

## 基于双变异策略的自适应骨架差分进化算法

刘会宇<sup>1,2</sup>, 韩继红<sup>1</sup>, 袁霖<sup>1</sup>, 于波<sup>1,3</sup>

(1. 解放军信息工程大学, 河南 郑州 450001; 2. 中国人民解放军 75741 部队, 广东 广州 510510;  
3. 中国人民解放军国防信息学院, 湖北 武汉 430010)

**摘要:** 骨架差分进化算法能够较好规避差分进化算法控制参数和变异策略选择问题。针对基于双变异策略的经典骨架差分进化算法 (MGBDE) 没有根据个体进化差异选择适合的变异策略和考虑早熟收敛的问题, 提出一种改进算法。该算法引入变异策略选择因子, 并借鉴自适应差分进化算法的设计思想, 将选择因子随个体共同参与进化, 使个体执行当前最为适合的变异策略, 克服原始算法进化过程的盲目性, 同时选择因子的动态自适应特性保持了骨架算法近似无参数的优点; 该算法加入停滞扰动策略, 降低陷入局部最优的风险。采用 18 个标准测试函数进行实验, 结果表明, 新算法在收敛精度、收敛速度和顽健性上整体优于多种同类骨架算法以及知名的差分进化算法。

**关键词:** 差分进化; 骨架算法; 双变异策略; 自适应

中图分类号: TP18

文献标识码: A

## Self-adaptive bare-bones differential evolution based on bi-mutation strategy

LIU Hui-yu<sup>1,2</sup>, HAN Ji-hong<sup>1</sup>, YUAN Lin<sup>1</sup>, YU Bo<sup>1,3</sup>

(1. PLA Information Engineering University, Zhengzhou 450001, China; 2. PLA Unit 75741, Guangzhou 510510, China;  
3. Chinese People's Liberation Army National Defense Information Institute, Wuhan 430010, China)

**Abstract:** Bare-bones differential evolution (BBDE) can elegantly solve the selection problem of control parameters and mutation strategy in differential evolution (DE). MGBDE is a classical BBDE based on bi-mutation strategy. However, it randomly assigns a mutation strategy to each individual, not considering their differences during evolution process, meanwhile it may suffer from premature convergence. To overcome these drawbacks, a modified algorithm based on MGBDE was proposed. A mutation strategy choice factor that guided the individual to choose a preferable mutation strategy at each mutation operation was introduced, avoiding the evolution blindness brought by the random selection of mutation strategy. To retain the almost parameter-free characteristic of bare-bones algorithm, the tuning of choice factor to be adapted was involved in the individual evolution, inspired by the concept of self-adaptive DE. The algorithm also included a well-designed stagnation perturbation mechanism to reduce the risk of trapping into the local optimal. Experimental results on 18 benchmark functions show that the proposed algorithm generally achieves better performance than state-of-the-art BBDE variants and several well-known DE algorithms in terms of convergence and robustness.

**Key words:** differential evolution, bare-bones algorithm, bi-mutation strategy, self-adaptive

### 1 引言

差分进化 (DE, differential evolution) 算法是由 Storn 等<sup>[1]</sup>提出的一种简单高效的全局优化算法。研究表明, DE 算法对于求解多目标优化、约束优化、动态不确定性优化及多峰优化等问题表现出色, 是

一种十分具有竞争力的新型进化算法, 现已在众多领域得到广泛应用<sup>[2]</sup>。然而, DE 算法性能很大程度上取决于控制参数和变异策略的选取, Storn 等<sup>[3]</sup>基于大量实验, 给出了控制参数的取值建议和不同变异策略的适合场景。但此后研究人员发现, 当控制参数取值固定时, 优化结果并非十分理想, 所以

收稿日期: 2016-09-17; 修回日期: 2016-11-30

基金项目: 国家自然科学基金资助项目 (No.61309018)

**Foundation Item:** The National Natural Science Foundation of China (No.61309018)

陆续有研究者提出了一些控制参数和变异策略自适应调节的 DE 算法, 如 jDE 算法<sup>[4]</sup>、SaDE 算法<sup>[5]</sup>、JADE 算法<sup>[6]</sup>及上述算法的变体<sup>[7,8]</sup>等。这类算法的特点是将控制参数和变异策略的调节纳入进化过程, 记录每次产生优质解的经验, 为下一次迭代提供参考。

