

基于联合青睐概率的多目标粒子群优化算法研究

赵杉杉, 冯 敏*

重庆交通大学 数学与统计学院, 重庆

收稿日期: 2026年9月23日; 录用日期: 2026年10月16日; 发布日期: 2026年10月27日

摘 要

本文提出一种基于联合青睐概率精英竞争学习的多目标粒子群优化算法。该算法首先设计了一种联合青睐概率评价指标, 将个体的收敛性信息与分布性信息进行综合度量, 并根据当前种群的联合青睐概率选取粒子构成精英竞争组, 用于提供高质量的学习对象。其次, 构建了一种基于混合概率准则和参考向量关联的环境选择方案, 用于协同改进算法的收敛性和多样性。数值实验表明所提出的算法具有很好的收敛性和多样性, 在搜索效率方面表现良好, 尤其在超多目标优化问题中显示出良好的潜力。

关键词

多目标粒子群优化, 竞争学习, 联合青睐概率, 小生境保留策略, 多样性, 收敛性

A Multi-Objective Particle Swarm Optimization Algorithm Based on Total Preferable Probability

Shanshan Zhao, Min Feng*

School of Mathematics and Statistics, Chongqing Jiaotong University, Chongqing

Received: September 23, 2026; accepted: October 16, 2026; published: October 27, 2026

Abstract

This paper proposes a multi-objective particle swarm optimization algorithm based on elite competitive learning with total preferable probability. The proposed algorithm first develops a to-

*通讯作者。

tal preferable probability indicator to comprehensively evaluate the convergence and distribution characteristics of individuals. Based on the total preferable probabilities of the current population, a set of particles is selected to form an elite competition group, which provides high-quality learning exemplars for particle evolution. Furthermore, an environmental selection strategy combining total preferable probability criterion with reference-vector association is designed to achieve a coordinated improvement in convergence and diversity. Numerical experiments demonstrate that the proposed algorithm exhibits competitive convergence and diversity performance, together with favorable search efficiency, and shows particularly promising potential for solving many-objective optimization problems.

Keywords

Multi-Objective Particle Swarm Optimization, Competitive Swarm Optimizer, Total Preferable Probability, Niche Preservation Strategy, Diversity, Convergence

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

粒子群优化算法(PSO)是一类典型的群智能优化算法, 具有实现简单、参数较少、收敛速度较快等优点[1]。传统粒子群优化算法在单目标优化问题中主要依靠个体最优粒子和全局最优粒子引导种群搜索。因此, PSO 能够快速收敛的一个重要前提是个体最优粒子和全局最优粒子能够被明确确定。然而, 当 PSO 算法由单目标优化扩展到多目标优化领域后, 由于多个目标之间通常相互冲突, 不存在一个粒子能够在所有目标上同时达到最优, 导致传统个体最优和全局最优的区分变得困难。尤其是随着目标数量的增加, 种群中大量候选解趋于互不支配状态, 以及传统 Pareto 支配关系的区分能力减弱, 导致传统 PSO 更新公式中的学习对象不再明确。这使得 PSO 算法容易缺乏指向真实 Pareto 前沿的选择压力, 从而陷入停滞或者收敛性与多样性的失衡。

针对多目标粒子群优化算法(MOPSO)中领导粒子难以确定的问题, 已有研究提出了多种改进策略。常用的方法主要有两类: 一类方法是利用外部档案保存非支配解, 并结合 Pareto 排序[2]、拥挤距离[3]或网格密度[4]等机制从档案中选择全局引导粒子; 另一类方法是分解思想引入 MOPSO, 通过权重向量或参考向量将多目标问题转化为多个标量子问题[5], 从而在不同搜索方向上确定局部最优或全局最优的引导粒子。此外, 也有研究借助超体积指标、基于效用函数的 R2 指标等评价指标衡量粒子对解集质量的贡献, 用于辅助选择学习对象[6]-[8]。这些方法在一定程度上缓解了传统 PSO 在多目标场景下局部最优和全局最优粒子难以确定的问题。但在目标数量较多时, 这类方法仍然容易受到非支配解比例升高、选择压力不足以及收敛性和多样性难以平衡等问题的影响。

近年来, 竞争式粒子群优化算法(CSO)为解决上述问题提供了新的思路。与传统 PSO 直接依赖个体最优和全局最优粒子不同, CSO 通常通过粒子之间的竞争关系确定学习对象, 使竞争失败粒子向竞争获胜粒子学习[9]。已有研究表明, 竞争机制可以有效改善多目标粒子群优化算法的搜索能力。例如, CMOPSO [10]利用竞争机制和角度信息选择学习对象, 以增强算法在多目标问题上的收敛性和多样性; LMOCSO [9]将竞争机制扩展到大规模多目标优化问题, 通过两阶段更新策略提升搜索效率; 一些后续研究还从多种搜索策略、邻域搜索、自适应参数和多阶段竞争等方面对 CSO 进行了进一步改进[11]-[14]。

然而, 现有竞争类 MOPSO 在处理超多目标优化问题时仍然存在不足。首先, 在高维目标空间中, 大量粒子互不支配, 传统基于支配关系或随机配对的竞争方式难以准确判断粒子的综合优劣, 还可能导致质量较低的粒子被误选为学习对象。其次, 竞争机制的性能依赖于获胜粒子的评价和选择方式, 如果学习对象选择不稳定, 粒子更新方向也会变得不稳定, 从而削弱种群向真实 Pareto 前沿收敛的能力。此外, 在环境选择过程中, 若仅依靠非支配排序、垂直距离或随机小生境选择, 这可能会导致优秀个体被误删。因此, 如何构建一种能够合理评价粒子收敛性和分布性综合质量的标准, 并据此选择具有较强引导能力的学习对象, 是竞争类 MOPSO 算法需要解决的关键问题。

从本质上看, 竞争粒子或领导粒子的选择可以被理解作为一种多指标综合评价问题。一个优秀的学习对象不仅应在多个目标函数值上具有较好的收敛潜力, 还应能够为种群提供有利于维持分布性的搜索方向。概率基多目标优化[15]为这类综合评价问题提供了可借鉴的思想。该方法从系统论和概率论角度出发, 将多目标优化理解为多个目标事件的同时优化, 并通过偏青睐概率刻画候选对象在单个目标上的相对优势, 再以多个偏青睐概率的联合形式形成联合青睐概率。不过, 已有概率基多目标优化研究多用于材料加工、机械设计和化工机械正交试验等问题的固定候选方案或工艺参数组合的综合评价[16] [17], 尚未充分考虑其在竞争类粒子群优化中用于动态学习对象选择的作用。基于此, 本文将该思想引入竞争类 MOPSO 算法中作为候选方案综合优劣比较的标量评价依据。

综合上述分析, 本文利用概率基多目标优化中的总青睐概率评价粒子在多个目标上同时具有优势的可能性, 并进一步结合邻近距离构造联合青睐概率, 以同时反映粒子的收敛性和分布性。在此基础上, 本文提出一种基于联合青睐概率精英竞争学习的改进多目标粒子群优化算法 TP-CMOPSO。该算法首先依据联合青睐概率构建精英竞争组, 并在环境选择阶段融入联合青睐概率, 旨在协同增强算法在超多目标优化问题中的收敛压力的同时, 有效维持解集的多样性。本文的主要贡献如下。

a) 提出一种基于联合青睐概率的精英竞争组构建策略。从当前种群中选取联合青睐概率最高的前 10% 个体作为精英粒子集, 为粒子更新提供高质量学习对象, 该策略具有良好的鲁棒性, 且性能不受目标数量和范围的影响。

b) 设计一种基于角度最小准则的全粒子精英学习策略。对于种群中每个粒子, 从精英竞争组中随机选择两个候选精英, 并选取与当前粒子夹角更小的个体作为粒子学习对象, 以增强搜索方向的一致性, 以及提高粒子更新的稳定性。

c) 构建一种基于参考向量与联合青睐概率的小生境环境选择策略。该策略在非支配排序和参考向量关联的基础上, 优先填充关联数量较少的参考向量, 并在同一参考向量存在多个候选个体时选择联合青睐概率更优的解, 从而减少随机环境选择造成的高质量的解丢失, 以及提升外部档案的质量和解集的均匀性。

2. 相关工作

2.1. 多目标优化

本文研究的超多目标优化问题可视为目标数量较多的多目标优化问题[18], 其数学模型可表示为

$$\begin{aligned} \min f(\mathbf{x}) &= [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \\ \text{s.t. } &\begin{cases} g_i(\mathbf{x}) \leq 0, & i = 1, 2, \dots, p, \\ h_j(\mathbf{x}) = 0, & j = 1, 2, \dots, q, \end{cases} \end{aligned} \quad (1)$$

其中, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 是决策变量; $f_i(\mathbf{x})$ 表示第 i 个目标函数; m 表示目标数量, 在超多目标优化

问题中通常有 $m > 3$; $g_i(\mathbf{x})$ 和 $h_j(\mathbf{x})$ 分别表示不等式约束和等式约束, p 和 q 分别表示不等式约束和等式约束的数量。

多目标优化问题通常无法得到唯一的全局最优解, 而是得到一组相互折中的非支配解。对于两个可行解 $\mathbf{x}_a$ 和 $\mathbf{x}_b$, 若 $\mathbf{x}_a$ 在所有目标上都不劣于 $\mathbf{x}_b$, 且至少在一个目标上优于 $\mathbf{x}_b$, 则称 $\mathbf{x}_a$ 支配 $\mathbf{x}_b$, 记为 $\mathbf{x}_a \prec \mathbf{x}_b$, 其表现为

$$\begin{cases} f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b), & i=1,2,\dots,m, \\ \exists j \in \{1,2,\dots,m\}, & f_j(\mathbf{x}_a) < f_j(\mathbf{x}_b). \end{cases} \quad (2)$$

