

# 基于高斯分布估计的对位差分进化算法

方景远,季益胜,赵新超

(北京邮电大学 理学院,北京 100876)

**摘要:**针对差分进化算法探索能力不足、收敛慢等问题,提出一种基于高斯分布估计的对位差分进化算法.该算法在生成对位种群的同时还生成一个基于高斯分布估计的新种群,意在更充分地搜索解空间.在不满足跳转条件的情况下,算法给出一种基于高斯分布估计的种群跳转,增加了种群多样性.在选择操作时,将所有父代和子代个体混合起来择优选择,减少了部分优秀解和优秀基因的流失.最后在 CEC2014 标准函数中进行测试,与其他算法进行比较,验证了所提出的算法具有更好的搜索能力和收敛性.

**关键词:**差分进化算法;对位学习;高斯分布估计

**中图分类号:**TP278;TP391

**文献标志码:**A

差分进化(Differential Evolution, DE)是由文献[1]提出的一种基于群体搜索的全局优化算法,采用基于差分信息的变异操作和一对一的择优生存策略,实现了算法操作的简捷有效.差分算法特有的记忆能力使其可以动态跟踪当前群体的搜索状况,以调整其搜索策略,具有较强的全局收敛能力和鲁棒性,且不需要借助问题的特征信息.

差分算法自提出以来,引起广大学者的研究兴趣.基于对位学习(OBL)进行种群初始化和世代跳跃,文献[2]提出了一种对位差分进化算法.文献[3]提出了基于均匀邻域对位的自适应差分进化算法(NODE).文献[4]提出了一种新的差分进化算法 JADE,具有可选的外部存档和以自适应的方式更新控制参数.文献[5]将几种有效的测试向量生成策略与一些合适的控制参数设置相结合,提出了一种复合测试向量生成策略与控制参数的新算法(CoDE).文献[6]提出一种具有人工蜂群搜索策略的差分进化算法,利用人工蜂群搜索策略的探索能力,对种群进行引导以帮助算法快速跳出局部最优点.文献[7]提出一种新的基于变异的差分自适应全局优化的和谐搜索算法,文献[8]提出基于自上而下的策略失败记忆驱动的自适应差异进化.文献[9]针对差分进化算法易早熟、对参数设置敏感等问题,提出一种基于牛顿三次插值的自适应差分进化算法.文献[10]提出一种基于双群体搜索机制的差分进化算法用于求解约束多目标优化问题.为了将差分进化算法用于多目标优化问题,文献[11]提出了多目标混沌差分进化算法(CDEMO).为了更加有效地解决多目标优化,文献[12]提出了一种基于差分变异和扰动变异相结合的多目标克隆选择算法(MCSA-MDD).文献[13]提出了一种新的混合离散差分进化(HDDE)算法,用于求解阻塞流量车间调度问题,以最小化最大完成时间.文献[14]研究了一种基于自适应差分进化的分布式模型预测控制方法,用于处理复杂环境下多无人机飞行中实现避障和编队保持的多无人机飞行问题.

改进差分进化算法性能都有不同程度的提升,然而对位差分进化算法<sup>[2]</sup>依然存在探索能力不足、收敛速度慢、收敛精度差等问题.基于此,本文提出了一种基于高斯分布估计的对位差分进化算法,通过加入高斯分布估计策略和父子两代混合精英选择策略,扩大了探索能力,保护了优秀解,增大了种群多样性,从而达到加快收敛速度、提高收敛精度等目的.

