

## 多种群协方差学习差分进化算法

杜永兆<sup>①</sup> 范宇凌<sup>①</sup> 柳培忠<sup>①</sup> 唐加能<sup>\*①②</sup> 骆炎民<sup>③</sup>

<sup>①</sup>(华侨大学工学院 泉州 362021)

<sup>②</sup>(华侨大学机电及自动化学院 厦门 361021)

<sup>③</sup>(华侨大学计算机科学与技术学院 厦门 361021)

**摘要:** 种群多样性与交叉算子在差分进化(DE)算法求解全局优化问题中具有重要作用, 该文提出一种多种群协方差学习差分进化(MCDE)算法。首先, 采用多种群机制的种群结构, 利用每一子种群结合相应的变异策略保证进化过程个体多样性。然后, 通过种群间的协方差学习, 为交叉操作建立一个适当旋转的坐标系统; 同时, 使用自适应控制参数来平衡种群的勘测与收敛能力。最后, 在单峰函数、多峰函数、偏移函数和高维函数的25个基准测试函数上进行测试, 并同其他先进的进化算法对比, 实验结果表明该文算法相较于其他算法在求解全局优化问题上达到最优效果。

**关键词:** 差分进化; 多种群; 协方差学习; 自适应参数

中图分类号: TP18

文献标识码: A

文章编号: 1009-5896(2019)06-1488-08

DOI: 10.11999/JEIT180670

## Multi-populations Covariance Learning Differential Evolution Algorithm

DU Yongzhao<sup>①</sup> FAN Yuling<sup>①</sup> LIU Peizhong<sup>①</sup>

TANG Jianeng<sup>①②</sup> LUO Yanmin<sup>③</sup>

<sup>①</sup>(College of Engineering, Huaqiao University, Quanzhou 362021, China)

<sup>②</sup>(College of Mechanical Engineering and Automation, Huaqiao University, Xiamen 361021, China)

<sup>③</sup>(College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China)

**Abstract:** The diversity of the population and the crossover operator algorithm play an important role in solving global optimization problems in Differential Evolution (DE). The Multi-populations Covariance learning Differential Evolution (MCDE) algorithm is proposed. Firstly, the population structure is a multi-populations mechanism, and each subpopulation combines the corresponding mutation strategy to ensure the individual diversity in the evolutionary process. Then, the covariance learning establishes a proper rotation coordinate system for the crossover operation in the population. At the same time, the adaptive control parameters are used to balance the ability of population survey and convergence. Finally, the proposed algorithm is conducted on 25 benchmark functions including unimodal, multimodal, shifted and high-dimensional test functions and compared with the state-of-the-art evolutionary algorithms. The experimental results show that the proposed algorithm compared with other algorithms has the best effect on solving the global optimization problem.

**Key words:** Differential Evolution (DE); Multi-populations; Covariance learning; Self-adaptive parameter

收稿日期: 2018-07-06; 改回日期: 2019-01-28; 网络出版: 2019-02-18

\*通信作者: 唐加能 2812280164@qq.com

基金项目: 国家自然科学基金(61605048, 61231002, 51075068), 福建省教育厅项目(JA15035), 泉州市科技局项目(2014Z103, 2015Z114), 华侨大学研究生科研创新能力培养计划(1611422002)

Foundation Items: The National Natural Science Foundation of China (61605048, 61231002, 51075068), The Fujian Provincial Department of Education Project (JA15035), The Quanzhou Science and Technology Bureau Project (2014Z103, 2015Z114), Huaqiao University Graduate Research Innovation Capacity Development Program Funding Project (1611422002)

### 1 引言

差分进化(Differential Evolution, DE)算法是由美国学者Storn和Price<sup>[1]</sup>于1995年提出的一种模拟“优胜劣汰、适者生存”的仿生智能计算方法。DE采用变异、交叉和选择操作来模拟生物进化过程中的基因突变行为, 将适应性高的个体保留下来以此寻求最优解。针对种群收缩停滞和早熟收敛两个问题, 国内外研究学者主要集中在控制参数设置与进化策略选择<sup>[2]</sup>、交叉过程<sup>[3-5]</sup>以及种群结构<sup>[6,7]</sup>等研究。DE的核心算法是控制参数与进化策略选择。

第一, 控制参数放缩因子 $F$ 、交叉概率 $CR$ 和种群规模 $NP$ <sup>[8]</sup>的设置; 第二, 不同的策略用于解决不同的优化问题, 从一些可能的策略中选择一种最为合适的策略。参数的设置会影响到种群的多样性、前期的开发能力以及后期的收敛性。进化策略的选择是决定DE算法勘察能力与收敛能力平衡的关键, 不同的进化策略表现出不同的勘察能力与收敛倾向。

针对种群早熟收敛的问题, 参数控制和策略改进、交叉模型和种群结构变化越来越吸引学者的关注<sup>[9-11]</sup>。许多DE改进算法不断被提出, 有基于参数和策略改进DE算法, 如单一种群参数自适应的jDE(Differential Evolution algorithm with strategy adaptation)<sup>[9]</sup>、采用“current-to-pbest/1”策略和适应性参数的JADE(Adaptive Differential Evolution)<sup>[10]</sup>、差分策略自适应的SaDE(Self-adapting control parameters in Differential Evolution)<sup>[8]</sup>、适合试验向量代数策略和固定参数的CoDE(Differential Evolution with Composite trial vector)<sup>[11]</sup>、变异策略和控制参数集合的EPSDE(Differential Evolution algorithm with Ensemble of Parameters and mutation Strategies)<sup>[7]</sup>。基于交叉模型改进DE算法, 协方差学习和参数双峰分布的CoBiDE(Differential Evolution based on Covariance matrix learning and Bimodal distribution parameter setting)<sup>[12]</sup>。基于种群结构改进DE算法, 如多目标选择变异策略<sup>[13]</sup>、多种群和策略集合的MPEDE(Differential Evolution with Multi-Population based Ensemble of mutation strategies)<sup>[14]</sup>、主从模型、岛屿模型、元胞模型、等级模型及水池模型。近几年, 通过种群分离技术来提高进化算法的效果, 有粒子群优化算法、遗传算法以及进化算法<sup>[15-20]</sup>。