若不存在其他可行解支配 $\mathbf{x}^*$, 则称 $\mathbf{x}^*$ 为一个 Pareto 最优解。所有 Pareto 最优解在决策空间中构成 Pareto 最优解集, 其在目标空间中的映像称为 Pareto 前沿。

多目标优化问题的算法其性能一般同时关注两个性质: 一是具有较好的收敛性, 即解集应尽可能接近真实 Pareto 前沿; 二是具有较好的分布性, 即解集应尽可能均匀覆盖 Pareto 前沿。

2.2. 多目标竞争粒子群优化算法

粒子群优化算法是一种模拟鸟群觅食行为的元启发式群智能优化算法。在传统的单目标优化问题中, 粒子通过结合自身的飞行经验与整个种群的共享经验来动态调整自身的飞行速度与位置[19]。多目标粒子群优化算法是在传统粒子群优化算法基础上发展而来的。随着目标数量增加, 目标空间中大量解呈互不支配状态, 传统 Pareto 支配难以有效区分粒子优劣, 个体最优和全局最优的选择导向逐渐减弱。因此, 现有 MOPSO 通常结合分解策略、改进排序、评价指标或竞争策略等机制筛选学习样本, 以增强高维目标空间中的选择压力。

竞争机制粒子群优化算法是多目标粒子群优化算法中的一种典型改进形式。该类算法通过粒子间的成对竞争确定学习关系。竞争中综合表现较好的粒子作为获胜粒子, 表现较差的粒子作为失败粒子, 失败粒子向获胜粒子学习并完成速度与位置更新。此时待更新粒子 i 的速度和位置更新公式可表示为

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + cr(x_{win,j}(t) - x_{i,j}(t)), \quad (3)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1)\Delta t, \quad (4)$$

其中, $x_{i,j}(t)$ 表示第 t 次迭代时粒子 i 在第 j 维上的位置; $v_{i,j}(t)$ 表示粒子 i 在第 j 维上单位离散迭代间隔内的位置变化量, 即粒子位置相对于迭代进程的变化率; ω 为惯性权重, c 为学习因子, r 为 $[0,1]$ 内的随机数; $x_{win,j}(t)$ 表示竞争获胜粒子在第 j 维上的位置; Δt 表示相邻两次迭代之间的离散迭代间隔, 即 $\Delta t=1$ 。该机制弱化了传统 PSO 对个体最优和全局最优的依赖, 能够在一定程度上缓解高维目标下学习对象选择压力不足的问题。

3. 基于联合青睐概率的多目标粒子群优化算法

本文基于联合青睐概率能够从概率角度综合刻画个体多目标优化潜力的特点, 提出一种基于联合青睐概率的精英竞争多目标粒子群优化算法, 主要包括联合青睐概率评价、精英竞争学习、环境选择三个部分。

3.1. 联合青睐概率评价方法

在超多目标优化问题中, 由于目标数量非常大, 导致大量个体粒子之间呈现出互不支配关系, 从而仅依靠 Pareto 支配关系难以形成有效的选择压力。联合青睐概率源于概率基多目标优化思想, 其基本思想是将“多目标同时优化”理解为多个目标偏好事件的联合发生, 通过概率计算为每个候选解计算出相

对应的综合评价值, 这种粒子评价方法可以有效增强对候选解优劣的区分能力。具体流程是先计算粒子在每个目标上的偏青睐概率, 再将多个目标上的偏青睐概率进行联合, 从而得到粒子在多个目标上同时保持较优表现的概率评价值。

设当前种群规模为 N , 目标空间维数为 m , X_{ij} 表示第 i 个粒子在第 j 个目标上的函数值, $X_{j\max}$ 和 $X_{j\min}$ 分别表示当前种群在第 j 个目标上的最大值和最小值。若第 j 个目标为效益型指标, 即数值越大越好, 则偏青睐概率可定义为

$$P_{ij} = \frac{X_{ij}}{\sum_{k=1}^n X_{kj}}; \quad (5)$$

若第 j 个目标为成本型指标, 即数值越小越好, 根据概率基多目标优化中的偏青睐概率定义[15], 成本型指标首先通过 $X_{j\max} + X_{j\min} - X_{ij}$ 对原始目标值进行反向线性映射, 使较小的目标值对应较大的青睐程度; 随后对所有候选个体的映射值进行归一化, 得到偏青睐概率:

$$P_{ij} = \frac{X_{j\max} + X_{j\min} - X_{ij}}{n(X_{j\max} + X_{j\min}) - \sum_{k=1}^n X_{kj}}. \quad (6)$$

其中, 分母 $n(X_{j\max} + X_{j\min}) - \sum_{k=1}^n X_{kj}$ 等价于当前种群所有个体反向映射值之和, 用于保证同一目标下各候选个体的偏青睐概率之和为 1。该定义沿用了概率基多目标优化方法中成本性指标的处理方式, 其理论基础及具体推导见文献[15]。

在获得各目标的偏青睐概率后, 粒子 i 的总青睐概率定义为

$$P_i = \prod_{j=1}^m P_{ij}. \quad (7)$$

P_i 越大表示粒子 i 在多个目标上同时保持较优表现的可能性越高, 因而 P_i 可作为精英竞争组构建中的收敛性评价依据。为进一步在环境选择中兼顾解集分布性, 本文计算粒子在目标空间中的 k -近邻距离 D_i , 并令 $k = \lfloor \sqrt{N} \rfloor$ 。先计算粒子 i 与粒子 j 在目标空间中的欧氏距离:

$$d_{ij} = \sqrt{\sum_{\ell=1}^m (X_{i\ell} - X_{j\ell})^2}, \quad i \neq j, \quad (8)$$

将粒子 i 与其余 $N-1$ 个粒子之间的距离按照升序排列, 得到

$$d_i^{(1)} \leq d_i^{(2)} \leq \dots \leq d_i^{(N-1)}, \quad (9)$$

则粒子 i 的 k -近邻距离可表示为

$$D_i = d_i^{(k)}, \quad (10)$$

其中, $d_i^{(k)}$ 表示粒子 i 与其他粒子之间第 k 小的欧氏距离。 D_i 越大表示该粒子所在区域越稀疏, 可用于反映解集的分布性。将总青睐概率 P_i 与 k -近邻距离 D_i 均视为效益型指标进行二次融合, 得到联合青睐概率

$$TP_i = \frac{P_i}{\sum_{j=1}^N P_j} * \frac{D_i}{\sum_{j=1}^N D_j}, \quad (11)$$

其中, TP_i 同时包含收敛性和分布性信息。本文在后续粒子学习阶段采用 TP_i 筛选精英个体, 在环境选择

关键层采用 TP_i 辅助筛选候选解。

式(11)采用乘法方式融合归一化后的总青睐概率与 k -近邻距离, 与线性加法模型相比, 乘法聚合能够降低不同指标之间的补偿效应, 当粒子在收敛性或分布性中的任一方面表现较差时, 其综合评价值均会受到明显抑制, 因此更强调两类指标的同时满足。为进一步验证该聚合方式的合理性, 本文构造等权加法模型作为对照, 并在实验部分对两种聚合方式进行比较。

$$TP_i^{add} = 0.5P_i + 0.5D_i, \quad (12)$$

Beyer [20]和 Aggarwal 等[21]指出, 随着空间维数增加, 样本间欧氏距离的相对差异会逐渐减小, 使近邻关系的区分变得困难, 即出现距离集中现象。因此, 本文采用 k -近邻欧氏距离在高维目标空间中也存在局部分布判别能力下降的潜在局限。尽管如此, 从本文 3~15 目标测试结果来看, TP-CMOPSO 在多数 DTLZ 和 WFG 测试实例上仍取得了较好的 IGD 和 HV 结果, 表明该距离度量在当前实验范围内仍能够提供有效的分布性信息, 其高维距离集中效应尚未对算法整体性能造成明显影响。

3.2. 基于联合青睐概率构建的精英竞争组

传统竞争粒子群算法通常通过随机确定成对竞争的获胜粒子和失败粒子, 并令失败粒子向获胜粒子学习。该方式能够弱化对全局最优粒子的依赖。但在超多目标优化中, 随机竞争可能导致学习对象质量不稳定, 进而削弱粒子向真实 Pareto 前沿收敛的能力。为此, 本文根据联合青睐概率构建精英竞争组, 使粒子优先从综合表现较好的个体中选择学习对象。

具体而言, 在每一代进化过程中, 首先根据当前种群的目标函数值计算所有粒子的联合青睐概率 TP_i , 并按 TP_i 从大到小排序, 然后选取排序靠前的 10%粒子组成精英竞争组 E 。精英竞争组中的粒子在多个目标上具有更高的联合优势, 能够为种群提供更稳定的学习方向。与直接从整个种群中随机选择学习对象相比, 该策略提高了学习样本的质量, 有助于增强算法的收敛性。

