

# 基于精英差异反馈策略的自适应差分进化算法

王 毅<sup>1</sup>, 彭钰博<sup>1</sup>, 黄 欣<sup>2</sup>, 郑宏志<sup>1</sup>, 张鹏辉<sup>1</sup>

(1. 西北大学信息科学与技术学院, 陕西西安 710127; 2. 西安交通大学第一附属医院, 陕西西安 710049)

**摘要:** 为提升差分进化的收敛性与多样性能力, 分析并归纳了影响差分进化算法性能的参数自适应调节机制与改进策略, 提出一种基于精英差异反馈策略的自适应差分进化算法. 首先, 通过种群个体适应度评估, 找出当前种群全局最优个体, 并与待进化个体判定生成全局精英差异; 其次, 提出基于多样性扰动邻域生成策略, 构建个体邻域, 并基于邻域最优个体与待进化个体判定生成邻域精英差异; 最后通过全局精英差异与邻域精英差异指导 DE (Differential Evolution) 算法交叉策略, 平衡了 DE 算法的收敛性与多样性. 同时, 本算法步长因子  $F$  采用自适应调整策略, 进而降低算法陷入搜索停滞的可能. 通过本文在 CEC2017 测试集上与其他主流改进算法相比, 实验结果表明该算法在寻优效率与收敛精度上显著优于对比方法.

**关键词:** 差分进化; 精英差异; 收敛性; 多样性; 邻域; 全局最优解

中图分类号: TP182; TP391

文献标识码: A

文章编号: 0372-2112(XXXX)XX-0001-11

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20210513

## Adaptive Differential Evolution Algorithm Based on Elite Differential Feedback Strategy

WANG Yi<sup>1</sup>, PENG Yu-bo<sup>1</sup>, HUANG Xin<sup>2</sup>, ZHENG Hong-zhi<sup>1</sup>, ZHANG Peng-hui<sup>1</sup>

(1. School of Information Science and Technology, Northwestern University, Xi'an, Shaanxi 710127, China;

2. The First Affiliated Hospital of Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

**Abstract:** In order to improve the algorithm convergence and diversity capabilities in differential evolution, the parameter adaptive adjustment mechanism and improvement strategy that affect the performance of differential evolution algorithm are analyzed and summarized, and an adaptive differential evolution algorithm based on elite differential feedback strategy is proposed. First, through population individual fitness assessment, find the global optimal individual of the current population, and determine the global elite difference with the individual to be evolved; second, propose a strategy based on the diversity perturbation neighborhood generation strategy to construct individual neighborhoods based on neighborhoods. The optimal individual and the individual to be evolved are determined to generate neighborhood elite differences; finally, the crossover strategy of DE algorithm is guided by global elite differences and neighborhood elite differences, which balances the convergence and diversity of DE algorithm. At the same time, the step factor  $F$  of this algorithm adopts an adaptive adjustment strategy, thereby reducing the possibility of the algorithm getting stuck in search. By comparing this article with other mainstream improved algorithms on the CEC2017 test set, the experimental results show that the algorithm is significantly better than the comparison method in terms of optimization efficiency and convergence accuracy.

**Key words:** differential evolution; elite difference; convergence; diversity; neighborhood; global optimal solution

## 1 引言

差分进化算法 (Differential Evolution, DE)<sup>[1]</sup> 在众多群体智能优化算法中表现优异, 体现在收敛性能好、控制参数较少、适用范围广和鲁棒性强等优点, 但在某些

情况下依然存在局部收敛能力弱、算法后期效率差等问题. 为此, 国内外许多学者从不同角度对 DE 算法进行了改进, 通过现有文献分析, 可以总结划分为以下三类:

收稿日期: 2021-04-21; 修回日期: 2021-08-05; 责任编辑: 梅志强

基金项目: 国家重点研发计划项目 (No.2018YFC1504705); 国家重大仪器专项 (No.42027806); 国家重点自然科学基金 (No.61731015); 国家自然科学基金 (No.61402517); 国家重点实验室基金 (No.2016ADL-DW0302); 陕西省重点研发项目 (No.2022GY-331)

第1类 参数自适应调节:在DE算法寻优过程中,大多数算法通过自适应地调整算法参数,动态改变搜索范围,增强算法的搜索精度<sup>[2]</sup>.然而为了获得不同阶段个体的最优参数,引入了历史信息,通过对先验知识的学习,并根据与个体当前状态的启发式信息来动态调整变异策略和相关参数,以提高算法多样性<sup>[3]</sup>.Chang等人用一种DE算法去搜索另一种DE算法的控制参数,通过建立两级DE,具有不同的变异和交叉方案,通过适应度评估结果反馈调整DE算法的控制参数,以达到寻优目的<sup>[4]</sup>.此外,利用两个随机个体之间的差异来生成一种新的步长因子<sup>[5]</sup>,将当前种群个体的分布情况反馈于步长因子,促使变异过程中个体突变步长更合理,解决了因步长较长而错失最优值的不足.

第2类 变异策略:在求解约束优化问题中,通过信用分配机制选择变异策略,根据信用值计算变异策略的质量并反馈于策略选择<sup>[6]</sup>.此外,还可通过择优选择变异策略通过找出比自身好的所有个体参与变异个体的生成,增加了算法的收敛速度<sup>[7]</sup>;而双变异策略骨架差分进化算法则通过选择因子随个体共同进化,执行当前最为适合的变异策略,克服了进化策略选择的盲目性,同时保持了骨架算法近似无参数的优点<sup>[8]</sup>.

第3类 种群结构改进:随着问题规模增大以及种群多样性的降低,从种群结构出发进行多方位改进自成一统.借鉴自组织映射网络的思想,提出一种自组织映射邻域的结构,并根据邻域生成变异个体,使DE算法搜索更精细<sup>[9]</sup>.将种群划分为若干子种群结构,通过子种群之间信息交互与协同进化,增强了个体的多样性<sup>[10]</sup>.通过主从多种群分布式框架<sup>[11]</sup>,将种群分为探索种群、开发种群和平衡种群三个种群协同进化,不同种群会根据进化状态估计自适应地选择合适变异策略,达到充分利用个体和整个种群的反馈信息;基于竞争的分布式差分进化算法<sup>[12]</sup>,从种群结构上将种群分为多个子种群,基于竞争策略来保证子种群之间全面交互,设计反向入侵算子和交叉入侵算子,反向入侵算子实现优势子种群到劣势子种群的入侵,交叉入侵算子有助于子种群信息融合,维持了种群多样性.

通过对上述研究成果的总结与分析发现,大多数算法对变异策略进行了改进,生成了更适合种群进化的变异个体,然而却忽略了DE中交叉策略的重要性.交叉策略控制待进化个体与变异个体基因组合,决定了实验个体基因组成,很大程度上影响了子代个体的优劣程度,因此,对交叉策略的研究至关重要.

本研究在以上理论分析基础上,基于交叉策略,提出一种基于精英差异反馈策略的自适应差分进化算法(adaptive Differential Evolution algorithm based on Elite Differential Feedback strategy, EDF-DE),通过精英差异

来对待进化个体和变异个体的交叉进行控制,变异个体中符合精英差异的优秀基因将会被选择进行交叉,使交叉更有目的性;传统交叉在标准DE中为二项式交叉,通过交叉率CR进行基因选择,存在因选择较差基因而丧失优秀基因的缺点,而精英差异交叉克服了其不足;全局精英差异和邻域精英差异协同交互,维持种群多样性的同时提高算法收敛性,促使算法进行深度与广度搜索.

