

基于多种群多策略的竞争粒子群算法

李媛媛,李文博,尚志豪

(大连交通大学 软件学院,辽宁 大连 116028)

摘要:针对标准粒子群算法遇到的易陷入局部最优、收敛差、求解精度低等问题,提出了多种群多策略竞争粒子群优化算法.新算法将每一代粒子根据适应度排序,将其划分为不同的子种群,并引入非线性 Logistic 混沌映射权重、收缩因子和混合高斯-柯西扰动机制来更新这些子种群.使用不同的粒子更新方式平衡了算法整个时期的全局搜索和局部开发能力,从而加快了收敛速度.最后,将多种群多策略竞争粒子群优化算法与标准粒子群算法和其它优化算法在 11 个测试函数上进行对比,结果表明,新算法在跳出局部最优解、和寻优精度方面显著优于标准粒子群算法,且有更快的收敛速度.在寻优能力和算法稳定性上大幅度强于其它对比算法.

关键词:种群划分;多策略;混合变异;混沌映射

中图分类号:TP311 **文献标志码:**A **文章编号:**1672-8513(2024)01-0079-08

粒子群优化算法是一种基于群体智能的优化算法,它通过模拟智能群体中每个个体的行为来解决优化问题.主要应用在工程设计^[1]、控制工程^[2]、路径规划^[3]、计算机视觉^[4]等方面.文献[1]为了在锻坯过程中找到最佳的工艺参数,改善传统数值模拟方法的不足,采用粒子群算法对参数进行优化,从材料本身和锻压成形节能的角度出发,采用该算法找到最合适的参数结果.文献[2]使用非线性 PID (NLPID) 控件取代了传统的 PID 控件,提出了一种基于广义对立学习算法的粒子群优化算法来优化 NLPID 控制器,成功抑制了系统超调.文献[4]提出了一种灵活的卷积自编码器,利用粒子群优化设计了一种架构发现方法,该方法能够自动搜索所提出的柔性卷积自编码器的最优架构,无需任何人工干预,大大减少计算资源,解除了传统卷积自编码器对卷积层和池化层数量的禁锢,并证明了加入粒子群算法后的新的图像分类算法远优于其他同类算法.

标准粒子群优化算法(standarding particle swarm optimization, SPSO)最初由 Shi 和 Eb-erhart 等^[5]在 1998 年提出,它拥有惯性权重,是为了解决原始 PSO 算法易陷入局部最优值而进行的改进.带有压缩因子的粒子群优化算法在 1999 年由 Clerc 等^[6]提出,目的是在帮助粒子跳脱局部最优值的同时加快整个优化过程的收敛速度.这两种改进方式是针对原始 PSO 最经典的改进方法.其他的改进算法通常都是在此基础上进行改进,方法包括:改变粒子拓扑结构、与其他算法结合、引入新的机制、或是对参数进行修改.例如:为了克服传统的 Pareto 最优前沿形状变化分解方法的不稳定性,Zheng Jinhua 等^[7]提出了 1 种基于对抗分解和邻域演化的动态多目标粒子群优化算法;针对存在多个 pareto 最优解且适应度值相同的多模态多目标优化问题,Liang Jing 等^[8]提出了一种具有自组织机制的多目标粒子群优化算法;Liu Yaxian 等^[9]通过将强化学习算法与粒子群算法结合起来,得到了自适应参数;为了使算法寻优过程中更轻松地跳出局部最优值,徐利锋等^[10]在带有收缩因子的粒子群算法基础上引入了多级扰动机制.

粒子群优化算法的优势在于它可以快速收敛到最优解,同时具有较好的全局搜索能力.在众多应用中,粒子群优化算法已经取得了良好的效果,但是在实际应用中还是会出现易陷入局部最优^[11-12]、收敛性差^[13-14]、求解精度低^[15]等问题.为了减少这些问题对算法的影响,作者对标准粒子群算法进行改进,提出基于 Logistic 混沌映射权重及混合高斯、柯西扰动变异,同时使用了收缩因子的多种群多策略竞争粒子群算法(multi-swarm multistrategy competitive particle swarm optimization, MMCPSO).为了获得比标准粒子群算法更

收稿日期:2023-01-12.

基金项目:辽宁省教育厅项目(LJKZ0481, LJKMZ20220838).