**收稿日期:**2020-09-12;**修回日期:**2021-03-28.

**基金项目:**国家自然科学基金(61973042);北京市自然科学基金(1202020);北京邮电大学提升科技创新能力行动计划项目(2020XD-A01-1).

**作者简介(通信作者):**赵新超(1976—),男,河南商丘人,北京邮电大学教授,博士,博士生导师,主要研究方向为群体智能与运筹优化,E-mail: xcbupt@126.com.

## 1 差分进化算法

差分进化算法(DE)是一种基于种群差异化信息的进化搜索的方法,通过群体内个体之间的相互合作与竞争产生的群体信息指导优化搜索方向.算法基本思想是:随机产生一个初始种群;任取种群中两个不同的个体向量,进行差分运算并进行缩放,再将之加到第3个随机选取的个体基向量上,得到一个变异个体向量;以一定的概率将父代个体与变异个体进行交叉运算生成子代个体;最后父代与子代之间通过贪婪选择策略保留较优个体.差分进化算法的变异操作有很多种,如DE/rand/1/bin,DE/rand/2/bin,DE/best/1/bin,DE/best/2/bin,DE/current-to-rand/1/bin等.本文以DE/rand/1/bin为例介绍,具体步骤如下.

### 1.1 初始化

生成一个种群规模为 $N$ 、维度为 $D$ 的随机种群,每个个体 $x_i$ 按照(1)式生成,

$$x_{ij} = a_j + \text{rand}(0,1) \times (b_j - a_j), \quad (1)$$

其中 $x_{ij}$ 是第 $i$ 个体向量的第 $j$ 个分量, $i = \{1, 2, \dots, N\}$ , $j = \{1, 2, \dots, D\}$ . $\text{rand}(0,1)$ 是 $[0,1]$ 之间的均匀随机数, $b_j$ 和 $a_j$ 分别是解向量在第 $j$ 维的上下界.

### 1.2 变异操作

对于每个个体 $x_i$ ,随机选择3个不同个体 $x_{r1}, x_{r2}, x_{r3}$ , $F \in [0,2]$ 是放缩因子,

$$v_i = x_{r1} + F(x_{r2} - x_{r3}). \quad (2)$$

### 1.3 杂交操作

父代 $x_i$ 和变异个体 $v_i$ 之间按照(3)式进行杂交,

$$U_i = \begin{cases} v_i, & \text{当 } \text{rand}(0,1) < C_r \text{ 或 } j = j_{\text{rand}} \text{ 时,} \\ x_i, & \text{其他情况,} \end{cases} \quad (3)$$

$j_{\text{rand}} \in \{1, 2, \dots, D\}$ 是一个预先给定的随机整数, $C_r \in [0,1]$ 是杂交概率.

### 1.4 选择操作

按照(4)式进行一对一择优选择操作,得到新个体,

$$x_i = \begin{cases} U_i, & \text{当 } f(U_i) \leq f(x_i) \text{ 时,} \\ x_i, & \text{其他情况.} \end{cases} \quad (4)$$

## 2 对位差分进化算法

文献[2]在基本差分进化算法的基础上引入了对位学习思想,得到对位差分进化算法,基本流程如下.

### 2.1 初始化

按照(1)式进行初始化,得到种群 $P$ .

### 2.2 生成对位种群

将种群 $P$ 中的每一个体向量按照公式进行对位操作,最终得到对位种群 $OP$ ,

$$ox_{ij} = a_j + b_j - x_{ij}, \quad (5)$$

从 $\{P, OP\}$ 中选出 $N$ 个适应值最高的个体组成新种群.

### 2.3 变异,交叉和选择

按照(2)~(4)式依次进行变异、交叉和选择操作.

### 2.4 种群跳转

如果产生的 $[0,1]$ 间随机数小于跳转概率 $J_r$ ,则进行种群跳转得到 $OP^*$ ,种群中每个个体 $x_i$ 按照(6)式跳转生成新的对位种群,

$$ox_{ij}^* = \max_j + \min_j - x_{ij}, \quad (6)$$

其中 $\max_j$ 和 $\min_j$ 分别是第 $j$ 维分量的最大值和最小值,然后从 $\{P, OP^*\}$ 中选出 $N$ 个适应值最高的个体组成新种群.

