

# 基于最优交叉的广泛学习粒子群优化

陈小斌, 杨利华, 汤可宗

(景德镇陶瓷大学信息工程学院, 江西景德镇 333403)

**摘要:** 粒子群优化算法实现简单、便于操作, 近年来已被广泛应用于资源分配等大规模复杂问题, 但算法收敛速度慢、求解精度低等问题也制约着其进一步应用。针对以上问题, 引入遗传算法的染色体交叉特性, 结合广泛学习粒子群优化算法, 提出一种基于最优交叉的广泛学习粒子群优化算法。通过全局最优粒子位置与个体历史最优位置执行最优交叉操作得到更优个体, 加快算法收敛速度, 提高对问题的求解精度。基准测试函数实验结果表明, 该算法相较于原算法具有更快的收敛速度和优化精度, 同时Friedman检验和Wilcoxon符号秩检验结果表明, 基于最优交叉的广泛学习粒子群优化算法具备较好的稳定性, 优化精度和收敛速度有了较大提升。

**关键词:** 粒子群优化; 遗传算法; 广泛学习策略; 最优交叉; Friedman检验; Wilcoxon符号秩检验

DOI: 10.11907/rjdk.222375

开放科学(资源服务)标识码(OSID):

中图分类号: TP301

文献标识码: A

文章编号: 1672-7800(2023)012-0132-07



## Comprehensive Learning Particle Swarm Optimization Based on Optimal Crossover

CHEN Xiaobin, YANG Lihua, TANG Kezong

(School of Information Engineering, Jingdezhen Ceramic University, Jingdezhen 333403, China)

**Abstract:** Particle swarm optimization (PSO) algorithm has been widely used in large-scale complex problems such as resource allocation in recent years because of its simple implementation and easy operation. However, the slow convergence speed and low solution accuracy of the algorithm also restrict its further application. In order to solve the above problems, this paper introduces the chromosome crossing characteristics of genetic algorithm, and combines with comprehensive learning particle swarm optimization, proposes a comprehensive learning particle swarm optimization based on optimal crossing. It can improve the convergence speed of the algorithm and the accuracy of solving the problem by performing the optimal crossover operation between the global optimal particle position and the historical optimal position of the individual to obtain a better individual. The experimental results of benchmark function show that the proposed algorithm has faster convergence speed and optimization accuracy than the original algorithm, and the results of Friedman test and Wilcoxon signed-rank test show that the proposed algorithm has better advantages than other comparison algorithms.

**Key Words:** particle swarm optimization; genetic algorithm; comprehensive learning strategy; optimal crossover; Friedman test; Wilcoxon signed-rank test

## 0 引言

粒子群优化(Particle Swarm Optimization, PSO)是进化计算的重要分支之一<sup>[1]</sup>, 其是通过模拟自然界生物活动的群体智能行为, 同时在借鉴人工生命<sup>[2]</sup>、鸟群觅食、鱼群学习等群智能行为基础上提出的一种启发式搜索优化算

法<sup>[3]</sup>。相比于其他启发式搜索算法, 由于PSO具有易于实现、操作简单等优点, 多年来已成功应用于机器学习、光伏参数识别、资源调度分配等现实世界的诸多优化问题中<sup>[4-10]</sup>。

然而, PSO的启发式搜索特性使得该算法存在早熟收敛、收敛速度慢、求解精度低等问题<sup>[11]</sup>。针对这些问题, 不同学者从粒子活动行为、粒子性能评价以及惯性权重自适应

收稿日期: 2022-11-21

基金项目: 江西省教育厅科学技术研究项目(GJJ210731, GJJ211331)

作者简介: 陈小斌(1997-), 男, 景德镇陶瓷大学信息工程学院硕士研究生, 研究方向为计算智能; 杨利华(1981-), 男, 硕士, 景德镇陶瓷大学信息工程学院副教授, 研究方向为统计计算与机器学习; 汤可宗(1978-), 男, 博士, 景德镇陶瓷大学信息工程学院教授, 研究方向为计算智能、模式识别与智能控制。

应调整等方面进行了不少改进工作。文献[12]在基础 PSO 算法中提出使用惯性权重调整算法的勘探和开发进度, 并提出基于惯性权重  $\omega$  的粒子群优化算法(Omega Particle Swarm Optimizer, OPSO), 更好地实现了初期勘探与后期开发之间的平衡, 有效提升了优化精度; 文献[13]通过分析粒子在离散时间运动时的轨迹, 并将其发展到连续时间, 将勘探与开发任务统一, 使用固定参数收缩因子取代惯性权重, 提出基于收缩因子的粒子群优化算法(Constriction Factor in Particle Swarm Optimization, CPSO), 有效提高了算法优化能力; 文献[14]–[17]则分别讨论了 PSO 的粒子动态跟踪以及模糊系统, 并结合粒子种群个体与集体的信息传递方式, 研究了不同种群拓扑结构对 PSO 优化能力的影响。

