

引用格式:吴聪聪,贺毅朝,赵建立,等. 基于精英反向学习的阶段性变异杂草算法[J]. 微电子学与计算机,2019,36(3):32-37. [WU Cong-cong, HE Yi-chao, ZHAO Jian-li, et al. Elite opposition-based learning multi-stage mutated invasive weed optimization algorithm[J]. Microelectronics & Computer,2019,36(3):32-37.]

## 基于精英反向学习的阶段性变异杂草算法

吴聪聪<sup>1</sup>,贺毅朝<sup>1</sup>,赵建立<sup>1,2</sup>,李 宁<sup>1</sup>

(1 河北地质大学 信息工程学院,河北 石家庄 050031;2 全北国立大学 电子信息工程学院,全州 韩国 54896)

**摘 要:** 提出了基于精英反向学习的阶段性变异杂草算法(Elite opposition-based learning multi-stage mutated invasive weed optimization, EOMMIWO). 该算法将正态分布的标准差初始值和终止值由固定设置转化为根据问题自适应产生;在杂草进化过程中让精英个体反向学习,提高了算法的勘探能力;另外,在算法进化的不同阶段对杂草实施不同变异,增强个体交流,提高算法的开发能力. 通过 8 个经典的 Benchmark 函数测试,结果表明该算法提高了杂草算法的求解精度和收敛速度,具有很强的鲁棒性.

**关键词:** 杂草算法;自适应标准差;精英反向学习;阶段性变异

**中图分类号:** TP181

**文献标识码:** A

**文章编号:** 1000-7180(2019)03-0032-06

## Elite opposition-based learning multi-stage mutated invasive weed optimization algorithm

WU Cong-cong<sup>1</sup>, HE Yi-chao<sup>1</sup>, ZHAO Jian-li<sup>1,2</sup>, LI Ning<sup>1</sup>

(1 College of Information Engineering, Hebei GEO University, Shijiazhuang 050031, China;

2 School of Electronics & Information Engineering, Chonbuk National University, Jeonju 54896, Korea)

**Abstract:** An elite opposition-based learning multi-stage mutated invasive weed optimization algorithm(EOMMIWO) was proposed. First, the algorithm adaptively generated the initial value and the ending value of the standard deviation of the normal distribution according to the search area of the problem; Second, the elite individuals opposition-based learned which improved the exploration ability; Finally, different mutations were adopted in different stages the algorithm, which enhanced the communication among the individuals and improves the ability in exploitation. The test and comparison results of several complex Benchmark functions show that the proposed algorithm not only improves the solution accuracy and convergence speed of IWO algorithm, but also strengthens robustness.

**Key words:** invasive weed optimization algorithm; adaptive standard deviation; elite opposition-based learning; multi-stage mutation

### 1 引言

杂草算法(Invasive Weed Optimization, IWO)<sup>[1]</sup>是 Mehrabian 和 Lucas 在 2006 年提出的一

种启发式搜索算法,其思想源于自然界杂草的繁殖生长. 算法充分利用种群中的优秀个体来指导群体的进化,子个体以正态分布的方式在父个体周围产生. IWO 算法结构简单,参数少且鲁棒性较好,在很

收稿日期:2018-07-14;修回日期:2018-08-01

基金项目:国家社会科学基金(17BGL202);河北省高等学校科学研究计划项目(ZD2016005);河北省自然科学基金(F2016403055);河北省教育厅科学技术研究重点项目(ZD2018043)

多领域得到了有效的应用<sup>[2-4]</sup>.但在解决一些较复杂问题时,IWO存在收敛速度慢,易陷入局部极值和寻优精度不高的缺陷.本文通过分析IWO算法的特征,引入反向学习理论,利用精英个体的反向学习提高杂草的全局勘探能力;用自适应的标准差初始值和终止值,代替原算法的固定值,使算法能根据问题调整搜索步长;另外,按照种群的进化进程,在不同阶段对杂草个体实施不同变异:进化前期着重个体的多样性,通过变异提高全局搜索能力;进化中期兼顾勘探能力和开发能力,通过变异提高算法跳出局部极值的能力;进化后期更注重在精英个体周围的搜索,提高算法的求解精度和收敛速度.通过8个复杂的Benchmark函数的测试表明,本文提出的各种策略大大提高了杂草算法的求解精度和收敛速度,并且使算法具有了更强的鲁棒性.