本文内容安排: 第2节介绍所研究算法背景, 第3节详细阐述了算法的改进工作, 第4节通过实验数据来分析算法的显著性, 第5节对于提出的算法进行总结。

## 2 MCDE算法

在本文算法中, 通过将整体种群划分为多个子种群, 给予相应变异策略。然后通过进化中子种群的协方差学习和参数自适应, 建立每个子种群的交叉坐标系统。最后, 将得到的交叉向量进行选择操作, 把适应值更好的向量个体保留下来, 使整体种群搜索朝着全局最优解前进。

### 2.1 多种群变异策略集成方法

由于不同的变异策略适合求解不同的优化函数, 研究学者主要集中研究多种变异策略集合方

法。即使对于一个特定的优化问题, 在进化过程的不同阶段, 最合适的变异策略也可能是不同的。因此, 在DE算法中, 变异策略是保证效果显著的一个重要指标。在进化过程中, 通过对变异策略的不同性能要求, 本文选取“当前最优策略”、“当前随机策略”和“随机策略”作为多种群变异策略组。“当前随机策略”和“随机策略”参与变异的向量均以随机的方式被选择, 因此能够在进化前期进行全局寻优。“当前最优策略”通过当前种群最优解来寻求全局最优解, 在进化过程能将搜索范围缩小至最优解附近并加快算法的收敛速度。

当前最优策略:

$$v_i^G = x_i^G + F \cdot (x_{pb}^G - x_i^G + x_{r_1}^G - x_{r_2}^G) \quad (1)$$

当前随机策略:

$$v_i^G = x_i^G + K \cdot (x_{r_1}^G - x_i^G) + F \cdot (x_{r_2}^G - x_{r_3}^G) \quad (2)$$

随机策略:

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G) \quad (3)$$

其中,  $x_i^G$ 表示当前进化个体,  $pb$ 表示当前种群最优个体,  $r_1 \neq r_2 \neq r_3$ 是3个随机选取介于1和 $D$ 之间的参数,  $v_i^G$ 表示变异后得到的个体,  $K$ 是调整因子。由于3个变异策略有各自的优点, 但存在一定的差异。因此, 本文引入了多种群策略机制, 将整体种群 $P_0$ 划分为3个子种群 $P_{01}$ ,  $P_{02}$ ,  $P_{03}$ , 认定其中种群规模最大的子种群 $P_{01}$ 为试验群, 并与相应的变异策略结合。在进化过程中, 将试验群分配给进化结果较好的变异策略。种群结构中有种群规模比率 $\sigma$ , 种群规模 $NP$ , 种群结构表示如式(4)

$$\left. \begin{aligned} P_0 &= \bigcup_{i=1,2,3} P_{0i} \\ NP_i &= \sigma_i \cdot NP \\ \bigcup_{i=1,2,3} \sigma_i &= 1 \end{aligned} \right\} \quad (4)$$

其中,  $P_{01}$ 为子种群中试验群, 那么 $\sigma_1 > \sigma_2 = \sigma_3$ 且 $\sigma_i \in [0, 1]$ 。

种群结构设计好后, 要给出子种群的分配规则。首先, 子种群 $P_{01}$ ,  $P_{02}$ ,  $P_{03}$ 结合相应的变异策略。然后, 种群进行变异、交叉、选择操作。最后, 通过计算和比较相邻两代种群的适应度值, 统计父代里3个子种群经过每一次种群进化后保留下优良个体的总和 $bd_i$ 。即子种群的优良率 $br_i$ 可以表示为

$$br_i = \frac{bd_i}{NP_i} \quad (5)$$

计算每一代3个子种群对应变异策略的优良率 $br_i$ , 在下一代进化初始化阶段根据优良率 $br_i$ 重新

将3个变异策略按高到低分别分配给子种群 $P_{O_1}$ ,  $P_{O_2}$ ,  $P_{O_3}$ 。多种群变异策略集合方法充分利用了3个变异策略各自的优势, 调节和平衡种群多样性和算法收敛速度之间的矛盾, 从后面的实验结果更可以看出它的优越性。

2.2 协方差学习

上述提到交叉算子主要依赖于坐标系统, 而种群的分布信息一定程度上反映进化的方向。在进化过程中往往忽略了种群的分布情况, 导致种群可能陷入局部最优。本文采用方差和协方差分析种群分布, 形成协方差矩阵来反映种群多样性信息。在本文中, 协方差矩阵学习包括两个相关技术: 协方差矩阵的特征分解和坐标变换。协方差矩阵学习步骤如下:

步骤 1 通过子种群信息, 计算各个子种群的协方差矩阵  $C$ ;

步骤 2 求协方差矩阵的特征值  $\lambda$  和协方差矩阵特征向量  $R$ ;