将 DE 算法与其他算法混合是 DE 算法改进的另一思路<sup>[9-11]</sup>。BBDE 算法是 DE 算法与 PSO 算法的结合, 最早由 Omran 等<sup>[12]</sup>提出。该算法借鉴 BBPSO 算法<sup>[13]</sup>中关于种群粒子的收敛性结论, 在基于 DE/rand/1 策略 DE 算法的变异向量生成中引入当前全局最优  $\mathbf{g}_{\text{best}}$  和个体最佳  $\mathbf{p}_{\text{best}}$  的随机数权值的求和运算。随后, Wang 等<sup>[14]</sup>提出了 BBDE 算法的高斯变异形式——GBDE 算法, 其主体思想是种群个体收敛于以  $\mathbf{g}_{\text{best}}$  为参数的高斯分布函数。此外, 他们还给出了一个改进的 GBDE 版本——MGBDE。最近, 彭虎等<sup>[15]</sup>对 BBDE 算法进行了梳理, 并提出了基于三角高斯变异策略的 tBBDE 算法, 以 3 个随机个体的几何中心值代替传统 BBDE 算法中的  $\mathbf{g}_{\text{best}}$  和  $\mathbf{p}_{\text{best}}$ , 旨在增强算法的全局搜索能力。本文将 BBDE 算法、GBDE 算法、MGBDE 算法和 tBBDE 算法统称为 BBDE 算法家族, 这类算法最大的优点在于较好地规避了标准 DE 算法中控制参数和变异策略选取问题; 另外由于 PSO 算法自身的简单性, BBDE 算法家族在实现上也易于一些自适应 DE 算法, 本文的工作也属于这一范畴。

基于高斯变异策略的经典 MGBDE 算法在执行变异操作时, 以随机等概率方式为个体指定变异策略, 没有考虑个体在进化过程中的差异性, 容易导致进化过程盲目。此外, 算法缺乏对早熟收敛和搜索停滞问题的应对措施。对此, 本文提出一种 MGBDE 的改进算法 self-adaptive MGBDE (SMGBDE)。该算法引入新的控制参数—变异策略选择因子, 基于适应值引导个体选择当前更为适合的变异策略; 借鉴自适应 DE 算法的设计思想, 让选择因子随个体共同进化寻优, 实现选择因子的自适应调节, 保持了骨架算法近似无参数的优点; 算法加入了停滞扰动策略, 观察种群进化过程, 当出现早熟收敛和搜索停滞征兆时, 及时对种群进行扰动, 从而尽可能跳出该局部极值区域。实验证明 SMGBDE 算法在收敛精度、收敛速度和顽健性上整体优于同族算法和一些知名的 DE 算法。

## 2 相关背景介绍

### 2.1 标准 DE 算法

DE 算法首先在问题的可行解空间随机生成大小为  $NP$  的初始种群:  $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ , 其中,  $i = 1, 2, \dots, NP$ ,  $D$  为问题维数。然后迭代执行变异、交叉和选择操作, 直到满足预设的终止条件。

差分变异是 DE 算法的思想核心。在 DE 算法中, 种群内不同个体的差分向量经过缩放后, 与种群内另一相异个体(称为目标向量)相加生成变异向量  $\mathbf{v}_{i,g}$ 。根据变异向量生成方式的不同, 形成多种变异策略。常用变异策略主要有以下 5 种。

1) DE/rand/1

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_1,g} + F(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) \quad (1)$$

2) DE/best/1

$$\mathbf{v}_{i,g} = \mathbf{x}_{\text{best},g} + F(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) \quad (2)$$

3) DE/target-to-best/1

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_1,g} + F(\mathbf{x}_{\text{best},g} - \mathbf{x}_{r_1,g}) + F(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) \quad (3)$$

4) DE/rand/2

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_1,g} + F(\mathbf{x}_{r_2,g} - \mathbf{x}_{r_3,g}) + F(\mathbf{x}_{r_4,g} - \mathbf{x}_{r_5,g}) \quad (4)$$

5) DE/best/2

$$\mathbf{v}_{i,g} = \mathbf{x}_{\text{best},g} + F(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}) + F(\mathbf{x}_{r_3,g} - \mathbf{x}_{r_4,g}) \quad (5)$$