### 3 基于高斯分布估计的对位差分进化算法

#### 3.1 生成高斯分布估计对位种群

针对经典对位操作可能对解空间探索不够充分的问题,本文增加了对种群进行高斯分布估计从而产生新的种群,新种群重点对个体分布的中心区域进行探索,这样可能对所有可能的潜力区域强化搜索,从而加快算法收敛,同时结合对位种群又可以有效避免陷入局部最优区域.对于种群  $P$  的每个个体按照(5)式进行对位操作得到对位种群  $OP$ ,再将每个个体按照(7)式生成一个新个体,最终得到高斯分布估计种群  $EP$ .

$$ex_{ij} = \text{normrnd}(\text{mean}_j, \text{std}_j), \quad (7)$$

$\text{mean}_j$  是当前种群第  $j$  维分量的均值,  $\text{std}_j$  是当前种群第  $j$  维分量的标准差.从  $\{P, OP, EP\}$  中选出适应值最高的  $N$  个个体组成新种群.

#### 3.2 种群选择

经典差分算法的选择操作是在父代个体和与其生成的新子代个体之间一对一的择优选择,这样可能导致很多优秀解的有益信息损失,因此本文选择将所有的父代个体和所有新生成的子代个体混合在一起,选择其中适应值最高的  $N$  个个体组成新种群,这样能够确保优秀的个体与优秀解信息被保留.

#### 3.3 高斯分布估计跳转

为了加强对解空间的探索,本文在算法不满足原本跳转条件的情况下添加了一种基于高斯分布估计的操作进行探索.具体操作如下:先产生一个  $[0, 1]$  间的随机数,如果小于跳转概率  $J_r$ ,则种群中每个个体按照(6)式进行跳转操作,最终得到种群  $OP^*$ ,从  $\{P, OP^*\}$  中选出适应值最高的  $N$  个个体组成新种群.如果产生的随机数不小于  $J_r$ ,则种群中每个个体按照(7)式生成高斯分布估计种群  $EP^*$ ,从  $\{P, EP^*\}$  中选出适应值最高的  $N$  个解组成新群体.

#### 3.4 算法流程

基于高斯分布估计的对位差分进化算法的伪代码如下,其中  $\text{maxFES}$  是最大函数值计算次数,  $FES$  是当前函数值计算次数.

|    |  |    |   |
|----|--|----|---|
| 1  | 初始化,根据(1)式生成随机初始种群                       | 13 | for $i=1$ to $N$  |
| 2  | while $FES < \text{maxFES}$              | 14 | for $j=1$ to $D$  |
| 2  | for $i=1$ to $N$                         | 15 | 按照(6)式计算 $ox_{ij}^*$                                    |
| 3  | for $j=1$ to $D$                         | 16 | end   |
| 4  | 根据(5)式计算 $ox_{ij}$                       | 17 | end   |
| 5  | 根据(7)式计算 $ex_{ij}$                       | 18 | else  |
| 6  | end                                      | 19 | for $i=1$ to $N$  |
| 7  | end                                      | 20 | for $j=1$ to $D$  |
| 8  | 在 $\{P, OP, EP\}$ 中选出适应值最高的 $N$ 个个体组成新种群 | 21 | 按照(7)式计算 $ex_{ij}^*$                                    |
| 9  | 根据(2)式进行变异操作                             | 22 | end   |
| 10 | 根据(3)式进行杂交操作                             | 23 | end   |
| 11 | 将父代和子代混合在一起选出适应值最高的 $N$ 个个体组成新种群         | 24 | end   |
| 12 | if( $\text{rand}(0,1) < J_r$ )           | 25 | 在 $\{P, OP^*\}$ 或者 $\{P, EP^*\}$ 中选出适应值最高的 $N$ 个个体组成新种群 |
|    |  | 26 | end   |

### 4 仿真实验与结果分析

为了验证本文所提出的基于高斯分布估计的对位差分进化算法(EODE)的性能,将其与经典 DE(DE/rand/1/bin)算法<sup>[1]</sup>和 ODE 算法<sup>[2]</sup>在 CEC2014 测试函数集上进行了仿真实验对比.