此外, 粒子种群间的交流学习近年来成为改进的流行方向。文献[18]尝试让粒子个体结合种群历史最优位置的优良信息进行更新, 提出一种广泛学习粒子群优化算法(Comprehensive Learning Particle Swarm Optimizer, CLPSO)。实验发现广泛学习策略可以较长时间保持种群的多样性, 对搜索空间进行更加充分地探索, 能够较好地避免种群过早收敛。文献[19]在 CLPSO 的基础上, 通过在粒子速度更新时增加一个动态扰动项, 实现粒子种群在开发与勘探阶段的平衡, 从而解决 CLPSO 的求解精度问题。文献[20]则利用广泛学习策略生成两个子种群, 提出一种增强探索和开发的异构综合学习粒子群优化算法(Heterogeneous Comprehensive Learning Particle Swarm Optimization with Enhanced Exploration and Exploitation, HCLPSO)。其中, 开发种群和勘探种群通过异步学习更新各种最优位置信息, 在保持算法开发性能充分发挥的同时, 也维系了粒子群的多样性。文献[21]利用一种信息共享机制(Information Sharing Mechanism, ISM)来提高 PSO 的性能, 提出具有信息共享机制的竞争与合作粒子群优化算法(Competitive and Cooperative Particle Swarm Optimization with Information Sharing Mechanism, CCPSO-ISM), 通过各粒子之间的信息共享来增强粒子间的沟通, 以提高算法的搜索能力。另一方面, 通过分析粒子群搜索过程中的运动行为也不失为一种好的改进方式。文献[22]通过动态评估粒子种群分布与适应度等方式对粒子群的惯性权重以及速度更新方式进行自适应调整, 以此改进 PSO 的收敛速度和全局搜索能力; 文献[23]则通过分析粒子运动状态提出双中心粒子群优化算法(Double Center Particle Swarm Optimization, DCPSO), 在不额外增加算法实现复杂度的情况下, 改善算法的收敛速度和精度。

除分析 PSO 自身的启发式优化特性进行改进外, 也有不少学者根据其他优化算法获得改进灵感。文献[24]利用高斯预过程拟合方法对当前问题优化空间最优解方向进行预测, 并引导粒子群搜索, 加快搜索速度, 同时增强粒子群的多样性; 文献[25]通过一种 sigmoid 函数的加权策

略来自适应地调整加速系数, 提出自适应权重粒子群优化算法(Adaptive Weighting Particle Swarm Optimization, AW-PSO), 以此提高算法的收敛速度。

遗传算法(Genetic Algorithm, GA)是一种通过借鉴进化生物学遗传中包括突变、自然选择以及杂交等现象发展而来的优化算法<sup>[26]</sup>。相对于 PSO 而言, 该算法可通过突变维持染色体的多样性, 同时通过交叉操作进行基因重组, 以获得更好的结果。因此, 许多学者都会借鉴 GA 的相关概念对 PSO 进行改进。文献[27]借鉴 GA 提出另一种具有重组学习和混合变异的动态多种群粒子群优化算法, 通过重组学习和混合变异提高算法的优化能力; 文献[28]则借鉴 GA 个体繁殖的概念, 结合 PSO 粒子的社会学习操作, 提出一种遗传学习粒子群优化算法。

CLPSO 的广泛学习策略能够在搜索过程中较好地勘探整个问题空间, 但收敛速度较慢成为算法进一步广泛应用的掣肘。为进一步提高 CLPSO 的求解精度, 同时加快算法收敛速度, 本文借鉴 GA 中的染色体交叉概念, 提出一种基于最优交叉的广泛学习粒子群优化(Comprehensive Learning Particle Swarm Optimization Based on Optimal Crossover, CLPSO-OC)算法。该算法利用 CLPSO 杰出的全局勘探能力, 通过粒子历史最优位置与全局最优位置执行交叉操作, 产生更优异的结果, 以进一步提升算法的综合性能。

## 1 算法基本理论

### 1.1 粒子群优化算法

PSO 是一种结合粒子社会学习行为和群体合作交互实现问题求解的启发式优化算法。该算法将种群中每个粒子个体抽象为具有速度  $V_i$  和位置  $X_i$  的一个质点。伴随着优化过程, 各粒子在全局最优位置  $gbest$  和个体历史最优位置  $pbest$  的引领下按式(1)、式(2)更新粒子速度与位置。

$$V_i = V_i + c_1 r_1 * (pbest_i - X_i) + c_2 r_2 * (gbest - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

其中,  $c_1$ 、 $c_2$  分别代表各粒子向  $pbest$  和  $gbest$  临近区域移动的学习因子,  $r_1$ 、 $r_2$  则为介于  $[0, 1]$  之间的随机数。 $pbest$  和  $gbest$  对粒子的引导使得粒子能够在更广泛的空间内进行探索, 同时赋予粒子一定的智能, 使粒子能够利用自身历史最优位置与全局最优位置不断地引导自身搜索, 让种群得以持续优化。

为进一步平衡粒子群的全局搜索与局部搜索能力, Shi 等<sup>[12]</sup>在式(1)的基础上引入惯性权重  $\omega$ , 使用式(3)更新速度。此改进大大提高了算法的优化能力, 并为之后的研究提供了新方向。

$$V_i = \omega V_i + c_1 r_1 * (pbest_i - X_i) + c_2 r_2 * (gbest - X_i) \quad (3)$$

### 1.2 遗传算法

GA 是受进化启发而提出的一种优化模型, 该算法在

简单类似染色体的数据结构上对特定问题的潜在方案进行编码,并将重组算法应用于这些结构以保留关键信息。GA通过随机初始化一定数量的染色体,通过特定方式评估这些结构并进行繁殖,此时那些代表目标问题更好解决方案的染色体比那些解决方案较差的染色体有更多繁殖机会。最优解决方案通常来源于当前染色体集合。在GA中,主要有以下3种操作:选择、交叉和变异。在选择操作中,可以选择轮盘赌、玻尔兹曼选择等方法;交叉操作中同样具有大量可选的交叉技术,例如循环交叉、多点交叉、均匀交叉等,但主流的操作可分为两种:单点交叉和两点交叉,如图1所示。

