650e-2	0.4964e-1	0.5921e-1	1.0594e-1	0.4279e-2	0.9791e-1
	Std	(1.04e-1)	(6.83e-2)	(0.31e-2)	(5.40e-2)	(4.94e-2)	(7.07e-2)	(5.75e-2)	(0.82e-2)	(1.88e-2)
UF8	Mean	0.8964e-1	5.7774e-2	0.5452e-1	8.9854e-3	1.8270e-1	0.6427e-1	1.7333e-1	0.7775e-2	0.2423e-1
	Std	(5.10e-2)	(0.34e-2)	(0.79e-2)	(1.08e-2)	(0.50e-2)	(0.76e-2)	(4.52e-2)	(0.73e-2)	(1.50e-2)
UF9	Mean	0.3084e-1	1.1386e-1	0.4877e-1	0.5878e-2	0.6995e-1	0.6939e-1	0.0772e-1	7.0068e-2	5.9052e-1
	Std	(7.06e-2)	(0.95e-2)	(0.70e-2)	(0.71e-2)	(5.89e-2)	(0.66e-2)	(0.81e-2)	(4.56e-2)	(0.07e-2)
UF10	Mean	0.0000e+0	0.0000e+0	1.2411e-5	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	1.3239e-1
	Std	(0.00e+0)	(0.00e+0)	(6.80e-5)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(0.00e+0)	(7.13e-2)
Best/All		1/15	0/15	0/15	0/15	1/15	2/15	0/15	1/15	10/15

3.4 Friedman 秩实验

在全面评估算法性能的过程中,采纳了权威的 Friedman 秩检验^[30]统计方法,对 CSMOPSO 算法在表 3 和表 4 中的表现进行了深入地对比分析。具体而言,针对 ZDT 和 UF 基准测试集上的 IGD 指标实施了 Friedman 检验,并据此得出了各参与算法的平均排名,详细结果可见于表 5。同时,对于 HV 指标的比较排名情况,请参见表 6。这些排名数据全面覆盖了 9 种不同算法在 15 个测试问题上的性能展现。在对比 IGD 和 HV 这两个核心性能指标时,清晰地观察到 CSMOPSO 算法展现出了压倒性的优势。无论是在 IGD 还是 HV 指标上,无论是针对 ZDT、UF 还是整体的基准测试集,CSMOPSO 算法均取得了最为优异的排名。这一卓越表现充分有力地证明了 CSMOPSO 在 IGD 和 HV 指标上的出色性能。

综合 IGD、HV 以及 Friedman 秩检验的结果,可以得出:相较于其他参与对比的算法,所提出的 CSMOPSO 算法在求解多目标优化问题(MOP)时,展现

出了更为卓越的综合性能,能够稳定地获得既收敛又具备高度多样性的更优解。

表 5 Friedman 秩实验对比分析 CSMOPSO 与所有对比算法的 IGD 数值

Table 5 Comparative analysis of the IGD values of CSMOPSO and all comparison algorithms using the Friedman rank test

IGD	ZDT1-4,6		UF1-10		Overall	
	Friedman test	rank	Friedman test	rank	Friedman test	rank
NMPSO	0.4	2	0.1	3	0.20	2
MPSOD	5.8	6	6.1	5	6.00	5
MOPSO	8.2	8	5.9	4	6.67	7
MOPSOCD	0.8	3	7.3	7	6.13	6
SPEAR	5.2	4	0.1	3	0.80	4
NSGAIII	0.8	3	0.0	2	0.27	3
IDBEA	7.6	7	6.4	6	6.80	8
DGEA	5.6	5	8.2	8	7.33	9
CSMOPSO	1.6	1	1.9	1	1.80	1

表 6 Friedman 秩实验对比分析 CSMOPSO 与所有对比算法的 HV 数值

Table 6 Comparative analysis of HV values between CSMOPSO and all comparison algorithms using the Friedman rank test

HV	ZDT1-4,6		UF1-10		Overall	
算法	Friedman test rank	Friedman test rank	Friedman test rank	Friedman test rank	Friedman test rank	Friedman test rank
NMPSO	0.1	2	0.95	2	0.00	2
MPSOD	5.3	6	6.15	6	5.87	6
MOPSO	8.2	9	5.85	5	6.63	7
MOPSOCD	0.5	3	6.55	7	5.53	5
SPEAR	5.2	5	0.50	3	4.07	4
NSGAIII	4.6	4	0.55	4	0.9	3
IDBEA	7.4	8	6.75	8	6.97	8
DGEA	6.1	7	1.55	9	7.07	9
CSMOPSO	1.6	1	0.15	1	1.97	1

3.5 图形比较分析

为了直观地展现各算法间的性能差异,图 4、图 5 分别呈现了在二维目标测试函数 ZDT3 和三维目标测试函数 UF9 上,各算法的帕累托前沿对比结果。从这些精细

的对比图中,可以清晰地观察到,本文所提出的 CSMOPSO 算法所得到的帕累托最优解集最接近真实的 PF 前沿。在 ZDT3 的测试中,尽管 NMPSO 算法得到的 Pareto 最优解较为接近真实的 PF,但其分布却呈现出明显的不均匀性,主要集中于中间区域;而 MOPSOCD 和 NSGAIII 算法虽然能够获得分布较为均匀的 Pareto 最优解,但这些解却远离了真实的 PF。同样地,在 UF9 的测试中,CSMOPSO 算法所得到的 Pareto 最优解也展现出了对理想状态的出色逼近能力。相比之下,其他对比算法的表现则略显逊色,它们的帕累托解集要么过于分散,缺乏紧凑性,要么陷入了局部最优的困境,都未能像 CSMOPSO 算法那样均匀且精确地逼近真实的 PF。从图 4、图 5 的直观展示中,可以明确地看出,CSMOPSO 算法在收敛性和分布性这两个关键方面,相对于其他算法而言,都展现出了更为突出的优势。这充分说明了 Cubic-Sine 混合映射和 Centre 策略在平衡收敛性和多样性方面所发挥的积极作用,使得 CSMOPSO 算法在多目标优化问题的求解中表现得更加出色。

