

基于动态分区与收敛速度控制器的改进竞争群优化算法

张伟^{1,2†}, 伊杰昌¹

(1. 河南理工大学 电气工程与自动化学院, 河南 焦作 454003;
2. 河南省煤矿装备智能检测与控制重点实验室, 河南 焦作 454003)

摘要: 为提升竞争群优化 (CSO) 算法在解决复杂高维优化问题时的性能, 提出一种基于分区策略与收敛速度控制器的改进竞争群优化 (PCSCCSO) 算法. 首先, 采用适应度变化率驱动的动态分区策略, 以增强算法的收敛性和搜索效率; 然后, 提出一种快速 CSO 策略, 通过三重竞争机制增强算法的寻优能力: 获胜粒子通过对立学习策略更新, 失败粒子向获胜子群平均位置学习, 劣败粒子通过变异增强局部搜索, 这些策略能够有效平衡全局探索与局部开发, 提高算法的寻优效率; 最后, 结合粒子与全局最优解间的余弦相似度以及停滞计数, 设计自适应的收敛速度控制器, 用以调节粒子的搜索行为, 从而避免粒子陷入局部最优解, 加速全局收敛. 理论分析验证了所提出算法的稳定性和收敛性. 实验结果表明, 与其他改进算法相比, PCSCCSO 算法在处理复杂高维优化问题时具有更好的收敛精度和收敛效率.

关键词: 竞争群优化算法; 高维优化; 种群分区; 三重竞争; 快速 CSO 策略; 收敛速度控制器

中图分类号: TP183 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.0304

引用格式: 张伟, 伊杰昌. 基于动态分区与收敛速度控制器的改进竞争群优化算法 [J]. 控制与决策, 2025, 40(10): 3019-3028.

Improved competitive swarm optimization algorithm based on dynamic partition strategy and convergence speed controller

ZHANG Wei^{1,2†}, YI Jie-chang¹

(1. College of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China;
2. Henan Key Laboratory of Intelligent Detection and Control of Coal Mine Equipment, Jiaozuo 454003, China)

Abstract: To enhance the performance of the competitive swarm optimization (CSO) algorithm in solving complex high-dimensional optimization problems, an improved CSO algorithm is proposed based on the partition strategy and convergence speed controller (PCSCCSO). First, a dynamic partition strategy driven by fitness change rate is introduced to improve the convergence and search efficiency. Then, a fast CSO strategy is designed with a triple competition mechanism to strengthen the optimization capability. Winning particles are updated via an opposition-based learning strategy, defeated particles learn from the average position of the winning subgroup and inferior particles undergo mutation-enhanced local search. These mechanisms can effectively balance global exploration and local exploitation, and can enhance the optimization efficiency. Finally, an adaptive convergence speed controller is developed by integrating the cosine similarity between particles and the global best solution with stagnation counting, which can dynamically regulate particle search behaviors to avoid local optima and accelerate global convergence. Theoretical analysis demonstrates the stability and convergence of the proposed algorithm. Experiment results indicate that, compared to other improved algorithms, the PCSCCSO algorithm can obtain superior convergence accuracy and efficiency in solving complex high-dimensional optimization problems.

Keywords: competitive swarm optimization algorithm; high-dimensional optimization; population partitioning; triple competition; fast CSO strategy; convergence speed controller

0 引言

粒子群优化 (PSO) 算法^[1] 概念简单且搜索效率

高, 在各领域得到了广泛研究和应用^[2-5]. 然而, PSO 算法仍然存在一定局限性, 可能无法充分探索整个

收稿日期: 2025-03-25; 录用日期: 2025-05-21.

基金项目: 国家自然科学基金项目 (61703145); 河南省科技攻关项目 (222102210213).

责任编辑: 刘向杰.

†通信作者. E-mail: zwei1563@126.com.

本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

搜索空间,且易陷入局部最优,尤其是在复杂的高维搜索空间中,这一问题更为突出^[6].

为平衡种群的探索与开发,消除全局最优粒子和个体最优粒子对种群的影响,Cheng等^[7]提出了竞争群优化(CSO)算法,采用两两竞争的机制确定获胜者和失败者,通过获胜者和种群平均位置引导失败者学习的过程更新种群,强调种群开发与勘探的平衡.由于CSO的卓越性能,现已成为解决高维优化问题的有效选择之一.

CSO算法每次迭代只更新一半的粒子,失败者向不同方向探索的获胜者学习,虽然这种方法有助于保持多样性,但是会减慢收敛速度.Chu等^[8]提出对于任何算法,在一类问题上任何性能的提升均会被另一类问题上的性能所抵消,因此,任何算法的开发均应描述算法为具体的问题而设计;Mohapatra等^[9]提出了一种改进的竞争群优化(MCSO)算法,使用三重竞争机制每次迭代更新2/3的粒子,提高了算法的寻优效率.在经典的CSO中,获胜者直接进入下一代,以防止有前途的优秀粒子退化.然而,获胜者在没有任何更新的情况下进入下一次迭代,会限制粒子的优化潜力^[10].为进一步加强学习,Wang等^[11]引入了一种基于集群的CSO,获胜者采用PSO的认知成分和社会成分更新速度.Musikawan等^[12]采用了基于对立学习的方法来更新获胜者,可减轻具有误导性信息的获胜者对种群进化造成的负面影响,加快了收敛速度.Xiong等^[13]提出了“前进两步,后退一步”的现象,为减缓这个现象,失败者只以获胜者为引导,删去了种群平均位置一项,虽然能够提升收敛速度,但是也会导致丧失多样性.张伟等^[14]提出了获胜子群自竞争和劣败粒子基于获胜者变异的操作,增强了粒子局部开发能力,避免了算法陷入局部最优.Chen等^[15]提出了一种改进的竞争群优化算法(ImCSO),首先,采用排序配对策略^[16]对失败者进行有效的指导,失败者根据自己的排名向获胜者进行点对点学习;然后,使用差分进化策略^[17]对获胜者进行更新,提高了算法的搜索能力,具有较高的寻优效率和收敛精度.然而,失败者直接向获胜者学习,会错过一些好的失败者的指导.Xiong等^[18]提出了一种结合正交学习策略的竞争群优化(OLCSO)算法,算法中的正交策略将获胜者和失败者组成选定的组合更新失败者,解决了CSO的局限性,提高了失败者的决策效率.这些多样化的粒子更新策略从不同程度提高了算法的鲁棒性和探索能力.

为进一步提升算法的性能,Wang等^[19]采用三重竞争机制增加更新粒子的比例,并提出了两种不

同的机制来更新失败者,促进了探索与开发间的良好平衡;Borowska^[20]提出了一种基于粒子群优化方法和竞争机制的学习竞争群优化(LCSO)算法,利用种群分区策略,在每次迭代中多次进行种群内学习,能够有效保持种群多样性,但是子种群间信息交流较少,不利于子种群间探索与开发的平衡;针对传统CSO存在失败者调整方向和自身速度方向不一致,导致搜索效率降低的问题,Tian等^[21]采用两阶段的更新策略,先通过粒子自身速度预更新位置,再根据预更新的位置与获胜者的位置差异进行调整,在一定程度上减轻了方向对齐带来的影响,提升了搜索效率并优化了解的分布质量,但是对速度方向一致性有所依赖,局部搜索能力不足;为克服文献^[21]的局限性,Liu等^[22]提出了一种全面竞争学习(CCLCSO)算法,通过引入环境竞争、社会竞争和认知竞争3种竞争机制,使得失败者能够从多个获胜者中学习,增强了失败者的学习效率,提升了全局和局部搜索能力,但是其竞争机制增加了计算复杂度,不利于提升高维空间中的收敛效率.