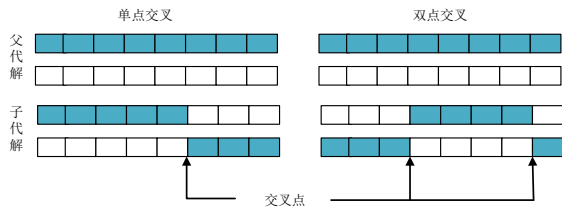


Fig. 1 Crossover operation

图1 交叉操作

交叉操作完成后,GA对于对应染色体中的每个位会以一些低概率的 $P_m$ 进行变异。通常,突变率的应用概率小于1%。然而,在实值函数优化中,GA染色体长度与求解精度紧密相关。此外,优化过程中对优化结果进行评估时需要进行编解码操作,因此大大增加了其实现难度。

## 2 基于最优交叉的广泛学习粒子群优化

### 2.1 广泛学习策略

在经典PSO中,每个粒子的更新同时受到全局极值 $gbest$ 和个体极值 $pbest$ 的影响。由于整个种群粒子速度的更新受到全局极值 $gbest$ 的影响,因此导致经典PSO算法快速收敛。然而,粒子收敛速度过快在一定程度上会造成粒子种群早熟收敛,无法更加广泛地勘探搜索空间。因此,在这些算法的单峰函数中,往往能够相对快速地得到一个精度较高的解。然而,在多模态函数优化中,过快收敛可能会导致算法陷入局部最优,以至于算法停留在局部最优位置,而无法达到真正的全局最优区域。广泛学习策略为了给予PSO在多模态函数中更好的优化能力,使用式(4)进行粒子速度更新。此时粒子的更新不再受全局最优 $gbest$ 影响,而是由 $pbest_{f_i(d)}^d$ 进行引导,其中 $pbest_{f_i(d)}^d$ 的计算规则如式(5)所示。通过使用新的速度更新方式控制粒子群对优化空间的搜索,使得各粒子能够更充分地勘探周边区域。

$$v_i^d = \omega v_i^d + cr_i^d * (pbest_{f_i(d)}^d - x_i^d) \quad (4)$$

$$pbest_{f_i(d)}^d = \begin{cases} pbest_i^d, & r_i^d < Pc_i \\ pbest_x^d, & otherwise \end{cases} \quad (5)$$

其中, $pbest_{f_i(d)}^d$ 的选择通过初始概率 $Pc_i$ 决定。当随机概率 $r_i^d$ 大于 $Pc_i$ 时,则由当前粒子本身的历史最优位置进

行引导,否则由其他粒子的历史最优位置进行引导。当粒子进行广泛勘探而发生停滞时,说明由式(4)进行更新的粒子可能陷入局部最优。因此,当停滞次数 $m$ 达到固定次数时,变更为由式(3)进行粒子速度更新,此时由当前全局最优位置对粒子群体进行引导,跳出停滞区域,并让粒子前往其他区域进行广泛搜索。

广泛学习策略伪代码如下:

Algorithm: CLS

1 if  $m > FixedTimes$  then

2  $V = \omega V + c_1 r_1 (pbest - X) + c_2 r_2 (gbest - X)$ ;

3 else

4 for  $i=1:N$  do

5 for  $d=1:dim$  do

6 if  $r_i^d > Pc_i$  then

7  $pbest_{f_i(d)}^d = pbest_i^d$ ;

8 else

9  $x = randperm(N, 1)$ ;

10  $pbest_{f_i(d)}^d = pbest_x^d$ ;

11 end

12  $v_i^d = \omega v_i^d + cr_i^d (pbest_{f_i(d)}^d - x_i^d)$ ;

13 end

14 end

15 end

### 2.2 最优交叉操作

在PSO中,寻优粒子个体是独立且自主的,在行为上会受到全局最优粒子的影响,但粒子间无法将可能存在的最优信息进行共享,因此优化效率和优化精度都会在一定程度上受到影响。而GA中通过染色体交叉的方式可以对两个染色体上的对应位置进行交换,以此让优化个体间进行直接的信息交流。据此,本文为进一步改善粒子群的收敛速度和优化精度,从粒子群的社会学习方式出发,提出一种最优交叉操作,该操作只发生在优化过程后期的全局最优位置与个体历史最优位置之间。由于这两个位置都包含了各粒子的优良信息,因此通过交叉操作可以更好地进行优良信息的学习,从而有可能得到更好的最优位置。交叉操作执行公式如式(6)所示,其中 $temp_{cross}$ 是最优位置 $best_i$ 和随机个体最优位置对应的交叉结果。

$$best_i = argmin \{ F(best_i), F(temp_{cross}) \} \quad (6)$$

在交叉操作中,通过一个随机概率 $r$ 选择交叉操作类型。当随机概率 $r$ 大于0.5时执行单点交叉操作,小于等于0.5时执行双点交叉操作,最优交叉伪代码如下:

Algorithm: CLS

1 if  $r > 0.5$  then

2  $[pbest_{cross}, gbest_{cross}] = SinglePointCross(pbest, gbest)$ ;

3 else

4  $[pbest_{cross}, gbest_{cross}] = DoublePointCross(pbest, gbest)$ ;

5 end

6  $pbest = argmin \{ F(pbest), F(pbest_{cross}) \}$ ;

7  $gbest = argmin \{ F(gbest), F(gbest_{cross}) \}$ ;