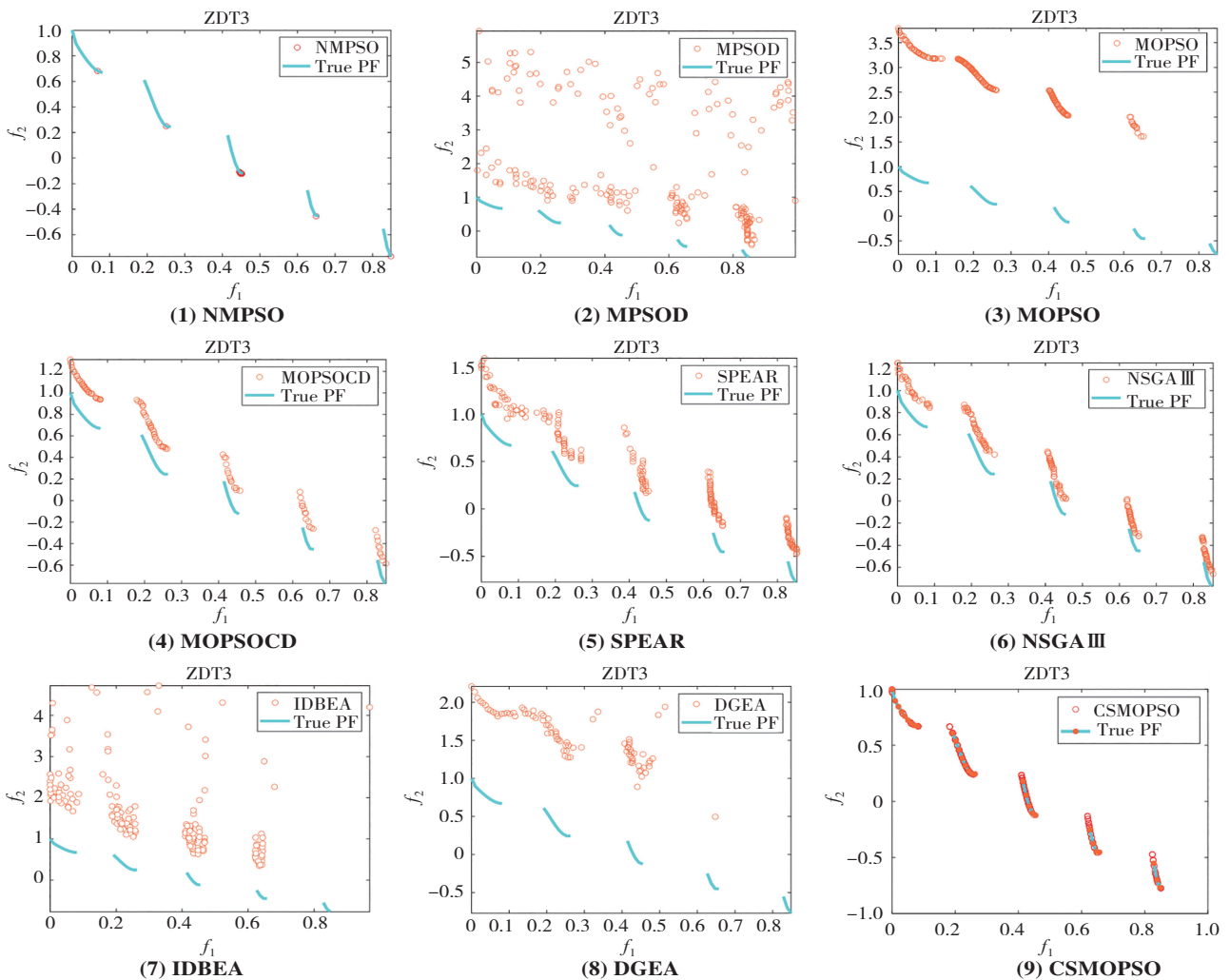


图 4 9 个算法在 ZDT3 测试问题上的 Pareto 前沿可视化图
Fig. 4 Visualization of the Pareto front of nine algorithms on the ZDT3 test problem