## 2 EDF-DE 算法

本算法思想示意图如图1所示,可以看出,EDF-DE主要包括6个部分:初始化、精英差异生成、步长因子更新、变异、交叉与选择操作.算法由初始化种群开始,经精英差异生成全局精英与邻域精英差异(具体过程为:对当代种群所有个体适应度评估找出全局精英,同时针对每一个待进化个体,通过本文邻域生成策略构建邻域,结合适应度值以邻域内最优个体作为邻域精英;最后使用精英差异生成方法将待进化个体分别与全局精英和邻域精英判定生成精英差异),更新步长因子 $F$ ,经具有外部档案变异策略生成变异个体,采用二项式交叉和本文所提精英差异交叉两种策略择一生成实验个体,通过选择策略更新种群,若未满足终止条件,将进行下一次迭代.二项式交叉保留了传统交叉随机性的优点,精英差异交叉具有学习功能,向全局最优和邻域最优学习,两种策略交替运行、协同合作,平衡了算法的收敛性与多样性.

### 2.1 具有外部档案变异策略

本文变异策略采用JADE<sup>[13]</sup>算法具有外部档案变

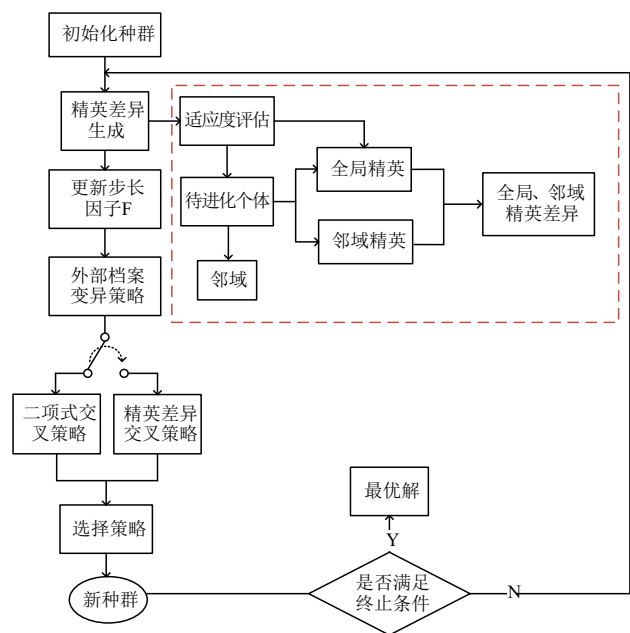


图1 算法思想示意图

异策略,具体公式如式下:

$$v_{i,j} = x_i + F \times (x_{best}^p - x_i) + F \times (x_{r1} - x_{r2}) \quad (1)$$

其中  $x_i$  为父代个体,  $x_{best}^p$  为种群前 top 个个体中随机选择,  $x_{r1}$  为当前种群随机选择,  $x_{r2}$  为当前种群与进化失败个体档案并集中选择.

## 2.2 参数自适应调节

$F$  是 DE 算法的步长因子,用来控制两个基向量差的缩放程度. 前期需要进行全局勘探,因此需要较大步长才可以探索到较多区域;而在后期,大部分个体都围绕到种群最优解附近,因此需要在局部精细搜索进而获得局部最优,此时,  $F$  的值应该较小. 为此本文提出一种伴随迭代次数  $F$  自适应变化的方法,具体公式如下:

$$F = \max[F_0 \times (1 - \frac{g}{G_{\max}}), F_1] \quad (2)$$

其中  $F$  为自适应变化的步长因子,  $F_0$  为步长因子上界,  $F_1$  为步长因子下界,  $F \in [F_1, F_0]$ .  $g$  为当前迭代次数,  $G_{\max}$  为算法最大迭代次数. 由公式可知,在迭代前期  $F$  接近于步长因子上限,实现全局勘探;随着迭代次数增加,  $F$  逐渐自适应地减少,最终到达步长因子下限,进行局部开发,实现精细搜索.

## 2.3 全局精英差异

精英差异是一种离散指引算子,全局精英差异旨在加入全局最优个体的影响机制,促进个体在全局最优值附近开发,以精英差异值为指导,进行矢量化交叉,具体示意如图 2 所示(图中精英个体和待进化个体的数值为迭代过程中的随机数值). 以下以 4 维问题的精英差异生成为例. 图中分别为待进化个体、精英个体和精英差异,其中待进化个体的 4 个维度分别设为: 3、2、17、-8,精英个体代表当前迭代种群中最优个体或邻域中最优个体,分别设为: 7、-3、12、-8. 精英个体与待变异个体对应维度求差,若差值大于 0,则对应精英差异维度值为 1;若差值小于 0,则对应精英差异维度值为 -1;若差值等于 0,则对应精英差异维度值为 0.

初始化全局精英差异  $E_g = \{e_1, e_2, \dots, e_{NP}\}$ , NP 代表种群数量. 每个精英差异  $e_i, e_i = \{e_{i1}, e_{i2}, \dots, e_{ij}\}$ , 其中  $j$  代表种群个体维度.  $E_g$  的生成方法如下:

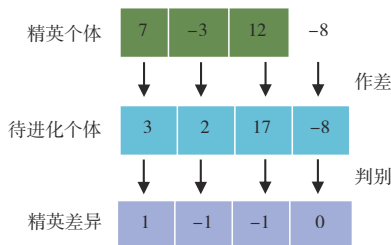


图 2 精英差异生成示例

求解维度差,计算种群中个体  $x_i$  与全局最优个体  $x\_gbest$  之间的维度差记作  $Cg$ ,计算方法如下:

$$Cg(i,j) = x\_gbest_{i,j} - x_{i,j} \quad (3)$$

式(3)可求得个体与最优值之间每个维度的差值,根据  $Cg$  生成全局精英差异:

$$Eg(i,j) = \begin{cases} 1, & \text{if } Cg(i,j) > 0 \\ -1, & \text{if } Cg(i,j) < 0 \\ 0, & \text{else} \end{cases} \quad (4)$$

如式(3)和式(4)可求出全局精英差异. 待进化个体在某维度差异值为 1 时,代表全局最优解在此维度大于待进化个体,待进化个体在此维度将会向正方向进化;当差异值为 -1 时,将会向负方向进化;差异值为 0 时,则代表待进化个体在此维度不需改变.

## 2.4 邻域精英差异

全局精英差异指导个体向最优个体所在方向进化,当所有个体都学习最优个体时,种群多样性就会逐步降低,且容易陷入浅层搜索的死循环. 因此本文提出一种邻域精英差异结合全局精英差异对个体交叉进行指导. 并提出基于个体多样性扰动构建邻域模型,从个体所在邻域中找出最优值,与待变异个体判定求邻域精英差异. 其中多样性扰动构建邻域通过权重因子加权欧式距离来实现,有效维护个体维度多样性,促进算法进行广度搜索开发.

### 2.4.1 加权欧氏距离

为了保持种群在迭代过程中的多样性,引入个体维度权重因子  $p$  对距离计算进行扰动,权重因子较大时,代表个体多样性较大,反之个体多样性较小.  $mean\_d = \{mean\_d_1, mean\_d_2, \dots, mean\_d_j\}$  代表每个维度所有个体的平均值,  $p_i = \{p_{i1}, p_{i2}, \dots, p_{ij}\}$  分别代表第  $i$  个个体在第  $j$  个维度的权重因子,维度均值  $mean\_d$  和权重因子  $p$  的计算公式如下:

$$mean\_d_j = (\sum_{i=1}^{pop} x_{i,j}) / pop \quad (5)$$

$$p_{i,j} = (|x_{i,j} - mean\_d_j| + |mean\_d_j|) / |mean\_d_j| \quad (6)$$

其中  $j$  代表个体的维度,  $i$  代表种群个体,如式(6)所示,当个体某个维度与对应维度均值之间的差异越大,权重因子  $p$  会越大. 由此可见,  $p$  代表个体的维度偏离维度均值的程度,当个体所有的维度都保持较大的  $p$  值,表示此个体较偏离大部分个体聚集处,即具备较大多样性.