作者简介:李媛媛(1980-),女,博士,副教授.主要从事智能算法、人工智能研究.

好的寻优性能,作者将每一代的粒子群划分为不同的子种群,并使用不同的更新机制来更新这些子种群,从而使粒子的全局搜索能力和局部开采能力在寻优过程中达到平衡。

1 标准粒子群算法

在粒子群算法中,每个粒子的位置代表了给定问题的潜在解决方案,并使用适应度函数来评判当前位置的优劣.群体中的粒子会通过信息共享机制来更新自身的速度和位置,从而更新整个群体.群体在迭代过程中不断追寻最优粒子,在解空间内进行搜索运动,从而逐渐从无序向有序演变,最终达到在限制条件下求得待解决问题的帕累托最优解的目的。

粒子速度和位置更新公式:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_1(p_{bestij}(t) - x_{ij}(t)) + c_2r_2(g_{bestij}(t) - x_{ij}(t)). \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \quad (2)$$

其中, $v_{ij}(t)$ 表示第 t 代粒子 i 在第 j 维度上的速度, $wv_{ij}(t)$ 部分表示上一代历史速度对当前速度的影响, 惯性权重 w 用来调节此影响的大小, 从而调节粒子在解空间的搜寻范围, 使粒子全局搜索和局部开采能力达到平衡. $c_1r_1(p_{bestij}(t) - x_{ij}(t))$ 为粒子的自我认知部分, $c_2r_2(g_{bestij}(t) - x_{ij}(t))$ 为社会认知部分. $p_{bestij}(t)$, $g_{bestij}(t)$ 分别为粒子的个体历史最优位置和全局历史最优位置. 加速度因子 c_1, c_2 分别用来调节粒子向自己历史最优位置和全局最优位置学习的步长. 随机数 r_1, r_2 都取值 $[0, 1]$ 内, 用以增加粒子搜索的随机性。

2 改进的粒子群算法 MMCP SO

标准 PSO 寻优过程一直伴随着局部搜索能力不够强, 搜索精度差, 处理复杂非线性多峰问题常陷入局部最优等问题. 为了摆脱这个困扰, MMCP SO 根据同代种群粒子的适应度值将粒子分别划为 3 个子种群: 优等子群 (superiors)、普通子群 (ordinaries)、劣等子群 (inferiors); 针对不同子种群粒子的特点分别加入扰动变异、Logistic 混沌映射、收缩因子 3 种不同策略来进行粒子的更新; 不同子种群产生的新一代粒子通过参与适应度值竞争排序后, 更新到不同的子种群; 算法中的每个子种群会通过粒子更新公式, 不同程度的参与引领整个种群更新. 不同于标准 PSO 的所有粒子只对本身历史最优和全局历史最优进行追逐, 这种新的更新策略使整体算法寻优全程拥有较强的全局和局部寻优能力并兼具了易跳出局部最优、保持种群多样性的特性, 弥补了标准 PSO 的不足. 下面将详细介绍 MMCP SO 的种群划分方式和不同子种群的更新策略。

2.1 种群划分

以求最小值问题为例, 在 MMCP SO 中, 每一代的所有粒子按照适应度从低到高进行竞争排序后, 求得当代种群适应度的平均值和标准差. 在求最小值的优化问题中: 取平均适应度一倍标准差内的粒子组成普通子群; 取适应度值小于普通子群的粒子组成优等子群; 取适应度值大于普通子群的粒子为劣等子群. 劣等子群向优等子群和普通子群两个子群按照合理的权重学习更新, 尽快向两个区域靠拢; 普通子群使用带有 w 惯性的更新公式, 平衡普通子群粒子的全局和局部探索能力。

适应度平均值和标准差的计算公式如下:

$$\bar{f} = \frac{\sum_{i=1}^n f_i}{N}. \quad (3)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^n (f_i - \bar{f})^2}. \quad (4)$$

其中, 第 i 个粒子的适应度值为 f_i , 粒子个数为 N , 适应度值的平均值为 \bar{f} , 适应度值的标准差为 σ . 粒子按适应度排序后, 子种群划分方式如图 1 所示。