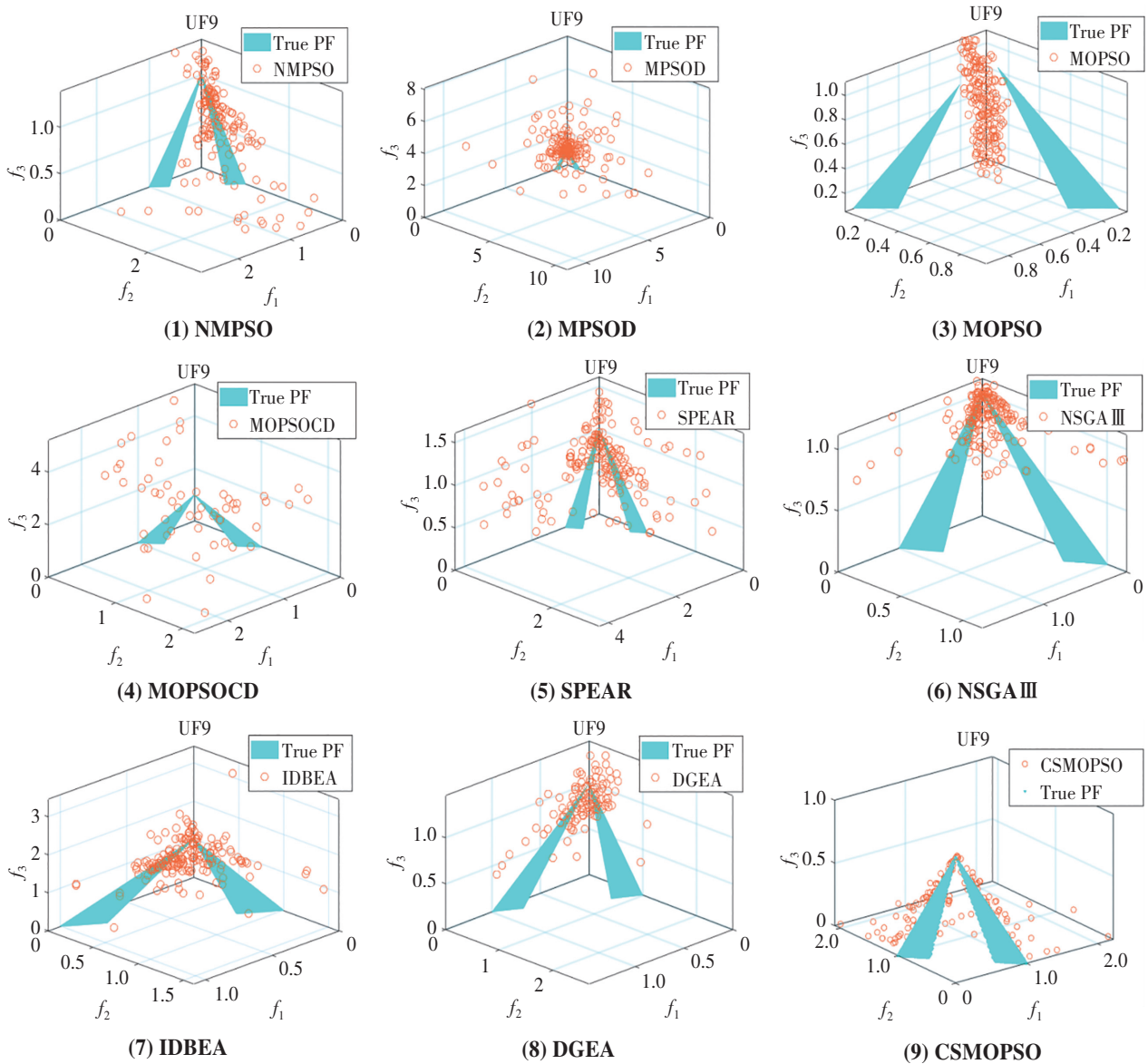
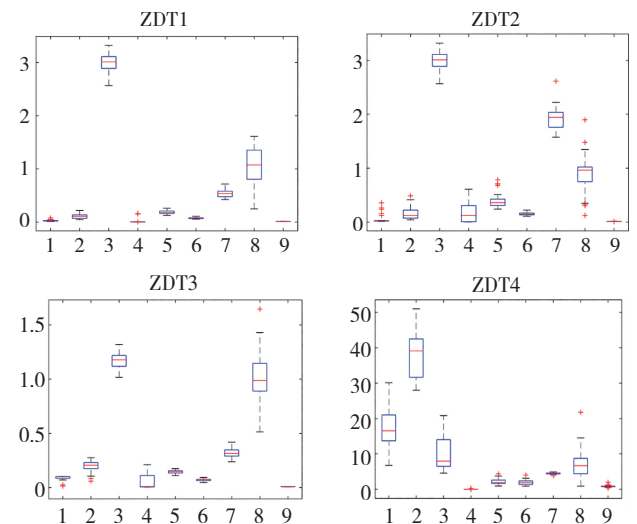


图 5 9 个算法在 UF9 测试问题上的 Pareto 前沿可视化图

Fig. 5 Visualization of the Pareto front of nine algorithms on the UF9 test problem

为了直观地衡量各算法在 ZDT 和 UF 系列测试问题上的稳定性能,图 6 采用了统计箱型图的形式,展示了 CSMOPSO 算法与其他 8 个算法在多个测试问题上各自独立运行 30 次后,所得到的 IGD 值的分布情况。箱型图的横坐标依次标记了 NMPSO、MPSOD、MOPSO、MOPSOCD、SPEAR、NSGAIII、IDBEA、DGEA 及 CSMOPSO 这 9 种算法,而纵坐标则代表算法的 IGD 值。观察这些图表,可以发现图 6 中的结果与表 3 中的对比结果相吻合。重要的是箱型图中 IGD 值越低、图框越短,意味着算法的平均 IGD 值表现越佳。作为箱型图的最后一个方框,本文提出的 CSMOPSO 算法在多数测试问题上展现出了比其他算法更稳定的表现。这意味着,在解的质量和算法稳定性方面,CSMOPSO 算