### 2.3 CLPSO-OC 算法

CLPSO-OC 借助广泛学习策略优良的广域勘探能力,提高算法在多模态问题模型中的优化能力,减少算法落入局部最优的风险。同时结合遗传算法中的染色体交叉概念,提出一种最优交叉方法,通过全局最优位置随机与种群中任意粒子历史最优位置的交叉操作来交换彼此的优良信息,以促进粒子间的学习交流,加快算法收敛速度,并有效提高求解精度。算法总体流程如下:

Step 1: 初始化参数设定,给每个粒子赋予随机的初始位置和速度。

Step 2: 将粒子个体的历史最优位置初始化为粒子的初始位置,同时根据适应度函数评价出全局最优位置。

Step 3: 执行广泛学习策略并更新粒子位置。

Step 4: 从  $X_t$  时刻的粒子位置和  $pbest_{t-1}$  时刻的粒子个体历史最优位置中评价出  $pbest_t$  时刻的粒子个体历史最优位置。

Step 5: 从  $pbest_t$  时刻的粒子个体历史最优位置和  $gbest_{t-1}$  时刻的全局最优位置中更新  $gbest_t$ 。

Step 6: 执行最优交叉操作。

Step 7: 判断是否满足结束操作,若是,则执行 Step 10, 否则执行 Step 3。

Step 8: 输出当前全局最优位置以及相对应的适应度值。

CLPSO-OC 相对于原算法而言,并未增加算法的实现难度,也没有额外增加时间复杂度。算法流程如图 2 所示。

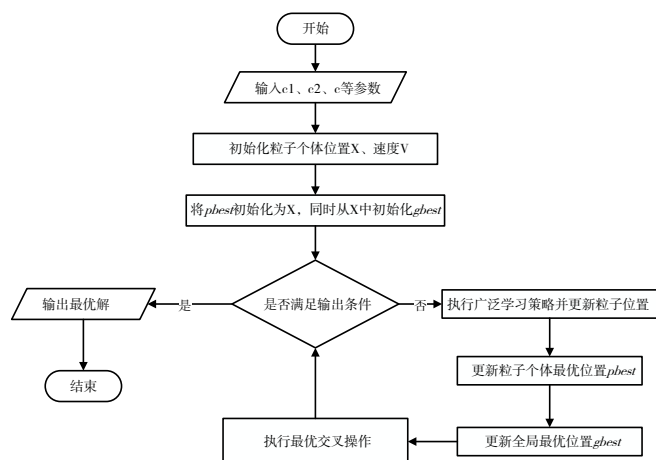


Fig. 2 Flow of CLPSO-OC  
图2 CLPSO-OC 执行流程

## 3 实验结果与分析

### 3.1 平台搭建与测试前的准备

本次测试采用的软件环境和硬件环境配置为: Windows11 操作系统, Matlab2020b 运行环境, 处理器为 Intel (R) Core (TM) i7-9700 CPU, 主频 3.00 GHz, 内存为 8GB。

为了突出对比,本文选取国外较为流行的典型改进 PSO 进行对比,分别是 OPSPSO<sup>[12]</sup>、CPSO<sup>[13]</sup>、CLPSO<sup>[18]</sup>、HCLPSO<sup>[20]</sup>、CCPSO-ISM<sup>[20]</sup>、DCPSO<sup>[23]</sup>、AWPSO<sup>[25]</sup>。其中, OPSPSO 和 CPSO 是 PSO 算法的经典之作,两者从粒子种群空间与时间角度对算法进行改进; CLPSO、HCLPSO 以及 CCPSO-ISM 则从粒子群学习方式出发对算法进行改进; AWPSO 使用 Sigmoid 函数对惯性权重进行自适应调整; DCPSO 则从粒子种群搜索在空间上的分布角度出发,使用双中心方法提升算法的优化效率。以上算法都涵盖了 PSO 较为代表性的改进方向,也被大部分学者所认可,因此本文选择以上算法进行对比实验。各算法参数、粒子种群大小及最大迭代次数设置如表 1 所示。从 CEC2017 中选取部分函数,如 Sphere、Rosenbrock 等经典优化问题模型作为基准测试函数,具体如表 2 所示。其中,  $f_1 \sim f_6$  和  $f_7 \sim f_{12}$  分别是无模态和多模态函数,表中  $D$  代表问题的求解维度,  $X$  代表搜索区间,且满足  $X \in R^D$ 。  $f^*$  指对应优化问题模型在  $D$  维度下的理论最优值。

Table 1 Parameter setting of algorithms

表 1 算法参数设置

算法	对应参数	种群规模 /N	最大迭代次 数
OPSPSO	$\omega=[0.9, 0.4], c1=2, c2=2$	30	5 000
CPSO	$\chi=0.729\ 844, c1=2.05, c2=2.05$	30	5 000
CLPSO	$\omega=[0.9, 0.4], c=1.494\ 45, c1=2, c2=2, m=7$	30	5 000
DCPSO	$\omega=[0.95, 0.3], c1=2, c2=2$	30	5 000
HCLPSO	$\omega=[0.99, 0.2], c1=[2.5, 0.5], c2=[0.5, 2.5], c=[3, 1.5]$	30	5 000
CCPSO-ISM	$\omega=0.6, c=2, p=0.05$	30	5 000
AWPSO	$\omega=[0.9, 0.4], a=0.000\ 035m, b=0.5, c=0, d=1.5$	30	5 000
CLPSO-OC	$\omega=[0.9, 0.4], c=1.494\ 45, c1=2, c2=2, m=7$	30	5 000