步骤 3 通过基于特征协同系统对目标个体和变异个体进行更新操作

$$\left. \begin{aligned} x_i^G &= R^{-1} x_i^G = R^T x_i^G \\ v_i^G &= R^{-1} v_i^G = R^T v_i^G \end{aligned} \right\} \quad (6)$$

步骤 4 种群进行交叉和选择操作适应度更好的个体保留下来, 并旋转回到原坐标系统。

$$u_i^G = R \cdot v_i^G \quad (7)$$

基于上述4个步骤来建立种群特征坐标系统, 图1(a)是种群进化初始坐标系, 图1(b)是特征坐标系。通过分析种群特征来得到  $ox'_1 x'_2$  坐标系, 发现能更快找到全局最优。

2.3 自适应控制参数设置

研究学者已提出许多参数适应方法<sup>[8-10]</sup>, 不同的控制参数与变异策略结合对优化问题的求解会得出不同的结果。本文令每一尺度策略都有其独立的控制参数, 并将不同的技术应用于本文算法。文献<sup>[10]</sup>方法更适合本文算法, 因此通过改进其技术适应到算法中。

在进化过程中, 缩放因子  $F$  对基向量的搜索范

围起决定作用。在标准DE算法中,  $F$  的取值是一个固定值, 不能适用于求解所有的全局优化函数。本文放缩因子  $F$  主要采用柯西逆累积分布函数, 假定  $F_{i,j}$  表示个体中每一维数的交叉概率。即  $F_{i,j}$  表示如式(8)

$$F_{i,j} = Cy_{i,j}(Fm_j, 0.1) \quad (8)$$

其中,  $Fm_j$  是柯西逆累积分布函数的位置参数和当前个体的放缩因子,  $Fm_j$  初始值定为0.5, 0.1是柯西逆累积分布函数的尺度参数。为了更好适用于种群进化, 引入加权因子  $c$  将父代放缩因子和下一代放缩因子结合。即  $Fm_j$  表示如式(9)

$$Fm_j = (1 - c) \cdot Fm_j + c \cdot \text{mean}_F(S_{F,j}) \quad (9)$$

其中,  $c \in [0, 1]$ , 而父代放缩因子采用Power均值求取  $S_{F,j}$ 。即Power均值表示为

$$\text{mean}_F(S_{F,j}) = \left( \frac{1}{NP} \cdot \sum_{NP} F^n \right)^{\frac{1}{n}} \quad (10)$$

其中,  $NP$  为种群规模,  $n$  是Power均值的指数值, 通过Power均值来量化父代放缩因子对子代的影响。

在DE算法中, 交叉概率  $CR$  决定目标个体从变异个体  $v_i^G$  中继承基因的可能性。在本文中交叉概率  $CR$  主要采用正态分布函数,  $CR_{i,j}$  表示个体中每一维数的交叉概率。即  $CR_{i,j}$  表示为

$$CR_{i,j} = \text{randn}_{i,j}(CRm_j, 0.1) \quad (11)$$

其中,  $CRm_j$  是个体交叉概率的平均值,  $CRm_j$  初始值定为0.5, 并把正太分布标准偏差值设为0.1。为了更好继承父代基因, 引入加权因子  $c$  将父代交叉概率和下一代交叉概率结合。即  $CRm_j$  表示为

$$CRm_j = (1 - c) \cdot CRm_j + c \cdot \text{mean}_{CR}(S_{CR,j}) \quad (12)$$

其中,  $c \in [0, 1]$ , 而父代交叉概率采用Lehmer均值求取  $S_{CR,j}$ 。即Lehmer均值表示为

$$\text{mean}_{CR}(S_{CR,j}) = \frac{\sum CR^2}{\sum CR} \quad (13)$$

Lehmer均值方法能根据父代交叉概率灵活的调整  $CR$  的取值。



图 1 种群进化过程坐标系

### 3 实验与结果

MCDE算法在IEEE CEC 2005的25个基准函数上测试, 这25个测试函数主要包括: 单峰函数 $F_1 \sim F_5$ 、基本型多峰函数 $F_6 \sim F_{12}$ 、偏移型多峰函数 $F_{13}, F_{14}$ 、高维型函数 $F_{15} \sim F_{25}$ , 详细信息请参考文献<sup>[21]</sup>。本文算法MCDE相关参数设置: 种群规模 $NP = 250$ ; 子种群比例 $\sigma_1 = 0.6, \sigma_2 = \sigma_3 = 0.2$ 。实验环境: 操作系统是win7专业版64位, CPU是Core i7(3.40 GHz), RAM是8 GB, 编译器为MATLAB R2014b。

#### 3.1 与改进DE算法比较

为了验证MCDE算法的性能, 从表1检测结果可以看出, Wilcoxon's检测在 $\alpha = 0.05$ 和 $\alpha = 0.10$ 时, MCDE比JADE, CoDE和CoBiDE算法效果更

表1 在 $D=30$ 下3种算法与MCDE的Wilcoxon's检测结果比较

比较算法	$R^+$	$R^-$	$P$ 值	$\alpha=0.05$	$\alpha=0.10$
JADE	240.5	59.5	0.007012	是	是
CoDE	264.5	60.5	0.005181	是	是
CoBiDE	251.0	74.0	0.016633	是	是

加显著。Wilcoxon's秩检验是通过两算法实验结果差异性求得差值变量的秩, 分布计算正号秩总和 $R^+$ 和负号秩总和 $R^-$ 。概率 $P$ 值表示显著性水平, 若 $P$ 值小于 $\alpha$ , 则该算法表现显著。根据表2在 $D=30$ 下Friedman平均排名可知, MCDE在各种类型的测试函数里表现显著并取得了最好的排名。Friedman的显著值表示与最优适应值之间的差距, 得出最后算法的排名。