法相较于其他 8 种算法具有显著优势。



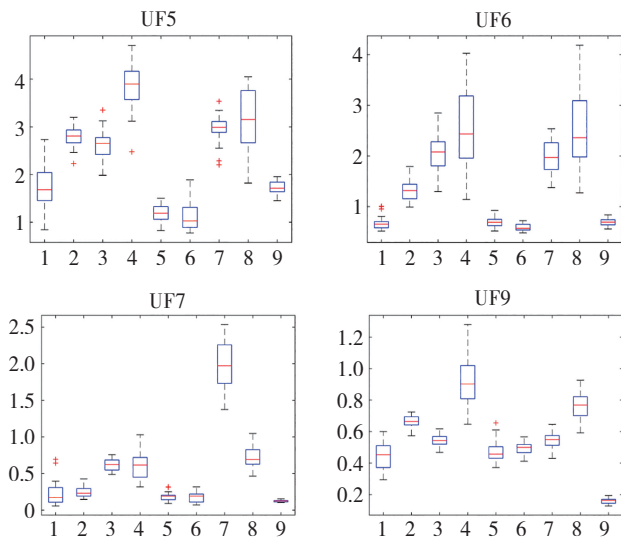


图 6 9 个算法针对多个测试问题所得到的 IGD 值的箱型分布图

Fig. 6 Box plots of IGD values obtained by nine algorithms on multiple test problems

为了深入探究该算法的收敛速率,本文选取了 3 个特定的测试函数进行评估。图 7 展示了 CSMOPSO 算法与另外 8 种对比算法在 ZDT2、UT8 以及 UF10 这 3 个测试问题上的收敛速度对比。从图 7 中可以观察到,该算法在迭代的初期阶段便迅速逼近真实的 PF,展现出了卓越的收敛性能。

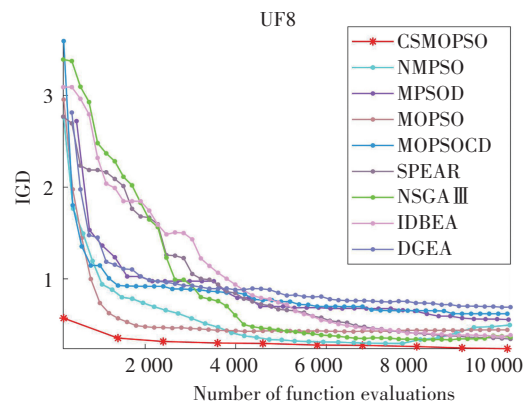


图 7 9 个算法在 ZDT2、UF8 和 UF10 测试问题上的 IGD 收敛轨迹

Fig. 7 IGD convergence trajectories for nine algorithms on ZDT2, UF8, and UF10 test problems

综上所述,结合 PF 图、箱型图以及收敛轨迹图的全面评估结果,可以确信:相较于其他几个同样具备竞争力的算法,所提出的 CSMOPSO 算法展现出了更为卓越的综合性能。这一卓越表现充分证明了 CSMOPSO 算法所采用的 Cubic-Sine 混合映射与 Centre 策略在有效平衡算法的收敛速度与多样性方面展现出了更为出色的能力。

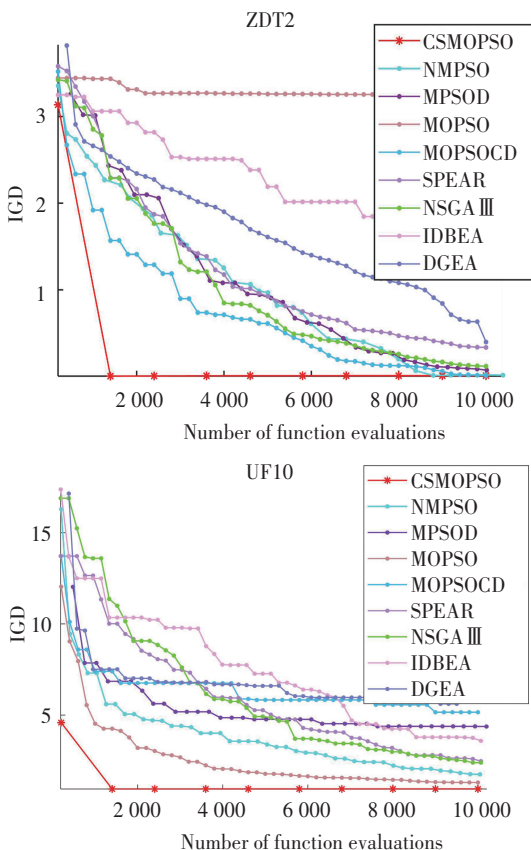
3.6 算法的有效性分析

3.6.1 CSMOPSO 及其变体的消融实验

为了严谨地检验 CSMOPSO 中策略的有效性,我们设计了一系列消融实验,通过逐一去除策略来评估其对算法性能的影响。这种方法的核心在于控制变量,从而明确每个组件在系统中的作用和价值。在本研究中设立了两组实验,每组实验都专注于探究 CSMOPSO 算法中一个关键策略的重要性。第一组实验(CSMOPSO-1)去除了 Cubic-Sine 混合映射,而第二组实验(CSMOPSO-2)则没有采用 Centre 策略维护外部存档。这两组实验均保持了与原始 CSMOPSO 算法相同的参数配置,以便将 CSMOPSO 作为对照组进行比较,从而具体评估每个被排除策略的重要性。

算法的两个变体和 CSMOPSO 本身在 ZDT 和 UF 测试套件上各自独立运行 30 次,并计算了 IGD 和 HV 指标的统计平均值与标准差。若实验组的性能相较于对照组 CSMOPSO 出现明显下滑,则直接证明被排除的策略对提升算法性能至关重要。表 6 和表 7 详细列出了 IGD 和 HV 的比较结果,其中最优结果已被加粗标记。

分析表 6 和表 7 的数据后,可以清晰地得出:CSMOPSO 的变体算法(CSMOPSO-1 和 CSMOPSO-2)与完整的 CSMOPSO 在性能指标上存在显著差距。特别是在 HV 指标上,这两个变体在 15 个测试问题中仅