## 2 基本杂草算法

杂草是大自然中一种较顽强的植物,靠着适者生存的本能不断的繁衍子孙,扩大它们的生存范围.IWO算法<sup>[1]</sup>正是受大自然中杂草繁衍的启发而设计.杂草根据各自生命力的强弱(即适应度高低)按比例产生草种子,生命力越强的(适应度越高)杂草产生的种子越多;相反,生命力越弱的(适应度越低)产生的种子越少.每个杂草产生的种子按照正态分布分散在它的周围,正态分布的步长随着代数的增加而变化:刚开始步长大,而后越来越小.杂草和种子都是种群中的个体,如果杂草和种子的总个数大于预先设定的种群最大规模P\_MAX,则选择其中适应度高的P\_MAX个个体作为下一代的杂草.基本杂草算法流程如下<sup>[1]</sup>:

Step1:初始化种群.在 $d$ 维空间上随机产生 $N$ 个可行解 $X_1, X_2, \dots, X_N, X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}\}, i = 1, 2, \dots, N$ ,并计算适应度.

Step2:繁殖.按式(1)计算每个杂草 $X_i$ 所能产生的种子数:

$$w_1 = \frac{f(X_i) - f_{\min}}{f_{\max} - f_{\min}}(s_{\max} - s_{\min}) + s_{\min} \quad (1)$$

式中: $f(X_i)$ 是杂草 $X_i$ 的适应度; $f_{\max}$ 和 $f_{\min}$ 分别是种群中杂草对应的最大适应度和最小适应度; $s_{\max}$ 和 $s_{\min}$ 是一个杂草所能产生草种子的最大数目和最小数目,由算法预先设定.

Step3:散布草种子.每个杂草 $X_i$ 将 $w_i$ 个草种按 $N(0, \sigma_{\text{iter}})$ 的正态分布散布在自己周围,其中每代标准差按式(2)计算得到:

$$\sigma_{\text{iter}} = \frac{(\text{iter}_{\max} - \text{iter})^n}{(\text{iter}_{\max})^n}(\sigma_{\text{initial}} - \sigma_{\text{final}}) + \sigma_{\text{final}} \quad (2)$$

式中,iter是进化代数;iter<sub>max</sub>为最大进化代数; $\sigma_{\text{initial}}$ 和 $\sigma_{\text{final}}$ 分别是标准差的最初值和最终值; $n$ 为非线性调和因子,一般情况下 $n = 3$ <sup>[1]</sup>.

Step4:个体选择.对种群中杂草和种子按照适应度大小进行排序.算法进化过程中,杂草和种子的总数目如果达到了种群的最大规模P\_MAX,选取适应度较好的前P\_MAX个个体,淘汰其余个体.

## 3 基于精英反向学习的阶段性变异杂草算法

### 3.1 自适应标准差设置

IWO算法按正态分布在每个杂草周围产生种子,正态分布的标准差按式(2)变化.从较大标准差变化到较小标准差,使得算法的搜索范围从开始的松散,到后来的聚集,从大范围的全局搜索到细密的局部搜索,标准差的变化范围是由最初值 $\sigma_{\text{initial}}$ 和最终值 $\sigma_{\text{final}}$ 决定的.而现实中,对于不同的求解问题,可行解的约束范围是不同的.如果对不同问题设定相同的 $\sigma_{\text{initial}}$ 和 $\sigma_{\text{final}}$ ,即对不同问题按照相同的起始标准差、终止标准差和标准差变化率产生种子,显而易见是不合理的.针对此问题,基本IWO<sup>[1]</sup>采用的对策是对不同问题设置不同的 $\sigma_{\text{initial}}$ 和 $\sigma_{\text{final}}$ ,但是,这种对每个问题设置参数的方法太繁琐.本文受文献[5]启发,根据问题搜索的范围 $[x_{\min}, x_{\max}]$ 来设置标准差的初始值和终止值,如式(3)所示,其中 $\alpha$ 和 $\beta$ 为标准差参考系数,均为0到1之间的实数, $\alpha$ 用于控制初始标准差与合法搜索宽度的关系.根据正态分布的特点,如果 $\alpha$ 设置过大,算法初期会产生过多的超出搜索范围的种子;设置过小,会减弱算法勘探能力,经过多次实验,建议 $\alpha$ 设置在0.3附近,本文设置为0.3. $\beta$ 控制终止标准差与初始标准差的关系,它体现了算法搜索步长的变化和最终步长,本文设置为0.05.这样,算法能够针对不同问题,根据其定义域,自适应的产生与之相对应的标准差初始值和终止值.