表2 在 $D=30$ 下各算法的Friedman平均排名

算法	显著值	最终排名
JADE	3.78	3
CoDE	3.80	4
CoBiDE	3.34	2
MCDE	2.74	1

与JADE<sup>[10]</sup>, CoDE<sup>[11]</sup>, CoBiDE<sup>[12]</sup>3种经典的DE改进算法比较。JADE是两种代表性算法且被大量引用。CoDE和CoBiDE是基于种群结构改进的算法。表3提供了以上4种算法在 $D=30$ 和最大评

表3 30次独立运行在4种算法的最优解平均值及标准差

函数	JADE	CoDE	CoBiDE	MCDE
$F_1$	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	<b>0.00E+00±0.00E+00</b>
$F_2$	1.26E-28±1.22E-28+	6.77E-15±3.44E-15-	1.60E-12±2.90E-12-	8.49E-28±3.75E-28
$F_3$	8.42E+03±6.58E+03-	5.65E+05±5.66E+04-	7.26E+04±5.64E+04-	<b>2.74E-12±2.82E-11</b>
$F_4$	4.13E-16±3.45E-16-	6.21E-03±4.67E-02-	1.16E-03±2.74E-03-	<b>7.57E-22±4.26E-21</b>
$F_5$	7.59E-08±5.65E-07-	3.16E+02±3.62E+02-	8.03E+02±1.51E+01-	<b>5.38E-10±7.12E-10</b>
$F_6$	1.16E+01±3.16E+01-	3.32E-01±6.57E-01-	4.13E-02±9.21E-02+	3.19E-01±1.09E-01
$F_7$	8.27E-03±8.22E-03-	7.39E-03±6.45E-03-	1.77E-03±3.73E-03-	<b>1.52E-03±4.11E-03</b>
$F_8$	2.09E+01±1.68E-01≈	2.01E+01±1.25E-01+	2.07E+01±3.75E-01+	2.09E+01±4.21E-02
$F_9$	0.00E+00±0.00E+00+	0.00E+00±0.00E+00+	0.00E+00±0.00E+00+	2.64E-07±5.87E-07
$F_{10}$	2.42E+01±5.44E+00-	4.21E+01±2.84E+01-	4.41E+01±1.29E+01-	<b>2.28E+01±4.27E+00</b>
$F_{11}$	2.57E+01±2.21E+00-	1.24E+01±3.55E+00+	5.62E+00±2.19E+00+	1.51E+01±6.81e+00
$F_{12}$	6.45E+03±2.89E+03-	3.21E+03±4.48E+03-	2.94E+03±3.93E+03-	<b>2.12E+03±1.34E+03</b>
$F_{13}$	1.47E+00±1.15E-01+	1.66E+00±3.25E-01+	2.64E+00±1.13E+00-	1.74E+00±2.04E-01
$F_{14}$	1.23E+01±3.21E-01≈	1.23E+01±3.56E-01≈	1.23E+01±4.90E-01≈	<b>1.23E+01±2.66E-01</b>
$F_{15}$	3.61E+02±2.24E+02+	4.00E+02±5.24E+01≈	4.04E+02±5.03E+01-	4.00E+02±1.09E+02
$F_{16}$	9.33E+01±1.31E+02-	7.25E+01±6.22E+01+	7.38E+01±3.66E+01-	<b>5.37E+01±3.01E+01</b>
$F_{17}$	1.21E+02±1.08E+02-	7.16E+01±2.35E+01-	7.25E+01±2.02e+01-	<b>6.36E+01±6.41E+01</b>
$F_{18}$	9.04E+02±1.24E-01≈	9.04E+02±1.34E+00≈	9.03E+02±1.05E+01≈	9.03E+02±6.01E-01
$F_{19}$	9.04E+02±8.32E+00≈	9.04E+02±3.22E-01≈	9.03E+02±1.04E+01≈	9.03E+02±2.31E-01
$F_{20}$	9.04E+02±7.65E-01≈	9.04E+02±7.11E-01≈	9.04E+02±5.95E-01≈	9.03E+02±2.45E-01
$F_{21}$	5.00E+02±4.67E-13≈	5.00E+02±4.68E-13≈	5.00E+02±4.62E-13≈	<b>5.00E+02±4.51E-14</b>
$F_{22}$	8.68E+02±2.24E+01≈	8.78E+02±3.54E+01≈	8.69E+02±2.80E+01≈	8.69E+02±1.89E+01
$F_{23}$	5.48E+02±8.62E+01-	5.34E+02±4.45E-04≈	5.34E+02±1.30E-04≈	<b>5.34E+02±2.49E-13</b>
$F_{24}$	2.00E+02±2.12E-14≈	2.00E+02±2.62E-14≈	2.00E+02±2.90E-14≈	2.00E+02±2.90E-14
$F_{25}$	2.11E+02±7.35E-01-	2.11E+02±6.82E-01-	2.10E+02±7.73E-01-	<b>2.09E+02±2.78E-01</b>
+/-/≈	<b>3/13/9</b>	<b>5/10/10</b>	<b>4/13/8</b>	