获得了 6 次最优表现,而集成了多种策略的 CSMOPSO 则展现出了更强的性能,在 IGD 和 HV 指标上分别取得了 10 次最佳成绩,凸显了其整体上的优越性。

研究结果显示:混沌初始化策略和锥域划分的外部存档策略对于提升 CSMOPSO 算法的性能而言,其涉及的活跃粒子发挥着至关重要的作用。通过深入对比实验组与对照组的表现,我们证实了所采纳策略的有效性,并明确了每种改进手段都能对原始的 MOPSO 算法产生正面效应。这一发现不仅强调了综合策略运用的必要性,也为未来多目标进化算法(MOEA)的设计指明了方向,提供了宝贵的见解。

表 6 CSMOPSO 及其变体的 IGD 值比较结果

Table 6 Comparison of IGD values of CSMOPSO and its variants

测试问题	CSMOPSO-1			CSMOPSO-2			CSMOPSO		
		IGD		IGD		IGD		IGD	
ZDT1	Mean	1.09e-02		1.09e-02		1.10e-02			
	Std	0.61e-04		0.26e-04		0.21e-04			
ZDT2	Mean	9.15e-03		9.21e-03		9.15e-03			
	Std	1.22e-04		0.16e-04		1.63e-04			
ZDT3	Mean	9.67e-03		9.91e-03		9.55e-03			
	Std	8.74e-04		1.24e-03		1.18e-03			
ZDT4	Mean	8.12e-01		7.46e-01		8.41e-01			
	Std	0.50e-01		0.38e-01		0.72e-01			
ZDT6	Mean	6.96e-03		6.93e-03		6.93e-03			
	Std	7.12e-05		5.55e-05		5.33e-05			
UF1	Mean	1.56e-01		1.50e-01		1.50e-01			
	Std	1.16e-02		1.07e-02		1.44e-02			
UF2	Mean	8.25e-02		8.28e-02		8.19e-02			
	Std	5.80e-03		4.51e-03		5.76e-03			
UF3	Mean	0.49e-01		0.41e-01		0.48e-01			
	Std	4.33e-02		0.52e-02		0.66e-02			
UF4	Mean	6.46e-02		6.63e-02		6.47e-02			
	Std	6.83e-03		8.31e-03		5.08e-03			
UF5	Mean	1.73e+00		1.67e+00		1.73e+00			
	Std	0.19e-01		1.83e-01		1.27e-01			
UF6	Mean	7.60e-01		7.17e-01		6.97e-01			
	Std	8.13e-02		7.82e-02		6.79e-02			
UF7	Mean	1.27e-01		1.27e-01		1.25e-01			
	Std	1.53e-02		1.58e-02		1.31e-02			
UF8	Mean	0.73e-01		0.87e-01		0.47e-01			
	Std	4.86e-02		6.80e-02		0.70e-02			
UF9	Mean	1.57e-01		1.64e-01		1.59e-01			
	Std	1.51e-02		1.71e-02		1.80e-02			
UF10	Mean	5.71e-01		6.77e-01		6.76e-01			
	Std	0.00e-01		0.19e-01		0.23e-01			
Best/all		10		9		10			

表 7 CSMOPSO 及其变体的 HV 值比较结果

Table 7 Comparison of HV values of CSMOPSO and its variants

测试问题	CSMOPSO-1			CSMOPSO-2			CSMOPSO		
		HV		HV		HV		HV	
ZDT1	Mean	7.12e-01		7.12e-01		7.12e-01			
	Std	0.94e-04		0.75e-04		0.08e-04			
ZDT2	Mean	4.40e-01		4.40e-01		4.40e-01			
	Std	1.83e-04		0.85e-04		0.43e-04			
ZDT3	Mean	5.96e-01		5.96e-01		5.96e-01			
	Std	4.19e-04		5.04e-04		4.50e-04			
ZDT4	Mean	1.54e-01		1.95e-01		1.43e-01			
	Std	1.59e-01		1.29e-01		1.18e-01			
ZDT6	Mean	0.85e-01		0.85e-01		0.85e-01			
	Std	0.18e-04		1.78e-04		1.42e-04			
UF1	Mean	4.79e-01		4.89e-01		4.89e-01			
	Std	1.69e-02		1.64e-02		0.17e-02			
UF2	Mean	6.26e-01		6.26e-01		6.27e-01			
	Std	5.57e-03		4.11e-03		5.90e-03			
UF3	Mean	0.05e-01		0.11e-01		0.03e-01			
	Std	0.78e-02		0.00e-02		0.23e-02			
UF4	Mean	0.54e-01		0.52e-01		0.54e-01			
	Std	8.96e-03		1.07e-02		6.79e-03			
UF5	Mean	0.00e+00		0.00e+00		0.00e+00			
	Std	0.00e+00		0.00e+00		0.00e+00			
UF6	Mean	1.83e-03		0.51e-03		0.61e-03			
	Std	0.38e-03		6.50e-03		6.23e-03			
UF7	Mean	0.96e-01		0.96e-01		0.98e-01			
	Std	0.07e-02		0.01e-02		1.88e-02			
UF8	Mean	0.28e-01		0.23e-01		0.24e-01			
	Std	1.84e-02		1.38e-02		1.50e-02			
UF9	Mean	5.95e-01		5.85e-01		5.91e-01			
	Std	0.00e-02		0.14e-02		0.07e-02			
UF10	Mean	1.53e-01		1.27e-01		1.32e-01			
	Std	7.64e-02		5.69e-02		7.13e-02			
Best/all		6		6		10			