综上所述,为提高CSO算法在求解复杂高维优化问题中的能力,本文提出一种基于分区与收敛速度控制器的改进竞争群优化(PCSCSO)算法.本文主要内容如下:1)根据适应度变化率和停滞计数器动态地对种群进行分区,更加灵活地应对种群停滞的情况,提高种群多样性,并加速收敛;2)为使得算法在处理复杂高维优化问题时能够保持较强的指引能力,提出一种快速CSO策略,只通过获胜者的平均位置来引导失败者进行更新,失败者更精确地朝着当前解空间更优的区域搜索;3)设计一种收敛速度控制器,通过衡量粒子与全局最优解的余弦相似度,粒子采用不同的探索策略,在探索与开发间保持平衡,避免过早收敛;4)给出所提出算法的稳定性和收敛性证明,以确保算法的理论可行性.最终通过仿真实验进一步验证了所提出算法的性能.

1 CSO算法及其问题分析

在CSO算法中,假设种群有 n 个粒子,被随机分为 $n/2$ 组,每组中的两个粒子间进行竞争,根据适应度值将种群分为获胜者和失败者,获胜者直接进入下一代, $X_w(t)$ 、 $V_w(t)$ 、 $X_l(t)$ 、 $V_l(t)$ 分别为第 t 次迭代时获胜者和失败者的位置和速度,失败者的速度和位置按照下式进行更新:

$$V_l(t+1) = R_1(t)V_l(t) + R_2(t)(X_w(t) - X_l(t)) + \varphi R_3(t)(\bar{X}(t) - X_l(t)), \quad (1)$$

$$X_l(t+1) = X_l(t) + V_l(t+1). \quad (2)$$

其中: $\bar{X}(t)$ 为第 t 次迭代后种群的平均位置, φ 为控制 $\bar{X}(t)$ 影响的参数, $R_1(t)$ 、 $R_2(t)$ 、 $R_3(t)$ 为均匀分布在 $(0, 1)$ 内的随机数. 尽管标准CSO通过引入随机寻优机制来降低种群过早收敛的概率, 但是作为PSO的变体, 如何平衡种群多样性与收敛性的问题仍然是CSO优化的难点. 当种群多样性不足时, 算法易陷入局部最优, 而过度强调多样性时, 又可能会导致收敛速度显著下降, 影响整体性能.

在高维优化问题中, CSO算法面临更加严峻的挑战. 高维空间中的极值分布通常更复杂, 粒子可能会在局部极值区域停滞, 而难以跳出这些局部解, 导致算法的全局搜索能力下降. 同时, 维度的增加使得粒子间的距离变大, 传统的局部竞争机制无法有效

地引导粒子进行全局探索, 从而使得算法的收敛速度显著下降. 此外, 过度强调种群的多样性, 可能会导致粒子的更新步长过小, 搜索过程缓慢, 进一步加剧维度灾难的问题. 因此, 在高维优化问题中, CSO算法易遭遇局部最优困境, 缺乏足够的全局探索能力, 进而影响其在复杂优化问题上的表现.

2 PCSCCSO 算法

为提高CSO解决复杂高维优化问题时的能力, 本文提出一种基于分区与收敛速度控制器的改进竞争群优化算法, 整体结构如图1所示. 通过动态种群分区、改进的粒子更新策略和收敛速度控制器等部分协同, 共同指导粒子寻优过程, 提升算法性能.