估次数MaxFES=300000的实验结果,表中数值的形式是平均误差±标准偏差,标记“+/-/≈”分别表示比较算法性能好于MCDE、差于MCDE和近似于MCDE。根据图2和表3数据,得出以下结论:单峰测试函数 $F_1 \sim F_5$ :在比较算法里面JADE的效果最好,由于采用贪婪策略“当前最优策略”,使算法能快速收敛和高精度。然而,MCDE多尺度策略将全局搜索和局部搜索,在测试函数 $F_3$ (图2(a))、 $F_4$ (图2(b))、 $F_5$ (图2(c))求解精度上效果好于JADE。基本型多峰测试函数 $F_6 \sim F_{12}$ :表中表现最好的算法是CoBiDE,在测试函数 $F_6, F_8, F_9, F_{11}$ 上效果比MCDE好, $F_7$ (图2(d))、 $F_{10}$ (图2(e))、 $F_{12}$ (图2(f))测试效果差于MCDE。整体上,MCDE算法在这一类测试函数上和CoBiDE算法效果差不多。偏移型多峰函数 $F_{13}, F_{14}$ :4种算法的测试的平均误差在

1个数量级上,但是JADE,CoDE和MCDE的效果比CoBiDE算法稍微好一些。高维型函数 $F_{15} \sim F_{25}$ :这一类的测试函数数量比其他类型的测试函数多,MCDE效果明显好于JADE,CoDE和CoBiDE算法。根据上述4点结论和表1的Wilcoxon's检验可以得出MCDE算法在4类测试函数里表现出显著的效果。最后可得,MCDE算法在 $D=30$ 维的实验结果比JADE,CoDE和CoBiDE分别要差3,5,4个测试函数,要好13,10,13个测试函数,相似9,10,8个测试函数。

### 3.2 与相关进化算法比较

为了进一步评估MCDE算法,将其与CLPSO(Comprehensive Learning Particle Swarm Optimizer)<sup>[22]</sup>,CMA-ES(Completely derandomized self-adaptation in evolution strategies)<sup>[23]</sup>,GL-25