3.3. 面向全体粒子的精英竞争学习策略

为进一步提高种群整体搜索活性, 本文在粒子更新阶段不再仅更新竞争失败粒子, 而是使种群中所有粒子均参与学习。对于任意待更新粒子 $\mathbf{x}_i$, 算法从精英竞争组 E 中随机选择两个候选精英粒子 $\mathbf{x}_a$ 和 $\mathbf{x}_b$, 并分别计算其与当前粒子在目标空间中的夹角:

$$\theta_{i,a} = \arccos(F_i^T F_a), \quad (13)$$

$$\theta_{i,b} = \arccos(F_i^T F_b). \quad (14)$$

其中, $F_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$, $F_a = \frac{\mathbf{x}_a}{\|\mathbf{x}_a\|}$, $F_b = \frac{\mathbf{x}_b}{\|\mathbf{x}_b\|}$ 。算法选择夹角较小的候选精英作为当前粒子的学习对象, 即:

$$\hat{\mathbf{x}}_i = \begin{cases} \mathbf{x}_a, & \theta_{i,a} \leq \theta_{i,b}, \\ \mathbf{x}_b, & \theta_{i,a} > \theta_{i,b}. \end{cases} \quad (15)$$

该策略使粒子优先向目标空间中方向更接近的精英个体学习, 在保持搜索方向一致性的同时减少盲目学习带来的扰动。设最终选定的学习对象为 $\hat{\mathbf{x}}_i$, 则粒子的速度和位置更新方式为

$$\mathbf{v}_{i,j}(t+1) = \omega \mathbf{v}_{i,j}(t) + cr(\hat{\mathbf{x}}_{i,j} - \mathbf{x}_{i,j}(t)), \quad (16)$$

$$\mathbf{x}_{i,j}(t+1) = \mathbf{x}_{i,j}(t) + \mathbf{v}_{i,j}(t+1)\Delta t, \quad (17)$$

其中, ω 为惯性权重, c 为学习因子, r 为 $[0,1]$ 内的随机数; $\hat{\mathbf{x}}_{i,j}$ 表示粒子 $\mathbf{x}_i$ 的学习对象在第 j 维上的位

置, Δt 表示相邻两次迭代之间的离散迭代间隔。更新完成后, 对粒子位置进行边界处理, 并采用多项式变异增强局部搜索能力, 从而降低算法陷入局部最优的风险。

3.4. 基于联合青睐概率的环境选择策略

完成粒子更新和变异后, 将父代种群与新生成的子代种群合并, 并对合并种群执行非支配排序。在构造下一代种群时, 按照非支配层编号从小到大的顺序依次选择粒子。若将当前非支配层中的全部粒子加入后, 下一代种群规模仍未达到预设规模, 则完整保留该层, 并继续考察下一非支配层。当某一非支配层中的全部粒子加入后会使得下一代种群规模超过预设规模时, 将该层定义为关键层, 记为第 k 层 F_k 。此时, 第 k 层之前的所有非支配层均被完整保留, 而关键层中的粒子需要进一步执行基于参考向量的小生境选择策略, 以筛选出部分粒子填补下一代种群中的剩余位置。

在参考向量关联过程中, 首先根据理想点对目标函数值进行平移, 并利用极端点构造超平面对目标空间进行归一化, 以减弱不同目标量纲差异对选择结果的影响; 然后计算每个候选粒子到各参考向量的垂直距离, 并将其关联到距离最近的参考向量。其中, 参考向量集合用 V 表示, $V = \{Z_1, Z_2, \dots, Z_N\}$ 。

粒子 x_i 到参考向量 Z_j 的垂直距离可表示为

$$d_{\perp}(x_i, Z_j) = \|x_i\| \sqrt{1 - \left(\frac{F_i^T Z_j}{\|Z_j\|} \right)^2}, \quad (18)$$

其中, $F_i = x_i / \|x_i\|$ 。对于关键层中的候选粒子, 若某一参考向量尚未关联任何已选个体, 则优先选择与该参考向量垂直距离最近的粒子, 以保证解集覆盖更多区域; 若该参考向量已有关联个体, 则从对应候选粒子中选择联合青睐概率 TP_i 最大的个体, 以在维持分布性的基础上进一步提高收敛质量。

如图 1 所示, 对于尚未关联已选个体的参考向量, 优先保留垂直距离最小的候选粒子; 对于已有个体关联的参考向量, 则优先保留联合青睐概率较大的候选粒子, 从而兼顾参考方向覆盖和候选解质量。

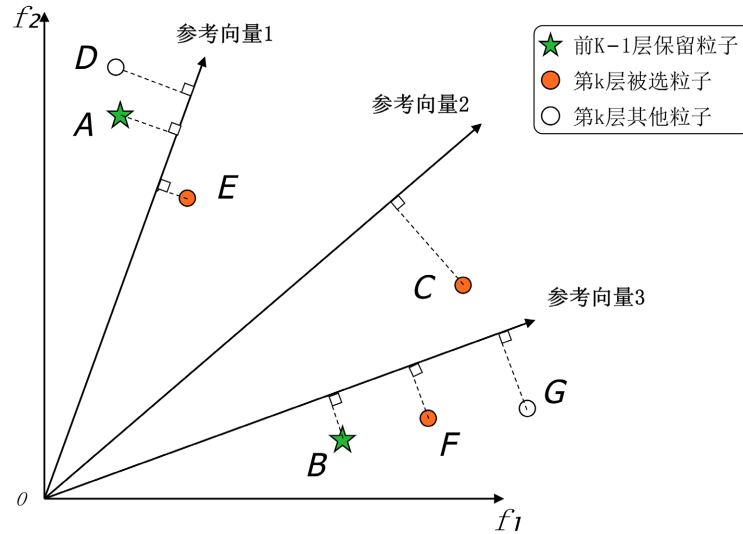


Figure 1. Niche-based environmental selection strategy
图 1. 小生境环境选择策略

3.5. TP-CMOPSO 算法流程

TP-CMOPSO 的算法步骤如下.

算法 1. TP-CMOPSO 算法的基本流程

输入: 种群规模 N , 参考向量 V , 最大迭代次数 $t_{\max}$.

输出: 外部档案 A .

1 初始化种群 P_0 , 生成参考向量 V , 计算理想点 Z^* .

2 从初始种群 P_0 中选择非支配解, 构成外部档案 A_0 .

3 $t = 0$.

4 **While** $t < t_{\max}$ **do**.

/* 计算联合青睐概率并构成精英竞争组 */.

5 计算种群 P_t 中所有粒子的联合青睐概率 TP_i :

(a) 基于目标值计算偏青睐概率和总青睐概率; 计算 k -近邻距离;

(b) 融合得到联合青睐概率 TP_i .

6 依据 TP_i 排序结果选择前 10% 个体作为精英竞争组 E .

/* 粒子更新 */.

For $i = 1$ **to** N **do**.

7 从 E 中随机选择两个候选精英粒子 x_a 和 x_b .

8 分别计算 x_a 、 x_b 与当前粒子 x_i 在目标空间中的夹角.

9 选择夹角较小的候选精英粒子 $\hat{x}_i$ 作为当前粒子的学习对象.

10 更新粒子速度和位置.

End For.

11 对更新后的粒子执行边界处理和多项式变异, 得到子代种群 Q_t .

/* 环境选择策略 */.

12 合并父代种群、子代种群和外部档案: $R_t = P_t \cup Q_t \cup A_t$. 并对 R_t 进行非支配排序.

13 采用基于参考向量和联合青睐概率的小生境选择策略.

14 得到更新后的外部档案 A_{t+1} .

15 更新理想点 Z^* .

16 $t = t + 1$.

End While.

Return A .

3.6. 计算复杂度分析

TP-CMOPSO 的计算复杂度主要与联合青睐概率计算、精英竞争学习和环境选择有关。设种群规模为 N , 目标维数为 M , 决策变量维数为 D 。在联合青睐概率计算阶段, 计算各粒子在所有目标上的偏青睐概率和总青睐概率需要的计算量 $O(MN)$; 为了获得 k -近邻距离, 需要计算种群个体之间的两两欧氏距离, 其时间复杂度为 $O(MN^2)$, 随后对每个粒子的邻域距离进行排序, 其时间复杂度为 $O(N^2 \log N)$ 。在精英竞争学习阶段, 根据联合青睐概率对粒子进行排序需要 $O(N \log N)$ 的计算量, 对所有粒子进行候选精英夹角计算需要 $O(MN)$, 粒子的速度、位置更新及变异操作需要 $O(ND)$ 。在环境选择阶段, 非支配排序以及参考向量关联和小生境选择的最坏时间复杂度均为 $O(MN^2)$ 。因此, TP-CMOPSO 单代进化的最坏时间复杂度可表示为

$$O(MN^2 + N^2 \log N + ND) \quad (19)$$

可以看出, 算法的主要计算开销来自粒子间距离计算、邻域距离排序以及环境选择过程。实际运行时间进一步验证了上述分析。在 10 目标和 15 目标测试中, TP-CMOPSO 的平均运行时间分别为 3.9349 s 和 4.0500 s, 低于 CMOPSO 和 MOPSOCD, 但高于 MMOPSO 和 SMPSO。随着目标数及种群规模增加, 其运行时间增幅较小, 说明 TP-CMOPSO 在引入精英竞争学习和多样性保持机制的同时, 仍具有可接受的实际运行效率。

4. 实验设计

4.1. 实验环境与参数设置

实验在配 Intel Core i5-10210U CPU、16GB 内存和 Microsoft Windows 11 操作系统的环境下执行。选择基于竞争机制的多目标粒子群优化(CMOPSO)算法[10]、多种搜索策略的多目标粒子群优化(MMOPSO)算法[12]、基于拥挤距离的多目标粒子群优化(MOPSOCD)算法[3]以及速度受限的多目标粒子群优化(SMPSO)算法[22]与本文 TP-CMOPSO 算法在相同环境下进行性能对比, 相关参数按照原始论文的建议值设置, 如表 1 所示。对比过程在 MATLAB R2024b 中用 PlatEMO-3.5 实现[23], 为保证实验公平性, 最大评价次数统一设置为 20000, 每个算法在每个测试问题中均独立运行 30 次。