2.2 优等子群更新策略

优等子群粒子已经获得了较优的适应度值, 所以优等子群进行自我学习更新. 同时, 该种群粒子聚集在局部最优解附近. 为了避免陷入局部最优, 同时又使粒子具备好的局部寻优能力, 作者设计了带有局部开发能力强的高斯变异和具有两翼分布概率且更易跳出局部最优的柯西变异的粒子更新方式。

高斯 - 柯西变异算子 GC 定义式:

$$GC = \alpha G + (1 - \alpha) C, 0 < \alpha < 1. \quad (5)$$

$$G = \text{Gaussian}(0, 1) = \text{rand}(0, 1) \sim N(0, 1). \quad (6)$$

$$C = \text{Cauchy}(0, 1) = \tan\left(\left(\text{rand}(0, 1) - \frac{1}{2}\right) \cdot \pi\right). \quad (7)$$

优等子群粒子更新公式:

$$x_{ij}^S(t+1) = x_{ij}^S(t) \cdot (1 + GC) \cdot n_i. \quad (8)$$

$$n_i = \frac{t_{\max} - t}{t_{\max}}. \quad (9)$$

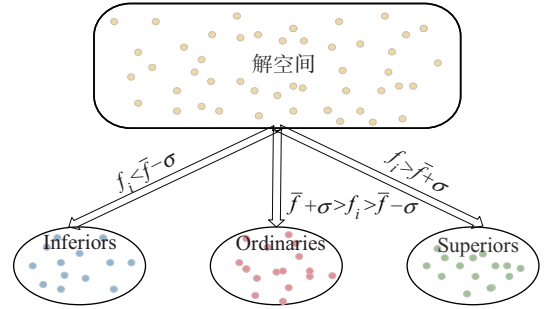


图 1 子种群划分方式

其中, x_{ij}^S 表示优等子群中的粒子 i 在第 j 维度上的位置. GC 为高斯 - 柯西变异算子, G 是根据公式(6)产生的服从高斯分布的随机数, C 是根据公式(7)产生的服从柯西分布的随机数. t_{\max} 为预设的种群最大迭代次数, t 为当前种群迭代次数. 通过 t_{\max} 和 t 求得的 n_i 为控制变异步长的自适应参数, 在优等子群的早期迭代中, 使用较大的变异因子, 较容易让粒子拜托局部最优; 反之, 在迭代后期, 缩小变异因子的取值, 会加速算法收敛. 根据公式(9), 随着迭代次数的增加, 自适应参数 n_i 会变小. 作者通过使用自适应变异步长 n_i 来平衡当前种群粒子的开发和探索能力.

2.3 劣等子群更新策略

劣等子群中的个体通过主要向优等子群学习, 兼顾受种群中心平均值牵引的方式更新. 采用收缩因子对整个更新过程进行压缩, 使劣等子群的粒子能快速脱离劣势区域向优势区域收敛, 同时又能对各个学习因子进行调节, 均衡了该阶段算法的收敛性能, 避免在快速靠近优等子群的过程中丧失了开发能力.

收缩因子 φ 定义式:

$$c = c_1 + c_2 + c_3, c \geq 4. \quad (10)$$

$$\varphi = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|}. \quad (11)$$

劣等子群更新公式:

$$x_{ij}^I(t+1) = \varphi [c_1 \cdot x_{ij}^I(t) + c_2 \cdot (x_{ij}^S(t) - x_{ij}^I(t)) + c_3 \cdot (\bar{f} - x_{ij}^I(t))]. \quad (12)$$

其中, x_{ij}^I 表示劣等粒子的位置. c_1, c_2, c_3 分别为劣等粒子向自身、优等子群和种群平均值的学习因子. 调整 3 个学习因子的参数值可以控制粒子对不同引导部分学习的权重, 在这部分需设置较大的 c_2 值, 达到主要向优等种群学习的意图. 设置收缩因子 φ 将学习因子控制在合理范围内, 为了避免部分学习因子或全部学习因子设置过大, 破坏了收敛的平衡性.

2.4 普通子群更新策略

普通子群粒子处于解空间合理位置范围内, 无明显优劣势, 该子群进化过程中需要平衡算法的勘探和开采能力. SPSO 算法使用的线性递减权重 w 在一定程度上平衡了粒子的全局探索和局部开发, 但线性的调整方式在多维复杂非线性函数的优化过程中常陷入局部最优. 混沌映射作为非线性映射方式的一种, 其产生的随机序列具有良好的空间便利性. 因此, 作者在 SPSO 算法基础上对权重 w 加入 Logistic 混沌映射, 用非线性权重 w_L 对粒子的速度进行更新, 使算法搜索能力均衡的同时又能很好地遍历解空间, 不易陷入局部最优.