#### 4.1 参数设置和函数选择

本节设计了两组实验,实验一的种群规模  $N=50$ ,搜索空间维度  $D=50$ ,实验二种群规模  $N=100$ ,搜索

空间维度  $D=100$ .参与仿真实验算法的其余参数设置如下:搜索空间 $[-100,100]^D$ ,缩放因子  $F=0.5$ ,交叉概率  $C_r=0.9$ ,跳转概率  $J_r=0.3$ ,最大评估次数  $\max FES=10\ 000 * D$ ,算法独立运行次数  $RUN=30$ .

本文在 CEC2014 测试函数集<sup>[15]</sup>上选择 2 个单峰函数  $F1$  和  $F3$ ,4 个多峰函数  $F5, F12, F14$  和  $F16$ ,2 个混合函数  $F18$  和  $F20$ ,2 个组合函数  $F25$  和  $F28$ ,依次重新标记为  $f1 \sim f10$ .所有单峰函数都有偏移量,多峰函数、混合函数和组合函数都是经过偏移和旋转的.

#### 4.2 数值实验结果

经过两组仿真实验后,DE,ODE 和 EODE 3 个算法在每一个测试函数上 30 次独立运行中获得最终结果的最优值、平均值和标准差的结果统计见表 1、表 2.其中实验一结果统计见表 1,实验二结果见表 2,各组的最优结果均用黑体表示.

表 1 DE,ODE,EODE 3 种算法实验结果对比 ( $N=50, D=50$ )

Tab. 1 Experimental results comparison among DE, ODE and EODE algorithms ( $N=50, D=50$ )

| 函数   | 指标  | DE                 | ODE                | EODE               | 函数    | 指标  | DE          | ODE                | EODE               |
|------|-----|--------------------|--------------------|--------------------|-------|-----|-------------|--------------------|--------------------|
| $f1$ | 最优值 | <b>1.561 5E+05</b> | 2.481 6E+05        | 2.435 9E+05        | $f7$  | 最优值 | 1.819 5E+03 | 1.816 8E+03        | <b>1.812 4E+03</b> |
|      | 平均值 | 6.260 5E+05        | <b>5.962 1E+05</b> | 6.618 7E+05        |       | 平均值 | 2.885 3E+03 | 2.741 2E+03        | <b>2.724 7E+03</b> |
|      | 标准差 | 3.512 1E+05        | 2.840 5E+05        | <b>2.596 4E+05</b> |       | 标准差 | 1.266 9E+03 | 1.286 4E+03        | <b>9.784 6E+02</b> |
| $f2$ | 最优值 | 4.049 1E+02        | 4.248 8E+02        | <b>3.043 7E+02</b> | $f8$  | 最优值 | 2.157 8E+03 | 2.145 7E+03        | <b>2.014 1E+03</b> |
|      | 平均值 | 2.216 2E+03        | 1.931 8E+03        | <b>6.403 3E+02</b> |       | 平均值 | 2.844 1E+03 | 2.793 9E+03        | <b>2.035 8E+03</b> |
|      | 标准差 | 2.251 3E+03        | 1.076 2E+03        | <b>5.331 5E+02</b> |       | 标准差 | 7.152 7E+02 | 6.304 8E+02        | <b>1.862 7E+01</b> |
| $f3$ | 最优值 | 5.210 6E+02        | 5.205 3E+02        | <b>5.200 5E+02</b> | $f9$  | 最优值 | 2.705 2E+03 | <b>2.700 0E+03</b> | <b>2.700 0E+03</b> |
|      | 平均值 | 5.211 2E+02        | 5.208 6E+02        | <b>5.202 7E+02</b> |       | 平均值 | 2.706 9E+03 | 2.706 2E+03        | <b>2.700 0E+03</b> |
|      | 标准差 | 3.648 0E-02        | 1.982 1E-01        | <b>1.366 4E-01</b> |       | 标准差 | 1.370 5E+00 | 3.100 8E+00        | <b>0.000 0E+00</b> |
| $f4$ | 最优值 | 1.202 9E+03        | <b>1.200 1E+03</b> | <b>1.200 1E+03</b> | $f10$ | 最优值 | 3.897 2E+03 | <b>3.834 6E+03</b> | 3.852 1E+03        |
|      | 平均值 | 1.203 4E+03        | 1.201 1E+03        | <b>1.200 2E+03</b> |       | 平均值 | 4.080 8E+03 | 3.952 3E+03        | <b>3.922 5E+03</b> |
|      | 标准差 | 2.163 1E-01        | 7.256 5E-01        | <b>1.108 9E-01</b> |       | 标准差 | 1.557 6E+02 | 8.301 8E+01        | <b>3.681 7E+01</b> |
| $f5$ | 最优值 | 1.400 3E+03        | <b>1.400 2E+03</b> | <b>1.400 2E+03</b> | 排名最   | 最优值 | 1           | 4                  | 8                  |
|      | 平均值 | 1.400 4E+03        | 1.400 4E+03        | <b>1.400 3E+03</b> | 高的    | 平均值 | 0           | 1                  | 9                  |
|      | 标准差 | 1.669 0E-01        | 1.250 1E-01        | <b>3.171 5E-02</b> | 次数    | 标准差 | 1           | 0                  | 9                  |
| $f6$ | 最优值 | 1.621 6E+03        | 1.618 5E+03        | <b>1.614 7E+03</b> | 小计    |     | 2           | 5                  | 26                 |
|      | 平均值 | 1.622 1E+03        | 1.620 8E+03        | <b>1.620 0E+03</b> |       |     |             |                    |                    |
|      | 标准差 | <b>2.610 0E-01</b> | 1.131 2E+00        | 1.985 4E+00        |       |     |             |                    |                    |