Table 1. Parameter settings of the algorithms

表 1. 各算法参数设置

算法	参数设定
CMOPSO	$r_1, r_2 \in [0, 1], \gamma = 10, \eta_m = 20$
MMOPSO	$w \in [0.1, 0.5], c_1, c_2 \in [1.5, 2], p_c = 0.9, p_m = 1/n, \eta_c = \eta_m = 20$
MOPSOCD	$w \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], p_c = 0.9, p_m = 1/n, \eta_c = \eta_m = 20$
SMPSO	$w \in [0.1, 0.5], c_1, c_2 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$
TP-CMOPSO	$w \in [0.1, 0.5], c_1 \in [1.5, 2.5], p_m = 1/n, \eta_m = 20$

4.2. 测试问题和性能指标

为了评估 TP-CMOPSO 算法的有效性, 本文选取 DTLZ [24]和 WFG [25]系列基准测试函数进行实验。DTLZ 系列测试问题具有线性、球面、退化或断开等不同 Pareto 前沿结构, 适合用于考察算法的收敛性和分布性。WFG 系列测试问题包含偏置、多峰、欺骗性、不可分性以及退化、断开等复杂特征, 能够更全面地检验算法在复杂优化场景下的综合性能。本文选取反世代距离(Inverted Generational Distance, IGD) [26]和超体积指标(Hypervolume, HV) [27]作为评价指标, 其中 IGD 越小、HV 越大, 表明算法在测试问题上具有更优的综合性能。考虑到高维 HV 计算开销太大, 本文在 M= 3, 5, 8, 10, 15 时同时统计 IGD 和 HV, 在 M= 20、25 时主要采用 IGD 进行评价。

5. 实验结果与分析

对于每个测试问题, 算法独立运行 30 次, 最大评估次数设置为 20000, 记录评价指标值的均值和标准差, 并通过显著性水平为 0.05 的 Wilcoxon 秩和检验对实验进行统计分析。Wilcoxon 检验结果用符号“+”、“-”和“=”分别表示对比算法的性能显著更好、显著更差以及与 TP-CMOPSO 算法性能相当。

5.1. 综合实验结果与分析

表 2 和表 3 给出了 TP-CMOPSO 与四种对比算法在 DTLZ 和 WFG 测试集上的 IGD 结果。总体来看, 在 48 组有效 DTLZ 测试实例中, TP-CMOPSO 取得 31 个最优 IGD 结果, 尤其在 DTLZ1、DTLZ2、DTLZ3 和 DTLZ5 的多数高目标实例上表现较好, 其中 DTLZ2 在 3~25 目标下均取得最优 IGD 值。相比之下, 算法在 DTLZ6 和 DTLZ7 上的优势相对有限。WFG 结果表明, TP-CMOPSO 在 WFG2、WFG3 及部分 WFG6 实例上具有较好的性能, 而在 WFG1 及 20、25 目标下的部分复杂 WFG 问题上优势减弱。结合表 4 中 3~15 目标下的 HV 结果可以看出, 本文算法在规则或部分退化 Pareto 前沿问题上具有较好的综合优化性能, 但在复杂、不规则及高度偏置前沿上仍存在一定局限。

Table 2. IGD values of the proposed algorithm and comparison algorithms on DTLZ1-DTLZ7

表 2. DTLZ1-DTLZ7 上所提出的算法和各比较算法的 IGD 值

Problem	M	CMOPSO	MMOPSO	MOPSOCD	SMPSO	TP-CMOPSO
DTLZ1	3	1.4181e-1 (3.87e-1) =	3.6910e-2 (4.96e-2) =	1.0141e+1 (3.72e+0) -	1.1972e+0 (2.83e+0) -	4.7102e-2 (7.92e-2)
	5	7.2398e+0 (3.72e+0) -	9.7951e-1 (1.75e+0) -	1.5059e+1 (4.29e+0) -	3.3535e+0 (4.09e+0) -	1.2964e-1 (1.21e-1)
	8	2.8405e+1 (1.04e+1) -	7.0712e+0 (5.12e+0) -	2.0522e+1 (2.28e+0) -	1.1437e+1 (8.89e+0) -	2.0025e-1 (1.68e-1)
	10	9.3097e+1 (2.80e+1) -	1.0284e+1 (5.71e+0) -	2.0284e+1 (3.48e+0) -	2.4641e+1 (9.66e+0) -	4.1748e-1 (6.02e-1)
	15	1.0640e+2 (1.88e+1) -	1.5399e+1 (7.11e+0) -	2.0486e+1 (2.75e+0) -	3.7850e+1 (7.66e+0) -	4.3473e-1 (6.79e-1)
	20	8.6955e+1 (1.78e+1) -	1.2549e+1 (7.58e+0) -	2.2833e+1 (3.46e+0) -	6.0947e+1 (1.36e+1) -	2.9143e+0 (5.37e+0)
	25	6.3413e+1 (1.49e+1) -	1.7811e+1 (6.23e+0) -	2.2469e+1 (3.19e+0) -	5.2186e+1 (1.12e+1) -	5.0490e+0 (6.38e+0)
DTLZ2	3	5.7645e-2 (1.05e-3) -	7.2628e-2 (2.98e-3) -	6.5890e-2 (2.20e-3) -	6.8207e-2 (2.17e-3) -	5.4539e-2 (2.12e-5)
	5	3.4902e-1 (2.75e-2) -	2.5056e-1 (9.25e-3) -	4.1535e-1 (1.85e-2) -	4.7061e-1 (3.85e-2) -	1.9541e-1 (2.57e-4)
	8	2.3056e+0 (2.99e-2) -	7.2640e-1 (5.33e-2) -	8.5292e-1 (4.26e-2) -	1.2143e+0 (2.39e-1) -	3.5927e-1 (2.19e-2)
	10	1.4399e+0 (3.92e-1) -	1.2859e+0 (1.64e-1) -	1.0614e+0 (7.44e-2) -	1.1759e+0 (8.93e-2) -	5.6908e-1 (4.53e-2)
	15	1.4039e+0 (1.38e-1) -	1.5764e+0 (1.55e-1) -	1.3002e+0 (8.93e-2) -	1.2630e+0 (7.20e-2) -	8.3099e-1 (4.61e-2)
	20	2.1365e+0 (1.42e-1) -	1.9947e+0 (2.91e-1) -	1.6523e+0 (1.04e-1) -	1.4543e+0 (7.94e-2) -	1.1026e+0 (5.32e-2)
	25	1.9424e+0 (1.22e-1) -	1.8003e+0 (2.14e-1) -	1.6428e+0 (1.23e-1) -	1.4827e+0 (7.41e-2) -	1.1207e+0 (5.12e-2)
DTLZ3	3	1.3010e+2 (3.14e+1) -	7.7241e+1 (2.87e+1) =	1.1176e+2 (3.93e+1) -	3.1456e+1 (4.95e+1) +	6.7371e+1 (2.90e+1)
	5	2.1298e+2 (2.95e+1) -	1.5191e+2 (3.89e+1) -	1.6662e+2 (2.89e+1) -	1.5546e+2 (5.17e+1) -	8.0095e+1 (2.96e+1)
	8	4.3938e+2 (7.96e+1) -	1.9839e+2 (2.57e+1) -	1.9392e+2 (2.26e+1) -	4.7565e+2 (7.60e+1) -	9.8589e+1 (3.34e+1)
	10	4.9282e+2 (9.44e+1) -	2.2007e+2 (1.62e+1) -	2.0981e+2 (1.05e+1) -	5.3979e+2 (9.52e+1) -	1.1806e+2 (3.26e+1)
	15	6.4361e+2 (9.04e+1) -	2.3331e+2 (1.77e+1) -	2.2422e+2 (1.24e+1) -	5.9190e+2 (1.24e+2) -	1.3843e+2 (4.64e+1)
	20	8.2568e+2 (1.22e+2) -	2.4264e+2 (1.53e+1) -	2.2892e+2 (1.43e+1) -	5.4005e+2 (1.13e+2) -	8.9661e+1 (3.98e+1)
	25	8.1076e+2 (9.43e+1) -	2.4472e+2 (1.40e+1) -	2.3867e+2 (1.31e+1) -	4.8779e+2 (1.05e+2) -	1.1615e+2 (4.23e+1)
DTLZ4	3	9.0001e-2 (1.62e-1) +	1.1611e-1 (1.79e-1) =	2.8719e-1 (4.51e-2) =	5.1729e-1 (1.86e-1) =	3.4467e-1 (2.84e-1)
	5	3.9655e-1 (2.77e-2) -	2.7602e-1 (1.58e-2) =	6.0418e-1 (2.30e-2) -	5.3985e-1 (3.61e-2) -	3.1461e-1 (8.32e-2)
	8	7.2569e-1 (3.30e-2) -	1.1221e+0 (1.71e-1) -	8.2330e-1 (3.21e-2) -	7.8779e-1 (5.00e-2) -	4.3376e-1 (1.86e-2)
	10	7.8968e-1 (3.27e-2) -	1.1973e+0 (1.18e-1) -	9.4123e-1 (3.60e-2) -	8.6547e-1 (3.61e-2) -	5.3245e-1 (1.82e-2)
	15	8.9014e-1 (3.14e-2) -	1.3755e+0 (7.99e-2) -	1.0688e+0 (3.42e-2) -	9.7134e-1 (3.39e-2) -	6.3521e-1 (9.16e-3)