3.6.2 Cubic-Sine 混合映射及其变体的消融实验

为了严谨地确认 Cubic-Sine 混合映射在 CSMOPSO 算法中初始化种群位置的有效性,我们策划了一系列消融实验,目标是衡量该策略中各个组成部分对算法性能的单贡献。这些实验通过逐一移除策略中的成分来系统地探究其对整体性能提升的影响。

本研究中设计了两组实验来分别探究 CSMOPSO 算法中 Sine 和 Cubic 混沌映射初始化种群策略的重要性。第一组组仅采用 Sine 混沌映射,而第二组仅采用 Cubic 混沌映射。这两组实验保持与 Cubic-Sine 混合映射相同的参数,以便将 CSMOPSO 作为对照组,准确评估每种策略的作用。在 ZDT 和 UF 测试套件上,我们分别独立运行了算法的两种映射(Sine 和 Cubic)以及

Cubic_Sine 混合映射各 30 次,并计算了 IGD 和 HV 指标的平均值和标准差。如果实验组的性能相较于对照组 Cubic-Sine 混合映射有所下降,那么这就直接证明了被排除的策略对于提升算法性能是至关重要的。表 8 和表 9 详细展示了 IGD 和 HV 的比较结果,其中最佳结果已被突出显示。

表 8 Cubic-Sine 混合映射及 Sine 映射和 Cubic 映射的 IGD 值比较结果

Table 8 Comparison results of IGD values among Cubic-Sine hybrid mapping, Sine mapping, and Cubic mapping

测试问题		Sine 映射	Cubic 映射	Cubic-Sine 混合映射
		IGD	IGD	IGD
ZDT1	Mean	1.06e-02	1.08e-02	1.10e-02
	Std	4.37e-04	0.87e-04	0.21e-04
ZDT2	Mean	9.12e-03	9.15e-03	9.15e-03
	Std	1.96e-04	1.65e-04	1.63e-04
ZDT3	Mean	9.99e-03	9.49e-03	9.55e-03
	Std	1.41e-03	1.03e-03	1.18e-03
ZDT4	Mean	6.95e-01	1.33e+00	8.41e-01
	Std	0.30e-01	1.20e+00	.72e-01
ZDT6	Mean	6.93e-03	6.94e-03	6.93e-03
	Std	4.80e-05	7.02e-05	5.33e-05
UF1	Mean	1.60e-01	1.51e-01	1.50e-01
	Std	1.39e-02	1.14e-02	1.44e-02
UF2	Mean	7.87e-02	8.03e-02	8.19e-02
	Std	0.43e-03	4.41e-03	5.76e-03
UF3	Mean	0.51e-01	0.43e-01	0.48e-01
	Std	0.82e-02	0.72e-02	0.66e-02
UF4	Mean	5.61e-02	6.16e-02	6.47e-02
	Std	1.24e-03	0.89e-03	5.08e-03
UF5	Mean	1.67e+00	1.69e+00	1.73e+00
	Std	0.44e-01	1.98e-01	1.27e-01
UF6	Mean	7.28e-01	7.27e-01	6.97e-01
	Std	8.40e-02	6.91e-02	6.79e-02
UF7	Mean	1.32e-01	1.15e-01	1.25e-01
	Std	1.47e-02	1.35e-02	1.31e-02
UF8	Mean	0.60e-01	0.30e-01	0.47e-01
	Std	4.77e-02	6.78e-02	0.70e-02
UF9	Mean	1.61e-01	0.28e-01	1.59e-01
	Std	0.01e-02	0.46e-02	1.80e-02
UF10	Mean	6.48e-01	5.35e-01	6.76e-01
	Std	1.52e-01	1.75e-01	0.23e-01
Best/all		10	8	10

通过深入分析表 8 和表 9 的数据可以明确看出, CSMOPSO 的变体算法(即 Sine 混沌映射和 Cubic 混沌映射)与 Cubic_Sine 混合映射在性能指标上存在显著差异。特别是在 HV 指标上,Sine 映射在 15 个测试问题中有 7 次表现最优,而 Cubic 映射仅有 5 次。虽然 IGD 指标上三者差异不大,但鉴于单一的混沌映射初始化种群方法已被广泛使用,我们提出了将这两种映射混合起来进行初始化种群的新方法。这种方法不仅新颖,而且效果显著,使得 Cubic-Sine 混合映射在 IGD 和 HV 指标上分别取得了 10 次最佳成绩,进一步凸显了其在整体性能上的优越性。

表 9 Cubic-Sine 混合映射及 Sine 映射和 Cubic 映射的 HV 值比较结果

Table 9 Comparison results of HV values among Cubic-Sine hybrid mapping, Sine mapping and Cubic mapping