$$\begin{cases} \sigma_{\text{initial}} = (x_{\max} - x_{\min})\alpha \\ \sigma_{\text{final}} = \sigma_{\text{initial}}\beta \end{cases} \quad (3)$$

### 3.2 反向学习

2005年Tizhoosh<sup>[6]</sup>提出了反向学习(Opposition-based Learning, OBL)的概念,认为在智能搜索中,不断的学习、优化和搜索是最根本的任务.反向学习的目的是扩大估计范围(同时评价当前解和反向解)从而得到更优秀的近似解.OBL能够有效的

提高种群的多样性,对算法跳出局部极值有很大帮助.反向学习因其有效性被学者们应用于各种算法中.杂草算法的种子是在父草周围按正态分布产生的,所以反向学习使杂草对合法空间的搜索更全面和精细.

### 定义 1 反向点<sup>[6]</sup>

设  $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$  是  $d$  维搜索空间中的可行解,  $x_{i,j} \in [a_j, b_j], x_{i,j} \in R, \forall j \in (1, 2, \dots, d)$ , 那么它的反向点(反向解)  $\tilde{x}_{i,1}, \tilde{x}_{i,2}, \dots, \tilde{x}_{i,d}$ .

$$\tilde{x}_{i,j} = a_j + b_j - x_{i,j} \quad (4)$$

### 定义 2 浮动反向点

$X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$  是  $d$  维搜索空间中的可行解,  $x_{i,j} \in [a_j, b_j], x_{i,j} \in R, \forall j \in (1, 2, \dots, d)$ , 那么它的浮动反向点(浮动反向解)  $\tilde{\tilde{x}}_{i,1}, \tilde{\tilde{x}}_{i,2}, \dots, \tilde{\tilde{x}}_{i,d}$ .

$$\tilde{\tilde{x}}_{i,j} = a_j + b_j - x_{i,j} \quad (5)$$

$\theta \in [0, 1]$  是符合均匀分布的随机数. 这样在求  $X_i$  反向点的时候, 先将  $X_i$  做微小的随机浮动, 使得反向点更具有探测性, 并且, 同一个点  $X_i$  在不同迭代次数求得的浮动反向点会不同, 进一步增强了算法的勘探能力.

在求浮动反向点过程中如果出现不可行解, 可按如下方法处理:

$$\text{if}(\tilde{\tilde{x}}_{i,j} > a_j) \tilde{\tilde{x}}_{i,j} = a_j$$

$$\text{if}(\tilde{\tilde{x}}_{i,j} < b_j) \tilde{\tilde{x}}_{i,j} = b_j$$

本文所使用的反向学习策略就是对个体求其浮动反向点. 如果浮动反向点的适应度好于原个体, 则由浮动反向点代替原个体.

根据种群中进行反向学习的个体范围可分为三种反向学习:(1) 种子反向学习: 在种子产生的时候对种子个体浮动反向, 同时测试当前种子和反向种子的优劣;(2) 精英反向学习: 对杂草(父种群) 浮动反向, 这相当于让群体中的精英个体进行反向学习;(3) 全体反向学习: 对进化中杂草和种子都进行浮动反向学习. 为了比较三种反向策略的有效性, 本文从 8 个测试函数中随机选择了两个单峰函数和两个多峰函数对它们进行了测试(第 4 节中  $f_1, f_2, f_3, f_6$ ). 测试在相同的条件下进行, 图 1 是杂草算法在三种反向学习下, 对每个函数独立运行 50 次得到的最优值(求最小值问题) 构成的箱图.