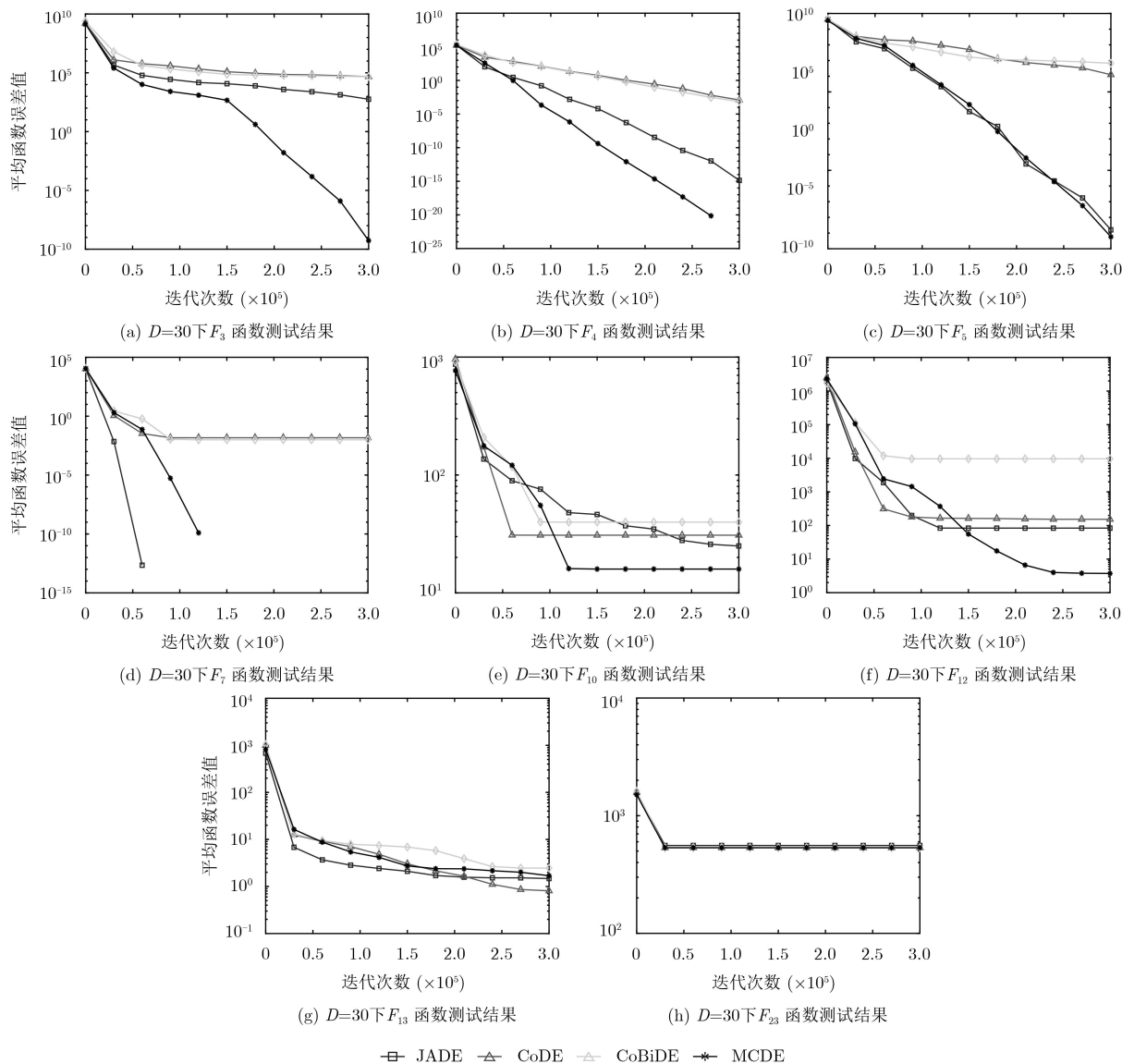


图2 4种演化算法在8个测试函数上的平均函数误差

(Global and local real-coded genetic algorithms)<sup>[24]</sup>算法进行比较。CLPSO算法是局部版本的PSO (Particle Swarm Optimizer)算法, 采用新的学习策略机制。CMA-ES算法是采用协方差矩阵自适应进化策略, 主要用于解决连续优化问题。GL-25算法是基于新的交叉算子的全局和局部实数编码遗传算法。表4提供了MCDE, CLPSO, CMA-ES和GL-25算法在 $D=30$ 维和最大评估次数 $\text{MaxFES}=300000$ 的实验结果, 表中数值的形式是平均误差 $\pm$ 标准偏差, 标记“+/-/ $\approx$ ”分别表示比较算法性能好于MCDE、差于MCDE和近似于MCDE。

根据表4给出的数据, 本文可以得出: MCDE算法在单峰类测试函数( $F_2 \sim F_5$ )的效果最为突出, 比其他演化算法的平均误差有数量级的小。在基本型多峰函数中,  $F_{10}$ 和 $F_{12}$ 比其他算法明显更优。对于扩展型多峰函数和复合型函数的测试, 绝大部分函数( $F_{14}, F_{16}, F_{17}, F_{21}, F_{23}, F_{24}$ 和 $F_{25}$ )效果显著。

最后可得, MCDE算法在 $D=30$ 维的实验结果比CLPSO, CMA-ES和GL算法分别要差2, 5, 1个测试函数, 要好19, 15, 21个测试函数, 相似4, 5, 3个测试函数。

#### 4 结论

本文算法MCDE提出多种群策略集成机制, 包括局部变异策略和全局变异策略, 以此来扩大种群的搜索范围。在进化过程中, 通过协方差学习矩阵将初始化坐标系统适当将目标个体与变异个体旋转。在协方差学习期间, 通过Lehmer均值和Power均值分别来继承上代优秀的交叉概率CR和缩放因子 $F$ 。

在CEC 2005测试函数上, 将MCDE算法与JADE, CoDE和CoBiDE比较, 发现对于求解全局优化问题有显著的效果。进一步验证本文算法, 将其与其他类型的演化算法CLPSO, CMA-ES, GL-25