续表

	20	1.3096e+0 (6.97e-2) -	2.3886e+0 (3.62e-1) -	1.2560e+0 (4.40e-2) -	1.2656e+0 (5.75e-2) -	7.8879e-1 (1.37e-2)
	25	1.2692e+0 (6.25e-2) -	2.0086e+0 (3.20e-1) -	1.2357e+0 (3.79e-2) -	1.2118e+0 (4.47e-2) -	8.0724e-1 (7.80e-3)
DTLZ5	3	7.4641e-3 (4.33e-4) +	6.4283e-3 (4.93e-4) +	2.5256e-2 (9.01e-3) -	5.7911e-3 (3.28e-4) +	1.3960e-2 (5.93e-4)
	5	2.9421e-1 (7.38e-2) -	9.2052e-2 (3.76e-2) -	1.9517e-1 (3.85e-2) -	1.1349e-1 (2.67e-2) -	3.1773e-2 (8.45e-3)
	8	8.8514e-1 (2.33e-1) -	1.6435e-1 (6.12e-2) -	2.7448e-1 (3.21e-2) -	1.7724e-1 (3.85e-2) -	4.8118e-2 (2.63e-2)
	10	7.5581e-1 (2.13e-1) -	1.4451e-1 (5.04e-2) -	2.9564e-1 (6.27e-2) -	1.9376e-1 (4.17e-2) -	6.2306e-2 (1.84e-2)
	15	9.1551e-1 (3.44e-1) -	1.8385e-1 (8.28e-2) -	3.3818e-1 (5.51e-2) -	2.4146e-1 (5.22e-2) -	6.6160e-2 (1.66e-2)
	20	1.6229e+0 (4.74e-1) -	7.6904e-1 (9.11e-2) -	4.2462e-1 (8.42e-2) -	3.2207e-1 (9.96e-2) -	2.2880e-1 (5.16e-2)
	25	1.5578e+0 (4.34e-1) -	7.7035e-1 (7.88e-2) -	4.3987e-1 (8.19e-2) -	3.3756e-1 (8.13e-2) -	2.1219e-1 (6.11e-2)
DTLZ6	3	1.4336e-1 (3.55e-1) -	6.8612e-3 (1.13e-3) +	7.0336e-2 (2.03e-1) -	1.4196e+0 (1.04e+0) -	5.1537e-2 (1.77e-1)
	5	6.1872e+0 (6.76e-1) -	1.9541e-1 (1.82e-1) +	7.2340e-2 (2.97e-2) =	4.3013e+0 (1.39e+0) -	2.6576e-1 (3.88e-1)
	8	8.2426e+0 (1.09e+0) -	1.1819e+0 (8.01e-1) -	1.6931e-1 (5.85e-2) =	5.6516e+0 (1.11e+0) -	2.3009e-1 (2.28e-1)
	10	8.2966e+0 (8.00e-1) -	1.3562e+0 (5.87e-1) -	1.9931e-1 (5.85e-2) =	6.4510e+0 (1.05e+0) -	3.5182e-1 (3.27e-1)
	15	8.6307e+0 (5.20e-1) -	1.3942e+0 (9.73e-1) -	2.4717e-1 (9.56e-2) =	6.9788e+0 (1.04e+0) -	2.6265e-1 (2.65e-1)
	20	9.3203e+0 (3.41e-1) -	1.4327e+0 (6.79e-1) -	4.5235e-1 (2.07e-1) +	6.2411e+0 (1.43e+0) -	5.2040e-1 (6.68e-1)
	25	9.2585e+0 (2.84e-1) -	1.3961e+0 (6.45e-1) -	5.5053e-1 (2.01e-1) -	6.6760e+0 (1.14e+0) -	4.8979e-1 (6.76e-1)
DTLZ7	3	1.3794e-1 (1.95e-1) +	2.5474e-1 (2.69e-1) +	8.6545e-2 (7.48e-3) +	6.0206e-1 (3.15e-1) +	1.0149e+0 (5.54e-2)
	5	9.1181e-1 (2.01e-1) +	4.5517e-1 (1.95e-1) +	4.9990e-1 (1.30e-1) +	1.6386e+0 (8.80e-1) +	2.3321e+0 (5.98e-2)
	8	1.4347e+1 (3.32e+0) -	1.5262e+0 (4.55e-1) +	1.9376e+0 (5.54e-1) +	4.0131e+0 (1.97e+0) +	4.6366e+0 (9.82e-2)
	10	2.4338e+1 (3.72e+0) -	2.6933e+0 (1.39e+0) +	2.1670e+0 (2.91e-1) +	5.9562e+0 (2.19e+0) =	6.0648e+0 (1.15e-1)
	15	4.6178e+1 (4.00e+0) -	5.6449e+0 (2.65e+0) +	3.7267e+0 (6.69e-1) +	1.6476e+1 (4.45e+0) -	1.1999e+1 (1.70e-1)
	20	7.6036e+1 (6.82e+0) -	3.4917e+0 (4.34e-1) +	4.1731e+0 (5.09e-1) +	1.0477e+1 (5.65e+0) +	1.6135e+1 (2.37e-1)
	25	—	—	—	—	—
	+/-/=	4/43/1	9/35/4	7/36/5	6/40/2	

注: DTLZ7 在 M = 25 时由于实验平台内存限制未获得有效结果。

Table 3. IGD values of the proposed algorithm and comparison algorithms on WFG1-WFG7

表 3. WFG1-WFG7 上所提出的算法和各比较算法的 IGD 值

Problem	M	CMOPSO	MMOPSO	MOPSOCD	SMP SO	TP-CMOPSO
WFG1	3	1.5391e+0 (1.95e-2) -	9.8633e-1 (7.84e-2) +	2.0906e+0 (3.61e-2) -	1.5404e+0 (1.32e-2) -	1.4937e+0 (2.28e-2)
	5	2.1765e+0 (5.42e-2) -	1.8720e+0 (4.10e-2) +	2.5202e+0 (3.62e-2) -	2.1499e+0 (4.55e-2) -	2.0017e+0 (2.99e-2)
	8	2.8736e+0 (3.57e-2) =	2.6060e+0 (5.53e-2) +	3.0494e+0 (2.68e-2) -	2.7688e+0 (3.64e-2) +	2.8504e+0 (7.22e-2)
	10	3.2550e+0 (2.83e-2) -	3.0017e+0 (3.10e-2) +	3.3581e+0 (3.15e-2) -	3.1436e+0 (3.01e-2) +	3.1968e+0 (5.54e-2)
	15	4.1524e+0 (2.27e-2) +	3.9681e+0 (4.76e-2) +	4.2401e+0 (2.65e-2) -	4.0786e+0 (2.53e-2) +	4.1961e+0 (6.80e-2)
	20	5.4504e+0 (5.96e-2) -	5.2130e+0 (2.39e-1) -	5.6768e+0 (7.64e-2) -	5.3371e+0 (4.54e-2) -	5.0048e+0 (1.65e-1)
	25	5.9960e+0 (6.76e-2) -	4.7654e+0 (1.81e-1) +	6.1996e+0 (1.00e-1) -	5.8218e+0 (6.15e-2) -	5.2193e+0 (3.29e-1)