从图 1 可以看出, 精英反向和全部种群反向的求解效果相当, 而只对种子反向的策略求解精度不如精英反向和全部种群反向. 考虑到计算成本, 全部

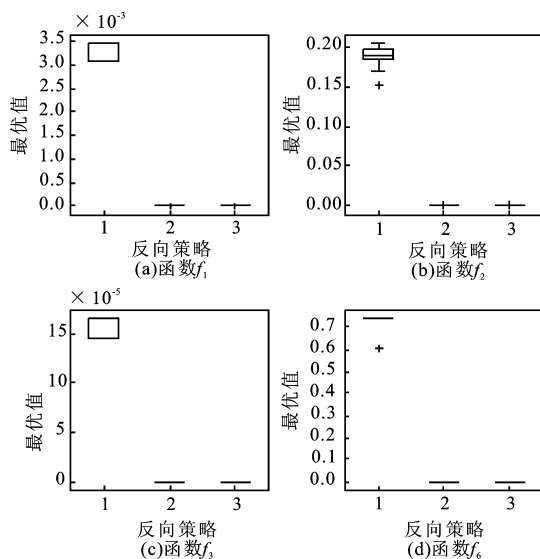


图 1 三种反向学习策略性能的比较

种群反向比精英反向需要耗费更多的时间, 所以采用精英反向是最恰当的选择.

### 3.3 阶段性变异

在杂草算法中, 种子围绕杂草个体正态分布, 种子的产生完全依靠杂草个体, 而杂草个体之间没有交流, 缺少社会性, 使得算法在中后期容易陷入局部极值. 为此, 本文引入差分演化中的变异算子. 文献[7]将差分的变异和交叉算子引入杂草算法, 对新产生的种子进行变异交叉操作, 虽然在一定程度上提高了种群多样性, 但是, 由于新产生的种子优劣无法确定, 不能保证变异后得到更优秀个体. 本文让种群的精英个体(杂草)之间进行交叉变异, 使得优秀个体之间相互学习, 增大产生更优秀个体的可能性, 不但能提高种群的多样性, 而且在提高算法寻优能力和收敛速度上也起到重要作用. 文献[8]分析了差分演化算法的多种进化模式, 并将它们分为三类: 第一类为 DE/rand/1/bin 和 DE/rand/2/bin; 第二类为 DE/best/1/bin 和 DE/best/2/bin; 第三类是 DE/rand-to-best/1/bin 模式. 每类进化模式都有各自的优势和不足. 文献[8]指出: 第一类模式, 群体的自由探索性强, 全局搜索能力强, 使得算法不易陷入局部极值, 但是收敛速度会减慢; 第二类模式, 因为以全局最优为基准, 所以局部探索性好, 收敛速度变快, 但容易陷入局部极值; 第三类模式以个体  $X_i$  为基准, 均衡局部探索和全局探索能力, 但寻优的鲁棒性不够. 本文根据这三类模式的特点, 将它们分别用于杂草算法的进化初期, 中期和后期, 充分发挥三种模式的优势: 算法进化初期, 需要个体有很高的探索性, 要求个体在空间充分扩散, 以提高全局搜索能力, 此时使用第一类模式; 算法中期, 既

需要强调个体的局部探索,又需要加强群体的全局搜索,第三类模式更适合;算法后期,加强算法在局部搜索,尤其是在优秀个体周围的搜索,更有利于提高算法寻优能力和收敛速度,此时使用第二类模式.本文选取每类模式中一种变异,式(6)属于第一类模式,式(7)属于第二类模式,式(8)属于第三类模式.关于模式分类更详尽的内容请参阅文献[8].

$$X' = X_{k1} + F * (X_{k2} - X_{k3}) \quad (6)$$

$$X' = X_g + F * (X_{k2} - X_{k3}) \quad (7)$$

$$X' = X_i + F * (X_g - X_i + F(X_{k1} - X_{k2})) \quad (8)$$

式中, $X'$ 是变异得到的临时个体; $X_{k1}, X_{k2}, X_{k3}$ 是与当前个体 $X_i$ 不同的互不相同三个杂草个体; $X_g$ 是目前群体中的最优杂草个体; $F$ 是调节系数,在 $[0, 1]$ 范围取值.