从表 1 和表 2 统计结果看出,本文提出的算法 EODE 在不同维度下得到的最优值、平均值和标准差都要远远优于 ODE 和 DE,因此 EODE 算法的综合性能表现最好.

#### 4.3 3 种算法进化趋势对比

考虑篇幅原因,本文在 4 类测试函数中各选择一个作为仿真实验对象,选择的 4 个函数为  $f2, f3, f8, f9$ .3 种算法的平均进化趋势如图 1 所示,横坐标为函数值评估次数,纵坐标为 30 次独立运行中在同一次迭代中的最优适应值的平均适应值.

从进化趋势对比曲线图可以看出本文提出的 EODE 算法的收敛速度和收敛精度都优于 ODE 和 DE.

## 5 结 论

本文在对位差分进化算法的基础上,在生成对位种群的同时生成了一个基于高斯分布估计的种群,增加了对于解空间的多样化探索.在选择操作时选用了所有父代和子代个体组成的并集中进行择优选择的策略,保留了新生成的优秀解和可能的优秀基因信息得以保留.算法在不满足群体跳转条件时也给出了一种高斯分布估计的跳转策略,丰富了跳转形式,增加了种群新解构成的多样性.最后选择 CEC2014 不同类型测试函数中具有代表性的 10 个函数进行了仿真对比实验,验证了本文所提出的基于高斯分布估计的对位差分进化算法 EODE 优于对位差分进化算法 ODE 和差分进化算法 DE.

表 2 DE,ODE,EODE 3 种算法实验结果对比 ( $N=100,D=100$ )