Logistic 混沌映射惯性权重 w_L 定义式:

$$r(t+1) = 4r(t)(1 - r(t)), r(0) = \text{rand}(0, 1) \text{ 且 } r(0) \neq \{0, 0.25, 0.75, 1\}. \quad (13)$$

$$w_L(t) = r(t)w_{\min} + \frac{(w_{\max} - w_{\min})t}{t_{\max}}. \quad (14)$$

普通子群粒子更新公式:

$$v_{ij}^O(t+1) = w_L(t)v_{ij}^O(t) + c_1 r_1(p_{\text{best}ij}(t) - x_{ij}^O(t)) + c_2 r_2(g_{\text{best}ij}(t) - x_{ij}^O(t)). \quad (15)$$

$$x_{ij}^O(t+1) = x_{ij}^O(t) + v_{ij}^O(t+1). \quad (16)$$

式(14)中, w_{\max} 等于 0.9, w_{\min} 等于 0.4. v_{ij}^O 为普通子群粒子 i 在第 j 维上的速度, x_{ij}^O 为普通子群中粒子 i 在第 j 维上的位置. 基于混沌映射的惯性权重 w_L 随迭代次数的变化如图 2 所示.

3 实验

3.1 标准函数测试

选取 11 个基本测试函数与标准粒子群算法从寻优精度、寻优速度,跳出局部最优的能力等方面进行对比,来验证 MMCPSO 算法的有效性. 11 个标准测试函数的基本信息由表 1 给出, f_{opt} 为函数最优值. 函数 $f_1 \sim f_6$ 是用来测试算法寻优的快慢和所得解优劣的单峰函数, 函数 $f_7 \sim f_{11}$ 为测试算法跳出局部最优, 避免过早收敛的能力的多峰函数.

3.2 算法性能测试

将 MMCPSO 与 SPSO 进行对比实验, 给予 2 种算法相同的种群大小 $N=90$ 和最大迭代次数 $t_{max}=150$, 令二者在一个 30 维的解空间内对 11 个基本测试函数进行最小值寻优. 在相同的硬件条件下, 运行两个算法 50 次, 记录两种算法 50 次的寻优结果, 分别求取平均值 (Mean) 和标准差 (standard deviation, SD) 作为评价算法性能的指标. 算法参数设置见表 2.

2 种算法对基本测试函数的寻优结果见表 3.

表 1 11 个标准测试函数

Name	Function	Formulation	Range	f_{opt}
f_1	Shpere	$f_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	0
f_2	Schwefel's 2. 22	$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	0
f_3	Schwefel's 1. 2	$f_3 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	0
f_4	Schwefel's 2. 21	$f_4 = \max_i x_i , 1 \leq i \leq n$	$[-100, 100]^n$	0
f_5	Rosenbrock's	$f_5 = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	$[-30, 30]^n$	0
f_6	Step	$f_6 = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]^n$	0
f_7	Ackley's	$f_7 = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i \right) + 20 + e$	$[-32, 32]^n$	0
f_8	Rastrigin's	$f_8 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	0
f_9	Griewank's	$f_9 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^n$	0
f_{10}	Penalized1	$f_{10} = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4),$ $y_i = 1 + \frac{1}{4}(x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^n$	0

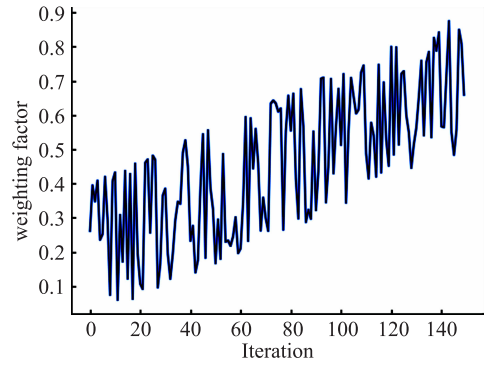


图 2 惯性权重随迭代次数变化图

Name	Function	Formulation	Range	f_{opt}
f_{11}	Penalized2	$f_{10} = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4),$ $y_i = 1 + \frac{1}{4}(x_i + 1),$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^n$	0