在杂草个体每次变异后,对变异得到的临时个体 $X' = [x'_1, x'_2, \dots, x'_d]$ 进行二项交叉,如式(9),其中 $Cr \in [0, 1]$ 为交叉因子, $\text{rand}$ 为在 $[0, 1]$ 上的随机数;对交叉后得到的临时个体计算其适应度,如果适应度优于当前个体 $X_i$ ,则替换当前个体,否则 $X_i$ 保持不变.

$$X'_j = \begin{cases} X'_j & \text{if } (\text{rand} > Cr) \\ x_{i,j} & \text{others} \end{cases} \quad j = \{1, 2, \dots, d\} \quad (9)$$

### 3.4 基于精英反向学习的阶段性变异杂草算法

将前面提到的自适应标准差应用到基本 IWO 算法中,并加入精英反向学习策略和阶段性变异,就形成了基于精英反向学习的阶段性变异杂草算法 (Elite opposition-based learning mutil-stage mutated IWO, EOMMIWO). 图 2 给出了算法 EOMMIWO 的执行流程. 其中, EOMMIWO 算法的参数包括算法的最大迭代次数 MAXiter, 产生种子的最大个数  $s_{\max}$  和最小个数  $s_{\min}$ , 标准差参考系数  $\alpha$  和  $\beta$ , 变异的调节系数  $F$ , 交叉因子  $Cr$  和杂草种群的最大规模  $P_{\max}$ .

## 4 仿真实验与结果分析

### 4.1 测试函数和实验环境

为了评估本文提出 EOMMIWO 算法的优化性能,使用文献[9]中 8 个国际通用 Benchmark 函数进行了测试. 仿真实验所使用环境为: intel(R) Core (TM) i5-7Y54 CPU-1.2 Hz, 内存为 8 GB (7.89 GB 可用), 操作系统为 Windows 10, 编程环境 Microsoft VS 2010, 所有代码使用 C++ 编写.