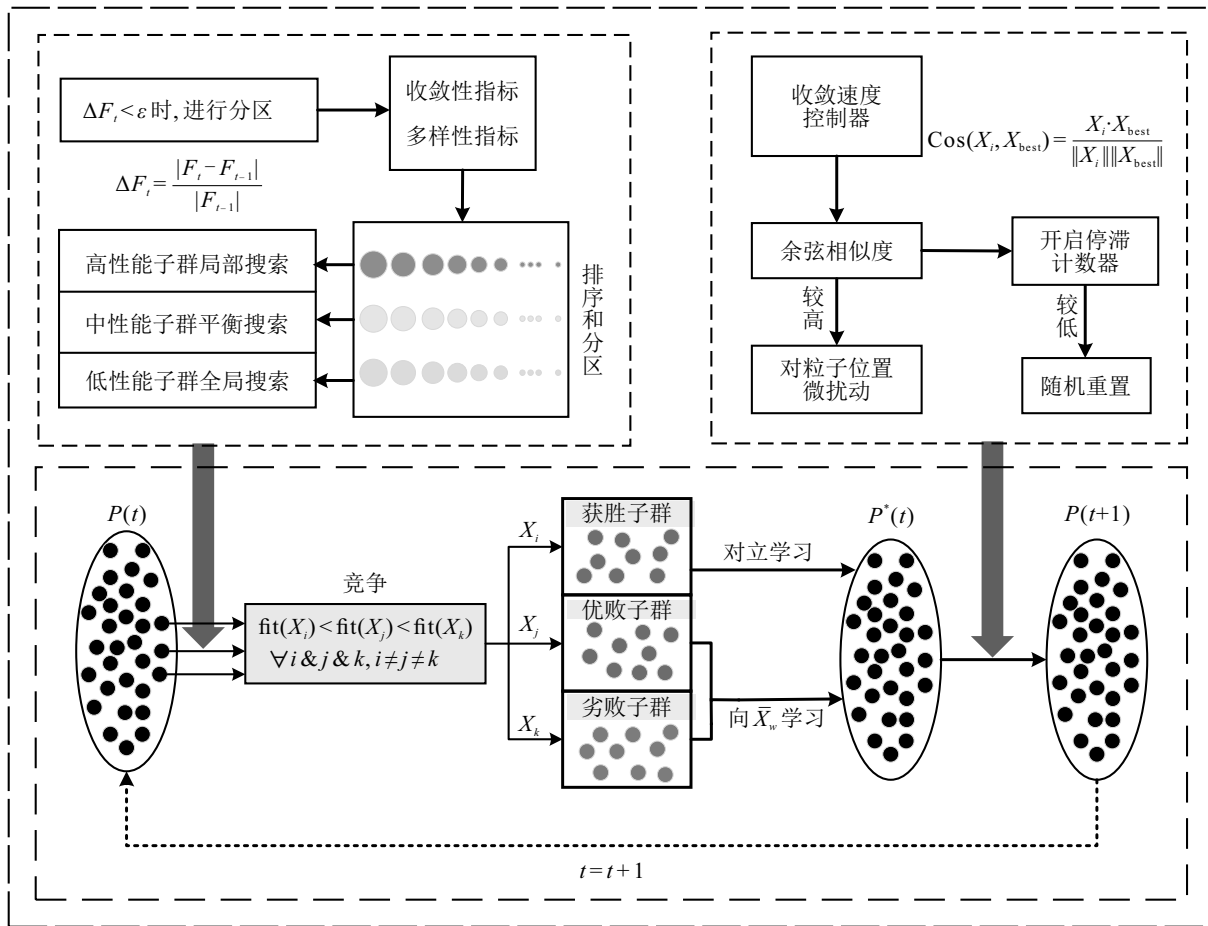


图1 PCSCCSO 算法整体框架

由于基本CSO算法的搜索机制存在局限性, 在解决高维优化问题时寻优效率较低, 需要对其竞争机制、获胜粒子的选择、种群平均位置的计算方式以及粒子变异操作等方面制定优化策略, 以更有效地引导粒子搜索. 根据适应度值变化率对种群动态分区, 每个区域粒子根据自身特性进行更新. 快速CSO策略保证所提出算法在高维优化问题中仍然具有较快的收敛速度. 收敛速度控制器能够有效防止种群

陷入局部最优.

2.1 动态种群分区

在传统CSO算法中, 种群更新通常采用统一策略, 忽略了不同粒子在搜索过程中的角色差异. 为解决这一问题, 本文提出了一种基于融合收敛性和多样性指标的动态种群分区策略, 通过将种群划分为多个子群体, 并针对不同子群体应用不同的进化策略, 提升算法在高维复杂优化问题中的求解效率.

2.1.1 子群体划分

动态种群分区根据种群的适应度分布和多样性进行动态划分,保证每个子种群的进化策略和变异操作能够适应其当前的特性.

采用适应度标准差衡量粒子收敛性,标准差越大,种群差异越大,探索性越强;标准差越小,种群越趋于收敛,开发性越强.如下所示:

$$\sigma_f = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - \bar{f})^2}. \quad (3)$$

其中: f_i 为第 i 个个体适应度值, \bar{f} 为种群适应度均值.

采用种群在每个维度上的方差作为种群多样性的评价指标,方差越大,种群的多样性越高,越有助于算法进行全局探索;方差越小,种群越趋于收敛,越易进入局部最优.如下所示:

$$\sigma_{d,j}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_{i,j} - \bar{X}_j)^2. \quad (4)$$

其中: $X_{i,j}$ 表示第 i 个个体在第 j 维上的位置, \bar{X}_j 为种群在第 j 维位置的均值.

根据适应度标准差和种群位置方差,计算每个个体的融合评分,将各粒子的融合评分由低到高排序,平均分为 3 个子种群.该评分结合了个体的收敛性和多样性.融合评分越低,个体越适合当前子种群的进化策略.如下所示:

$$f^* = \alpha \cdot \sigma_f + \beta \cdot \sigma_{d,i}, \quad (5)$$

其中 α 和 β 分别为收敛性和多样性指标的权重系数.

在迭代过程中,通过适应度变化率判断是否需要种群进行分区.适应度变化率计算为

$$\Delta F_t = \frac{|F_t - F_{t-1}|}{|F_{t-1}|}, \quad (6)$$

其中 F_t 和 F_{t-1} 分别为粒子在第 t 次和第 $t-1$ 次迭代时的适应度值.通过下式判断粒子陷入局部最优:

$$\Delta F_t < \varepsilon_1, \quad (7)$$

这里 ε_1 为适应度变化率阈值,设置为 0.01.当适应度变化率过小时,表明粒子陷入局部最优,这时分区操作会被触发,鼓励粒子继续探索其他可能解的区域.

2.1.2 不同子群进化策略

对于高性能子群体,通过高斯变异增强局部搜索能力,以更精细地探索解空间中的局部最优区域,该子群体粒子按照下式进行更新:

$$X_i^{\text{new}} = X_i + F_1 \cdot N(0, \sigma) \cdot m_i. \quad (8)$$

其中: X_i 为当前子群体中第 i 个粒子的位置; $N(0, \sigma)$ 为标准正态分布随机变量,表示每个维度的高斯噪

声; m_i 为变异掩码,表示解向量需要变异.

对于中等性能子群体,通过差分进化策略中的变异、交叉和选择操作来平衡全局探索与局部开发能力,在保证种群多样性的同时,引导粒子朝着更优的解进行探索,该子群体粒子按照下式进行更新:

$$X_i^{\text{new}} = X_i + F_2 \cdot (X_{r_1} - X_{r_2}), \quad (9)$$

其中 X_{r_1} 和 X_{r_2} 为从群体中随机选择两个不同个体.

对于低性能子群体,采用柯西变异使得粒子进行更大范围的搜索,从而跳出局部最优,增强算法的全局探索能力,该子群体粒子按照下式进行更新:

$$X_i^{\text{new}} = X_i + F_3 \cdot \text{trnd}(1, d). \quad (10)$$

其中: F_1 、 F_2 、 F_3 为控制变异幅度的缩放因子; $\text{trnd}(1, d)$ 为从柯西分布中采样的随机数, d 为搜索空间的维度.

综上所述,动态种群分区通过对种群的实时调整,在求解复杂的优化问题时能够更好地平衡探索与开发,提高全局搜索能力和局部优化效果.

2.2 快速 CSO 策略

在 CSO 中,每次迭代只更新一半的粒子,群体中剩余一半粒子无法向较优的解决方案学习.因此,群体中高度多样性无法避免,导致算法寻优效率缓慢. PCSCCSO 算法采用三重竞争机制,所有粒子根据适应度值从小到大排序,将种群分为获胜子群、优败子群和劣败子群,并为每个子群设计不同的更新策略,以增强算法的适应性和收敛速度.

2.2.1 获胜粒子更新策略

在每轮迭代中,各子种群粒子与其他子种群粒子竞争,进行位置和速度的更新.然而,经典 CSO 算法中部分获胜者的适应度值并不具有代表性,可能会误导其他粒子的搜索方向.为避免这一问题, PCSCCSO 采用对立学习策略来更新获胜粒子的位置.通过生成与当前获胜粒子位置对立的解,并根据对立解的适应度值与当前解的适应度值比较,若对立解的适应度值更优,则将解更新为对立解.这一策略通过提供多角度的优化视角,防止单一获胜者对其他粒子的搜索产生过度影响,有助于探索当前解的对立区域,从而提高全局搜索能力,加速找到更优解.获胜粒子按照下式进行更新:

$$X'_w(t) = X_{\max} + X_{\min} - X_w(t). \quad (11)$$

其中: X_{\max} 和 X_{\min} 分别为解空间的最大边界和最小边界, $X'_w(t)$ 为当前获胜者的对立位置.