测试问题		Sine 映射	Cubic 映射	Cubic-Sine 混合映射
		HV	HV	HV
ZDT1	Mean	7.12e-01	7.12e-01	7.12e-01
	Std	4.71e-04	4.17e-04	0.08e-04
ZDT2	Mean	4.40e-01	4.40e-01	4.40e-01
	Std	0.85e-04	0.40e-04	0.43e-04
ZDT3	Mean	5.95e-01	5.96e-01	5.96e-01
	Std	0.89e-04	4.33e-04	4.50e-04
ZDT4	Mean	0.16e-01	8.95e-02	1.43e-01
	Std	1.57e-01	8.89e-02	1.18e-01
ZDT6	Mean	0.85e-01	0.85e-01	0.85e-01
	Std	1.74e-04	1.66e-04	1.42e-04
UF1	Mean	4.76e-01	4.89e-01	4.89e-01
	Std	0.07e-02	1.58e-02	0.17e-02
UF2	Mean	6.32e-01	6.28e-01	6.27e-01
	Std	0.54e-03	4.66e-03	5.90e-03
UF3	Mean	0.99e-01	0.07e-01	0.03e-01
	Std	0.46e-02	0.09e-02	0.23e-02
UF4	Mean	0.64e-01	1.58e-01	0.54e-01
	Std	0.41e-03	0.37e-03	6.79e-03
UF5	Mean	0.00e+00	0.00e+00	0.00e+00
	Std	0.00e+00	0.00e+00	0.00e+00
UF6	Mean	1.38e-03	1.86e-03	0.61e-03
	Std	0.34e-03	0.63e-03	6.23e-03
UF7	Mean	0.87e-01	4.12e-01	0.98e-01
	Std	0.01e-02	1.90e-02	1.88e-02
UF8	Mean	0.36e-01	0.86e-01	0.24e-01
	Std	1.27e-02	5.54e-02	1.50e-02
UF9	Mean	5.95e-01	0.74e-01	5.91e-01
	Std	1.88e-02	0.87e-02	0.07e-02
UF10	Mean	9.97e-02	1.64e-01	1.32e-01
	Std	0.37e-02	8.15e-02	7.13e-02
Best/all		7	5	10

3.7 算法的灵敏度分析

本节旨在详细研究 Sine 混沌映射中的关键参数 r 如何影响 CSMOPSO 算法的性能。具体而言,参数 r 的值直接关联到算法的输出效率:较小的 r 值会限制算法生成的解的数量,这有可能使算法过早陷入局部最优解,从而失去寻找全局最优解的机会;相反,如果 r 值设置得过大,则可能会使搜索过程变得过于冗长和低效。

在本文的研究中,特别关注了参数 $r = 0.9$ 对 CSMOPSO 算法性能的影响。具体来说,表 10 详尽展示了在处理 ZDT3、ZDT6、UF1、UF6、UF8、UF9 和 UF10 这 7 个测试函数时,CSMOPSO 算法在 r 取不同值(0.7、0.8、0.9、4.0、4.1)下的 IGD 性能指标。为了凸显最佳表现,表 10 中对每个测试函数的最低 IGD 值进行了加粗处理。每个测试函数都经过算法 30 次的独立运行,以确保所得结果的可靠性和稳健性。通过对表 10 数据的深入分析,在 r 设置为 0.9 时,CSMOPSO 算法在 7 个测试函数中有 4 个实现了最优的 IGD 值,这一成绩明显优于 r 取其他值时的表现($r=0.7、0.8、4.1$ 时各有一个最优 IGD 值,而 $r=4.0$ 时则未获得任何最优 IGD 值)。这一发现强有力地证明了在 $r=0.9$ 这一参数配置下,CSMOPSO 算法的性能达到了最优水平,进一步凸显了精确选择参数对于提升算法性能的重要性。

表 10 IGD 指标中参数 r 的灵敏度分析

Table 10 Sensitivity analysis of parameter r in IGD index

测试问题	$r=0.7$	$r=0.8$	$r=0.9$	$r=4.0$	$r=4.1$
ZDT3	9.76e-03	9.68e-03	9.55e-03	9.56e-03	9.81e-03
ZDT6	6.95e-03	6.93e-03	6.93e-03	6.94e-03	6.92e-03
UF1	1.51e-01	1.53e-01	1.50e-01	1.51e-01	1.53e-01
UF6	7.24e-01	7.25e-01	6.97e-01	7.07e-01	7.31e-01
UF8	0.63e-01	0.50e-01	0.47e-01	0.73e-01	0.97e-01
UF9	1.60e-01	1.57e-01	1.59e-01	1.60e-01	1.61e-01
UF10	5.82e-01	7.25e-01	6.76e-01	6.72e-01	5.92e-01
Best/all	1	1	4	0	1

4 结论

本文介绍了一种创新的多目标粒子群优化算法——CSMOPSO,该算法巧妙地融合了 Cubic-Sine 混合映射与 Centre 策略,这一结合不仅确保了算法的快速收敛,还显著提升了解决方案的多样性。通过将 Cubic 混沌映射与 Sine 混沌映射融入算法设计中,CSMOPSO 有效增强了种群的多样性,提升了全局搜索的效率,并优化了初始种群的质量。此外,算法还采用

了锥域对外部存档进行划分,以高效存储非支配解。当外部存档达到容量限制时,CSMOPSO 会运用 Centre 策略剔除表现较差的解,确保外部存档中保留的解均为高质量的非支配解,CSMOPSO 实现了算法收敛性与多样性的出色平衡。为了全面且深入地验证 CSMOPSO 算法的性能表现,本文采用了两个至关重要的性能指标:IGD 和 HV,并在 ZDT 与 UF 这两个广泛认可的测试集上,将 CSMOPSO 与另外 8 个算法进行了详尽的仿真对比实验。同时还引入了 Friedman 秩检验方法、算法消融实验以及对参数的灵敏度分析,以综合且客观地评估 CSMOPSO 算法与其他对比算法的整体性能。

实验结果显示:CSMOPSO 算法在综合性能方面展现出了显著的优势,它不仅能够确保解决方案的高收敛性,还呈现出了出色的多样性特征,因此,CSMOPSO 算法能够高效且有效地应对和解决绝大多数复杂的多目标优化问题。

参考文献(References):