本文将维度权重因子  $p$  应用于欧氏距离,用来计算个体之间的距离. 定义于欧氏空间之中,在  $N$  维空间向量  $x = \{x_1, \dots, x_n\}$  和向量  $y = \{y_1, \dots, y_n\}$  之间的距离为:

$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (7)$$

将 $p$ 应用于欧氏距离,构成多样性加权欧式距离公式如下:

$$d_p(\mathbf{x}_c, \mathbf{x}_i) = \left( \sum_{j=1}^d \frac{1}{p_{i,j}} \|\mathbf{x}_{c,j} - \mathbf{x}_{i,j}\|^2 \right)^{1/2} \quad (8)$$

式(8)用于计算个体 $x_c$ 和个体 $x_i$ 的加权欧氏距离, $p_{i,j}$ 对应于第 $i$ 个个体的第 $j$ 个维度的权重因子. $p_{i,j}$ 较大时,个体之间的距离 $d_p$ 会变小,即求 $x_c$ 与个体 $x_i$ 的距离时,当 $x_i$ 对应的 $p$ 值较大时,说明个体 $x_i$ 距种群聚集点距离较远,则个体 $x_c$ 与 $x_i$ 之间的加权欧式距离会更小,因此实现了通过个体多样性来对欧氏距离加权扰动.

#### 2.4.2 多样性扰动构建邻域策略

构建邻域按式(8)多样性加权欧式距离计算邻域中心与种群个体之间的距离,经权重因子扰动,距邻域中心较近个体多样性较强,选择与邻域中心距离小于邻域半径的个体构建邻域,以此来实现多样性扰动构建邻域的目的.创建个体邻域示意图如图3所示,绿色个体为邻域中心,红色个体为多样性较大的个体,黑色个体为多样性较小的个体.多样性较小的黑色个体相应的 $p$ 值小,经扰动与邻域中心计算的距离会变大,多样性较大的红色个体 $p$ 值大,与邻域中心计算的距离会变小,最终得到经多样性扰动所构建的邻域.

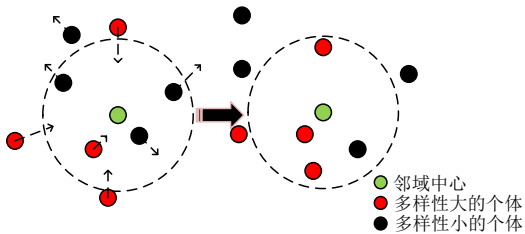


图3 多样性扰动邻域构建示意图

#### 2.4.3 邻域精英差异生成策略

在本文邻域精英差异生成过程中,首先针对每个个体采用多样性扰动邻域构建策略构建邻域,以个体 $x_i$ 作为邻域中心,给定邻域半径 $m$ ,构建邻域中心为 $x_c$ ,邻域半径为 $m$ 的邻域,记作 $U(x_c, m)$ .对邻域 $U(x_c, m)$ 中个体进行适应度值从低到高的排序,选择邻域中适应度值最小的个体为邻域精英 $x\_lbest$ ,由式(9)计算 $x\_lbest$ 与种群个体 $x_i$ 之间差异量 $Cl$ ,由式(10)计算邻域精英差异 $EI$ .

$$Cl(i, j) = x\_lbest_{i,j} - x_{i,j} \quad (9)$$

$$EI(i, j) = \begin{cases} 1, & \text{if } Cl(i, j) > 0 \\ -1, & \text{if } Cl(i, j) < 0 \\ 0, & \text{else} \end{cases} \quad (10)$$

通过式(9)和式(10)计算出邻域精英差异的值,邻域精英差异生成策略伪代码见算法1.

#### 算法1 邻域精英差异生成策略伪代码

```

输入:种群规模 pop,个体维度 d,
      个体适应度值 fitness,邻域半径 m
1: for j = 1 : d
2:   按式(5)计算每个维度的均值 mean_dj
3: end
4: for i = 1 : pop
5:   for j = 1 : d
6:     按式(6)计算个体维度权重因子 p_ij
7:   end
8: end
9: for i = 1 : pop
10:  按式(8)计算邻域中心与其他个体
    之间的加权欧氏距离 dp
11: end
12: for i = 1 : pop
13:  针对个体 x_i 为邻域中心,筛选欧式距离前
    m 个个体简历领域,并从 m 个邻域个体中
    选取最优个体作为邻域精英,通过式(9)
    和式(10)计算出邻域精英差异 EI
14: end
输出:EI

```

### 2.5 算法描述

#### 2.5.1 精英差异交叉策略

精英差异交叉策略以全局精英差异和邻域精英差异为指导,从待变异个体 $x_i$ 和变异个体 $v$ 之间选择分量形成实验个体 $u$ ,取代了二项式交叉采用交叉率 $CR$ 控制交叉.因全局精英差异和邻域精英差异结构相同,故用 $E$ 统一代表 $E_g$ 和 $E_l$ ,交叉公式示为:

$$u(i, j) = \begin{cases} v_{i,j}, & \text{if } v_{i,j} - x_{i,j} > 0 \text{ and } E(i, g) = 1 \\ v_{i,j}, & \text{if } v_{i,j} - x_{i,j} < 0 \text{ and } E(i, g) = -1 \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (11)$$

由式可看出,只会选择满足条件的变异个体基因组成实验向量,通过变异个体与待变异个体求差判定正负,再与精英差异值进行比对,可以有效从变异个体中选择出可以帮助待进化个体向精英个体学习的基因.

#### 2.5.2 算法流程

EDF-DE算法生成全局精英差异与邻域精英差异,对算法交叉阶段进行反馈控制,为平衡经典二项式交叉与精英差异反馈交叉选择比例,我们引入超参数 $v$ 控制选择传统二项式交叉或精英差异反馈交叉.用超参数 $t$ 平衡全局精英差异和邻域精英差异,进而平衡算法的收敛性与多样性,算法伪代码见算法2.