表 2 各算法参数表

Algorithm	Parameters
SPSO	$w = 0.65, c_1 = c_2 = 2$
MMCP SO	$w_{min} = 0.4, w_{max} = 0.9, c_1 = c_2 = 2, c_3 = 1$

表 3 2 种算法求解 11 个标准测试函数适应度的平均值和标准差

Name	SPSO		MMCP SO	
	Mean	SD	Mean	SD
f_1	1.776	0.814 1	1.827×10^{-5}	9.542×10^{-5}
f_2	38.86	33.82	1.399×10^{-6}	2.258×10^{-6}
f_3	48.27	17.90	8.529×10^{-12}	2.305×10^{-11}
f_4	2.675	0.654 8	3.710×10^{-9}	1.381×10^{-8}
f_5	444.2	291.7	28.76	0.121 5
f_6	1.687	0.929 0	0.861 1	0.759 9
f_7	4.530	0.820 9	2.546×10^{-5}	1.521×10^{-4}
f_8	172.1	41.30	3.117×10^{-6}	1.227×10^{-5}
f_9	0.106 7	0.036 0	3.479×10^{-9}	1.197×10^{-8}
f_{10}	1.136	0.840 9	0.076 6	0.056 3
f_{11}	1.073	0.465 1	0.089 5	0.055 0

绘制出 FPSO 和 MMCP SO 在求解 11 个标准测试函数时的适应度曲线, 以便更直观的对比观察 2 种算法的求解精度和收敛速度, 如图 3、图 4、图 5 所示。