2.2.2 失败粒子更新策略

CSO 算法通过结合获胜者位置和种群平均位置来更新失败者的位置,旨在平衡局部搜索与全局搜

索的能力. 种群平均位置能够代表种群的整体进化水平, 但是并不代表最优解的方向, 在群体适应度差异较大的情况下, 种群平均位置的引导可能会导致搜索过于宽泛, 影响种群的收敛速度. 特别是在求解复杂高维优化问题时, 难以达到预期的寻优效果. 为减少种群平均位置的干扰, 使得算法更精确地指导粒子朝着当前解空间中的较优区域进行搜索, 提高粒子更新的准确性, 提出了优败粒子仅向获胜子群平均位置学习的策略, 优败粒子的速度和位置按照如下方式进行更新:

$$V_i(t+1) = \omega R_4(t)V_i(t) + \varphi R_5(t)(\bar{X}_w(t) - X_i(t)), \quad (12)$$

$$X_i(t+1) = X_i(t) + V_i(t+1). \quad (13)$$

其中: $X_i(t)$ 和 $V_i(t)$ 分别为优败粒子的位置向量和速度向量, \bar{X}_w 为获胜子群的平均位置, ω 为记忆部分的权重系数, t 为迭代次数.

为使得劣败粒子从较差的位置跳出, 增加种群的多样性, 将劣败粒子的位置更新为获胜粒子的位置加上小的扰动. 劣败粒子的速度和位置更新为

$$V_j(t+1) = \omega R_6(t)V_j(t) + \varphi R_7(t)(\bar{X}_w(t) - X_j(t)), \quad (14)$$

$$X_j(t+1) = X_w(t) + \alpha V_j(t+1). \quad (15)$$

其中: $X_i(t)$ 和 $V_i(t)$ 分别为优败粒子的位置向量和速度向量; α 为非负实数, 其数值表示粒子变异强度大小. 这一策略有助于增强种群的多样性, 防止粒子群过早收敛, 同时保持了全局搜索能力. 在高维优化问题中, 能够保证种群在搜索过程中具有足够的跳跃能力, 避免过早收敛至局部最优.

通过多层次的粒子更新策略, PCSCCSO 算法能够在保持种群多样性的同时加快收敛速度. 在高维和多峰问题中, 能够更高效地寻找全局最优解. 针对不同类型粒子制定更新策略, PCSCCSO 算法能够有效克服标准 CSO 算法在高维问题中出现的收敛缓慢和过度多样性的问题, 从而提高算法的整体性能.

2.3 自适应收敛速度控制器

在寻优过程中, 粒子群可能会面临过早收敛、局部最优困境以及缺乏足够探索性等问题. 在优化的初期, 粒子通常需要广泛探索解空间, 若收敛速度过快, 则可能会导致粒子群过早地停滞在某些局部最优解附近, 无法发现潜在的更优解. 在优化的后期, 粒子则需要逐步收敛于全局最优解, 避免陷入局部最优解的同时, 确保最终解的精度. 若没有有效的控制策略, 则粒子可能会过度聚集在当前最优解附近,

忽略了全局搜索的重要性, 导致优化的精度和质量受到影响. 为解决上述问题, 本文提出了一种自适应收敛速度控制器.

收敛速度控制器的核心是通过余弦相似度来判断粒子与当前最优解间的相似程度, 余弦相似度计算为

$$\cos(X_k, X_{\text{best}}) = \frac{X_k \cdot X_{\text{best}}}{\|X_k\| \|X_{\text{best}}\|}. \quad (16)$$

其中: X_k 为种群内第 k 个粒子的位置, X_{best} 为当前全局最优解的位置, $\|X_k\|$ 和 $\|X_{\text{best}}\|$ 分别为粒子位置和全局最优解位置的欧几里得范数.

当余弦相似度较高时, 粒子已接近最优解的邻域, 为避免粒子陷入局部最优并继续探索更广泛的解空间, 通过对粒子位置进行随机扰动, 增加粒子的多样性, 从而防止过早收敛. 粒子按照下式进行扰动:

$$X_k^{\text{new}}(t) = X_k(t) + \gamma V_k(t). \quad (17)$$

其中: γ 为扰动系数, $V_k(t)$ 为当前粒子的速度. 相反, 当余弦相似度较低时, 粒子未接近最优解. 若粒子在多次迭代中未能获得显著的适应度提升, 则收敛速度控制器会激活停滞计数. 停滞计数更新如下所示:

$$\text{count}(i) = \begin{cases} 0, & \text{位置已被调整;} \\ \text{count}(i) + 1, & \text{otherwise.} \end{cases} \quad (18)$$

若停滞计数达到阈值 ε_2 , 则停滞计数会对粒子进行扰动, 以避免其陷入局部最优解, 并促进粒子在更广泛的解空间中进行探索. ε_2 根据实验经验设置为 20, 此时粒子按照下式进行扰动:

$$X_k^{\text{new}}(t) = r_1 \cdot \text{bound}. \quad (19)$$

其中: r_1 为生成一个随机位置向量, bound 为解空间的边界. 收敛速度控制器通过调整粒子的搜索行为, 避免过早收敛, 增强了算法的全局探索能力. 当粒子在解空间中长期停滞或与全局最优解的相似度过高时, 控制器将触发扰动或位置重置, 从而维持种群的多样性并加速收敛. 通过这种方式, 收敛速度控制器能够有效地提升算法的性能, 避免陷入局部最优解.

3 稳定性与收敛性分析

基于离散动力系统理论, 通过构建状态转移矩阵和特征值分析, 验证了 PCSCCSO 算法的动态稳定性. 算法通过竞争机制将粒子分为获胜者和失败者两类: 获胜者采用对立学习策略更新, 在增强全局搜索能力的同时维持算法动态平衡; 失败者的位置和速度更新过程主导系统整体稳定性, 因此, 稳定性分析聚焦于失败者. 系统稳定的充要条件为 $|\lambda| < 1$, 通过对状态转移矩阵进行特征值分解, 确定系统参数需要满足的约束条件, 使得状态转移矩阵的谱半

径保持有界, 确保粒子运动轨迹不会发散, 从而维持系统的动态平衡特性。

在系统稳定性保证的基础上, 进一步构建了获胜粒子平均位置与全局最优解间的差分向量, 采用Z变换将离散迭代系统转化为复平面传递函数模型, 结合朱利稳定判据对系统极点分布进行严格分析. 通过理论推导, 在满足稳定性条件的系统参数约束下, 系统传递函数的所有极点均位于单位圆内. 该数学特性保证算法解的极限存在且唯一, 最终确保粒子群能够收敛至全局最优解。

4 实验研究

为验证所提出 PCSCCSO 算法的寻优效率和收敛能力, 选用 9 个基准测试函数作为算法的测试函数, 如表 1 所示. 将所提出 PCSCCSO 算法与 CSO^[7]、PSO^[1]、MCSO^[9]、ImCSO^[15]、LCSO^[20]、ECSO^[19] 以及 WLCSSODGM^[14] 进行性能对比. 其他 7 种对比算法的参数设置与相应的初始文献一致。

表1 测试函数

测试函数	名称	搜索范围
f_1	Sphere Function	[-100, 100]
f_2	Schwefel Problem221	[-100, 100]
f_3	Rosenbrock Function	[-30, 30]
f_4	Griewank Function	[-600, 600]
f_5	Ackley Function	[-32, 32]
f_6	Alpine Function	[-10, 10]
f_7	RotateAckley Function	[-32, 32]
f_8	RotateGriewank Function	[-600, 600]
f_9	RotateRastrigin Function	[-5.12, 5.12]

本文设置种群粒子数量为 120, 维度为 1000, 算法最大评价次数为 1000. 实验结果为各算法在同一测试函数上独立运行 30 次的统计结果. 算法运行环境为 Intel i7-8750H CPU 220 GHz, 16 GB 内存, Windows 11 操作系统, Matlab 2024a.