**算法2 EDF-DE算法的伪代码**

**输入:**种群规模 pop, 迭代次数 Maxiter, 步长因子上界  $F_0$ ,

步长因子下界  $F_1$ , 超参数  $v$ , 超参数  $t$ , 交叉率 CR

```

1: 初始化:种群个体
2:  for j = 1: Maxiter Do
3:     for i = 1 : pop
4:         计算种群个体适应度值 fitness
5:     end
6:  计算种群全局最优个体 x_gbest, fitnessbest
7:     for i = 1: pop
8:         通过式(3)和式(4)计算
           全局精英差异 Eg 值
9:     end
10:  通过算法 1 计算邻域精英差异 EI
11:  通过式 (2) 更新步长因子 F
12:     for i = 1 : pop Do
13:         按式(1)进行变异操作,生成变异个体
14:     end
15:     for i = 1 : pop
16:         if rand < v
17:             经典二项式交叉策略
18:         else
19:             精英差异交叉策略
20:             if rand > t
21:                 Eg 结合式(11)指导交叉操作
22:             else
23:                 EI 结合式(11)指导交叉操作
24:             end
25:         end
26:     end
27:  选择操作
28:  更新种群
29: end

```

**输出:**fitnessgbest

### 3 实验与结果

#### 3.1 实验环境配置及参数设置

为了评估 EDF-DE 算法性能, 实验选取 IEEE CEC 2017 的 30 个基准函数包括单峰、多峰、混合以及复合函数四方面进行多方位测试, 其中测试函数  $f_1 \sim f_3$  为单峰函数,  $f_4 \sim f_{10}$  为多峰函数,  $f_{11} \sim f_{20}$  为混合函数,  $f_{21} \sim f_{30}$  为复合函数. 实验环境采用 Inter i7-6700CPU, 主频 3.40 GHz 和内存 8 GB 的 PC 机, 操作系统为 64 位的 Windows 10, 编程语言采用 MATLAB2019B. 本文算法 EDF-DE 相关参数设置如下: 步长因子下界  $F_1 = 0.1$ , 步长因子上界  $F_0 = 0.8$ , 邻域半径  $m = 7$ , 超参数  $v = 0.5$ , 超参数  $t = 0.7$ , 交叉概率  $CR = 0.5$ . 对比算法选取近年来主流的 DE 改进算法, 包括基于择优学习策略的差分进化算

法<sup>[7]</sup> (Perferred-Learning-Based Differential Evolution Algorithm, PL-DE)、双变异策略的自适应骨架差分进化算法<sup>[8]</sup> (Self-adaptive bare-bones differential evolution based on bi-mutation strategy, SMGBDE)、具有可选外部档案的自适应差分进化算法<sup>[13]</sup> (adaptive differential evolution with optional external archive, JADE)、基于状态估计反馈的策略自适应差分进化算法<sup>[14]</sup> (Strategy Self-adaptive Differential Evolution Algorithm Based on State Estimation Feedback, SEFDE) 和具有复合试验向量生成策略和控制参数的差分进化算法<sup>[15]</sup> (Differential evolution with composite trial vector generation strategies and control parameters, CODE), 其中 PL-DE、SMGBDE、SEFDE 是近几年提出的 DE 改进算法, 相较于其他改进算法更具优势, 而 JADE 和 CODE 算法是经典的 DE 算法的改进, 有显著的性能, 在众多实验中被用作对比算法.

为了公平比较各算法性能, 所有算法的公共参数设置相同. 对 CEC2017 测试集中的每个函数独立求解 30 次, 维度  $D = 30$ , 种群数量  $NP = 150$ , 最大迭代次数为  $Maxiter = 2000$ , 对比算法其他参数采用相应文献中推荐的最佳设置.

#### 3.2 实验结果与分析

##### 3.2.1 求解精度及稳定性分析

本次将 30 次实验结果取平均值和标准差, 并通过平均值评价算法收敛精度, 标准差评价算法稳定性, 如表 3 所示. 其中加黑数值表示各类算法在相应测试函数上的最优值, 最后一行为相应算法在所有测试函数上取得最优均值结果个数.

从表 3 可以看出, EDF-DE 算法在  $f_1 \sim f_{30}$  的测试函数上取得 16 个最优均值的结果, 而 SEFDE、PL-DE、JADE、CODE 和 SMGBDE 分别在 4 个、4 个、6 个、11 和 2 个测试函数上取得最优均值, 表明 EDF-DE 算法具有更好的综合寻优能力. 为了进一步分析算法性能, 这里对 6 类算法在单峰、多峰、混合与复合等 30 个测试函数上进行求解精度与稳定性分析.

(1) 单峰函数 ( $f_1 \sim f_3$ ): 从实验结果可以看出, CODE 和 SMGBDE 在 3 个测试函数中分别取得了 2 个最优均值结果; EDF-DE 表现不太理想, 与 JADE、SEFDE 和 PL-DE 算法分别取得 1 个最优均值; 然而 EDF-DE 在  $f_2, f_3$  两个单峰函数上取得了接近最优均值的结果, 其中在函数  $f_2$  上, 比 SEFDE、PL-DE 和 JADE 分别提高了 2 个、5 个和 9 个数量级.

在标准差所体现的稳定性上, JADE 在  $f_1$ 、SMGBDE 在  $f_2$ 、CODE 在  $f_3$  上分别获得了 1 个函数的最优稳定性. EDF-DE 虽未取得最优稳定性, 但在  $f_2$  和  $f_3$  上稳定性高于 SEFDE、PL-DE 和 JADE; 具体来讲, 在  $f_2$  上, EDF-DE

与CODE性能持平,相较于SEFDE、PL-DE和JADE的标准差分别提高4个、7个和10个数量级;在 $f_3$ 上,EDF-DE与SMGBDE相较差了2个数量级,但比SEFDE、PL-DE和JADE分别提高3个、3个和4个数量级.总体看来,在单峰测试函数上,EDF-DE表现出了不错的优化性能.

(2)多峰函数( $f_4 \sim f_{10}$ ):在7个多峰函数中,EDF-DE在4个多峰函数上取得最优均值结果,相较于其他的5种对比算法具有显著优势.这是由于多峰函数具有多个局部最优解,而EDF-DE的精英差异反馈策略能够维持种群多样性,促进算法在多局部进行超精细搜索.

在标准差上,EDF-DE、JADE和CODE算法皆取得了

2个函数的最优标准差,其他算法性能较弱;而EDF-DE在 $f_3$ 和 $f_7$ 上不仅取得了最好的标准差而且还获得了最优均值,这表明了EDF-DE算法在多峰函数上既有显著的全局寻优能力与求解精度,而且还具有较好的稳定性.

(3)混合函数( $f_{11} \sim f_{20}$ ):在10个混合函数上,EDF-DE在 $f_{11}$ 、 $f_{14}$ 、 $f_{15}$ 、 $f_{16}$ 、 $f_{17}$ 和 $f_{19}$ 这6个函数上获得最优均值,并在 $f_{20}$ 上也最为接近最优均值结果.CODE则在其他三个函数( $f_{13}$ 、 $f_{18}$ 、 $f_{20}$ )取得最优值,JADE在 $f_{12}$ 上取得最优值,而SEFDE、PL-DE与SMGBDE均没有获得最优值.算法在混合函数的寻优表现说明EDF-DE算法的邻域精英差异反馈策略的有效性,可以有效促进算法进行广度搜索,跳出局部最优解.