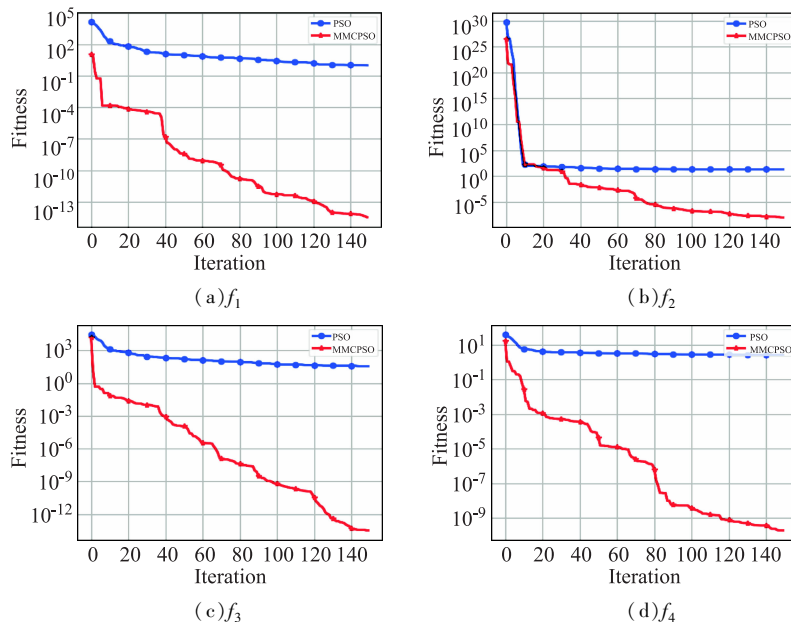


图 3 各算法 $f_1 \sim f_4$ 函数寻优适应度曲线

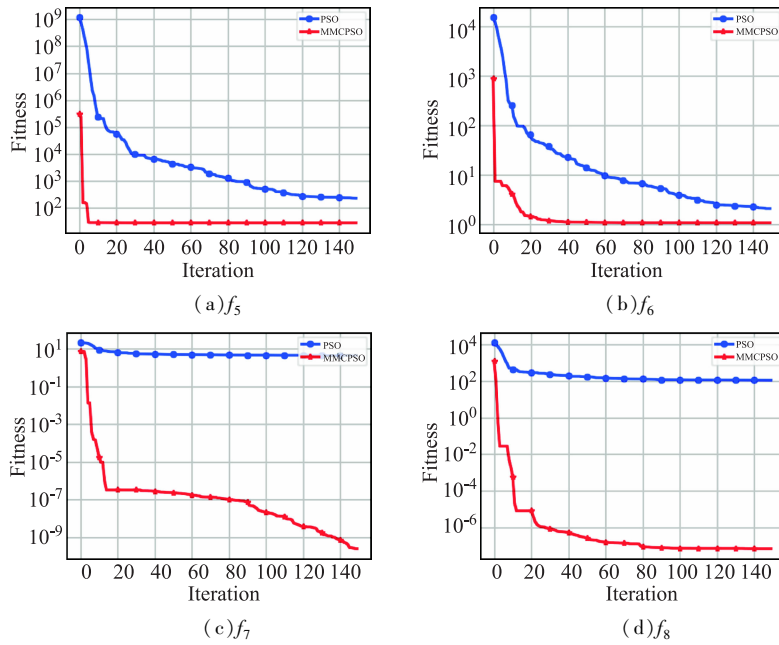


图 4 各算法 $f_5 \sim f_8$ 函数寻优适应度曲线

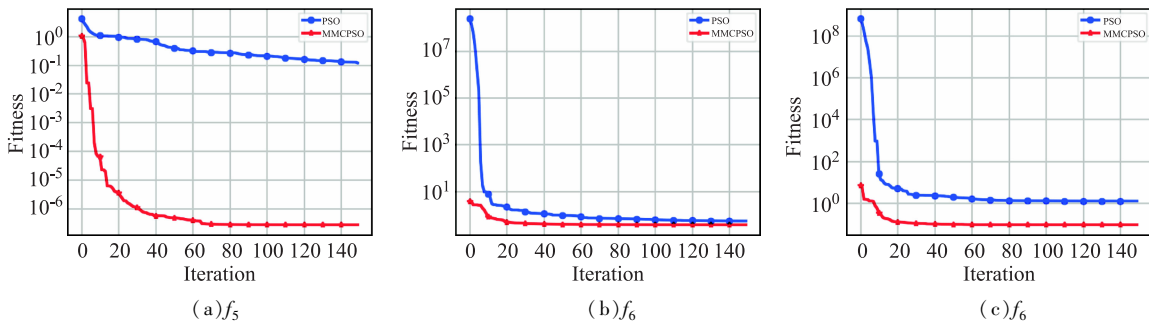


图 5 各算法 $f_9 \sim f_{11}$ 函数寻优适应度曲线

从表 3 可知,在标准测试函数 $f_1 \sim f_4, f_7 \sim f_9$ 上 MMCP SO 算法的寻优精度较 FPSO 算法有着明显的数量级优势,在剩余的测试函数上新算法求得适应度的平均值也更接近函数本身的最优值.从图 3、图 4、图 5 中各个测试函数的适应度曲线可以看出新算法收敛速度更加迅速,又得益于变异机制和混沌映射惯性权重,跳出局部最优解的能力也更强.

3.3 对比实验

为了证明 MMCP SO 算法的优越性,将其与 3 种典型的 PSO 变体进行比较,包括自适应惯性权重的全局 PSO(GPSO - AW)^[16]、动态维度自适应 PSO(DDAPSO)^[17]和局部竞争 PSO(LC - PSO)^[18].具体的参数设置如表 4 所示.为确保用不同算法进行实验比较的公正性,测试函数的维度 D 设置为 30,种群规模 N 设置为 90,最大迭代次数 t_{max} 设置为 150,每个算法独立运行 50 次.

表 5 给出了 MMCP SO 算法和当前 3 种 PSO 在 11 个基准测试函数上的性能比较结果.

表 4 算法参数表

Algorithm	Parameters
GPSO - AW	$w_{min} = 0.4, w_{max} = 0.9, c_1 = c_2 = 2, c_3 = 1$
DDAPSO	$w = 0.65, c_1 = c_2 = 2, c_3 = 1$
LC - PSO	$w_{min} = 0.4, w_{max} = 0.9, c_1 = c_2 = 2, c_3 = 1$
MMCP SO	$w_{min} = 0.4, w_{max} = 0.9, c_1 = c_2 = 2, c_3 = 1$

实验结果分析:

1) 算法寻优能力. 将 MMCP SO 算法与 GPSO - AW、DDAPSO、LC - PSO 3 个算法在多个单峰、双峰函数下

进行测试, MMCP SO 算法表现极佳. 无论是在单峰测试函数 f_1 、 f_3 、 f_4 、 f_5 , 还是多峰测试函数 f_2 、 $f_7 \sim f_9$ 上, 其求得的 Mean 值比其它算法更接近测试函数的最优值. 这说明该算法寻优能力出众.