续表

WFG2	3	1.8446e-1 (5.38e-3) +	2.4135e-1 (1.45e-2) -	3.6936e-1 (2.39e-2) -	2.9294e-1 (2.33e-2) -	1.8936e-1 (6.70e-3)
	5	6.4181e-1 (2.94e-2) -	7.6938e-1 (3.35e-2) -	1.0542e+0 (9.40e-2) -	9.0532e-1 (6.56e-2) -	4.8503e-1 (9.46e-3)
	8	1.2972e+0 (4.13e-2) -	1.3786e+0 (8.06e-2) -	2.0481e+0 (6.60e-2) -	1.6395e+0 (1.04e-1) -	1.0313e+0 (7.98e-2)
	10	1.7408e+0 (9.31e-2) -	1.5673e+0 (6.51e-2) -	2.4198e+0 (8.35e-2) -	2.1580e+0 (1.77e-1) -	1.2020e+0 (8.53e-2)
	15	2.8362e+0 (2.22e-1) -	2.5204e+0 (2.35e-1) -	3.2748e+0 (7.35e-2) -	3.4209e+0 (3.51e-1) -	2.0264e+0 (2.13e-1)
	20	4.9078e+0 (4.30e-1) =	5.2968e+0 (1.83e-1) =	4.9538e+0 (1.95e-1) =	5.1362e+0 (2.59e-1) =	6.2781e+0 (2.59e+0)
WFG3	25	6.0072e+0 (7.21e-1) =	4.8237e+0 (6.79e-1) =	5.3720e+0 (3.92e-1) =	5.8582e+0 (4.26e-1) =	6.4468e+0 (2.75e+0)
	3	2.2541e-1 (1.93e-2) -	1.7358e-1 (2.58e-2) -	4.1477e-1 (3.00e-2) -	2.8707e-1 (3.57e-2) -	1.2050e-1 (1.47e-2)
	5	1.0538e+0 (7.97e-2) -	7.4965e-1 (1.10e-1) -	8.8470e-1 (8.61e-2) -	6.3275e-1 (6.35e-2) -	3.0070e-1 (3.65e-2)
	8	2.0238e+0 (1.55e-1) -	1.3722e+0 (1.96e-1) -	1.2796e+0 (9.37e-2) -	9.6243e-1 (9.05e-2) -	7.8431e-1 (9.53e-2)
	10	2.5358e+0 (2.58e-1) -	1.4993e+0 (2.57e-1) -	1.4401e+0 (1.17e-1) -	1.0926e+0 (1.32e-1) =	1.0420e+0 (1.10e-1)
	15	3.5719e+0 (3.74e-1) -	2.3184e+0 (3.00e-1) -	1.7714e+0 (1.19e-1) +	1.4026e+0 (1.38e-1) +	2.0097e+0 (3.42e-1)
WFG4	20	5.7355e+0 (1.36e+0) -	9.5730e+0 (1.02e+0) -	2.1504e+0 (2.21e-1) +	1.6772e+0 (2.66e-1) +	4.7003e+0 (3.10e+0)
	25	5.5886e+0 (9.49e-1) =	1.1613e+1 (1.51e+0) -	2.2508e+0 (2.30e-1) +	1.6800e+0 (2.49e-1) +	5.2127e+0 (1.86e+0)
	3	2.7865e-1 (6.22e-3) -	3.1670e-1 (1.58e-2) -	3.9663e-1 (1.48e-2) -	3.3868e-1 (1.54e-2) -	2.7279e-1 (5.90e-3)
	5	1.0815e+0 (1.38e-2) +	1.2453e+0 (1.85e-2) -	1.4268e+0 (3.17e-2) -	1.3636e+0 (6.77e-2) -	1.1665e+0 (7.39e-3)
	8	2.8988e+0 (3.07e-2) +	3.3969e+0 (4.35e-2) -	3.3355e+0 (3.24e-2) -	3.5298e+0 (7.72e-2) -	2.9941e+0 (2.44e-2)
	10	4.1538e+0 (3.61e-2) +	4.6160e+0 (4.61e-2) -	4.5987e+0 (3.52e-2) -	4.7761e+0 (7.87e-2) -	4.3955e+0 (4.32e-2)
WFG5	15	7.7609e+0 (9.64e-2) +	8.2896e+0 (9.12e-2) -	7.8461e+0 (8.43e-2) +	8.4585e+0 (1.27e-1) -	8.0949e+0 (6.74e-2)
	20	1.3817e+1 (3.32e-1) +	1.5950e+1 (3.39e-1) +	1.2851e+1 (1.73e-1) +	1.4153e+1 (2.00e-1) +	1.7455e+1 (7.79e-1)
	25	1.7232e+1 (4.20e-1) +	1.9997e+1 (2.58e-1) +	1.5187e+1 (1.94e-1) +	1.7131e+1 (2.07e-1) +	2.1764e+1 (1.39e+0)
	3	2.6390e-1 (7.45e-3) -	2.9040e-1 (1.25e-2) -	2.9481e-1 (1.78e-2) -	3.2314e-1 (2.67e-2) -	2.4783e-1 (5.69e-3)
	5	1.0852e+0 (3.18e-2) +	1.2860e+0 (4.31e-2) -	1.4341e+0 (2.70e-2) -	1.5655e+0 (4.47e-2) -	1.1274e+0 (1.91e-2)
	8	3.0798e+0 (5.90e-2) +	3.5762e+0 (6.98e-2) -	3.6792e+0 (5.94e-2) -	3.7625e+0 (3.75e-2) -	3.1643e+0 (3.80e-2)
WFG6	10	4.3459e+0 (4.97e-2) +	4.7582e+0 (7.15e-2) -	5.0316e+0 (7.58e-2) -	5.0379e+0 (5.39e-2) -	4.4464e+0 (5.89e-2)
	15	7.8461e+0 (8.94e-2) +	8.3478e+0 (8.91e-2) =	9.8099e+0 (1.86e-1) -	8.9974e+0 (1.05e-1) -	8.3279e+0 (6.63e-2)
	20	1.4413e+1 (2.63e-1) +	1.7230e+1 (2.82e-1) =	1.8991e+1 (4.46e-1) -	1.6142e+1 (2.58e-1) +	1.7567e+1 (6.80e-1)
	25	1.7336e+1 (2.83e-1) +	2.1029e+1 (3.59e-1) +	2.4777e+1 (6.12e-1) -	1.9807e+1 (4.36e-1) +	2.1603e+1 (7.97e-1)
	3	2.8492e-1 (1.49e-2) -	3.2560e-1 (1.31e-2) -	3.1328e-1 (1.54e-2) -	3.4946e-1 (5.21e-2) -	2.7451e-1 (1.34e-2)
	5	1.2447e+0 (4.36e-2) -	1.3271e+0 (4.94e-2) -	1.5294e+0 (4.94e-2) -	1.5902e+0 (5.69e-2) -	1.1938e+0 (1.21e-2)
WFG6	8	3.2062e+0 (4.61e-2) -	3.5806e+0 (4.51e-2) -	3.4580e+0 (2.95e-2) -	3.7594e+0 (8.87e-2) -	3.1056e+0 (3.09e-2)
	10	4.4258e+0 (4.54e-2) -	4.7355e+0 (5.62e-2) -	4.6446e+0 (3.34e-2) -	5.0280e+0 (8.68e-2) -	4.3947e+0 (3.21e-2)
	15	7.8498e+0 (7.48e-2) +	8.2589e+0 (7.68e-2) -	8.0479e+0 (7.65e-2) +	8.7627e+0 (1.00e-1) -	8.1647e+0 (9.78e-2)
	20	1.4154e+1 (2.43e-1) +	1.6666e+1 (3.58e-1) +	1.3556e+1 (1.26e-1) +	1.4689e+1 (2.06e-1) +	1.7742e+1 (9.93e-1)
	25	1.7121e+1 (2.78e-1) +	2.0597e+1 (4.57e-1) +	1.6299e+1 (2.03e-1) +	1.8108e+1 (3.27e-1) +	2.2435e+1 (1.21e+0)

续表

	3	2.4461e-1 (5.44e-3) +	2.8725e-1 (1.19e-2) -	4.1372e-1 (2.05e-2) -	4.0357e-1 (2.32e-2) -	2.5266e-1 (4.76e-3)
	5	1.1622e+0 (2.98e-2) +	1.3023e+0 (4.08e-2) -	1.4883e+0 (2.58e-2) -	1.4128e+0 (2.53e-2) -	1.1720e+0 (1.16e-2)
	8	2.9695e+0 (2.57e-2) +	3.4497e+0 (6.13e-2) -	3.4569e+0 (2.85e-2) -	3.5189e+0 (6.62e-2) -	3.0095e+0 (1.99e-2)
WFG7	10	4.1886e+0 (3.84e-2) +	4.6018e+0 (4.76e-2) -	4.6368e+0 (4.41e-2) -	4.7897e+0 (6.35e-2) -	4.3697e+0 (3.24e-2)
	15	7.6076e+0 (8.94e-2) +	8.1601e+0 (9.46e-2) =	8.0110e+0 (6.01e-2) +	8.4943e+0 (8.31e-2) -	8.1159e+0 (1.04e-1)
	20	1.4228e+1 (4.01e-1) +	1.5884e+1 (3.56e-1) +	1.3726e+1 (2.28e-1) +	1.4852e+1 (3.12e-1) +	2.0126e+1 (1.57e+0)
	25	1.7673e+1 (3.42e-1) +	1.9545e+1 (4.16e-1) +	1.6692e+1 (4.66e-1) +	1.8492e+1 (5.21e-1) +	2.5262e+1 (2.03e+0)
		+/-/=	24/21/4	13/31/5	12/35/2	14/32/3

Table 4. Wilcoxon test results for HV on DTLZ and WFG
表 4. DTLZ 和 WFG 测试集上 HV 指标的 Wilcoxon 检验结果

Problem	CMOPSO	MMOPSO	MOPSOC	SMP
DTLZ	2/26/7	6/21/8	6/24/5	2/29/4
WFG	3/28/4	9/23/3	4/29/2	1/31/3

为了直观展示各算法所得解集的分布情况, 本文绘制了五种算法在 3 目标和 8 目标 DTLZ2 测试问题上的最优解分布图, 如图 2 和图 3 所示。DTLZ2 具有规则凹形 Pareto 前沿, 能够较好地检验算法在连续 PF 上的收敛性和分布性。从图 2 可以看出, 在 3 目标 DTLZ2 测试实例上, CMOPSO 和 TP-CMOPSO 均能够较好地收敛到真实 PF 附近。其中, CMOPSO 获得的解集整体分布较为均匀, 但在部分边界区域和中间区域仍存在一定疏密不均现象。相比之下, TP-CMOPSO 所得解集能够较完整地覆盖真实 PF, 且解在 PF 表面上的分布更加均衡, 说明本文算法在保持收敛性的同时具有更好的分布维护能力。MMOPSO、MOPSOC 和 SMP 的解集也基本能够靠近真实 PF, 但其分布效果相对不够稳定, 存在解点分布较为分散、部分区域覆盖不足的问题。

从图 3 可以看出, 在 8 目标 DTLZ2 测试实例上, CMOPSO、MOPSOC 和 SMP 所得解集分布较为分散, 部分曲线偏离主要前沿区域, 说明这些算法在该测试实例上难以获得兼具收敛性和分布性的解集。MMOPSO 相比上述算法有所改善, 但其解集仍存在一定不均匀现象, 整体分布效果不够稳定。相比之下, TP-CMOPSO 所得解集整体更加接近 DTLZ2 的真实 Pareto 前沿, 曲线分布也更加规整, 能够较好地覆盖主要前沿区域。该可视化结果与表中 IGD 和 HV 指标结果一致, 即 TP-CMOPSO 在 DTLZ2 的 8 目标实例上获得了最小 IGD 值和最大 HV 值。