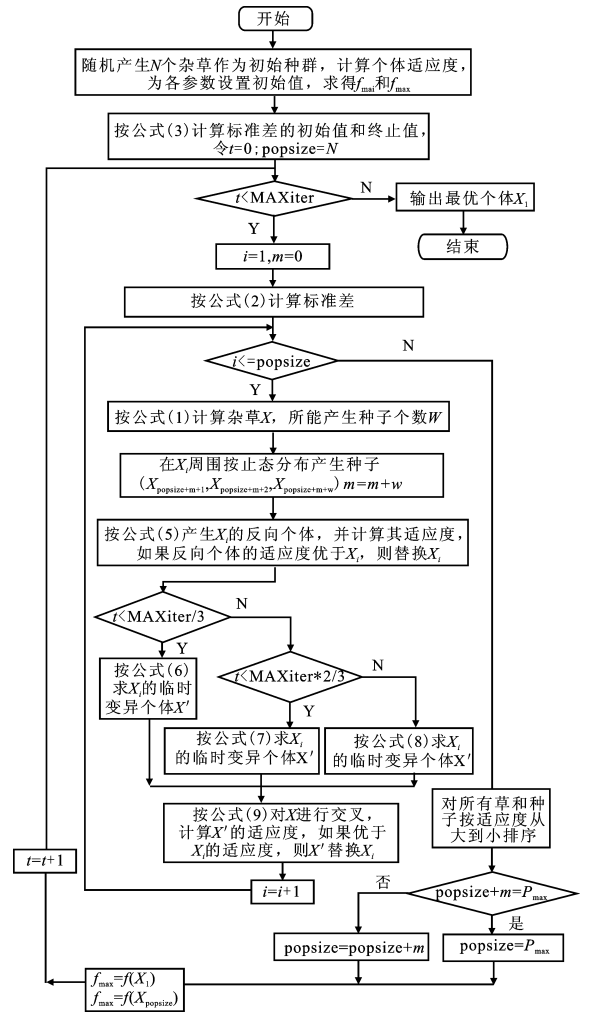


图 2 EOMMIWO 算法流程

### 4.2 实验结果分析

本文从运行结果的最优值 (Best)、平均值 (Mean)、最差值 (Worst) 和标准差 (Std) 四个方面对 DE, IWO, EOMMIOW 三个算法进行了比较. 三个算法的工作参数设置如表 1 所示. 每个算法最大迭代次数为 500 次, 对每个测试函数独立运行 50 次, 运行结果如表 2 所示.

表 1 三个算法的参数设置

| algorithm | parameter settings                                                                                         |
|-----------|------------------------------------------------------------------------------------------------------------|
| DE        | $N=40, F=0.8, Cr=0.3$                                                                                      |
| IWO       | $n=3, P_{\max}=40, N=10, \sigma_{\text{initial}}=5.12, \sigma_{\text{final}}=0.05, s_{\max}=3, s_{\min}=0$ |
| EOMMIWO   | $n=3, P_{\max}=40, N=10, \alpha=0.3, \beta=0.05, F=0.8, Cr=0.3$                                            |

从表 2 可以看出, 对于函数维度为 30 ( $f_8$  为 2 维) 和 100, EOMMIWO 算法对 8 个函数的求解在最优值、平均值、最差值和标准差上都远远优于基本的 IWO 算法和 DE 算法, 新算法的求解性能非

表 2 算法 DE、IWO、EOMMIWO 测试结果

| function | Algorithm | 30 维( $f_8$ 为 2 维) |           |           |           | 100 维     |           |           |           |
|----------|-----------|--------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
|          |           | Best               | Mean      | Worst     | Std       | Best      | Mean      | Worst     | Std       |
| $f_1$    | DE        | 2.7E-002           | 9.12E-002 | 2.27E-001 | 4.72E-002 | 2.82E+004 | 2.23E+004 | 1.56E+004 | 3.05E+003 |
|          | IWO       | 3.2E-002           | 3.33E-002 | 3.41E-002 | 6.40E-004 | 1.64E+004 | 2.44E+004 | 3.22E+004 | 5.91E+003 |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $f_2$    | DE        | 3.38E-002          | 6.68E-002 | 1.36E-001 | 1.93E-002 | 9.40E-002 | 6.28E-002 | 3.35E-002 | 1.48E-002 |
|          | IWO       | 8.97E-001          | 2.34E+00  | 1.15E+001 | 3.40E+0   | 2.25E+002 | 3.21E+002 | 4.18E+002 | 4.45E+001 |
|          | EOMMIWO   | 3.25E-211          | 4.97E-200 | 9.23E-199 | 0         | 1.07E-202 | 1.95E-193 | 4.14E-192 | 0         |
| $f_3$    | DE        | 6.34E-002          | 3.22E-001 | 6.47E-001 | 1.54E-001 | 2.85E+002 | 2.00E+002 | 1.47E+002 | 2.78E+001 |
|          | IWO       | 2.99E+002          | 3.48E+002 | 4.34E+002 | 5.27E+001 | 1.80E+003 | 1.94E+003 | 2.03E+003 | 8.79E+001 |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $f_4$    | DE        | 4.37E-014          | 2.70E-015 | 5.22E-018 | 6.56E-015 | 5.70E-014 | 2.95E-015 | 2.94E-018 | 8.46E-015 |
|          | IWO       | 5.23E-006          | 1.13E-005 | 1.22E-005 | 1.37E-006 | 1.94E-006 | 5.8E-006  | 2.44E-005 | 4.92E-006 |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $f_5$    | DE        | 2.11E-001          | 1.09E-001 | 5.99E-002 | 3.22E-002 | 1.57E+001 | 1.45E+001 | 1.29E+001 | 6.62E-001 |
|          | IWO       | 1.65E-001          | 2.49E-001 | 3.98E-001 | 1.11E-001 | 1.33E+001 | 1.69E+001 | 1.93E+001 | 2.22E+000 |
|          | EOMMIWO   | 4.44E-016          | 4.44E-016 | 4.44E-016 | 0         | 4.44E-016 | 4.44E-016 | 4.44E-016 | 0         |
| $f_6$    | DE        | 1.04E+002          | 1.25E+002 | 1.47E+002 | 9.59E+000 | 1.05E+003 | 9.95E+002 | 9.43E+002 | 2.97E+001 |
|          | IWO       | 6.41E+001          | 8.74E+001 | 1.01E+002 | 1.82E+001 | 7.40E+002 | 7.89E+002 | 8.90E+002 | 4.98E+001 |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $f_7$    | DE        | 9.94E-001          | 4.82E+00  | 1.01E+001 | 2.18E+00  | 1.35E+001 | 5.04E+000 | 9.39E-001 | 3.02E+000 |
|          | IWO       | 4.25E-003          | 4.25E-003 | 4.25E-003 | 0         | 3.76E-003 | 4.60E-003 | 4.99E-003 | 5.74E-004 |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | 0         | 0         | 0         | 0         |
| $f_8$    | DE        | 3.13e-019          | 2.89e-015 | 2.54e-014 | 5.94e-015 | —         | —         | —         | —         |
|          | IWO       | 1.75E-003          | 1.75E-003 | 1.75E-003 | 0         | —         | —         | —         | —         |
|          | EOMMIWO   | 0                  | 0         | 0         | 0         | —         | —         | —         | —         |