表4 30次独立运行在CLPSO, CMA-ES, GL-25, MCDE最优解平均值及标准差

Function	CLPSO	CMA-ES	GL-25	MCDE
$F_1$	0.00E+00 $\pm$ 0.00e+00 $\approx$	1.58E-25 $\pm$ 3.35E-26-	5.60E-27 $\pm$ 1.76E-26-	<b>0.00E+00<math>\pm</math>0.00E+00</b>
$F_2$	8.40E+02 $\pm$ 1.90E+02-	1.12E-24 $\pm$ 2.93E-25-	4.04E+01 $\pm$ 6.28E+01-	<b>8.49E-28<math>\pm</math>3.75E-28</b>
$F_3$	1.42E+07 $\pm$ 4.19E+06-	5.54E-21 $\pm$ 1.69E-21+	2.19E+06 $\pm$ 1.08E+06-	<b>2.74E-12<math>\pm</math>2.82E-11</b>
$F_4$	6.99E+03 $\pm$ 1.73E+03-	9.15E+05 $\pm$ 2.16E+06-	9.07E+02 $\pm$ 4.25E+02-	<b>7.57E-22<math>\pm</math>4.26E-21</b>
$F_5$	3.86E+03 $\pm$ 4.35E+02-	2.77E-10 $\pm$ 5.04E-11+	2.51E+03 $\pm$ 1.96E+02-	<b>5.38E-10<math>\pm</math>7.12E-10</b>
$F_6$	4.16E+00 $\pm$ 3.48E+00-	4.78E-01 $\pm$ 1.32E+00-	2.15E+01 $\pm$ 1.17E+00-	<b>3.19E-01<math>\pm</math>1.09E-01</b>
$F_7$	4.51E-01 $\pm$ 8.47E-02-	1.82E-03 $\pm$ 4.33E-03-	2.78E-02 $\pm$ 3.62E-02-	<b>1.52E-03<math>\pm</math>4.11E-03</b>
$F_8$	2.09E+01 $\pm$ 4.41E-02-	2.03E+01 $\pm$ 5.72E-01+	2.09E+01 $\pm$ 5.94E-02-	2.09E+01 $\pm$ 4.21E-02
$F_9$	0.00e+00 $\pm$ 0.00e+00+	4.45E+02 $\pm$ 7.12E+01-	2.45E+01 $\pm$ 7.35E+00-	2.64E-07 $\pm$ 5.87E-07
$F_{10}$	1.04E+02 $\pm$ 1.53E+01-	4.63E+01 $\pm$ 1.16E+01-	1.42E+02 $\pm$ 6.45E+01-	<b>2.28E+01<math>\pm</math>4.27E+00</b>
$F_{11}$	2.60E+01 $\pm$ 1.63E+00-	7.11E+00 $\pm$ 2.14E+00+	3.27E+01 $\pm$ 7.79E+00-	1.51E+01 $\pm$ 6.81e+00
$F_{12}$	1.79E+04 $\pm$ 5.24E+03-	1.26E+04 $\pm$ 1.74E+04-	6.53E+04 $\pm$ 4.69E+04-	<b>2.12E+03<math>\pm</math>1.34E+03</b>
$F_{13}$	2.06E+00 $\pm$ 2.15E-01-	3.43E+00 $\pm$ 7.60E-01-	6.23E+00 $\pm$ 4.88E+00-	<b>1.74E+00<math>\pm</math>2.04E-01</b>
$F_{14}$	1.28E+01 $\pm$ 2.48E-01-	1.47E+01 $\pm$ 3.31E-01-	1.31E+01 $\pm$ 1.84E-01-	<b>1.23E+01<math>\pm</math>2.66E-01</b>
$F_{15}$	5.77E+01 $\pm$ 2.76E+01-	5.55E+02 $\pm$ 3.32E+02-	3.04E+02 $\pm$ 1.99E+01+	4.00E+02 $\pm$ 1.09E+02
$F_{16}$	1.74E+02 $\pm$ 2.82E+01-	2.98E+02 $\pm$ 2.08E+02-	1.32E+02 $\pm$ 7.60E+01-	<b>5.37E+01<math>\pm</math>3.01E+01</b>
$F_{17}$	2.46E+02 $\pm$ 4.81E+01-	4.43E+02 $\pm$ 3.34E+02-	1.61E+02 $\pm$ 6.80E+01-	<b>6.36E+01<math>\pm</math>6.41E+01</b>
$F_{18}$	9.13E+02 $\pm$ 1.42E+00-	9.04E+02 $\pm$ 3.01E-01 $\approx$	9.07E+02 $\pm$ 1.48E+00-	<b>9.03E+02<math>\pm</math>6.01E-01</b>
$F_{19}$	9.14E+02 $\pm$ 1.45E+00-	9.16E+02 $\pm$ 6.03E+01-	9.06E+02 $\pm$ 1.24E+00-	<b>9.03E+02<math>\pm</math>2.31E-01</b>
$F_{20}$	9.14E+02 $\pm$ 3.62E+00-	9.04E+02 $\pm$ 2.71E-01 $\approx$	9.07E+02 $\pm$ 1.35E+00-	<b>9.03E+02<math>\pm</math>2.45E-01</b>
$F_{21}$	5.00E+02 $\pm$ 3.39E-13 $\approx$	5.00E+02 $\pm$ 2.68E-12 $\approx$	5.00E+02 $\pm$ 4.83E-13 $\approx$	<b>5.00E+02<math>\pm</math>4.51E-14</b>
$F_{22}$	9.72E+02 $\pm$ 1.20E+01-	8.26E+02 $\pm$ 1.46E+01+	9.28E+02 $\pm$ 7.04E+01-	8.69E+02 $\pm$ 1.89E+01
$F_{23}$	5.34E+02 $\pm$ 2.19E-04 $\approx$	5.36E+02 $\pm$ 5.44E+00 $\approx$	5.34E+02 $\pm$ 4.66E-04 $\approx$	<b>5.34E+02<math>\pm</math>2.49E-13</b>
$F_{24}$	2.00E+02 $\pm$ 1.49E-12 $\approx$	2.12E+02 $\pm$ 6.00E+01-	2.00E+02 $\pm$ 5.52E-11 $\approx$	<b>2.00E+02<math>\pm</math>2.90E-14</b>
$F_{25}$	2.00E+02 $\pm$ 1.96E+00+	2.07E+02 $\pm$ 6.07E+00 $\approx$	2.17E+02 $\pm$ 1.36E-01-	2.09E+02 $\pm$ 2.78E-01
+/-/ $\approx$	<b>2/19/4</b>	<b>5/15/5</b>	<b>1/21/3</b>	

在 $D=30$ 维的比较, MCDE算法效果最佳。在运行时间上, 本文算法介于比较算法的上流部分。综上所述, 无论在求解精度还是收敛速度方面都有所提高, 因此MCDE算法可以执行。