表1 6种算法在30个测试函数上运行结果

Fun	评价指标	EDF-DE	SEFDE		PL-DE		JADE		CODE		SMGBDE	
$f_1$	平均值	1.0000e+02	1.0000e+02	=	1.0000e+02	=	1.0000e+02	=	1.0000e+02	=	1.2143e+03	+
	标准差	3.2276e+00	2.3153e-10		3.2609e-13		4.5706e-15		0.1636e-03		2.1389e+03	
$f_2$	平均值	2.1286e+02	4.7085e+04		7.6575e+07		1.5719e+11		2.0997e+02		2.0000e+02	-
	标准差	2.1358e+01	1.3672e+05	+	4.1884e+08	+	7.1360e+11	+	2.0145e+01	-	2.6732e-05	
$f_3$	平均值	3.0525e+02	1.3281e+04		6.6493e+03		9.6592e+02		3.0000e+02		3.0000e+02	-
	标准差	5.0865e+00	2.4872e+03	+	2.3732e+03	+	1.2884e+04	+	9.0581e-05	-	0.1140e-02	
$f_4$	平均值	4.6542e+02	5.0781e+02		4.3954e+02		4.2920e+02		4.5893e+02		4.8899e+02	+
	标准差	2.6016e+01	1.4452e+01	+	3.3608e+01	-	3.1397e+01	-	1.4094e+00	-	2.9357e+01	
$f_5$	平均值	5.1895e+02	6.4825e+02		5.4725e+02		5.4220e+02		6.1285e+02		6.5336e+02	+
	标准差	6.2639e+00	1.0902e+01	+	2.7579e+01	+	7.0836e+00	+	8.6368e+00	+	2.6611e+01	
$f_6$	平均值	6.0000e+02	6.0000e+02	=	6.0000e+02	=	6.0000e+02	=	6.0000e+02	=	6.4674e+02	+
	标准差	1.4020e-05	2.4989e-08		0.1440e-03		3.7546e-08		3.7586e-05		6.1243e+00	
$f_7$	平均值	7.4883e+02	8.8384e+02		8.4039e+02		7.6936e+02		8.5714e+02		1.0570e+03	+
	标准差	4.9689e+00	8.4761e+00	+	5.5376e+01	+	4.9729e+00	+	9.4331e+00	+	7.1845e+01	
$f_8$	平均值	8.1808e+02	9.5014e+02		8.6065e+02		8.3937e+02		9.1508e+02		9.3811e+02	+
	标准差	5.9169e+00	7.3435e+00	+	3.2219e+01	+	4.4789e+00	+	9.5910e+00	+	2.6481e+01	
$f_9$	平均值	9.0075e+02	9.0000e+02		9.0095e+02		9.0116e+02		9.0000e+02		3.4671e+03	+
	标准差	0.1635e+00	0.1630e-01	-	0.7001e+00	+	7.5660e+00	+	9.1143e-06	-	6.3002e+02	
$f_{10}$	平均值	4.5920e+03	7.5439e+03		7.2642e+03		3.9020e+03		5.5059e+03		4.8252e+03	+
	标准差	1.6474e+03	3.3830e+02	+	9.7423e+02	+	3.2689e+02	-	2.0529e+03	+	6.6712e+02	
$f_{11}$	平均值	1.1191e+03	1.1426e+03		1.1662e+03		1.1384e+03		1.1376e+03		1.2254e+03	+
	标准差	2.3295e+01	2.3544e+01	+	3.5499e+01	+	2.6140e+01	+	9.5799e+01	+	4.1218e+01	
$f_{12}$	平均值	8.6118e+03	1.5978e+05		2.1410e+04		2.4670e+03		2.5548e+03		9.8386e+03	+
	标准差	5.7538e+03	1.2019e+05	+	1.1504e+04	+	5.8619e+03	-	2.3902e+04	-	6.1481e+03	
$f_{13}$	平均值	5.9623e+03	9.4516e+03		1.3055e+04		2.5910e+03		1.3671e+03		4.2209e+03	-
	标准差	5.7323e+00	3.1584e+03	+	1.3786e+04	+	5.0445e+03	-	9.4931e+00	-	3.9918e+03	
$f_{14}$	平均值	1.4346e+03	1.4929e+03		1.4821e+03		9.6685e+03		1.4389e+03		1.5180e+03	+
	标准差	7.5630e+00	7.4577e+00	+	2.0191e+01	+	1.0276e+04	+	5.4446e+01	+	3.9171e+01	
$f_{15}$	平均值	1.5120e+03	1.6559e+03		2.8815e+03		3.8028e+03		1.5216e+03		1.6713e+03	+
	标准差	1.5857e+01	2.6481e+01	+	3.5997e+03	+	4.0577e+03	+	2.5067e+01	+	6.5930e+01	
$f_{16}$	平均值	1.7127e+03	2.5819e+03		2.2951e+03		2.0346e+03		2.0018e+03		2.6939e+03	+
	标准差	1.0538e+02	1.4428e+02	+	3.6995e+02	+	1.2194e+02	+	1.2239e+02	+	2.9599e+02	
$f_{17}$	平均值	1.7451e+03	1.8653e+03		1.8335e+03		1.8024e+03		1.8060e+03		2.3343e+03	+

续表

Fun	评价指标	EDF-DE	SEFDE	PL-DE	JADE	CODE	SMGBDE
$f_{18}$	标准差	2.6888e+01	1.5019e+01	7.3945e+01	2.3043e+01	<b>1.4188e+01</b>	2.4696e+02
	平均值	8.5239e+03	1.2293e+05	1.6315e+05	4.4109e+04	<b>1.8296e+03</b>	6.1949e+03
	标准差	8.2251e+03	3.9069e+04	1.4439e+05	8.3368e+04	<b>1.4669e+00</b>	6.3159e+03
$f_{19}$	平均值	<b>1.9081e+03</b>	1.9480e+03	9.5882e+03	4.2432e+03	1.9200e+03	2.1229e+03
	标准差	<b>2.2345e+00</b>	4.8616e+00	1.2596e+04	4.0026e+03	1.8582e+01	7.9855e+02
$f_{20}$	平均值	2.0849e+03	2.2351e+03	2.1981e+03	2.1199e+03	<b>2.0674e+03</b>	2.5625e+03
	标准差	7.0931e+01	4.6391e+01	9.6073e+01	4.1137e+01	<b>1.5334e+01</b>	1.9654e+02
$f_{21}$	平均值	<b>2.3195e+03</b>	2.4458e+03	2.3567e+03	2.3383e+03	2.4110e+03	2.4515e+03
	标准差	5.7178e+00	9.1087e+00	4.0298e+01	<b>5.2321e+00</b>	8.2184e+00	2.6708e+01
$f_{22}$	平均值	2.3540e+03	<b>2.3000e+03</b>	<b>2.3000e+03</b>	<b>2.3000e+03</b>	<b>2.3000e+03</b>	4.5180e+03
	标准差	2.9585e+02	2.6703e-13	0.6336e+00	<b>0.00e+00</b>	3.2768e-08	2.1965e+03
$f_{23}$	平均值	<b>2.6602e+03</b>	2.7833e+03	2.6981e+03	2.6787e+03	2.7518e+03	3.1214e+03
	标准差	8.5434e+00	8.3550e+00	1.5102e+01	<b>6.0718e+00</b>	8.8766e+00	1.0261e+02
$f_{24}$	平均值	<b>2.8340e+03</b>	2.9710e+03	2.9117e+03	2.8440e+03	2.9093e+03	3.3979e+03
	标准差	6.4502e+00	8.7030e+00	5.7091e+01	3.5523e+00	<b>2.6180e+00</b>	9.6084e+01
$f_{25}$	平均值	2.8867e+03	2.8871e+03	<b>2.8779e+03</b>	2.8868e+03	2.8867e+03	2.8965e+03
	标准差	0.6442e+00	<b>0.4165e+00</b>	2.3204e+00	0.6613e+00	3.2400e+00	1.8234e+01
$f_{26}$	平均值	<b>3.6473e+03</b>	4.6219e+03	4.1418e+03	3.8466e+03	4.4465e+03	7.2123e+03
	标准差	8.8899e+01	3.5839e+02	1.9832e+02	<b>5.9382e+01</b>	5.2953e+02	9.5657e+02
$f_{27}$	平均值	3.2019e+03	3.2014e+03	3.2369e+03	3.2063e+03	<b>3.1836e+03</b>	3.6319e+03
	标准差	8.4243e+00	5.5406e+00	1.2429e+01	<b>4.0066e+00</b>	5.6388e+01	1.5008e+02
$f_{28}$	平均值	3.1583e+03	3.2103e+03	3.1291e+03	3.1110e+03	<b>3.1034e+03</b>	3.1779e+03
	标准差	4.8466e+01	<b>8.3997e+00</b>	5.0622e+01	3.3728e+01	1.8857e+01	5.3768e+02
$f_{29}$	平均值	<b>3.3509e+03</b>	3.6569e+03	3.4996e+03	3.4127e+03	3.4913e+03	4.1181e+03
	标准差	<b>3.9109e+01</b>	5.4218e+01	1.0476e+02	4.8055e+01	3.9338e+01	2.6055e+02
$f_{30}$	平均值	6.2854e+03	1.7623e+04	8.8386e+03	6.2644e+03	<b>4.9794e+03</b>	7.3405e+03
	标准差	1.1197e+03	4.7374e+03	2.7191e+03	3.3902e+03	<b>7.7769e+00</b>	1.9953e+03
t/w/l/t-w		NAN	25/3/2/22	24/4/2/20	21/7/2/14	15/12/3/3	26/4/0/22
最优均值个数		16	4	4	6	11	2