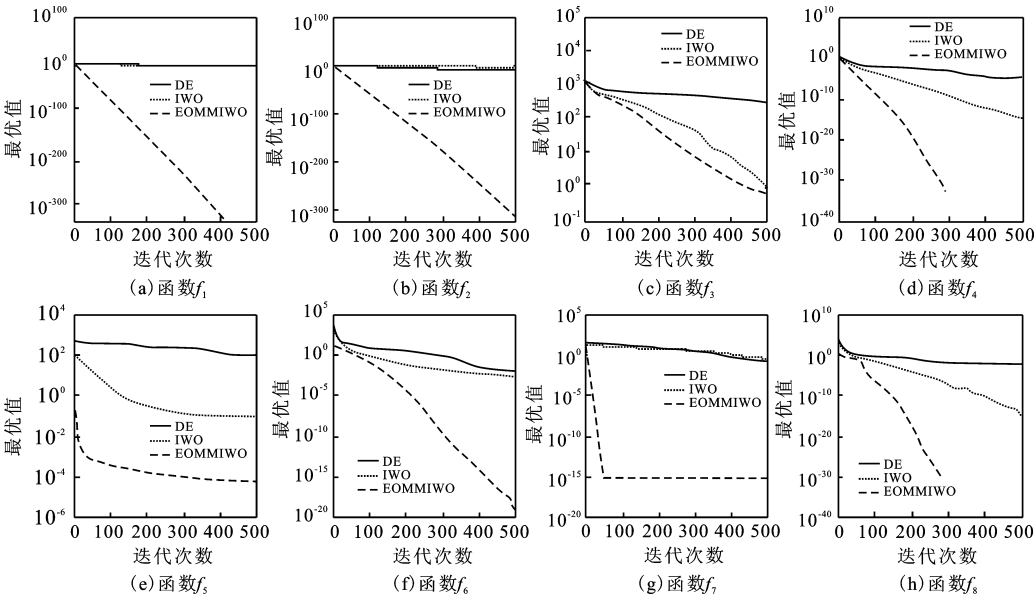


图 3 算法 DE、IWO、EOMMIWO 在函数  $f_1 \sim f_8$  上低维上的收敛曲线图

常好. 另外, EOMMIWO 的求解精度没有受到维度变化的影响, 而且求解的标准差都很低, 从这两方面看, 相对于基本的 IWO, EOMMIWO 的鲁棒性得到了很大提高. 图 3 是三个算法在低维下求解 8 个函数的平均收敛曲线图. 由于精英反向学习提高了 EOMMIWO 算法在整个进化过程中的勘探能力, 使得算法在求解精度上一直高于 IWO 和 DE; 阶段性变异在算法前期, 旨在提高个体多样性, 使 EOMMIWO 算法在进化前期就能快速向最优解收敛, EOMMIWO 算法的收敛速度明显高于 IWO 和 DE, 求解精度也远远高于他们.