Table 2 Information of benchmark test function

表 2 基准测试函数信息

类型	函数	名称	优化维度 $D$	理论最优 值/ $f^*$	搜索空间
Unimodal	$f_1(x)$	Quadric	30	0	$[-10, 10]$
	$f_2(x)$	Sphere	30	0	$[-5.12, 5.12]$
	$f_3(x)$	Schwefel's P22	30	0	$[-10, 10]$
	$f_4(x)$	Rosenbrock	30	0	$[5, 10]$
	$f_5(x)$	Sum Squares	30	0	$[-10, 10]$
	$f_6(x)$	Step	30	0	$[-100, 100]$
Multimodal	$f_7(x)$	Ackley	30	0	$[-32, 32]$
	$f_8(x)$	Michalewicz	10	-9.66	$[0, \pi]$
	$f_9(x)$	Schwefel	30	0	$[-500, 500]$
	$f_{10}(x)$	Dixon-Price	30	0	$[-10, 10]$
	$f_{11}(x)$	Griewank	30	0	$[-600, 600]$
	$f_{12}(x)$	Rastrigin	30	0	$[-5.12, 5.12]$

### 3.2 评价指标

本节选取各算法在基准测试函数独立重复实验中的最小值  $min$ 、平均值  $mean$  以及标准差  $std$  作为基本评价指

标,三者分别能够代表算法的最佳优化能力、平均优化效果和算法的优化稳定性,能够较为直观地反映出各算法在对应基准函数中的性能。

此外,统计检验方法是检验优化算法性能的关键方法之一,通过检验测试可以得出更可靠的结论。本文采用Friedman 检验各算法之间的差异,同时进一步使用Wilcoxon 符号秩检验来验证CLPSO-OC 与其他比较算法之间的差异显著性水平。通过以上两个统计检验方法验证CLPSO-OC 在求解经典优化问题上的普遍有效性。

### 3.3 算法测试结果分析

根据上文实验环境的设置,各算法在所有测试函数中完成了独立重复优化实验,详细实验结果如表3所示。从表3可以看出,CLPSO-OC 在 $f_3, f_9, f_{10}, f_{12}$  4个基准函数测试结果中的3项指标都优于其他7种改进PSO算法;在 $f_1, f_2$  两个基准函数中,CLPSO 具有更好的优化性能;CCPSO-ISM 则在 $f_8$  和 $f_{11}$  中表现出比其他改进PSO算法更优异的性能;

在基准函数 $f_5$  中,所有算法都能够得到最优结果,但OPSO、CLPSO、DCPSO、CCPSO-ISM 以及CLPSO-OC 表现更加稳定。

为了观察各优化算法的收敛状态,本文将收敛结果按照一定比例进行等比缩放,最终结果如图3所示。

从图3中各算法在对应函数中的收敛状态可以看出,在指定的迭代次数内,CLPSO-OC 在函数 $f_3, f_9$  中收敛速度和收敛精度的表现最为优异;在函数 $f_1$  中,HCLPSO 表现较好,在后期能较为快速地达到更高精度;CPSO 在函数 $f_4$  中展现出来的优化能力最强。综合来说,CLPSO-OC 相对于CLPSO 具有更快的收敛速度和更高的求解精度,同时与其他改进PSO算法相比具有较好的优化能力。