Tab. 2 Experimental results comparison among DE, ODE and EODE algorithms ( $N=100,D=100$ )

| 函数    | 指标  | DE                 | ODE                | EODE               | 函数       | 指标  | DE          | ODE                | EODE               |
|-------|-----|--------------------|--------------------|--------------------|----------|-----|-------------|--------------------|--------------------|
| $f_1$ | 最优值 | 3.401 9E+06        | <b>3.509 6E+06</b> | 3.622 4E+06        | $f_7$    | 最优值 | 1.844 7E+03 | 1.946 2E+03        | <b>1.824 1E+03</b> |
|       | 平均值 | 6.574 0E+06        | <b>7.622 7E+06</b> | 7.011 0E+06        |          | 平均值 | 4.084 6E+03 | 3.970 7E+03        | <b>2.197 7E+03</b> |
|       | 标准差 | 2.439 0E+06        | 2.467 9E+06        | <b>2.101 6E+06</b> |          | 标准差 | 2.509 9E+03 | 3.114 4E+03        | <b>4.688 8E+02</b> |
| $f_2$ | 最优值 | 4.625 7E+02        | 1.409 6E+03        | <b>3.000 9E+02</b> | $f_8$    | 最优值 | 2.550 7E+03 | 2.777 9E+03        | <b>2.267 4E+03</b> |
|       | 平均值 | 1.979 4E+03        | 4.658 7E+03        | <b>3.023 0E+02</b> |          | 平均值 | 2.987 6E+03 | 3.901 5E+03        | <b>2.325 5E+03</b> |
|       | 标准差 | 1.086 0E+03        | 2.281 9E+03        | <b>3.282 8E+00</b> |          | 标准差 | 3.139 7E+02 | 8.392 5E+02        | <b>2.531 9E+01</b> |
| $f_3$ | 最优值 | 5.212 6E+02        | 5.207 7E+02        | <b>5.200 5E+02</b> | $f_9$    | 最优值 | 2.718 7E+03 | <b>2.700 0E+03</b> | <b>2.700 0E+03</b> |
|       | 平均值 | 5.213 2E+02        | 5.210 1E+02        | <b>5.204 1E+02</b> |          | 平均值 | 2.723 7E+03 | 2.700 0E+03        | <b>2.700 0E+03</b> |
|       | 标准差 | 2.382 2E-02        | <b>1.129 9E-01</b> | 3.945 2E-01        |          | 标准差 | 2.383 1E+00 | 3.481 7E-13        | <b>0.000 0E+00</b> |
| $f_4$ | 最优值 | 1.203 5E+03        | 1.200 3E+03        | <b>1.200 0E+03</b> | $f_{10}$ | 最优值 | 4.894 3E+03 | <b>4.725 2E+03</b> | 4.769 0E+03        |
|       | 平均值 | 1.204 0E+03        | 1.201 2E+03        | <b>1.200 8E+03</b> |          | 平均值 | 5.167 3E+03 | 5.099 2E+03        | <b>4.894 7E+03</b> |
|       | 标准差 | 2.161 5E-01        | 8.496 7E-01        | <b>7.999 4E-01</b> |          | 标准差 | 3.602 9E+02 | 2.880 4E+02        | <b>7.136 0E+01</b> |
| $f_5$ | 最优值 | 2.189 2E+03        | 2.204 4E+03        | <b>1.984 4E+03</b> | 排名最      | 最优值 | 1           | 3                  | 8                  |
|       | 平均值 | <b>2.453 5E+03</b> | 2.558 3E+03        | 2.489 0E+03        | 高的       | 平均值 | 2           | 1                  | 8                  |
|       | 标准差 | <b>1.720 0E+02</b> | 1.824 3E+02        | 1.718 1E+02        | 次数       | 标准差 | 1           | 1                  | 8                  |
| $f_6$ | 最优值 | 1.202 9E+03        | <b>1.200 1E+03</b> | <b>1.200 1E+03</b> | 小计       |     | 4           | 5                  | 24                 |
|       | 平均值 | 1.203 4E+03        | 1.201 1E+03        | <b>1.200 2E+03</b> |          |     |             |                    |                    |
|       | 标准差 | 2.163 1E-01        | 7.256 5E-01        | <b>1.108 9E-01</b> |          |     |             |                    |                    |