5 结束语

本文针对 IWO 算法易早熟, 求解精度不高, 收敛速度慢的缺陷, 提出了基于精英反向学习的阶段性变异杂草算法 (EOMMIWO). EOMMIWO 对基本杂草算法进行了三个方面的改进: 第一, 提出了自适应标准差的策略, 将固定的初始标准差和终止标准差转化为根据问题的搜索区域而变化的值, 充分利用问题本身的条件在父个体周围产生子个体, 提高了搜索的精准度. 第二, 在杂草进化过程中让精英个体反向学习, 提高了算法的勘探能力和求解精度, 并且提出了浮动反向点的概念, 提升了反向学习的效果. 第三, 在算法进化的不同阶段, 对杂草群采用了不同变异策略, 不但增强了群体多样性和算法的全局寻优能力, 同时提高了算法的收敛速度. 通过 8 个 Benchmark 函数的测试表明, 该算法大大提高了 IWO 算法的求解精度和收敛速度.

参考文献:

[1] MEHRABIAN A R, LUCAS C. A novel numerical optimization algorithm inspired from weed colonization [J]. Ecological Informatics, 2006, 1(3):355-366.

[2] DORNELLES R, PAIM G, SILVEIRA B, et al. A power-efficient 4-2 adder compressor topology[C]// 2017 15th IEEE International New Circuits and Systems Conference (NEWCAS). Strasbourg, 2017: 281-284.

[5] PISHVAIE A, JABERIPUR G, JAHANIAN A. Re-designed CMOS (4; 2) compressor for fast binary multipliers[J]. Canadian Journal of Electrical and

[2] POUYA A R, SOLIMANPUR M, REZAEI M J. Solving multi-objective portfolio optimization problem using invasive weed optimization[J]. Swarm & Evolutionary Computation, 2016(28):42-57.

[3] 曹磊, 叶春明. 杂草-蚁群算法在应急管理中的应用[J]. 微电子学与计算机, 2016, 33(9):150-154.

[4] BARHAGHTALAB M H, MEIGOLI V, ZAVVAR M. Optimal fuzzy controller based on chaotic invasive weed optimization for damping power system oscillation[J]. Smart Science, 2017 (2) :1-16.

[5] BASAK A, MAITY D D. A differential invasive weed optimization algorithm for improved global numerical optimization[J]. Applied Mathematics and Computation, 2013, 219(12):6645-6668.

[6] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence[C]//Proceedings of IEEE Computational Intelligence for Modelling, Control and Automation Vienna, Austria, 2005:695-701.

[7] 李森, 任晓娜. 采用差分进化策略的入侵杂草改进算法及函数优化应用[J]. 计算机应用与软件, 2014, 31(6):285-259.

[8] 贺毅朝, 王熙照. 差分演化的收敛性分析与算法改进[J]. 软件学报, 2010, 21(5): 875-885.

[9] 陈欢, 周永权, 赵光伟. 基于混沌序列的多种群入侵杂草算法[J]. 计算机应用, 2012, 32(7):1958-1961.

作者简介:

吴聪聪 女, (1975-), 讲师. 研究方向为智能计算、信息检索、机器学习. E-mail: hebwucongcong@126.com.

贺毅朝 男, (1969-), 教授. 研究方向为进化算法理论与应用、算法设计与分析、计算复杂性理论与群测试理论.

赵建立 男, (1976-), 讲师. 研究方向为机器学习、智能计算.

李 宁 女, (1977-), 硕士, 副教授. 研究方向为智能计算、机器学习.

(上接第 31 页)

Computer Engineering, 2013, 36(3): 111-115.

作者简介:

仲 亚 男, (1994 -), 硕士研究生. 研究方向为超大规模集成电路设计、运算电路设计. E-mail: zhongya0614@qq.com.

叶瑶瑶 女, (1986-), 博士, 讲师. 研究方向为系统级芯片设计