在标准差体现的稳定性分析上,EDF-DE算法在6个混合函数( $f_{11}$ ,  $f_{12}$ ,  $f_{13}$ ,  $f_{15}$ ,  $f_{16}$ ,  $f_{19}$ )取得最优稳定性, CODE在 $f_{17}$ ,  $f_{18}$ ,  $f_{20}$ 取得最优稳定性,其它算法表现不佳. 相较于其他5类算法,EDF在处理混合函数优化问题时,不仅具有较高的收敛精度且能保持持续的高稳定性.

(4) 复合函数( $f_{21}$ ~ $f_{30}$ ): 在10个复合函数上,EDF-DE在( $f_{21}$ ,  $f_{23}$ ,  $f_{24}$ ,  $f_{26}$ ,  $f_{29}$ )上取得了5个最优均值. CODE在( $f_{22}$ ,  $f_{27}$ ,  $f_{28}$ ,  $f_{30}$ )取得了4个最优均值,位列第二. 说明EDF-DE算法采用的自适应步长因子 $F$ 和精英差异策略反馈交叉会在一定程度上保持多样性的同时促进收敛精度,从而提升在复杂函数中的优化性能.

在标准差体现的稳定性上,EDF-DE劣于其他算法,仅在 $f_{29}$ 上获得最优稳定性;而JADE在( $f_{21}$ ,  $f_{22}$ ,  $f_{23}$ ,  $f_{26}$ ,  $f_{27}$ )获得5个函数上的最优稳定性,排名第一. SEFDE在( $f_{25}$ ,  $f_{28}$ ), CODE在( $f_{24}$ ,  $f_{30}$ )上分别获得2个函数上的最优稳定性;PL-DE与SMGBDE没有获得最优稳定性. 通过实验

数据可以得出,JADE排名第一,获得5个函数上的最优稳定性,而EDF-DE仅在1个函数上的最为稳定,说明JADE的算法稳定性远高于EDF-DE算法,但JADE仅在函数 $f_{22}$ 上获得1个最优解,即JADE容易陷在目标函数上的某一位置,而不能进一步的搜索导致其“稳定”在某一非最优解上;而EDF-DE的平均值更小,标准差更大表明该算法往往能找到比均值更好的解,而出现少部分未找到最优解的情况致使平均值被拉高,这说明采用邻域精英差异反馈机制能在某一给定范围内搜索到更好解,相比于其它对比算法具有更好的局部搜索能力.

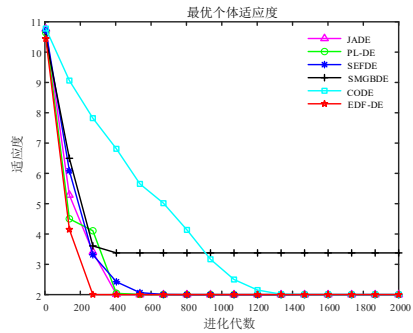
通过上述6种算法在4类函数上的实验分析结果可知,EDF-DE算法在多峰、混合函数与复合函数上具有最优性能,尤其在收敛精度方面,均高于其他对比算法;其中在混合函数上,具有最优稳定性.

### 3.2.2 收敛性分析

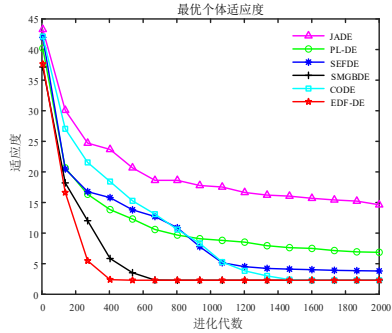
由于篇幅所限,为了验证EDF-DE算法的收敛性

能,从30个测试函数中选取了12个代表性的函数来观察EDF-DE的收敛性.其中2个单峰函数( $f_1$ 和 $f_2$ )、3个

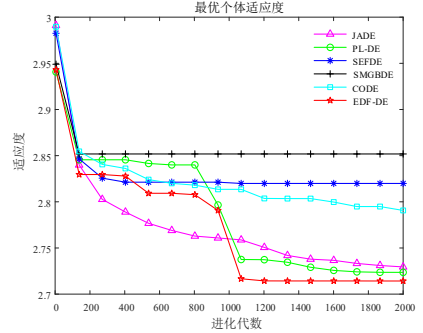
多峰函数( $f_5, f_6, f_7$ )、3个混合函数( $f_{15}, f_{16}, f_{17}$ )和4个复合函数( $f_{21}, f_{22}, f_{23}, f_{24}$ ).收敛性对比如图4所示,为了收敛