其中,  $g$  表示种群当前进化代数,  $\mathbf{x}_{r_1,g}$ 、 $\mathbf{x}_{r_2,g}$ 、 $\mathbf{x}_{r_3,g}$ 、 $\mathbf{x}_{r_4,g}$  和  $\mathbf{x}_{r_5,g}$  分别表示从当代种群中随机选取的互不相同的个体,  $\mathbf{x}_{\text{best},g}$  为当代种群中的最优个体,  $F$  为缩放因子。

为了增加种群潜在的多样性, 对变异向量执行交叉操作生成实验向量。DE 算法有 2 种交叉方式: 二项式交叉和指数交叉。其中, 二项式交叉对于多数问题求解表现较好, 在实际中多被采用。

二项式交叉操作的方程为

$$u_{i,j,g} = \begin{cases} v_{i,j,g}, & \text{rand}(0,1) \leq CR \text{ 或 } j = j_{\text{rand}} \\ x_{i,j,g}, & \text{其他} \end{cases} \quad (6)$$

其中,  $j_{\text{rand}}$  为  $[1, D]$  内随机选取的整数,  $CR$  为交叉率, 取值通常在  $(0, 1)$  之间。

选择 DE 算法采用贪婪选择策略, 根据目标向量  $\mathbf{x}_{i,g}$  和实验向量  $\mathbf{u}_{i,g}$  的适应值选择最优个体。对于

最小优化问题, 选择操作的方程为

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g}, \text{其他} \end{cases} \quad (7)$$

其中,  $\mathbf{x}_{i,g+1}$  为最终得到的下一代个体。

## 2.2 MGBDE 算法

Wang 等<sup>[14]</sup>于 2013 年提出了基于高斯变异的新 BBDE 算法 (GBDE) 及其改进版本 MGBDE。其中, MGBDE 算法将 GBDE 算法中的高斯变异策略和标准 DE 算法中 DE/best/1 策略进行组合, 种群中每一个体随机等概率地从中选择一种变异策略进行更新, 如式(8)所示。

$$M_i = \begin{cases} \text{DE/best/1}, & \text{rand}_i(0,1) \leq 0.5 \\ N\left(\frac{\mathbf{x}_{i,g} + \mathbf{g}_{\text{best}}}{2}, |\mathbf{x}_{i,g} - \mathbf{g}_{\text{best}}|\right), & \text{其他} \end{cases} \quad (8)$$

其中,  $M_i$  表示个体  $\mathbf{x}_i$  执行的变异策略。

## 3 SMGBDE 算法

本文算法是在 MGBDE 算法的基础上进行改进, 通过优化变异策略选择方式和加入停滞扰动策略, 改善算法的搜索性能。

### 3.1 MGBDE 算法分析

对式(8)所示的变异策略进行分析。对于高斯变异策略, 在寻优初期个体广布于解空间, 此时标准差较大, 算法偏向于全局搜索, 并逐渐向全局最优个体靠近; 随着优化的推进, 标准差逐渐减小, 算法主要围绕全局最优个体进行局部搜索以寻求更优解。因而, 高斯变异策略能够较好地实现算法前期探索能力与后期开发能力的调节。DE/best/1 策略由于利用当前种群最优个体信息引导进化过程, 使种群向最优个体收敛加快。因而, MGBDE 算法的收敛速度快于 GBDE 算法。

但 MGBDE 算法存在 2 点不足。1) 变异策略的指定未考虑个体当前特征, 进化过程存在一定盲目性。如对于当前最优个体, 应以较大概率选择 DE/best/1 策略, 在附近的小区域内进行搜索, 寻找更优解。若以等概率执行 2 种变异策略, 则可能会使该个体偏离当前区域, 错失最优值的寻找。2) 引入 DE/best/1 策略, 在加快算法寻优速度的同时, 也增加了早熟收敛的风险, 算法未给出相应的解决办法。基于上述 2 点, 本文对 MGBDE 算法进行改进。