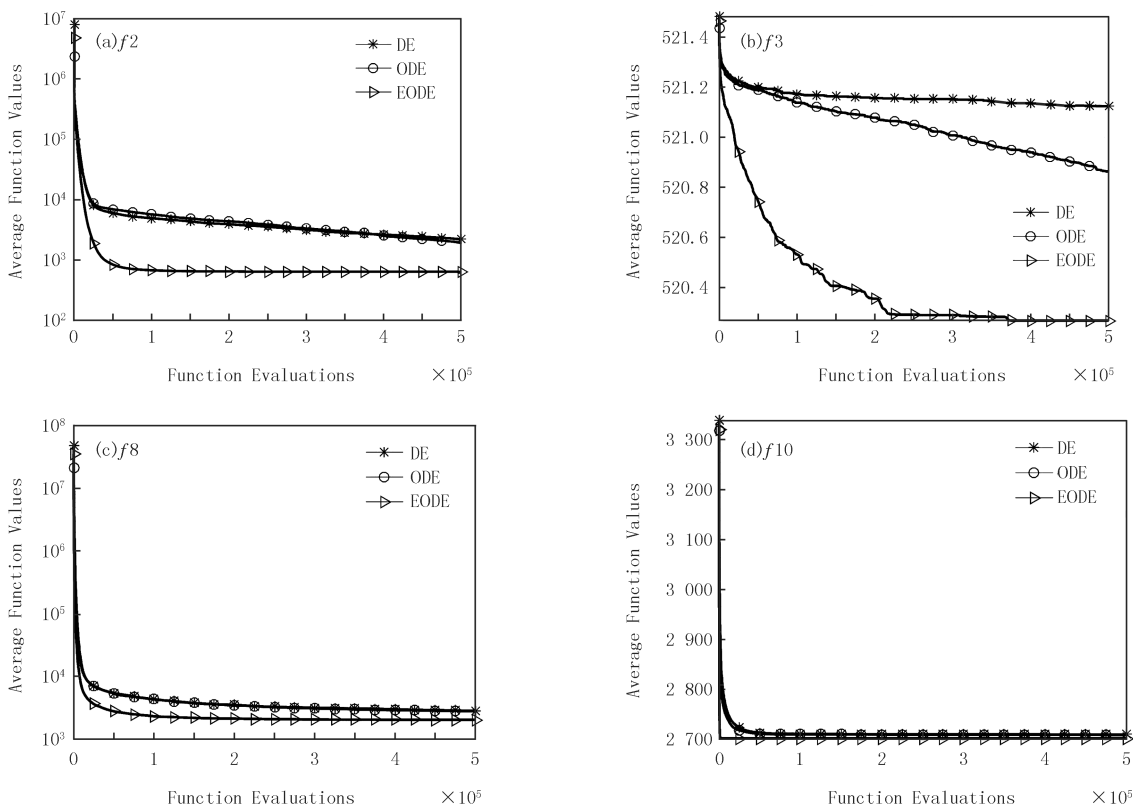


图1 3种算法进化趋势曲线图

Fig.1 Evolution curves comparison among three algorithms

参 考 文 献

[1] STORN R,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.