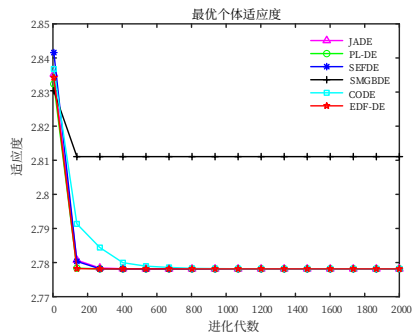
(a) 函数 $f_1$ 的收敛图



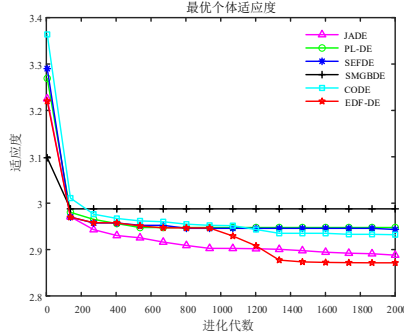
(b) 函数 $f_2$ 的收敛图



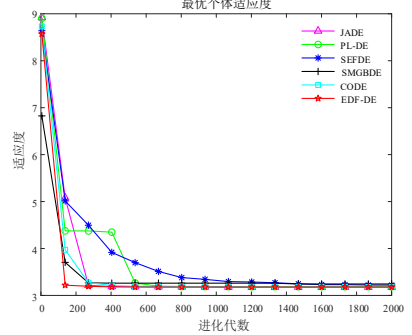
(c) 函数 $f_5$ 的收敛图



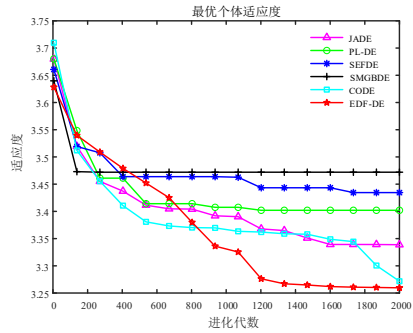
(d) 函数 $f_6$ 的收敛图



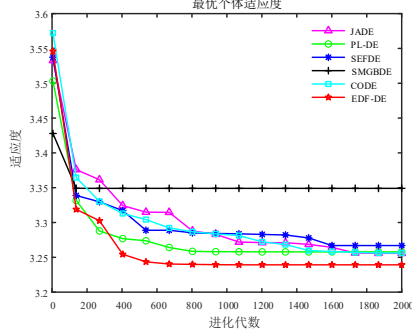
(e) 函数 $f_7$ 的收敛图



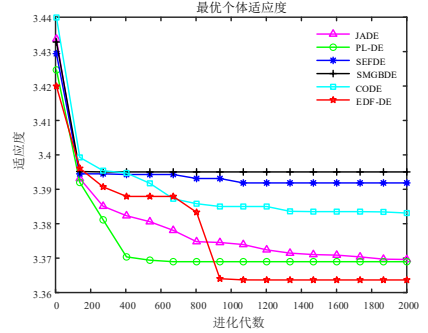
(f) 函数 $f_{15}$ 的收敛图



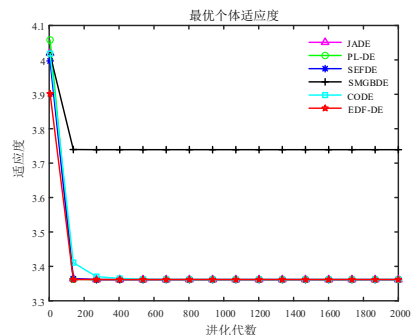
(g) 函数 $f_{16}$ 的收敛图



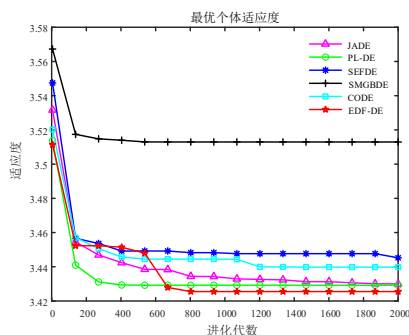
(h) 函数 $f_{17}$ 的收敛图



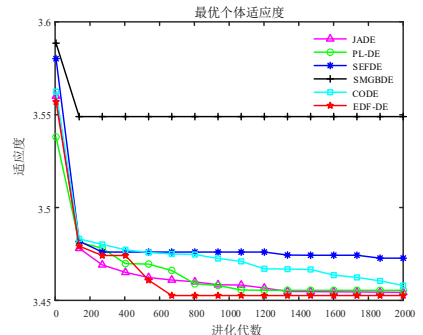
(i) 函数 $f_{21}$ 的收敛图



(j) 函数 $f_{22}$ 的收敛图



(k) 函数 $f_{23}$ 的收敛图



(l) 函数 $f_{24}$ 的收敛图

图4 各算法在部分函数上的收敛曲线图

图更加清晰显著,图中纵轴采用对数坐标,横坐标代表迭代次数.

从图4可以看出,无论在单峰、多峰、混合和复合函数上,EDF-DE具有较强的收敛能力.

具体来看,在单峰函数 $f_1$ 和 $f_2$ 上,EDF-DE的收敛速度要优于其他算法,尤其在 $f_2$ 上表现更为显著.各算法在进化初期,EDF-DE的收敛速度明显优于对比算法,这是由于算法前期步长因子较大,搜索范围广,并且交叉操作会有一定的概率被全局精英差异指导,提升了收敛速度与寻优效率,最终收敛于全局最优解.

在多峰函数 $f_5$ 、 $f_6$ 、 $f_7$ 的收敛曲线图中可知,EDF-DE在整个过程中皆处于深度搜索与广度搜索不断交替执行的寻优过程,尤其在函数 $f_5$ 、 $f_7$ 中的表现尤为明显,而在函数 $f_6$ 上表现较为平稳.

在混合函数 $f_{15}$ 、 $f_{16}$ 、 $f_{17}$ 中,各算法均能得到最优值,但EDF-DE在函数 $f_{15}$ 上收敛速度最快且能持续保持稳定;在 $f_{16}$ 上,虽然算法SMGBDE、CODE、SEFDE、PL-DE和JADE均先于EDF-DE算法收敛,但这些算法均没有找到最优解,尤其SMGBDE在200代时即收敛,陷入局部极值.而EDF-DE算法前期收敛速度虽稍慢于对比算法,但能够保持持续开采,直到算法收敛到全局最优解且搜索到最高精度值;在函数 $f_{17}$ 上,算法收敛过程与 $f_{16}$ 上类似,在此不再赘述.

在复合函数 $f_{21}$ 、 $f_{22}$ 、 $f_{23}$ 、 $f_{24}$ 中可以看出,EDF-DE算法在复合函数的收敛性能最优.在函数 $f_{22}$ 上,EDF-DE与大多数算法都能找到最优解,但收敛速度最快.在 $f_{21}$ 、 $f_{23}$ 和 $f_{24}$ 中,EDF-DE在算法前期收敛较快,在算法中期收敛速度趋于平缓,这是由于算法通过邻域精英差异,进而实现个体多样性扰动,指导种群进行局部开发实现广度搜索,提升多样性,当搜索到多个局部最优时进入算法后期,执行全局精英差异策略,实现深度搜索并快速收敛,获得最优解.通过上述分析可知,EDF-DE算

法在单峰、多峰、混合以及复合函数中,均表现出优异的求解精度与收敛性能,尤其在混合函数与复合函数上,这种优势体现的愈加明显.

### 3.3 显著性检验

(1) $t$ -检验:这里将EDF-DE算法与其他改进的DE算法在CEC2017的30个测试函数上的结果进行 $t$ -检验,检验结果如表3所示.表中符号说明:“+”、“-”、“=”分别代表EDF-DE相较对比算法性能“较好、较差、持平”. $t/w/l$ 则分别对应“+”、“-”、“=”的函数个数, $t-w$ 的差值表示EDF-DE算法优于对比算法个数减去劣于对比算法个数, $t-w$ 的差值越大,说明对应算法性能最差.

根据表3可以得出,EDF-DE相较于SEFDE、PL-DE、JADE、CODE和SMGBDE算法,分别有25、24、21、15和26个函数更具性能优势.在显著性比较时,显著性水平为0.05,即当 $t = n/2 + \sqrt{n}/2 \times 1.96$ ,可知 $n=30$ 可知 $t > 20.36$ 时,本算法显著优于其他算法,因此本算法显著优于SEFDE和SMGBDE算法;对于PLDE和JADE本算法也有很大的优势;而对于CODE算法相比,也在15个标准测试函数上取得了显著性优势.从表3来看,CODE取得了次优的排名,尽管EDF-DE有较优的寻优能力,但是在单峰函数上,对比结果的显著性并不明显,因此我们认为该算法在全局与邻域精英差异设置方面仍有待改进.