2) 算法稳定性. 在性能评价指标中的 SD 值也远小于其它算法, 这说明 MMCP SO 算法性能稳定. 美中不足的是在 f_6 和 f_{11} 上 MMCP SO 算法的表现都不如 GPSO - AW 算法.

表 5 不同算法性能比较

Name	对比标准	GPSO - AW	DDAPSO	LC - PSO	MMCP SO
f_1	MEAN	0.152 3	0.356 0	0.057 9	1.767×10^{-5}
	SD	0.008 9	0.285 4	0.142 5	8.827×10^{-5}
f_2	MEAN	3.912 4	0.224 1	2.997	2.617×10^{-6}
	SD	49.769	0.122 0	3.921	2.955×10^{-6}
f_3	MEAN	53.10	23.56	0.6720	4.773×10^{-12}
	SD	0.693 3	32.20	0.334 9	2.467×10^{-12}
f_4	MEAN	5.9125	56.102 3	11.034 1	4.204×10^{-9}
	SD	9.023 4	30.235 2	9.001 2	7.912×10^{-8}
f_5	MEAN	122.7	291.0	112.8	34.44
	SD	76.84	146.4	68.49	0.291 8
f_6	MEAN	1.444	10.41	1.337	5.580
	SD	0.981	8.476	2.990	1.373
f_7	MEAN	9.991 2	0.309 7	5.035 3	2.566×10^{-5}
	SD	8.915 6	0.682 4	6.092 3	2.346×10^{-4}
f_8	MEAN	16.34	10.201 2	0.102 4	2.618×10^{-6}
	SD	9.817	13.276 9	0.081 3	1.343×10^{-5}
f_9	MEAN	1.411×10^{-5}	0.024	0.037	3.485×10^{-8}
	SD	1.997×10^{-5}	0.014	0.071	1.229×10^{-7}
f_{10}	MEAN	0.467 6	0.077 4	1.217	0.067 0
	SD	0.719 3	0.047 8	0.471	0.076 4
f_{11}	MEAN	0.774 9	10.47	6.020 9	2.137
	SD	0.078 0	6.739	12.004 3	1.930

4 结语

设计了 1 种可以根据不同子种群状况, 采用不同更新策略的改进粒子群算法 MM - CPSO. MMCP SO 利用竞争学习机制和收缩因子加快了劣等子群学习速度; 通过引入融合的变异算子和使用自适应变异步长增大了优等子群中粒子局部开发能力和跳出局部最优的概率值; 加入 Logistic 混沌映射惯性权重令普通子群更好地遍历解空间. 新的更新策略有效地避免种群在优化单峰、多峰问题时早熟收敛和无法跳出局部最优解. 改进后的算法在 11 个标准测试函数上的优化表现表明采用不同子种群不同更新策略能够有效取得探索和开发能力的最佳平衡. 下一步的目标是将该算法引入到深度学习中, 帮助神经网络模型选取优秀的初始权重和高效的网络结构模型.

参考文献:

- [1] LI N. Parameter Optimization of the Forging and Forming Process Using Particle Swarm Optimization[J]. International Journal of Mechatronics and Applied Mechanics, 2022, 11: 249 - 258.
- [2] SUN X, LIU N, SHEN R, et al. Nonlinear PID Controller Parameters Optimization Using Improved Particle Swarm Optimization Algorithm for the CNC System[J]. Applied Sciences, 2022, 12, 10269:1 - 14.
- [3] LIAN J, YU W, XIAO K, et al. Cubic spline interpolation - based robot path planning using a chaotic adaptive particle swarm optimization algorithm[J]. Mathematical Problems in Engineering, 2020:1 - 20.
- [4] SUN Y, XUE B, ZHANG M, et al. A particle swarm optimization - based flexible convolutional autoencoder for image classifica-