### 3.4 基于Friedman 检验与Wilcoxon 符号秩检验的可靠性对比分析

各算法的Friedman 检验结果如表4所示,其中CLPSO-OC 在对比优化算法中的基准问题优化性能排名第一。

Table 3 Comparison of benchmark function test performance

表3 基准函数测试性能比较

Functions	Criterion	OPSO	CPSO	CLPSO	DCPSO	HCLPSO	CCPSO-ISM	AWPSO	CLPSO-OC
$f_1$	min	1.17E-03	1.32E-10	9.61E-04	1.62E-04	<b>3.65E-14</b>	3.79E+00	6.32E-06	2.27E-06
	mean	2.51E+00	2.56E+00	5.05E-03	2.53E-03	<b>2.63E-07</b>	8.39E+00	3.45E-01	4.37E-05
	std	7.35E+00	1.13E+01	3.90E-03	2.37E-03	<b>4.65E-06</b>	2.05E+00	5.96E+00	4.07E-05
$f_2$	min	2.63E-36	6.03E-92	9.25E-57	1.31E-61	<b>1.89E-100</b>	4.40E-25	1.47E-33	1.32E-71
	mean	4.36E-32	9.44E-77	4.66E-55	2.50E-55	<b>3.78E-80</b>	2.33E-21	7.41E-21	1.99E-67
	std	1.67E-31	8.54E-76	5.52E-55	2.08E-54	<b>3.74E-79</b>	1.48E-20	7.06E-20	4.66E-67
$f_3$	min	1.85E-22	1.85E-27	1.49E-32	1.30E-36	9.03E-18	9.20E-10	3.41E-13	<b>1.51E-40</b>
	mean	2.17E-19	1.29E-12	1.57E-31	7.31E-31	3.88E-09	2.66E-06	5.08E-07	<b>2.65E-39</b>
	std	1.02E-18	8.08E-12	1.07E-31	2.48E-30	2.08E-08	5.95E-06	2.18E-06	<b>4.36E-39</b>
$f_4$	min	4.83E+00	<b>1.24E-06</b>	3.12E-01	2.17E+01	1.19E-05	7.39E-01	2.86E-02	2.18E-01
	mean	1.37E+02	8.37E+01	4.76E+01	2.66E+01	3.60E+01	3.15E+01	3.66E+02	<b>2.58E+01</b>
	std	2.47E+02	3.67E+02	3.54E+01	<b>2.67E-01</b>	2.98E+01	7.07E-01	2.51E+03	3.40E+01
$f_5$	min	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	mean	<b>0.00E+00</b>	2.47E+00	<b>0.00E+00</b>	<b>0.00E+00</b>	6.00E-02	<b>0.00E+00</b>	4.00E-02	<b>0.00E+00</b>
	std	<b>0.00E+00</b>	4.554+00	<b>0.00E+00</b>	<b>0.00E+00</b>	2.39E-01	<b>0.00E+00</b>	1.97E-01	<b>0.00E+00</b>
$f_6$	min	1.86E-34	4.68E-89	1.28E-54	1.32E-101	<b>7.77E-102</b>	5.59E-24	2.53E-33	9.63E-69
	mean	6.60E-31	6.37E-75	1.82E-53	2.56E-54	<b>7.85E-87</b>	9.76E-20	5.00E-20	1.40E-65
	std	1.92E-30	4.29E-74	1.94E-53	1.45E-53	<b>7.12E-86</b>	6.99E-19	4.49E-19	6.30E-65
$f_7$	min	<b>8.88E-16</b>	4.44E-15	4.44E-15	4.44E-15	2.22E-14	1.71E-10	<b>8.88E-16</b>	7.99E-16
	mean	4.44E-15	1.16E-02	4.44E-15	<b>4.34E-15</b>	1.22E-01	2.92E-07	4.48E-15	1.11E-14
	std	5.05E-16	1.16E-01	<b>0.00E+00</b>	7.00E-16	3.78E-01	1.71E-06	7.98E-16	3.41E-15
$f_8$	min	-5.35E+00	-4.88E+00	-4.47E+00	-7.53E+00	-4.72E+00	<b>-8.07E+00</b>	-5.63E+00	-7.65E+00
	mean	-3.46E+00	-3.51E+00	-3.18E+00	-6.15E+00	-3.36E+00	<b>-7.14E+00</b>	-3.46E+00	-5.98E+00
	std	5.47E-01	4.01E-01	4.42E-01	4.74E-01	4.54E-01	<b>3.36E-01</b>	5.31E-01	6.87E-01
$f_9$	min	3.30E+03	4.34E+03	3.00E+03	1.76E+03	3.59E+03	2.80E+03	3.44E+03	<b>1.52E+03</b>
	mean	5.28E+03	5.98E+03	4.45E+03	2.86E+03	4.82E+03	4.12E+03	5.25E+03	<b>2.63E+03</b>
	std	6.51E+02	8.15E+02	4.51E+02	4.31E-02	6.61E+02	3.98E+02	7.08E+02	<b>4.23E+02</b>
$f_{10}$	min	<b>6.67E-01</b>	<b>6.67E-01</b>	<b>6.67E-01</b>	<b>6.67E-01</b>	2.59E+03	<b>6.67E-01</b>	6.67E-01	<b>6.67E-01</b>
	mean	<b>6.67E-01</b>	<b>6.67E-01</b>	<b>6.67E-01</b>	<b>6.67E-01</b>	6.60E-01	<b>6.67E-01</b>	8.87E+00	<b>6.67E-01</b>
	std	1.67E-14	1.20E-15	<b>1.77E-16</b>	1.92E-16	6.67E-02	5.93E-03	4.66E+01	2.56E-16
$f_{11}$	min	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	mean	1.80E-02	3.46E-02	2.61E-03	1.17E-02	1.78E-02	1.45E-03	1.71E-02	<b>3.65E-04</b>
	std	1.94E-02	6.53E-02	5.23E-03	1.52E-02	2.33E-02	6.32E-03	1.80E-02	<b>6.36E-04</b>
$f_{12}$	min	2.49E+01	2.09E+01	9.95E-01	4.98E+00	1.39E+01	1.62E+01	2.19E+01	<b>0.00E+00</b>
	mean	4.40E+01	4.54E+01	5.99E+00	2.28E+01	3.35E+01	3.27E+01	4.54E+01	<b>3.24E+00</b>
	std	1.33E+01	1.12E+01	3.50E+00	6.65E+00	9.44E+00	1.12E+01	1.60E+01	<b>1.02E+00</b>