(2)Friedman-排名:本文对6个算法在CEC2017的30个测试函数上进行Friedman-排名,如表4所示.Friedman排名是每个算法在30个测试函数上排名的平均值,排名越小代表算法综合能力越好,Statistic代表卡方统计量,如果卡方值大于5.99,则认为各算法存在明显差异.从表4可以看出,在函数 $f_1 \sim f_{30}$ 上,EDF-DE的Rank值最小,且卡方统计量大于5.99,证明各算法存在显著差异,综合来看,EDF-DE在30个测试函数上具有

表2 测试函数上均值结果的Friedman-检验分析

Average Rank	所有函数( $f_1 \sim f_{30}$ )		单峰函数( $f_1 \sim f_3$ )		多峰函数( $f_4 \sim f_{10}$ )		混合函数( $f_{11} \sim f_{20}$ )		复合函数( $f_{21} \sim f_{30}$ )	
	Algorithm	Rank	Algorithm	Rank	Algorithm	Rank	Algorithm	Rank	Algorithm	Rank
1	EDF-DE	2.10	EDF-DE	2.67	EDF-DE	2.14	EDF-DE	1.80	EDF-DE	2.20
2	CODE	2.53	CODE	2.67	JADE	2.14	CODE	1.80	JADE	2.25
3	JADE	2.85	SMGBDE	3.00	PL-DE	3.43	JADE	3.50	CODE	2.50
4	PL-DE	4.07	PL-DE	3.67	CODE	3.57	SEFDE	4.50	PL-DE	3.90
5	SEFDE	4.48	JADE	4.33	SEFDE	4.43	SMGBDE	4.60	SEFDE	4.45
6	SMGBDE	4.97	SEFDE	4.67	SMGBDE	5.29	PL-DE	4.80	SMGBDE	5.70
Statistic	58.4817		3.317		15.490		27.657		29.097	

较优性能. 并且在单峰、多峰和混合函数上分别与 CODE、JADE 和 CODE 取得并列第一排名, 在复合函数上取得第一排名. 其中单峰函数卡方统计量小于 5.99, 说明各算法在单峰函数上不存在显著性差异; 在多峰、混合和复合函数上卡方统计量分别为 15.490、27.657 和 29.097, 说明在这 3 种函数上, 存在显著性差异, 进一步证明 EDF-DE 相较于其他算法优势显著.

#### 4 结论

针对 DE 算法存在的不足, 本文提出了一种精英差异反馈策略调整交叉环节的方法. 本算法通过全局精英差异与邻域精英差异共同作用, 进行深度与广度搜索, 其中邻域精英差异引入了个体多样性扰动邻域构建策略, 通过邻域精英差异维持种群多样性, 促进算法多局部进行超精细搜索. 最后通过超参数控制, 在交叉阶段随机选择传统二项式交叉或精英差异策略控制交叉, 提高了算法搜索的能力, 实验结果表明本算法的有效性.

值得注意的是, 本算法在单峰函数以及稳定性方面依然存在不足, 不能根据待优化问题的分类自动进行调整全局精英差异与局部精英差异的作用比例, 导致整个迭代过程稍长, 这是一大难点, 也是本文作者在后续的工作中需要进行深入研究的方向.

#### 参考文献

- [1] STORN R, PRICE K. Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [2] MENG Z Y, PAN J S, KONG L P. Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution[J]. *Knowledge-Based Systems*, 2018, 141: 92-112.
- [3] LIU X F, ZHAN Z H, LIN Y, et al. Historical and heuristic-based adaptive differential evolution[J]. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019, 49(12): 2623-2635.
- [4] CHANG Y C. Parameter selection of differential evolution by another differential evolution algorithm[C]. 2019 IEEE International Conference on Systems, Man and Cybernetics. Piscataway: IEEE, 2019: 2506-2511.
- [5] HIBA H, MAHDAVI S, RAHNAMAYAN S. Differential evolution with self-adaptive mutation scaling factor[C]. 2017 IEEE Symposium Series on Computational Intelligence. Piscataway: IEEE, 2017: 1-8.
- [6] 大海, 李元香, 龚文引, 等. 一种求解约束优化问题的自适应差分进化算法[J]. *电子学报*, 2016, 44(10): 2535-2542.  
XIA Dahai, LI Yuanxiang, GONG Wenyin, et al. An adaptive differential evolution algorithm for constrained optimization problems[J]. *Acta Electronica Sinica*, 2016, 44(10): 2535-2542. (in Chinese)
- [7] 刘昊, 丁进良, 杨翠娥, 等. 基于择优学习策略的差分进化算法[J]. *上海交通大学学报*, 2017, 51(6): 704-708.  
LIU H, DING J L, YANG C E, et al. Preferred-learning-based differential evolution algorithm[J]. *Journal of Shanghai Jiao Tong University*, 2017, 51(6): 704-708. (in Chinese)
- [8] 刘会宇, 韩继红, 袁霖, 等. 基于双变异策略的自适应骨架差分进化算法[J]. *通信学报*, 2017, 38(8): 201-212.  
LIU H Y, HAN J H, YUAN L, et al. Self-adaptive barebones differential evolution based on bi-mutation strategy[J]. *Journal on Communications*, 2017, 38(8): 201-212. (in Chinese)
- [9] CAI Y Q, WU D W, ZHOU Y, et al. Self-organizing neighborhood-based differential evolution for global optimization[J]. *Swarm and Evolutionary Computation*, 2020, 56: 100699.
- [10] 王亚辉, 吴金妹, 贾晨辉. 基于动态种群多策略差分进化模型的多目标进化算法[J]. *电子学报*, 2016, 44(6): 1472-1480.  
WANG Y H, WU J M, JIA C H. Multi-objective evolutionary algorithm based on dynamic population multi-strategy differential models[J]. *Acta Electronica Sinica*, 2016, 44(6): 1472-1480. (in Chinese)
- [11] ZHAN Z H, WANG Z J, JIN H, et al. Adaptive distributed differential evolution[J]. *IEEE Transactions on Cybernetics*, 2020, 50(11): 4633-4647.
- [12] GE Y F, YU W J, ZHAN Z H, et al. Competition-based distributed differential evolution[C]. 2018 IEEE Congress on Evolutionary Computation (CEC). Piscataway: IEEE, 2018: 1-8.
- [13] ZHANG J Q, SANDERSON A C. JADE: adaptive differential evolution with optional external archive[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945-958.
- [14] 王柳静, 张贵军, 周晓根. 基于状态估计反馈的策略自适应差分进化算法[J]. *自动化学报*, 2020, 46(4): 752-766.  
WANG L J, ZHANG G J, ZHOU X G. Strategy self-

adaptive differential evolution algorithm based on state estimation feedback[J]. Acta Automatica Sinica, 2020, 46(4): 752-766. (in Chinese)

- [15] WANG Y, CAI Z X, ZHANG Q F. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55-66.

#### 作者简介



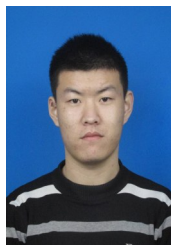
王毅 男, 1979年2月生, 上海人, 博士(后). 现为西北大学信息科学与技术学院副教授, 主要从事智能信息处理、深度学习与群体智能优化.



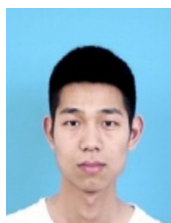
彭钰博 男, 1996年4月生, 陕西人. 西北大学信息科学与技术学院硕士研究生, 主要研究方向为群体智能优化算法、机器学习.



黄欣(通信作者) 女, 1979年6月生, 陕西人, 博士. 现为西安交通大学第一附属医院副教授, 主要从事血管腔内影像处理与优化研究.  
E-mail: hearthx@126.com



郑宏志 男, 1997年8月生, 山东人, 西北大学信息科学与技术学院硕士研究生, 主要从事深度学习, 群体智能与多目标优化.



张鹏辉 男, 1995年4月出生, 河南人, 现为西北大学信息科学与技术学院研究生, 主要研究深度学习和群体智能优化.