### 参考文献

- [1] STORN R and PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341–359. doi: [10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328).
- [2] PARK S Y and LEE J J. Stochastic opposition-based learning using a beta distribution in differential evolution[J]. *IEEE Transactions on Cybernetics*, 2016, 46(10): 2184–2194. doi: [10.1109/TCYB.2015.2469722](https://doi.org/10.1109/TCYB.2015.2469722).
- [3] ZHANG Xin and ZHANG Xiu. Improving differential evolution by differential vector archive and hybrid repair method for global optimization[J]. *Soft Computing*, 2017, 21(23): 7107–7116. doi: [10.1007/s00500-016-2253-4](https://doi.org/10.1007/s00500-016-2253-4).
- [4] SALLAM K M, ELSAYED S M, SARKER R A, *et al.* Landscape-based adaptive operator selection mechanism for differential evolution[J]. *Information Sciences*, 2017, 418–419: 383–404. doi: [10.1016/j.ins.2017.08.028](https://doi.org/10.1016/j.ins.2017.08.028).
- [5] MOHAMED A W and SUGANTHAN P N. Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation[J]. *Soft Computing*, 2018, 22(10): 3215–3235. doi: [10.1007/s00500-017-2777-2](https://doi.org/10.1007/s00500-017-2777-2).
- [6] YANG Ming, LI Changhe, CAI Zhihua, *et al.* Differential evolution with auto-enhanced population diversity[J]. *IEEE Transactions on Cybernetics*, 2015, 45(2): 302–315. doi: [10.1109/TCYB.2014.2339495](https://doi.org/10.1109/TCYB.2014.2339495).
- [7] MALLIPEDDI R, SUGANTHAN P N, PAN Q K, *et al.* Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. *Applied Soft Computing*, 2011, 11(2): 1679–1696. doi: [10.1016/j.asoc.2010.04.024](https://doi.org/10.1016/j.asoc.2010.04.024).
- [8] BREST J, GREINER S, BOSKOVIC B, *et al.* Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646–657. doi: [10.1109/TEVC.2006.872133](https://doi.org/10.1109/TEVC.2006.872133).
- [9] QIN K A, HUANG V L, and SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 398–417. doi: [10.1109/TEVC.2008.927706](https://doi.org/10.1109/TEVC.2008.927706).
- [10] ZHANG Jingqiao and SANDERSON A C. JADE: Adaptive differential evolution with optional external archive[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945–958. doi: [10.1109/TEVC.2009.2014613](https://doi.org/10.1109/TEVC.2009.2014613).
- [11] WANG Yong, CAI Zixing, and ZHANG Qingfu. Differential evolution with composite trial vector generation strategies and control parameters[J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 55–66. doi: [10.1109/TEVC.2010.2087271](https://doi.org/10.1109/TEVC.2010.2087271).
- [12] WANG Yong, LI Hanxiong, HUANG Tingwen, *et al.* Differential evolution based on covariance matrix learning and bimodal distribution parameter setting[J]. *Applied Soft Computing*, 2014, 18: 232–247. doi: [10.1016/j.asoc.2014.01.038](https://doi.org/10.1016/j.asoc.2014.01.038).
- [13] WANG Jiahai, LIAO Jianjun, ZHOU Ying, *et al.* Differential evolution enhanced with multiobjective sorting-based mutation operators[J]. *IEEE Transactions on Cybernetics*, 2017, 44(12): 2792–2805. doi: [10.1109/TCYB.2014.2316552](https://doi.org/10.1109/TCYB.2014.2316552).
- [14] WU Guohua, MALLIPEDDI R, SUGANTHAN P N, *et al.* Differential evolution with multi-population based ensemble of mutation strategies[J]. *Information Sciences*, 2016, 329: 329–345. doi: [10.1016/j.ins.2015.09.009](https://doi.org/10.1016/j.ins.2015.09.009).
- [15] XUE Yu, JIANG Jiongmeng, ZHAO Binping, *et al.* A self-adaptive artificial bee colony algorithm based on global best for global optimization[J]. *Soft Computing*, 2018, 22(9): 2935–2952. doi: [10.1007/s00500-017-2547-1](https://doi.org/10.1007/s00500-017-2547-1).
- [16] KIRAN M S and BABALIK A. Improved artificial bee colony algorithm for continuous optimization problems[J]. *Journal of Computer and Communications*, 2014, 2: 108–116. doi: [10.4236/jcc.2014.24015](https://doi.org/10.4236/jcc.2014.24015).
- [17] DU Wenbo, YING Wen, YAN Gang, *et al.* Heterogeneous strategy particle swarm optimization[J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2017, 64(4): 467–471. doi: [10.1109/TCSII.2016.2595597](https://doi.org/10.1109/TCSII.2016.2595597).
- [18] DONG Wenyong, KANG Lanlan, and ZHANG Wensheng. Opposition-based particle swarm optimization with adaptive mutation strategy[J]. *Soft Computing*, 2017, 21(17): 5081–5090. doi: [10.1007/s00500-016-2102-5](https://doi.org/10.1007/s00500-016-2102-5).
- [19] HAN Honggui, LU Wei, and QIAO Junfei. An adaptive multiobjective particle swarm optimization based on multiple adaptive methods[J]. *IEEE Transactions on Cybernetics*, 2017, 47(9): 2754–2767. doi: [10.1109/TCYB.2017.2692385](https://doi.org/10.1109/TCYB.2017.2692385).
- [20] HASSANAT A B A and ALKAFaweEN E. On enhancing genetic algorithms using new crossovers[J]. *International Journal of Computer Applications in Technology*, 2018, 55(3): 202–212. doi: [10.1504/IJCAT.2017.10005868](https://doi.org/10.1504/IJCAT.2017.10005868).
- [21] SUGANTHAN P N, HANSEN N, LIANG J J, *et al.* Problem definitions and evaluation criteria for the CEC

- 2005 special session on real-parameter optimization[R]. Technical Report, KanGAL Report #2005005, 2005: 1–50.
- [22] 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. doi: [10.1109/TEVC.2005.857610](https://doi.org/10.1109/TEVC.2005.857610).
- [23] HANSEN N and OSTERMEIER A. Completely derandomized self-adaptation in evolution strategies[J]. *Evolutionary Computation*, 2001, 9(2): 159–195. doi: [10.1162/106365601750190398](https://doi.org/10.1162/106365601750190398).
- [24] GARCÍA-MARTÍNEZ C, LOZANO M, HERRERA F, *et al.* Global and local real-coded genetic algorithms based on parent-centric crossover operators[J]. *European Journal of Operational Research*, 2008, 185(3): 1088–1113. doi: [10.1016/j.ejor.2006.06.043](https://doi.org/10.1016/j.ejor.2006.06.043).
- 杜永兆: 男, 1985年生, 副教授, 博士, 研究方向为智能计算、光学成像优化、医学图像处理.
- 范宇凌: 男, 1995年生, 硕士生, 研究方向为智能计算、图像处理.
- 柳培忠: 男, 1976年生, 副教授, 博士, 研究方向为智能计算、视觉媒体检索、深度学习、信息安全.
- 唐加能: 男, 1983年生, 副教授, 博士, 研究方向为智能计算、混沌同步和控制、网络同步和控制、信息安全、语音信号处理.
- 骆炎民: 男, 1975年生, 副教授, 博士, 研究方向为机器学习、图像处理、智能计算、模式识别.