- tion[J]. IEEE transactions on neural networks and learning systems, 2018, 30(8): 2295 – 2309.
- [5] SHI Y, EBERHART R. A modified particle swarm optimizer[C]//1998 IEEE international conference on evolutionary computation proceedings, IEEE, 1998: 69 – 73.
- [6] CLETC M, KENNEDY J. The particle swarm – explosion, stability, and convergence in a multidimensional complex space[J]. IEEE transactions on Evolutionary Computation, 2002, 6(1): 58 – 73.
- [7] ZHENG J, ZHANG Z, Zou J, et al. A dynamic multi – objective particle swarm optimization algorithm based on ad – versarial decomposition and neighborhood evolution[J]. Swarm and Evolutionary Computation, 2022, 69: 1 – 51.
- [8] LIANG J, GUO Q, YUE C, et al. A self – organizing multi – objective particle swarm optimization algorithm for mu – ltimodal multi – objective problems[C]//International Conference on Swarm Intelligence. Springer, Cham, 2018: 550 – 560.
- [9] LIU Y, LU H, CHENG S, et al. An adaptive online parameter control algorithm for particle swarm optimization based on reinforcement learning[C]//2019 IEEE congress on evolutionary computation (CEC). IEEE, 2019: 815 – 822.
- [10] 徐利锋, 黄祖胜, 杨中柱, 等. 引入多级扰动的混合型粒子群优化算法[J]. 软件学报, 2019, 30(6): 1835 – 1852.
- [11] PAN X, XUE L, LU Y, et al. Hybrid particle swarm optimization with simulated annealing[J]. Multimedia Tools and Applications, 2019, 78(21): 29921 – 29936.
- [12] LI M, CHEN H, WANG X, et al. An improved particle swarm optimization algorithm with adaptive inertia weights[J]. International Journal of Information Technology & Decision Making, 2019, 18(03): 833 – 866.
- [13] HARRISON K R, ENGELBRECHT A P, OMBUKI – BERMAN B M. Self – adaptive particle swarm optimization: a review and analysis of convergence[J]. Swarm Intelligence, 2018, 12(3): 187 – 226.
- [14] SUN L, SONG X, CHEN T. An improved convergence particle swarm optimization algorithm with random sampling of control parameters[J]. Journal of Control Science and Engineering, 2019, 2019:1 – 11.
- [15] GNANAPRAKASH V, VANATHI P T, SURESH G. Performance Analysis of Improved Particle Swarm Optimization for Fabric Defect Detection[C]//2021 Smart Technologies, Communication and Robotics (STCR). IEEE, 2021: 1 – 4.
- [16] 韩红桂, 卢薇, 乔俊飞. 一种基于种群多样性的粒子群优化算法设计及应用[J]. 信息与控制, 2017, 46(6): 677 – 684.
- [17] 宋美, 葛玉辉, 刘举胜. 基于协同进化的动态双重自适应改进 PSO 算法[J]. 计算机工程与应用, 2020, 56(13): 54 – 62.
- [18] CAI H H. On a Modified PSO Algorithm Based on Learning and Competitiveness[J]. Journal of Southwest China Normal University(Natural Science Edition), 2019, 44(5): 115 – 120.

A competitive particle swarm algorithm based on multiple swarm and strategy

LI Yuan-yuan, LI Wen-bo, SHANG Zhi-hao

(School of Software Engineering, Dalian Jiaotong University, Dalian 116028, China)

Abstract: The author proposes a multi – population and multi – strategy competitive particle swarm optimization (PSO) algorithm to address the issues faced by the standard PSO, such as falling into local optima, poor convergence, and low solution accuracy. In the new algorithm, each generation of particles is sorted based on fitness, divided into different sub – populations, and updated using nonlinear logistic chaotic mapping weights, contraction factors, and a hybrid Gaussian – Cauchy perturbation mechanism. The use of different particle update strategies balances the global search and local exploitation capabilities of the algorithm, thus accelerating the convergence speed. Finally, the multi – population and multi – strategy competitive PSO algorithm is compared with the standard PSO and other optimization algorithms on 11 test functions. The results demonstrate that the new algorithm significantly outperforms the standard PSO in terms of escaping local optima and solution accuracy, and it has a faster convergence speed. It also exhibits significantly higher optimization capability and algorithm stability compared to other benchmark algorithms.

Key words: swarm division; multi – strategy; hybrid mutation; chaotic mapping

(责任编辑 段 鹏)