测试函数 ($f_1 \sim f_3$) 为单峰函数, ($f_4 \sim f_6$) 为多峰函数, ($f_7 \sim f_9$) 为旋转多峰函数, 通过打破测试函数的对称性, 旋转能够有效地增加函数的复杂性, 使得优化算法不仅需要有较强的局部搜索能力, 还需要有全局搜索的能力, 能够更好地评估高维优

化算法的性能。

4.1 参数 ω 和 φ 对系统的影响分析

为研究惯性权重 ω 和控制因子 φ 对算法收敛速度和收敛精度的影响, 将所提出算法在 Schwefel221 测试函数上进行优化, 对粒子的惯性权重 ω 和控制因子 φ 进行网格搜索实验, 通过对比种群在不同参数组合下的最终适应度, 反映不同的收敛精度水平。

为验证参数组合对于算法收敛速度的影响, 统计算法运行精度达到 e^{-10} 时所需要的迭代次数, 绘制收敛速度热力图, 如图 2 所示。

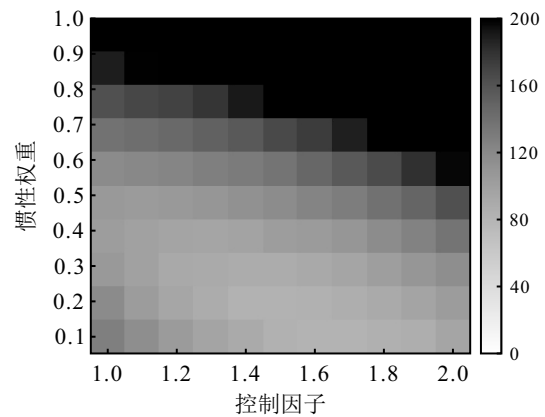


图2 不同参数组合收敛速度热力图

由图 2 可见, 收敛速度最快的区域集中在 $\omega \in [0.1, 0.2]$ 和 $\varphi \in [1.5, 1.7]$ 的范围内, ω 和 φ 分别取 0.1 和 1.6 时, 算法的收敛速度最快. 表 2 为适应度较好时截取的两个参数的取值范围, 评价次数为 200.

由表 2 的实验结果可知: 当 $\omega \in [0.1, 0.3]$ 且 $\varphi \in [1.5, 1.7]$ 时, 算法具有更高的收敛精度; 当 ω 和 φ 分别取 0.1 和 1.6 时, 算法的收敛精度最高. 综合图 2 和表 2 的实验结果, 最优参数区域的收敛速度和精度表现一致, 表明算法在该区域内具有较好的全局搜索能力, 能够快速找到全局最优解, 最优参数组合为 $\omega = 0.1$ 和 $\varphi = 1.6$.

4.2 实验结果和分析

为直观展示各算法性能差距, 图 3 给出了各算法在 9 个测试函数上的收敛曲线. 所提出算法与其他改进算法在 9 个测试函数上的实验数据如表 2 所示. 每种算法的性能通过平均适应度值、最小适应度值和标准差来衡量, 其中最佳值用粗体表示。

表2 不同参数组合下的适应度值

ω/φ	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
0.1	3.34e-21	4.97e-23	1.74e-25	1.10e-26	4.37e-28	5.80e-28	9.36e-27	1.70e-25
0.2	1.22e-23	7.55e-26	8.04e-27	3.21e-27	5.04e-27	2.44e-26	4.90e-25	8.16e-24
0.3	6.14e-25	2.05e-25	9.88e-26	2.73e-25	1.80e-24	2.39e-23	4.69e-22	6.54e-21
0.4	1.81e-23	2.43e-23	9.46e-23	3.97e-22	2.72e-21	3.30e-20	5.04e-19	1.43e-17
0.5	5.14e-21	2.59e-20	5.30e-20	2.72e-19	4.55e-18	4.22e-17	6.57e-16	8.77e-15