### 3.2 基于选择因子引导的自适应变异策略

MGBDE 算法最大的不足在于不能根据个体的进化状态选择最适合的变异策略。本文通过对 2 种变异策略的分析, 对于最优个体邻近的个体, 若执行 DE/best/1 策略, 能使其继续在最优个体附近寻优, 有利于更优解的开发; 而与最优个体相距较远的个体, 若执行高斯变异策略, 能够扩大最优解的搜索范围, 增强算法的探索能力。基于此, 本文根据个体适应值与当前全局最优值的比较确定个体将执行何种变异策略。引入新的控制参数—变异策略选择因子  $SA$ , 通过改变  $SA$  的大小, 使个体在不同阶段执行不同的变异策略。为使  $SA$  的调节与个体进化同步, 本文借鉴自适应 DE 算法中让控制参数随个体共同进化寻优的做法, 将  $SA$  与个体的进化信息绑定, 即

$$SA_i = 1 - e^{-|f(\mathbf{x}_i) - f(\mathbf{g}_{\text{best}})|} \quad (9)$$

因此, 式(8)所示的变异策略选择方式改为

$$M_i = \begin{cases} N\left(\frac{\mathbf{x}_{i,g} + \mathbf{g}_{\text{best}}}{2}, |\mathbf{x}_{i,g} - \mathbf{g}_{\text{best}}|\right), & \text{rand}_i(0,1) < SA_i \\ \text{DE/best/1}, & \text{其他} \end{cases} \quad (10)$$

分析发现, 在寻优初期, 个体适应值与全局最优值相差较大, 故  $SA_i$  较大, 种群中绝大部分个体执行高斯变异策略, 对解空间进行探索; 随着进化的推进, 个体不同程度地趋近最优个体, 此时根据个体适应值与最优适应值的比较, 为每一个体指定当前最适合的变异策略; 在进化后期, 大部分个体收敛于最优个体附近, 此时  $SA_i$  值较小, 转为执行 DE/best/1 策略, 算法聚焦于对最优值附近的开发。SMGBDE 算法充分利用了个体的进化信息, 使进化更具趋利性。此外, SMGBDE 算法继承了 BBDE 算法家族近似无参数的优点, 新引进的控制参数  $SA$  随进化过程自适应调节, 无需人工干预和任何先验知识。

### 3.3 停滞扰动策略

为解决 DE/best/1 策略潜在的早熟收敛问题, 需对种群进化状态进行监视。若在进化过程中某一次迭代结束时, 种群大部分个体未得到更新, 表明种群潜在的多样性减小, 算法可能陷入局部最优, 失去对解空间进一步探索的能力。因此, 本文加入停滞扰动策略, 通过观察每次种群更新比例  $r$ , 若其小于设定阈值  $\tau$ , 则调节  $SA_i$  的大小, 改变下一次迭代个体的进化方向, 尽最大努力跳出该局部极值区域。

种群更新比例  $r$  与  $SA_i$  的调节关系为

$$SA_{i,g+1} = \begin{cases} 0.5, & r \leq \tau \\ SA_{i,g}, & \text{其他} \end{cases} \quad (11)$$

其中, 若种群更新比例  $r \leq \tau$ , 则下一次迭代时, 所有个体以均等概率执行高斯变异策略和 DE/best/1 策略, 增强对个体的扰动性, 使其能扩大寻优范围, 跳出当前搜索区域, 否则, 维持  $SA_i$  大小不变。关于  $\tau$  的取值, 若  $\tau$  过大, 会影响种群在进化初期向利于进化的趋势演变, 而  $\tau$  取值过小, 则会错过跳出局部极值的较佳时机。基于大量实验, 本文发现  $\tau$  取值为 0.2 能对单峰问题和多峰问题均表现出良好的寻优性能, 本文实验结果也印证了该取值的合理性。

### 3.4 算法描述

算法 1 为 SMGBDE 算法的伪代码。它和 MGBDE 算法的框架基本类似, 主要有 3 处不同: 在步骤 2) 和步骤 18) 加入了变异策略选择因子的计算; 在步骤 20) 加入了对进化状态的监视, 观察本次迭代种群个体更新情况; 在步骤 21) 加入了对  $SA$  的更新判断, 确定是否需要执行停滞扰动。