- [1] CHEN Z, WU H, CHEN Y, et al. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm[J]. Applied Soft Computing, 2022, 116: 108192.
- [2] YUAN M, CHEN X, LIU G, et al. Coordinated allocation of water resources and wastewater emission permits based on multi-objective optimization model: From the perspective of conflict between equity and economic benefits[J]. Journal of Cleaner Production, 2022, 372: 133733.
- [3] LIN L, GUO J, LIU L. Multi-scene application of intelligent inspection robot based on computer vision in power plant[J]. Scientific Reports, 2024, 14: 10657.
- [4] NIE F, LIU M, ZHANG P. Multilevel thresholding with divergence measure and improved particle swarm optimization algorithm for crack image segmentation[J]. Scientific Reports, 2024, 14: 7642.
- [5] ACAMPORA G, SCHIATTARELLA R, VITIELLO A. Using quantum amplitude amplification in genetic algorithms[J]. Expert Systems with Applications, 2022, 209: 118203.
- [6] ZENG Z, ZHANG M, ZHANG H, et al. Improved differential evolution algorithm based on the sawtooth-linear population size adaptive method[J]. Information Sciences, 2022, 608: 1045-1071.
- [7] LIAN J, HUI G, MA L, et al. Parrot optimizer: Algorithm and applications to medical problems[J]. Computers in Biology and Medicine, 2024, 172: 108064.
- [8] ZHENG R, HUSSIEN A G, QADDOURA R, et al. A multi-strategy enhanced African vultures optimization algorithm for

- global optimization problems[J]. *Journal of Computational Design and Engineering*, 2023, 10(1): 329–356.
- [9] KENNEDY J, EBERHART R. Particle swarm optimization[C]// *Proceedings of the Proceedings of ICNN' 95-International Conference on Neural Networks*. Piscataway: IEEE Press, 1995: 1942–1948.
- [10] COELLO COELLO C A, LECHUGA M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]// *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*. Piscataway: IEEE Press, 2002: 1051–1056.
- [11] KHODADADI N, ABUALIGAH L, AL-TASHI Q, et al. Multi-objective chaos game optimization [J]. *Neural Computing and Applications*, 2023, 35(20): 14973–15004.
- [12] CHU H, YI J, YANG F. Chaos particle swarm optimization enhancement algorithm for UAV safe path planning[J]. *Applied Sciences*, 2022, 12(18): 8977.
- [13] XU F Q, ZOU D X, LI C, et al. An improved particle swarm algorithm introducing Circle mapping and sine-cosine factors [J]. *Journal of Computer Engineering & Applications*, 2015, 59(17): 80–90.
- [14] QU B, LI G, YAN L, et al. A grid-guided particle swarm optimizer for multimodal multi-objective problems[J]. *Applied Soft Computing*, 2022, 117: 108381.
- [15] ZHANG X, LIU Y, YANG J, et al. Handling multi-objective optimization problems with a comprehensive indicator and layered particle swarm optimizer[J]. *Mathematical Biosciences and Engineering*, 2023, 20(8): 14866–14898.
- [16] 陈飞, 刘衍民, 刘君, 等. 精英竞争和综合控制的多目标粒子群算法[J]. *重庆工商大学学报(自然科学版)*, 2024, 41(2): 74–85.
CHEN Fei, LIU Yan-min, LIU Jun, et al. Multi-objective particle swarm optimization with elite competition and comprehensive control[J]. *Journal of Chongqing Technology and Business University (Natural Science Edition)*, 2024, 41(2): 74–85.
- [17] LIU S, WANG H, YAO W, et al. Surrogate-assisted environmental selection for fast hypervolume-based many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2024, 28(1): 132–146.
- [18] 邹康格, 刘衍民. 基于自适应网格混合机制的多目标粒子群算法[J]. *重庆工商大学学报(自然科学版)*, 2022, 39(2): 14–23.
ZOU Kang-ge, LIU Yan-min. Multi-objective particle swarm optimization based on adaptive mesh mixing mechanism[J]. *Journal of Chongqing Technology and Business University (Natural Science Edition)*, 2022, 39(2): 14–23.
- [19] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results[J]. *Evolutionary Computation*, 2000, 8(2): 173–195.
- [20] ZHANG Q, ZHOU A, ZHAO S, et al. Multi-objective optimization test instances for the CEC 2009 special session and competition[R]. Colchester: University of Essex, 2008: 1–30.
- [21] LIN Q, LIU S, ZHU Q, et al. Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems[J]. *IEEE Transactions on Evolutionary Computation*, 2018, 22(1): 32–46.
- [22] DAI C, WANG Y, YE M. A new multi-objective particle swarm optimization algorithm based on decomposition [J]. *Information Sciences*, 2015, 325: 541–557.
- [23] RAQUEL C R, NAVAL P C Jr. An effective use of crowding distance in multiobjective particle swarm optimization [C]// *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*. New York: ACM, 2005: 257–264.
- [24] JIANG S, YANG S. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2017, 21(3): 329–346.
- [25] DEB K, JAIN H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577–601.
- [26] ASAFUDDOULA M, RAY T, SARKER R. A decomposition-based evolutionary algorithm for many objective optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2015, 19(3): 445–460.
- [27] HE C, CHENG R, YAZDANI D. Adaptive offspring generation for evolutionary large-scale multiobjective optimization[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022, 52(2): 786–798.
- [28] COELLO C A C, CORTÉS N C. Solving multiobjective optimization problems using an artificial immune system[J]. *Genetic Programming and Evolvable Machines*, 2005, 6(2): 163–190.
- [29] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257–271.
- [30] XIA X, GUI L, YU F, et al. Triple archives particle swarm optimization[J]. *IEEE Transactions on Cybernetics*, 2020, 50(12): 4862–4875.

责任编辑:陈 芳