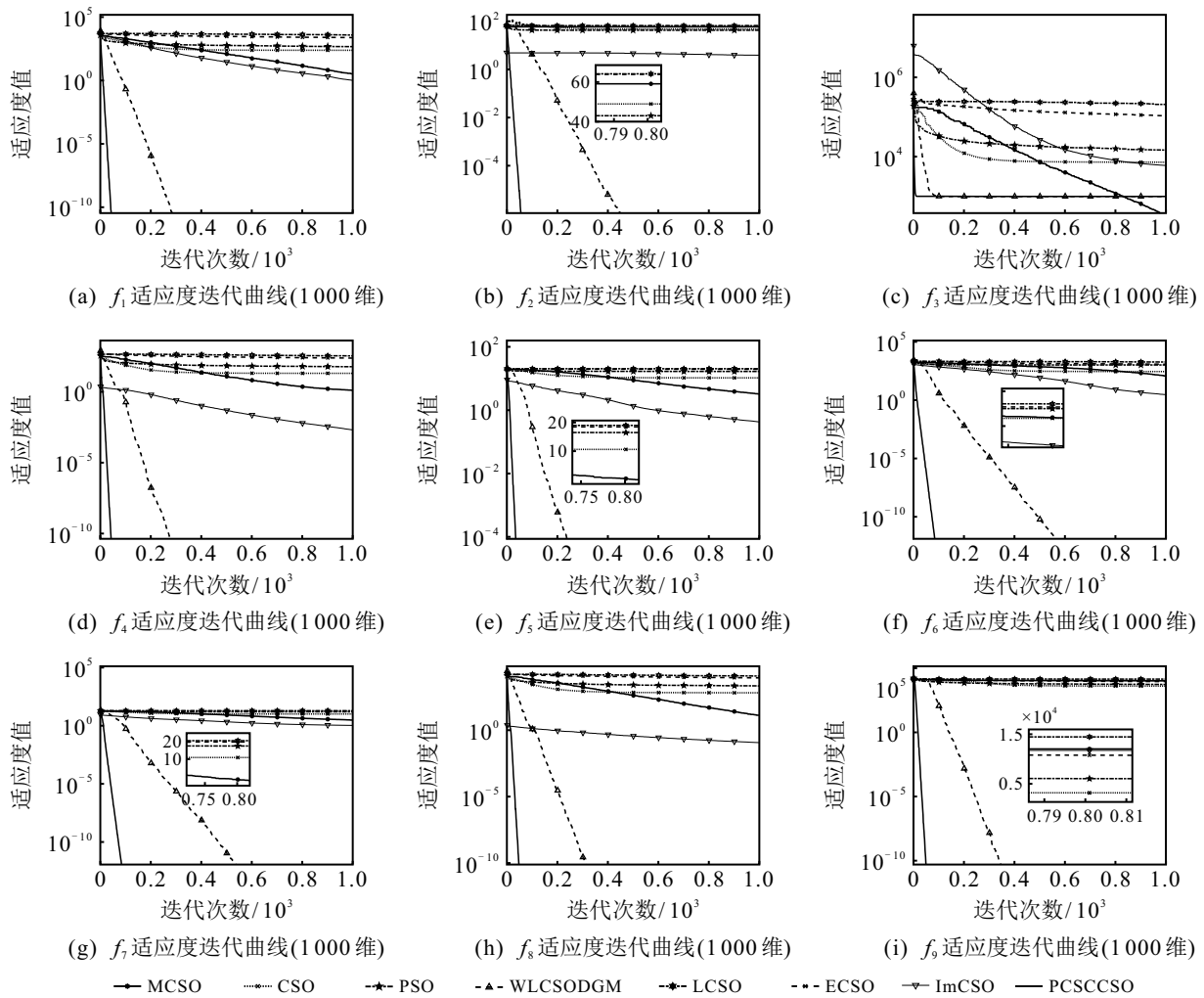


图3 各测试函数最优适应度迭代曲线

由图3可见,PCSCCSO 算法的收敛速度明显优于其他算法,体现了快速CSO策略在提升搜索效率方面的有效性,更精确地指导粒子向解空间较优区域进行搜索,避免对劣解空间过度探索.收敛速度控制器会根据粒子与当前最优粒子的余弦相似度而工作,有效避免陷入局部最优解,从而探索更广的搜索空间.在函数 f_3 上:PCSCCSO在迭代次数 $t < 840$ 时的收敛速度优于其他对比算法;当 $t > 840$ 后,PCSCCSO的收敛速度不如MCSO,但是仍然优于其他5种对比算法.在函数 f_1 、 f_2 上和 $f_4 \sim f_9$ 上:除PCSCCSO和WLCSODGM,其他对比算法均不同程度陷入了局部最优,而PCSCCSO的收敛速度比WLCSODGM更快,进化趋势更优.为进一步验证所提出算法在解决高维优化问题时的收敛精度,表3给出了各对比算法在9个标准测试函数上的实验结果.

由表3可知,PCSCCSO在8个测试函数上获得了更优的实验结果.在单峰测试函数 f_1 和 f_2 上,PCSCCSO的实验结果均优于其他7种对比算法;在函数 f_3 上,PCSCCSO的最小值和标准差相较于

WLCSODGM略差,但是优于其他6种对比算法. f_3 函数的全局最优解位于一个狭长的、弯曲的区域内,因此,要求优化算法能够沿着这个狭窄的通道精确地进行搜索.收敛速度控制器的扰动会导致粒子跳过较为精细的局部解,导致最终优化结果的精度不高,过多的扰动会导致粒子偏离最优解路径,从而降低收敛效率.在多峰测试函数 f_4 上,PCSCCSO和WLCSODGM实验结果均收敛至全局最优解,优于其他6种对比算法;在函数 f_5 上,只有PCSCCSO实验结果收敛至全局最优解;在函数 f_6 上,PCSCCSO实验结果优于其他7种对比算法;在旋转多峰函数 $f_7 \sim f_9$ 上:PCSCCSO均收敛至全局最优解,WLCSODGM在函数 f_8 和 f_9 上的实验结果也收敛至全局最优,均优于其他6种对比算法,但是PCSCCSO的收敛速度更快.以上实验结果表明,PCSCCSO在处理1000维复杂优化问题时,具有更高的稳定性和收敛精度,整体性能更为优越.

4.3 快速CSO与收敛速度控制器对优化性能的影响

为进一步表明快速CSO策略和收敛速度控制器的重要作用,选取9个有代表性的测试函数.实验

表3 各对比算法在9个标准测试函数上的实验结果(1000维)

函数	指标	MCSO	CSO	PSO	WLCSDGM	LCSO	ECSO	ImCSO	PCSCCSO
f_1	best	1.35e+02	3.98e+02	5.75e+02	2.02e-55	2.85e+03	2.45e+03	9.48e+00	1.01e-158
	mean	1.48e+02	4.40e+02	7.11e+02	1.16e-52	3.47e+03	2.77e+03	1.50e+01	4.79e-116
	std	9.08e+00	2.99e+01	6.54e+01	1.16e-52	2.49e+02	1.85e+02	2.83e+00	2.54e-115
f_2	best	6.09e+00	4.82e+01	3.91e+01	1.83e-21	6.39e+01	6.14e+01	3.84e+00	4.28e-71
	mean	6.75e+00	5.29e+01	4.39e+01	3.27e-19	6.64e+01	6.35e+01	4.14e+00	5.26e-55
	std	3.80e-01	1.98e+00	1.65e+00	4.54e-19	9.10e-01	8.97e-01	1.16e-01	2.60e-54
f_3	best	3.76e+05	1.09e+04	1.59e+04	9.90e+02	1.56e+05	9.47e+04	9.20e+03	9.90e+02
	mean	5.29e+05	1.25e+04	2.02e+04	9.90e+02	2.01e+05	1.21e+05	1.08e+04	9.91e+02
	std	8.01e+04	9.08e+02	2.96e+03	6.87e-02	1.62e+04	1.17e+04	8.75e+02	1.13e-01
f_4	best	2.35e-01	3.42e+01	5.34e+01	0.00e+00	2.64e+02	2.21e+02	2.18e-02	0.00e+00
	mean	2.57e-01	4.05e+01	6.73e+01	0.00e+00	3.21e+02	2.50e+02	2.90e-02	0.00e+00
	std	1.61e-02	2.52e+00	7.31e+00	0.00e+00	2.08e+01	1.80e+01	3.84e-03	0.00e+00
f_5	best	3.12e+00	1.19e+01	1.64e+01	3.55e-15	2.02e+01	1.94e+01	1.40e+00	0.00e+00
	mean	3.27e+00	1.29e+01	1.73e+01	3.55e-15	2.03e+01	1.96e+01	1.60e+00	0.00e+00
	std	6.70e-02	3.14e-01	4.21e-01	0.00e+00	9.21e-02	1.31e-01	9.68e-02	0.00e+00
f_6	best	5.56e+02	3.69e+02	9.19e+02	4.37e-28	1.70e+03	1.28e+03	9.63e+00	2.89e-73
	mean	6.33e+02	4.18e+02	1.07e+03	5.14e-27	1.89e+03	1.44e+03	1.28e+01	1.36e-57
	std	2.56e+01	1.91e+01	8.34e+01	4.75e-27	1.09e+02	6.78e+01	1.96e+00	5.66e-57
f_7	best	3.27e+00	1.25e+01	1.64e+01	3.55e-15	2.00e+01	1.94e+01	1.98e+00	0.00e+00
	mean	3.37e+00	1.31e+01	1.71e+01	3.55e-15	2.03e+01	1.96e+01	2.16e+00	0.00e+00
	std	4.17e-02	2.94e-01	3.33e-01	0.00e+00	9.87e-02	1.38e-01	9.12e-02	0.00e+00
f_8	best	7.42e-01	1.25e+03	1.66e+03	0.00e+00	9.94e+03	8.21e+03	1.46e-01	0.00e+00
	mean	7.70e-01	1.42e+03	2.39e+03	0.00e+00	1.16e+04	9.32e+03	1.71e-01	0.00e+00
	std	2.01e-02	9.09e+01	2.83e+02	0.00e+00	9.27e+02	6.08e+02	1.08e-02	0.00e+00
f_9	best	1.15e+04	4.91e+03	5.63e+03	0.00e+00	1.35e+04	9.35e+03	1.04e+04	0.00e+00
	mean	1.17e+04	5.17e+03	6.36e+03	0.00e+00	1.42e+04	9.92e+03	1.06e+04	0.00e+00
	std	1.08e+02	1.46e+02	5.68e+02	0.00e+00	3.22e+02	3.41e+02	9.43e+01	0.00e+00