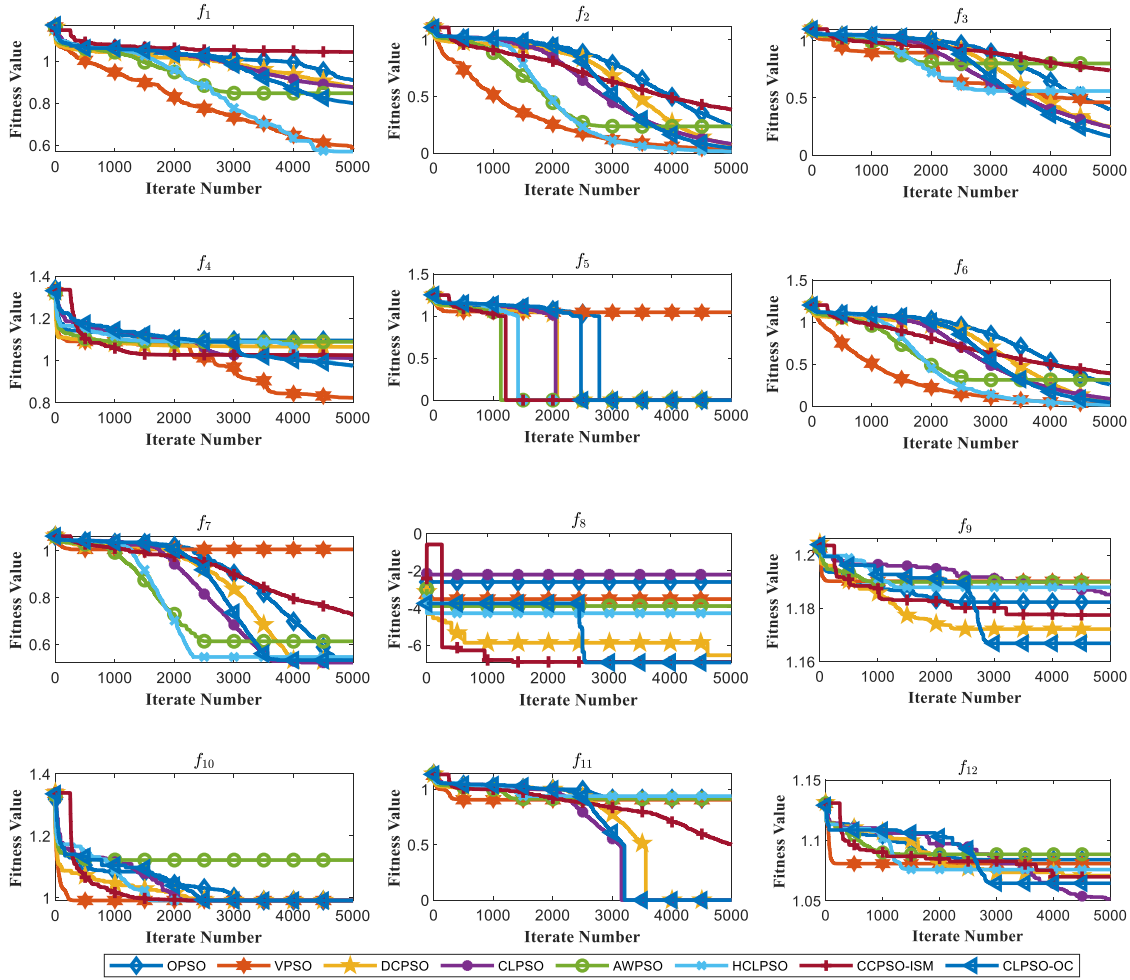


Fig. 3 Convergence state diagram of each improved algorithm under a fixed number of iterations

图3 各改进算法在固定迭代次数下的收敛状态图

Friedman 检验中的统计数据 p 值强烈表明 8 种算法优化结果之间存在显著差异。此外, CLPSO-OC 与其他参考算法的 Wilcoxon 符号秩检验结果如表 5 所示。检验结果表明, CLPSO-OC 与 OPPO、CPSO、CLPSO 以及 AWPPO 4 个改进 PSO 算法相比, 显示出显著性水平  $\alpha=0.05$  的明显改善, 其与 DCPSO、HCLPSO 和 CCPSO-ISM 之间的差异在统计学上并不显著, 但从基准函数测试结果可以看出, CLPSO-OC 在大部分基准测试函数上的优化效果与 DCPSO、HCLPSO 和 CCPSO-ISM 相比有一定的领先优势。此外, 从与原算法 CLPSO 的对比可以看出, CLPSO-OC 的优化精度和收敛速度有较为明显的改善。因此, Friedman 检验和 Wilcoxon 符号秩检验验证了 CLPSO-OC 在基准问题上的一般有效性。

不显著, 但从基准函数测试结果可以看出, CLPSO-OC 在大部分基准测试函数上的优化效果与 DCPSO、HCLPSO 和 CCPSO-ISM 相比有一定的领先优势。此外, 从与原算法 CLPSO 的对比可以看出, CLPSO-OC 的优化精度和收敛速度有较为明显的改善。因此, Friedman 检验和 Wilcoxon 符号秩检验验证了 CLPSO-OC 在基准问题上的一般有效性。

Table 4 Friedman statistical test results of various algorithms in benchmark function test

表 4 各算法在基准函数测试中的 Friedman 统计检验结果

检验	OPPO	CPSO	CLPSO	DCPSO	HCLPSO	CCPSO-ISM	AWPPO	CLPSO-OC	p-Value
Friedman ranks	5.958 3	6.458 3	4.708 3	3.291 7	4.916 7	5.416 7	7.250 0	3.125 0	0.000 8
Final rank	6	7	3	2	4	5	8	1	

Table 5 Statistical test results of Wilcoxon symbols for various algorithms in benchmark function test

表 5 各算法在基准函数测试中的 Wilcoxon 符号秩统计检验结果

CLPSO-OC vs.	OPPO	CPSO	CLPSO	DCPSO	HCLPSO	CCPSO-ISM	AWPPO
H-value	1	1	1	0	0	0	1
P-value	0.013 6	0.004 8	0.048 8	0.492 1	0.203 6	0.365 2	0.002 4

## 4 结论

针对 CLPSO 算法收敛速度慢、优化精度低等问题, 本文提出一种基于最优交叉的广泛学习粒子群优化算法, 实现在改善算法优化效率、提升求解精度的同时加快粒子群收敛。本文从 GA 中的染色体交叉获得灵感, 提出一种粒子群历史个体最优位置交叉操作, 用于改善粒子群收敛速