实验结果表明, TP-CMOPSO 在 DTLZ2 上表现较好, 而在具有退化和偏置特征的 DTLZ6 上性能相对较弱。为进一步分析算法在 DTLZ6 上表现不佳的原因, 本文选取 20 目标 DTLZ2 和 DTLZ6 作为表现较好与表现较弱的代表性实例, 对进化过程中精英组的多样性进行诊断。具体采用归一化目标空间中精英粒子的平均两两夹角衡量其方向多样性, 并进一步计算精英组平均夹角与整个种群平均夹角的比值, 以反映精英筛选后的相对多样性。结果表明, DTLZ2 上的精英组在进化过程中能够保持较高的方向多样性, 而 DTLZ6 上的精英组平均夹角整体较小, 且在搜索早期的相对多样性明显较低, 说明精英粒子更容易集中于少数相似搜索方向。由于 TP-CMOPSO 的粒子学习对象主要来源于精英竞争组, 这种精英方向的集中可能导致学习信息趋于单一, 从而限制种群对不同有效 PF 区域的探索, 这在一定程度上解释了算法在 DTLZ6 上的性能下降。

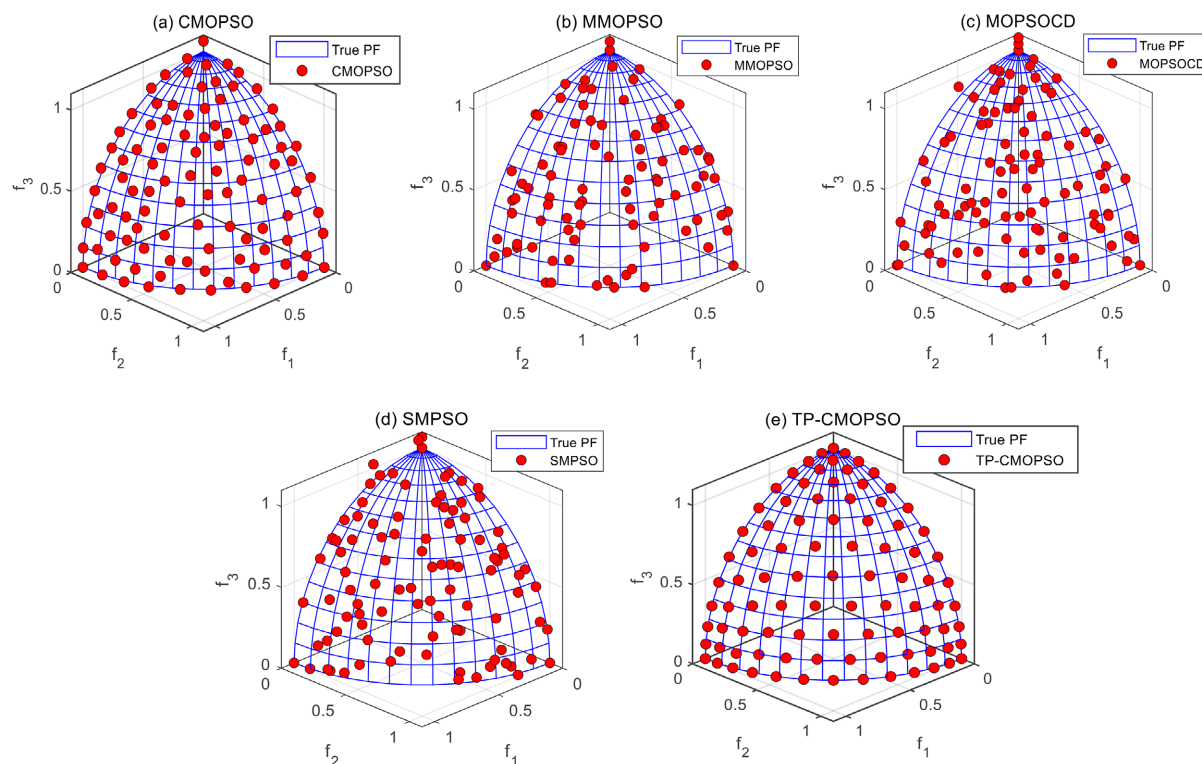


Figure 2. Distributions of the best solutions obtained by five algorithms on three-objective DTLZ2

图 2. 五种算法在 3 目标 DTLZ2 上的最优解分布图

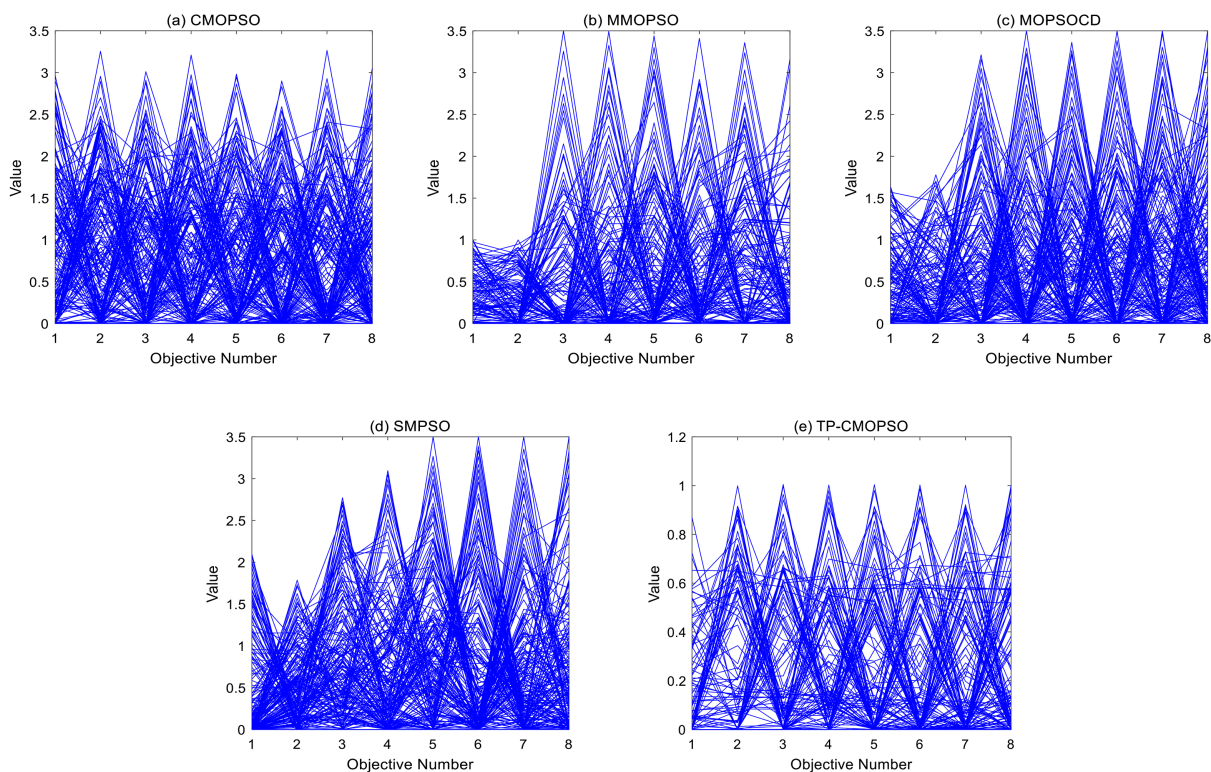


Figure 3. Distributions of the best solutions obtained by five algorithms on eight-objective DTLZ2

图 3. 五种算法在 8 目标 DTLZ2 上的最优解分布图

5.2. 不同聚合方式的对比分析

为分析不同聚合方式对算法性能的影响, 构造等权加法融合的 TP-CMOPSO-Add 算法。将乘法融合与等权加法融合分别在 DTLZ1-DTLZ7 和 WFG1-WFG7 上进行 30 次独立实验。表 5 中 Wilcoxon 秩和检验结果表明, 乘法融合在多数测试实例上表现出更好的稳定性, 尤其在 DTLZ 测试集上优势较为明显; WFG 测试集上的差异则具有更强的问题依赖性。加法模型允许收敛性和分布性之间进行一定程度的补偿, 而乘法模型对任一指标的较低取值更加敏感, 因此, 本文最终采用乘法方式构造联合青睐指标。

Table 5. Wilcoxon test results of different aggregation methods
表 5. 不同聚合方式的 Wilcoxon 检验结果

测试集	$M = 3$	$M = 5$	$M = 8$	$M = 10$	$M = 15$
DTLZ1~7	4/2/1	4/1/2	6/1/0	5/0/2	7/0/0
WFG1~7	5/1/1	4/0/3	3/2/2	3/2/2	4/2/1

注: “+/-/-” 分别表示 TP-CMOPSO 显著优于 TP-CMOPSO-Add、二者无显著差异以及 TP-CMOPSO 显著劣于 TP-CMOPSO-Add。

5.3. 参数分析

精英集大小是影响本文算法性能的重要参数之一。若精英集规模过小, 虽然能够增强优秀个体对种群的引导作用, 但学习对象过于集中, 容易降低种群多样性, 并增加算法陷入局部最优的风险; 若精英集规模过大, 则部分质量相对较低的个体也会被纳入精英集, 从而削弱精英个体的引导作用, 同时减少被更新粒子的数量, 影响算法的整体进化效率。本文选取种群规模的 5%、10%、15%和 20%作为候选参数, 为保证实验公平性, 除精英集大小外, 其他参数均保持一致。从表 6 的实验结果可以看出, TP-CMOPSO 在 14 个测试实例中有 8 个取得了最优 IGD 值, 整体表现优于其他对比算法。因此, 本文将精英竞争组规模设置为种群规模的 10%。