比较4种算法,分别为MCSO、PCSCCSO、MCSO-S1和MCSO-S2.在MCSO的基础上,改进粒子更新策略,实现MCSO-S1.在MCSO的基础上,引入收敛速度控制器,实现MCSO-S2.实验结果如表4所示.

表4实验结果表明,MCSO-S1在测试函数Schwefel221、Sphere、Schwefel12、Alpine和Zakharov中的最优适应度值优于MCSO与MCSO-S2,由于快速CSO策略,PCSCCSO的表现最好.这些函数的特点使得粒子能够在全局搜索与局部开发间保持较好的平衡.收敛速度控制器通过减缓粒子过早收敛的速度,能够在复杂的局部极值区域提供足够的全局探索能力,从而提高了最终找到全局最优解的概率.在Griewank、Ackley和Rastrigin中,MCSO-S1和PCSCCSO均能够收敛至全局最优解,进一步展现了快速CSO在多峰问题上的高效性.而在Quartic中,MCSO-S1效果最好,Quartic测试函数有多个局部极小值,且存在平坦区域,收敛速度控制器的加入反而减缓了算法的收敛速度,导致粒子在平坦区域无法有效地搜索到更优解,影响了算法的性能.综合对比MCSO-S1、MCSO-S2与PCSCCSO实验数据可知,MCSO-S1对于所提出算法的影响更

表4 快速CSO和收敛速度控制器策略的有效性验证

函数	MCSO	MCSO-S1	MCSO-S2	PCSCCSO
Schwefel221	8.84e+00	1.02e-84	6.42e-08	1.65e-93
Sphere	1.23e+01	3.82e-161	4.86e-15	1.42e-189
Quartic	5.70e+04	3.34e-07	4.01e-02	3.70e-03
Schwefel12	3.04e+05	4.45e-162	2.98e-07	3.07e-167
Griewank	2.10e-03	0.00e+00	0.00e+00	0.00e+00
Ackley	1.22e+00	0.00e+00	1.98e-09	0.00e+00
Rastrigin	1.07e+04	0.00e+00	6.70e-13	0.00e+00
Alpine	1.32e+02	6.50e-86	1.21e-07	1.71e-100
Zakharov	2.80e+04	3.14e-162	3.74e-05	0.00e+00

为显著,验证了快速CSO策略在CSO算法根本创新方面的有效性.综上所述,快速CSO策略与收敛速度控制器的结合能够在开发与勘探间实现良好的平衡,可有效提高算法的性能.

4.4 统计分析

为科学地验证PCSCCSO算法与其他对比算法间的性能差异,采用 t 检验作为统计分析方法,从而为进一步评估该算法的优越性提供统计依据. t 检验实验数据如表5所示, t 检验计算公式为

表5 各对比算法与 PCSCCSO 算法的 t 检验结果

算法	MCSO	CSO	PSO	WLCSODGM	LCSO	ECSO	ImCSO
f_1	-5.96e+01	-4.49e+01	-3.14e+01	-2.73e+00	-6.21e+01	-4.85e+01	-7.93e+00
f_2	-5.97e+01	-1.07e+02	-9.44e+01	-2.88e+00	-3.37e+02	-1.79e+02	-1.22e+02
f_3	1.59e+02	-4.91e+01	-3.09e+01	3.46e+01	-3.93e+01	-2.81e+01	-3.89e+01
f_4	-7.61e+01	-4.97e+01	-3.77e+01	N/A	-4.78e+01	-5.87e+01	-5.44e+00
f_5	-1.20e+02	-1.48e+02	-2.00e+02	6.55e+04	-5.05e+02	-3.96e+02	-7.84e+00
f_6	-3.28e+02	-8.05e+01	-5.01e+01	-4.00e+00	-8.91e+01	-1.05e+02	-6.94e+00
f_7	-9.18e+01	-1.41e+02	-2.23e+02	N/A	-9.46e+02	-4.26e+02	-4.07e+01
f_8	-7.40e+01	-4.18e+01	-3.85e+01	N/A	-5.51e+01	-5.30e+01	-3.92e+01
f_9	-5.86e+02	-1.09e+02	-6.98e+02	N/A	-2.22e+02	-1.13e+02	-3.70e+02

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2 - \sigma_2^2}{N}}}, \quad (20)$$

其中 μ_1 、 μ_2 和 σ_1 、 σ_2 分别为两种算法的适应度平均值和标准差。

由表 5 的 t 检验结果可知, 在统计意义上, PCSCCSO 算法的性能显著优于其他对比算法。为进

一步验证结论的稳健性, 采用非参数的 Wilcoxon 符号秩检验, 对 PCSCCSO 与各对比算法在多个测试函数上的配对结果进行统计分析, 实验结果如表 6 所示。在 63 组对比中, 共有 59 组的 $p < 0.5$, 表明 PCSCCSO 与其他对比算法的性能存在显著差异, 进一步验证了结论的可靠性。

表6 各对比算法与 PCSCCSO 算法的 Wilcoxon 符号秩检验结果 (p 值)

算法	MCSO	CSO	PSO	WLCSODGM	LCSO	ECSO	ImCSO
f_1	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07
f_2	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07
f_3	1.19e-07	1.19e-07	1.19e-07	2.70e-05	1.19e-07	1.19e-07	1.19e-07
f_4	1.19e-07	1.19e-07	1.19e-07	1.00e+00	1.19e-07	1.19e-07	1.19e-07
f_5	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07
f_6	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07
f_7	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07	1.19e-07
f_8	1.19e-07	1.19e-07	1.19e-07	1.00e+00	1.19e-07	1.19e-07	1.19e-07
f_9	1.19e-07	1.19e-07	1.19e-07	1.00e+00	1.19e-07	1.19e-07	1.19e-07