度慢的问题,同时提供各粒子之间优良信息传递的一个可行方法,既能保持CLPSO卓越的多模态开发能力以及较高的算法优化稳定性,又能提升算法的优化效率。

实验结果表明,本文提出的CLPSO-OC针对测试的基准函数相较于CLPSO有更好的优化表现。其在各基准测试函数中的搜索精度更高,平均优化效果和稳定性更好。相对于其他改进PSO算法,CLPSO-OC在综合性能上也有较大提升。最优交叉方法给当前PSO改进提供了一种新思路,同时给神经网络训练、大规模复杂网络等现实优化问题模型提供了一种新的解决方案。未来将进一步研究最优交叉的改进操作,以期再次提高算法优化效率。

#### 参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proceedings of ICNN'95-International Conference on Neural Networks, 1995: 1942-1948.
- [2] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory [C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995: 39-43.
- [3] KENNEDY J. Swarm intelligence [M]. Boston: Springer, 2006.
- [4] ANURAG A, PRIYADARSHI R, GOEL A, et al. 2-D coverage optimization in WSN using a novel variant of particle swarm optimisation [C]//2020 7th International Conference on Signal Processing and Integrated Networks, 2020: 663-668.
- [5] NUNES H G G, SILVA P N C, POMBO J A N, et al. Multiswarm spiral leader particle swarm optimisation algorithm for PV parameter identification [J]. Energy Conversion and Management, 2020, 225: 113388.
- [6] ZHU Y, LI G, WANG R, et al. Intelligent fault diagnosis of hydraulic piston pump combining improved LeNet-5 and PSO hyperparameter optimization [J]. Applied Acoustics, 2021, 183: 108336.
- [7] LIU W, WANG Z, ZENG N, et al. A PSO-based deep learning approach to classifying patients from emergency departments [J]. International Journal of Machine Learning and Cybernetics, 2021, 12(7): 1939-1948.
- [8] REN X, LIU S, YU X, et al. A method for state-of-charge estimation of lithium-ion batteries based on PSO-LSTM [J]. Energy, 2021, 234: 121236.
- [9] DOU J, LI J, XIA D, et al. A multi-objective particle swarm optimisation for integrated configuration design and scheduling in reconfigurable manufacturing system [J]. International Journal of Production Research, 2021, 59(13): 3975-3995.
- [10] CUONG-LE T, NGHIA-NGUYEN T, KHATIR S, et al. An efficient approach for damage identification based on improved machine learning using PSO-SVM [J]. Engineering with Computers, 2022, 38(4): 3069-3084.
- [11] SHI Y, EBERHART R C. Empirical study of particle swarm optimization [C]//Proceedings of the 1999 Congress on Evolutionary Computation, 1999: 1945-1950.
- [12] SHI Y, EBERHART R. A modified particle swarm optimizer [C]//1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998: 69-73.
- [13] EBERHART R C, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]//Proceedings of the 2000 Congress on Evolutionary Computation, 2000: 84-88.
- [14] SHI Y. Particle swarm optimization: developments, applications and resources [C]//Proceedings of the 2001 Congress on Evolutionary Computation, 2001: 81-86.
- [15] SHI Y, EBERHART R C. Fuzzy adaptive particle swarm optimization [C]//Proceedings of the 2001 Congress on Evolutionary Computation, 2001: 101-106.
- [16] KENNEDY J, MENDES R. Population structure and particle swarm performance [C]//Proceedings of the 2002 Congress on Evolutionary Computation, 2002: 1671-1676.
- [17] CLERC 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.
- [18] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [19] YU X, ZHANG X. Enhanced comprehensive learning particle swarm optimization [J]. Applied Mathematics & Computation, 2014, 242: 265-276.
- [20] LYNN N, SUGANTHAN P N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation [J]. Swarm and Evolutionary Computation, 2015, 24: 11-24.
- [21] LI Y, ZHAN Z H, LIN S, et al. Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems [J]. Information Sciences, 2015, 293: 370-382.
- [22] ZHAN Z H, ZHANG J, LI Y, et al. Adaptive particle swarm optimization [J]. IEEE Transactions on Systems, Man and Cybernetics, 2009, 39(6): 1362-1381.
- [23] TANG K Z, LIU B X, YANG J Y, et al. Double center particle swarm optimization algorithm [J]. Computer Research and Development, 2012, 49(5): 1086-1094.  
汤可宗,柳炳祥,杨静宇,等.双中心粒子群优化算法[J].计算机研究与发展,2012,49(5): 1086-1094.
- [24] JAKUBIK J, BINDING A, FEUERRIEGEL S. Directed particle swarm optimization with Gaussian-process-based function forecasting [J]. European Journal of Operational Research, 2021, 295(1): 157-169.
- [25] LIU W, WANG Z, YUAN Y, et al. A novel sigmoid-function-based adaptive weighted particle swarm optimizer [J]. IEEE Transactions on Cybernetics, 2021, 51(2): 1085-1093.
- [26] MIRJALILI S. Genetic algorithm [M]//Cham: Springer, 2019.
- [27] TANG K X, LIANG X L, ZHOU W F, et al. Dynamic multi-population particle swarm optimization algorithm with recombined learning and hybrid mutation [J]. Control and Decision, 2021, 36(12): 2871-2880.  
唐可心,梁晓磊,周文峰,等.具有重组学习和混合变异的动态多种群粒子群优化算法[J].控制与决策,2021,36(12): 2871-2880.
- [28] GONG Y J, LI J J, ZHOU Y, et al. Genetic learning particle swarm optimization [J]. IEEE Transactions on Cybernetics, 2015, 46(10): 2277-2290.

(责任编辑:黄健)