Table 6. Number of best IGD results under different elite competition group sizes
表 6. 不同精英竞争组规模下取得最优 IGD 的测试实例数

最优 IGD 数量	M	TP-CMOPSO_E5	TP-CMOPSO	TP-CMOPSO_E15	TP-CMOPSO_E20
DTLZ	8	0	4	1	2
WFG	8	0	4	1	2
合计	8	0	8	2	4

6. 结论

针对传统多目标粒子群优化算法在超多目标优化问题中容易出现 Pareto 支配关系区分能力减弱、学习对象选择不稳定以及收敛性和多样性难以平衡等问题, 本文提出了一种基于联合青睐概率精英竞争学习的多目标粒子群优化算法 TP-CMOPSO。该算法首先利用联合青睐概率对粒子在多个目标上的综合优化潜力进行评价, 并根据评价结果构建精英竞争组, 为粒子更新提供质量更高的学习对象; 其次, 采用基于角度最小准则的精英学习策略, 使粒子优先向目标空间中方向更接近的精英个体学习, 从而增强粒子更新方向的稳定性; 最后, 在环境选择阶段引入参考向量关联和联合青睐概率筛选机制, 使算法在保留收敛性较好个体的同时, 能够维持解集在不同参考方向上的分布均匀性。

通过在 DTLZ 和 WFG 系列标准测试问题上的对比实验可以看出, 本文算法在多数测试实例上表现出较好的综合优化性能。特别是在 DTLZ1、DTLZ2 等规则 Pareto 前沿问题上, TP-CMOPSO 能够获得较优的 IGD 和 HV 指标值, 说明联合青睐概率能够有效增强算法对高质量个体的选择压力, 提高种群向真实 Pareto 前沿收敛的能力。同时, 在 WFG2 和 WFG3 等具有不连续或退化 Pareto 前沿的测试问题上, 本文算法也表现出较好的收敛速度和解集分布效果, 表明参考向量小生境选择策略能够在一定程度上改善解集的覆盖性和均匀性。实验结果总体说明, 本文提出的联合青睐概率评价机制、精英竞争学习策略和环境选择策略能够有效协同, 提高多目标粒子群优化算法在收敛性和多样性方面的平衡能力。

然而, 本文算法在部分具有复杂混合特征、偏置特征或强变量依赖关系的 WFG 测试问题上仍存在一定不足。例如, 在 WFG1 等复杂不规则 Pareto 前沿问题中, 均匀参考向量难以完全匹配真实 PF 结构, 粒子学习方向的灵活性也有待进一步提高, 导致算法在部分指标上未能取得明显优势。因此, 后续工作可以从两个方面进一步改进: 一方面, 引入自适应参考向量调整策略, 使参考向量能够根据种群分布和 Pareto 前沿形状动态变化; 另一方面, 结合更加灵活的搜索算子或多策略协同机制, 增强算法对复杂多峰、退化和不规则前沿问题的探索能力, 从而进一步提升算法在复杂超多目标优化问题中的适应性和稳定性。

基金项目

国家自然科学基金(12201085); 重庆市自然科学基金(CSTB2023NSCQ-MSX0332)。

参考文献

- [1] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. Proceedings of ICNN'95 International Conference on Neural Networks, Perth, 27 November-1 December 1995, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>
- [2] 赵乃刚, 李勇, 王振荣. 可行性规则动态调整的多目标粒子群算法[J]. 计算机应用研究, 2018, 35(5): 1304-1306+1336.
- [3] Raquel, C.R. and Naval Jr, P.C. (2005) An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, Washington DC, 25-29 June 2005, 257-264. <https://doi.org/10.1145/1068009.1068047>
- [4] Coello, C.A.C., Pulido, G.T. and Lechuga, M.S. (2004) Handling Multiple Objectives with Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation*, 8, 256-279. <https://doi.org/10.1109/tevc.2004.826067>
- [5] Zhan, Z.H., Li, J., Cao, J., et al. (2013) Multiple Populations for Multiple Objectives: A Coevolutionary Technique for Solving Multiobjective Optimization Problems. *IEEE Transactions on Cybernetics*, 43, 445-463. <https://doi.org/10.1109/tsmcb.2012.2209115>
- [6] Zitzler, E. and Künzli, S. (2004) Indicator-Based Selection in Multiobjective Search. In: *Lecture Notes in Computer Science*, Springer, 832-842. https://doi.org/10.1007/978-3-540-30217-9_84
- [7] Garcia, I.C., Coello, C.A.C. and Arias-Montano, A. (2014) MOPSOhv: A New Hypervolume-Based Multi-Objective Particle Swarm Optimizer. 2014 *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, 6-11 July 2014, 266-273. <https://doi.org/10.1109/cec.2014.6900540>
- [8] 陈国玉, 李军华, 黎明, 等. 基于 R_2 指标和参考向量的高维多目标进化算法[J]. 自动化学报, 2021, 47(11): 2675-2690.
- [9] Cheng, R. and Jin, Y. (2014) A Competitive Swarm Optimizer for Large Scale Optimization. *IEEE Transactions on Cybernetics*, 45, 191-204. <https://doi.org/10.1109/tevc.2014.2322602>
- [10] Zhang, X., Zheng, X., Cheng, R., Qiu, J. and Jin, Y. (2018) A Competitive Mechanism Based Multi-Objective Particle Swarm Optimizer with Fast Convergence. *Information Sciences*, 427, 63-76. <https://doi.org/10.1016/j.ins.2017.10.037>
- [11] Gu, Q., Liu, Y., Chen, L. and Xiong, N. (2022) An Improved Competitive Particle Swarm Optimization for Many-Objective Optimization Problems. *Expert Systems with Applications*, 189, Article 116118. <https://doi.org/10.1016/j.eswa.2021.116118>
- [12] Lin, Q., Li, J., Du, Z., Chen, J. and Ming, Z. (2015) A Novel Multi-Objective Particle Swarm Optimization with Multiple Search Strategies. *European Journal of Operational Research*, 247, 732-744.

- <https://doi.org/10.1016/j.ejor.2015.06.071>
- [13] Qi, S., Zou, J., Yang, S., Jin, Y., Zheng, J. and Yang, X. (2022) A Self-Exploratory Competitive Swarm Optimization Algorithm for Large-Scale Multiobjective Optimization. *Information Sciences*, **609**, 1601-1620. <https://doi.org/10.1016/j.ins.2022.07.110>
 - [14] Huang, W. and Zhang, W. (2022) Multi-Objective Optimization Based on an Adaptive Competitive Swarm Optimizer. *Information Sciences*, **583**, 266-287. <https://doi.org/10.1016/j.ins.2021.11.031>
 - [15] Zheng, M., Teng, H., Yu, J., *et al.* (2023) Probability-Based Multi-Objective Optimization for Material Selection. Springer.
 - [16] Zheng, M. and Yu, J. (2024) An Appropriate Approach of Design and Machining of Mechanical Part by Means of Probabilistic Multi-Objective Optimization. *Materialwissenschaft und Werkstofftechnik*, **55**, 116-124. <https://doi.org/10.1002/mawe.202300032>
 - [17] 于洁, 贾璞, 肖超妮, 郑晓晖. 概率基多目标优化在化工机械正交试验及稳健设计中的应用[J]. 化工机械, 2023, 50(6): 775-783.
 - [18] 刘建昌, 李飞, 王洪海, 等. 进化高维多目标优化算法研究综述[J]. 控制与决策, 2018, 33(5): 879-887.
 - [19] 武保同, 舒若琦, 陈志祥. 基于环境识别策略的多目标自适应粒子群算法及应用[J]. 计算机应用研究, 2025, 42(10): 2980-2988.
 - [20] Beyer, K., Goldstein, J., Ramakrishnan, R. and Shaft, U. (1999) When Is “Nearest Neighbor” Meaningful? In: *Lecture Notes in Computer Science*, Springer, 217-235. https://doi.org/10.1007/3-540-49257-7_15
 - [21] Aggarwal, C.C., Hinneburg, A. and Keim, D.A. (2001) On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: *Lecture Notes in Computer Science*, Springer, 420-434. https://doi.org/10.1007/3-540-44503-x_27
 - [22] Nebro, A.J., Durillo, J.J., Garcia-Nieto, J., Coello Coello, C.A., Luna, F. and Alba, E. (2009) SMPSO: A New PSO-Based Metaheuristic for Multi-Objective Optimization. 2009 *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)*, Nashville, 30 March-2 April 2009, 66-73. <https://doi.org/10.1109/mcdm.2009.4938830>
 - [23] Tian, Y., Cheng, R., Zhang, X. and Jin, Y. (2017) PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Computational Intelligence Magazine*, **12**, 73-87. <https://doi.org/10.1109/mci.2017.2742868>
 - [24] Deb, K., Thiele, L., Laumanns, M. and Zitzler, E. (2005) Scalable Test Problems for Evolutionary Multiobjective Optimization. In: *Advanced Information and Knowledge Processing*, Springer, 105-145. https://doi.org/10.1007/1-84628-137-7_6
 - [25] Huband, S., Hingston, P., Barone, L. and While, L. (2006) A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, **10**, 477-506. <https://doi.org/10.1109/tevc.2005.861417>
 - [26] Bosman, P.A.N. and Thierens, D. (2003) The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, **7**, 174-188. <https://doi.org/10.1109/tevc.2003.810761>
 - [27] Zitzler, E. and Thiele, L. (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, **3**, 257-271. <https://doi.org/10.1109/4235.797969>