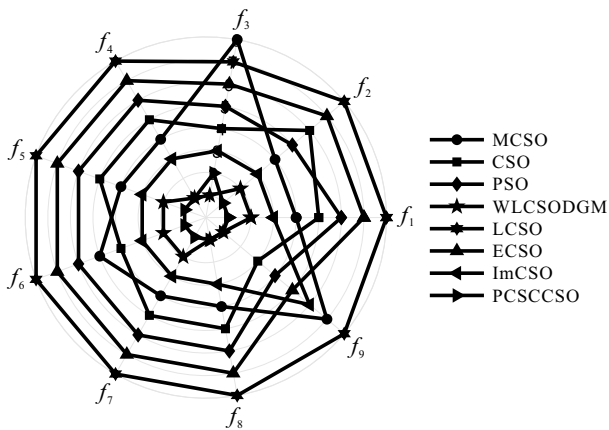


图4 各算法在不同测试函数上排名的雷达图

图 4 直观地展示了 PCSCCSO 算法与比较算法在 1000 维度上寻优能力的排名。图 4 中, PCSCCSO 算法处于最内层, 表明 PCSCCSO 平均排名最高, 与其他改进算法相比, 所提出算法更具竞争力, 寻优能力更强。

所提出 PCSCCSO 算法在大多数测试函数上均

表现出了好的效果, 虽然算法结构略为复杂, 但是快速 CSO 策略帮助加快了收敛速度, 使得粒子迅速向更精确的解空间搜索, 且动态分区策略和自适应收敛速度控制器增加了种群的多样性, 大幅提高了算法的收敛效率和收敛精度。

5 结论

针对复杂高维优化问题, 本文提出了一种基于动态分区和收敛速度控制器的改进竞争群优化算法。首先, 种群在优化过程中采用了动态分区策略, 将种群划分为 3 个子种群, 确保每个子种群能够独立进化, 同时, 通过三重竞争机制促进了不同子种群间的交流和协作, 降低了陷入局部最优解的风险, 增强了算法的收敛性和搜索效率; 然后, 本文提出了失败粒子仅向获胜子群平均位置学习的快速 CSO 策略, 更精确地指导粒子搜索, 防止陷入局部最优, 加快了算法的收敛速度; 接着, 引入收敛速度控制器进一步提高了算法的适应性, 在适应度变化率较小或出现停

滞时, 控制器通过余弦相似度和停滞计数对粒子进行调整, 避免了过早收敛, 增强了全局搜索能力; 最后, 通过对多个高维复杂优化问题的实验验证了所提出算法在收敛速度、求解精度以及鲁棒性方面均优于其他对比算法, 显示出了较强的性能优势. 然而, 所提出粒子学习策略中惯性权重和控制因子参数设定依赖于实验经验, 后续工作将引入参数的自适应调整, 以进一步增强算法的适应性.

参考文献 (References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proceedings of ICNN'95 — International Conference on Neural Networks. Perth, 1995: 1942-1948.
- [2] Ballerini L. Particle swarm optimization in 3D medical image registration: A systematic review[J]. *Archives of Computational Methods in Engineering*, 2025, 32(1): 311-318.
- [3] Pradhan A, Das A, Bisoy S K. Modified parallel PSO algorithm in cloud computing for performance improvement[J]. *Cluster Computing*, 2025, 28(2): 131.
- [4] 管军, 叶双慧, 易文俊. 基于 PSO-hpRPM 算法的滑翔制导炮弹弹道优化[J]. *控制与决策*, 2025, 40(5): 1733-1741.
(Guan J, Ye S H, Yi W J. Trajectory optimization of glide guidance projectile based on PSO-hpRPM hybrid algorithm[J]. *Control and Decision*, 2025, 40(5): 1733-1741.)
- [5] Guo H C, Huang R H, Cheng S Q. Scheduling optimization based on particle swarm optimization algorithm in emergency management of long-distance natural gas pipelines[J]. *PLoS One*, 2025, 20(2): e0317737.
- [6] Houssein E H, Gad A G, Hussain K, et al. Major advances in particle swarm optimization: Theory, analysis, and application[J]. *Swarm and Evolutionary Computation*, 2021, 63: 100868.
- [7] Cheng R, Jin Y C. A competitive swarm optimizer for large scale optimization[J]. *IEEE Transactions on Cybernetics*, 2015, 45(2): 191-204.
- [8] Chu W, Gao X G, Sorooshian S. A new evolutionary search strategy for global optimization of high-dimensional problems[J]. *Information Sciences*, 2011, 181(22): 4909-4927.
- [9] Mohapatra P, Nath D K, Roy S. A modified competitive swarm optimizer for large scale optimization problems[J]. *Applied Soft Computing*, 2017, 59: 340-362.
- [10] Chauhan D, Shivani, Cheng R. Competitive swarm optimizer: A decade survey[J]. *Swarm and Evolutionary Computation*, 2024, 87: 101543.
- [11] Wang X Y, Zhang B R, Wang J, et al. A cluster-based competitive particle swarm optimizer with a sparse truncation operator for multi-objective optimization[J]. *Swarm and Evolutionary Computation*, 2022, 71: 101083.
- [12] Musikawan P, Kongsorot Y, Muneesawang P, et al. An enhanced obstacle-aware deployment scheme with an opposition-based competitive swarm optimizer for mobile WSNs[J]. *Expert Systems with Applications*, 2022, 189: 116035.
- [13] Xiong G J, Zhang J, Shi D Y, et al. A simplified competitive swarm optimizer for parameter identification of solid oxide fuel cells[J]. *Energy Conversion and Management*, 2020, 203: 112204.
- [14] 张伟, 魏万峰, 黄卫民. 求解高维优化问题的 ITCSO 算法[J]. *控制与决策*, 2024, 39(2): 449-457.
(Zhang W, Wei W F, Huang W M. ITCSO algorithm for solving high dimensional optimization problem[J]. *Control and Decision*, 2024, 39(2): 449-457.)
- [15] Chen X, Tang G W. Solving static and dynamic multi-area economic dispatch problems using an improved competitive swarm optimization algorithm[J]. *Energy*, 2022, 238: 122035.
- [16] Deng H B, Peng L Z, Zhang H B, et al. Ranking-based biased learning swarm optimizer for large-scale optimization[J]. *Information Sciences*, 2019, 493: 120-137.
- [17] Bilal, Pant M, Zaheer H, et al. Differential evolution: A review of more than two decades of research[J]. *Engineering Applications of Artificial Intelligence*, 2020, 90: 103479.
- [18] Xiong G J, Shi D Y. Orthogonal learning competitive swarm optimizer for economic dispatch problems[J]. *Applied Soft Computing*, 2018, 66: 134-148.
- [19] Wang X Y, Zhang K, Wang J, et al. An enhanced competitive swarm optimizer with strongly convex sparse operator for large-scale multiobjective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2022, 26(5): 859-871.
- [20] Borowska B. Learning competitive swarm optimization[J]. *Entropy*, 2022, 24(2): 283.
- [21] Tian Y, Zheng X T, Zhang X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. *IEEE Transactions on Cybernetics*, 2020, 50(8): 3696-3708.
- [22] Liu J J, You Y, Gao S Q, et al. Identification of the protein glycation sites in human myoglobin as rapidly induced by d-ribose[J]. *Molecules*, 2021, 26(19): 5829.

作者简介

张伟 (1978-), 女, 教授, 博士, 主要研究方向为智能特征建模、神经网络结构优化、污水处理过程优化控制, E-mail: zwei1563@126.com;

伊杰昌 (2000-), 男, 硕士生, 主要研究方向为智能优化算法, E-mail: m15239151775@163.com.