- [2] RAHNAMAYAN S, TIZHOOSH H, SALAMA M. Opposition-based differential evolution[J]. IEEE Transaction Evolutionary Computation, 2008, 12(1): 64-79.
- [3] 赵新超, 熊卿, 冯帅. 基于均匀邻域对位的自适应差分进化算法[J]. 集美大学学报(自然科学版), 2021, 26(1): 72-81.  
ZHAO X Z, XIONG Q, FENG S. Opposition-based Differential Evolution Algorithm with Neighborhood-based self-adaptive Uniform Mutation[J]. Journal of Jimei University(Natural Science), 2021, 26(1): 72-81.
- [4] ZHANG J, SANDERSON A. JADE: Adaptive Differential Evolution With Optional External Archive[J]. IEEE Transaction Evolutionary Computation, 2009, 13(5): 945-958.
- [5] WANG Y, CAI Z, ZHANG Q. Differential Evolution with Composite Trial Vector Generation Strategies and Control Parameters[J]. IEEE Trans Evolutionary Computation, 2011, 15(1): 55-66.
- [6] 黄玲玲, 刘三阳, 高卫峰. 具有人工蜂群搜索策略的差分进化算法[J]. 控制与决策, 2012, 27(11): 1644-1648.  
HUANG L L, LIU S Y, GAO W F. Differential evolution with the search strategy of artificial bee colony algorithm[J]. Control and Decision, 2012, 27(11): 1644-1648.
- [7] ZHAO X, LI R, HAO J. A New Differential Mutation Based Adaptive Harmony Search Algorithm for Global Optimization[J]. Applied Sciences-basel, 2020, 10(8): 1-17.
- [8] ZHAO X, XU G, LI R. A failure remember-driven self-adaptive differential evolution with top-bottom strategy[J]. Swarm and Evolutionary Computation, 2019, 45: 1-14.
- [9] 陈恩茂, 徐志刚, 付源. 基于牛顿三次插值的自适应差分进化算法[J]. 计算机工程与设计, 2020, 41(8): 2170-2176.  
CHEN E M, XU Z G, FU Y. Adaptive differential evolution algorithm based on Newton cubic interpolation[J]. Computer Engineering and Design, 2020, 41(8): 2170-2176.
- [10] 孟红云, 张小华, 刘三阳. 用于约束多目标优化问题的双群体差分进化算法[J]. 计算机学报, 2008, 31(2): 228-235.  
MENG H Y, ZHANG X H, LIU S Y. A Differential Evolution Based on Double Populations for Constrained Multi-Objective Optimization Problem[J]. Chinese Journal of Computers, 2008, 31(2): 228-235.
- [11] 牛大鹏, 王福利, 何大阔, 等. 多目标混沌差分进化算法[J]. 控制与决策, 2009, 24(3): 361-364.  
NIU D P, WANG F L, HE D K, et al. Chaotic differential evolution for multiobjective optimization[J]. Control and Decision, 2009, 24(3): 361-364.
- [12] 叶志萍, 马军. 基于差分变异与扰动变异的多目标克隆选择算法[J]. 数学的实践与认识, 2018, 48(22): 127-134.  
YE Z P, MA J. Multi-objective colonial selection algorithm based on mutation of differential and disturbance[J]. Journal of Mathematics in Practice and Theory, 2018, 48(22): 127-134.
- [13] WANG L, PAN Q-K, SUGANTHAN P. A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems[J]. Computers & Operations Research, 2010, 37(3): 509-520.
- [14] ZHANG B, SUN X, LIU S. Adaptive Differential Evolution-Based Distributed Model Predictive Control for Multi-UAV Formation Flight[J]. International Journal of Aeronautical and Space Sciences, 2020, 21(2): 538-548.
- [15] LIANG J, QU B, SUGANTHAN P. Problem definitions and evaluation criteria for the CEC2014 special session and competition on single objective real-parameter numerical optimization[EB/OL]. [2020-09-05]. <https://max.book118.com/html/2019/0223/6051001233002011.shtm>.

## Opposition-based differential evolution algorithm with Gaussian distribution estimation

Fang Jingyuan, Ji Yisheng, Zhao Xinchao

(School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** Aiming at the problems of inefficient exploration and slow convergence, an Opposition-based Differential Evolution(ODE) with Gaussian distribution estimation is proposed in this paper. The algorithm generates a population based on Gaussian distribution estimation while it generates the opposite population, which fully explores the solution space. When the jumping condition is not met in ODE, a new jumping population is generated with Gaussian distribution estimation for well population diversity. During the selection operation, all the parents and children are mixed together for the best selection, which reduces the loss of some excellent solutions and genes. Finally, based on CEC2014 benchmark, the proposed algorithm is compared with other differential evolution algorithms. Experimental results indicate that the proposed algorithm has stronger search ability and better convergence property.

**Keywords:** differential evolution algorithm; opposition-based learning; Gaussian distribution estimation