#### 算法 1 SMGBDE 算法伪代码

输入 种群规模  $NP$

输出 全局最优解  $\mathbf{x}_{best,g}$

- 1) 初始化种群, 计算每个个体适应值, 迭代次数  $g = 0$ ;
- 2) 根据式(9)计算每个  $SA_i$ ;
- 3) 不满足停止条件
- 4) for  $i = 1$  to  $NP$  do
- 5) 根据式(10)生成变异向量  $\mathbf{v}_{i,g}$ ;
- 6) 根据式(6)生成实验向量  $\mathbf{u}_{i,g}$ ;
- 7) 计算实验向量  $\mathbf{u}_{i,g}$  的适应值;
- 8) if  $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})$  then
- 9)  $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$ ;
- 10) if  $f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{best,g})$  then
- 11)  $\mathbf{x}_{best,g} = \mathbf{u}_{i,g}$ ;
- 12) end
- 13) end
- 14) else
- 15)  $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$ ;
- 16) end

$$17) \quad CR_{i,g+1} = \begin{cases} CR_{i,g}, & f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g}) \\ N(0.5, 0.1), & \text{其他} \end{cases};$$

- 18) 根据式(9)计算每个  $SA_{i,g}$ ;
- 19) end
- 20) 计算本次迭代种群更新比例  $r$ ;
- 21) 根据式(11)更新  $SA_{i,g+1}$ ;
- 22) 进入下一轮迭代,  $g = g + 1$ ;
- 23) end

### 3.5 算法复杂度分析

根据算法 1 分析 SMGBDE 算法迭代一次的时间复杂度。已知标准 DE 算法迭代一次的时间复杂度为  $O(ND)$ , 其中,  $N$  为种群规模,  $D$  为问题维数。SMGBDE 算法与标准 DE 算法相比: 步骤 2) 和步骤 21) 加入变异策略选择因子的计算, 时间复杂度为  $O(ND)$ ; 步骤 5) 采用双变异策略, 但本质上仍是对个体的每一维向量执行一种变异操作, 复杂度为  $O(D)$ , 接下来的交叉、选择操作与标准 DE 算法相同,  $CR$  的更新为标量计算, 故步骤 4) 到步骤 19) 的总体复杂度为  $O(ND)$ ; 步骤 20) 的复杂度为  $O(D)$ 。综上, 省略低阶项后, SMGBDE 算法迭代一次的时间复杂度为  $O(ND)$ , 与标准 DE 算法相同, 可见对算法的改进没有带来过多的计算开销。

### 3.6 算法收敛性分析

文献[16]利用随机泛函理论证明了标准 DE 算法是渐近收敛的, 文献[15]采用同样的方法分析了基于三角高斯变异策略的 tBBDE 算法的渐近收敛性。分析发现, SMGBDE 算法较 tBBDE 算法有基于高斯抽样生成变异个体, 但也会以一定概率执行标准 DE 算法中 DE/best/1 策略, 而后续交叉和选择操作与标准 DE 算法相同。基于此, 本文也利用随机泛函理论分析 SMGBDE 算法的渐近收敛性。

令  $f(\mathbf{x})$  为最小优化问题, 其中,  $\mathbf{x} = (x_1, x_2, \dots, x_d)$  为  $d$  维实数编码向量,  $L_i \leq x_i \leq U_i$ , 则解空间  $S = \prod_{i=1}^d [L_i, U_i]$  是由  $d$  维实向量全体构成的集合。 $(\Omega, A, P)$  为完全概率测度空间,  $\Omega$  为非空抽象集合, 其元素  $\omega$  为基本事件;  $A$  是  $\Omega$  的某些子集所构成的  $\sigma$ -代数;  $P$  是  $A$  上的概率测度。

在每次迭代中, SMGBDE 算法依次进行变异、交叉和选择操作, 依据随机泛函理论可将一次迭代

抽象为上述 3 个操作算子合成的随机映射。

**定义 1** 变异算子  $\Psi_m$  是对个体向量按高斯分布  $N(\mu, \delta^2)$  抽样的过程。它是解空间上的一种随机映射： $\Psi_m : \Omega \times S \rightarrow S$ ，定义为

$$\Psi_m(\omega, \mathbf{x}) = \mathbf{v} \sim N(\mu, \delta^2) \quad (12)$$

其中， $\mu$  为  $\mathbf{x}_{\text{best},g}$  与  $\mathbf{x}_{\text{best},g}$  偏置的加权和值， $\delta$  为  $\mathbf{x}$  与  $\mathbf{x}_{\text{best},g}$  差的绝对值。

**定义 2** 交叉算子  $\Psi_c$  是标准 DE 算法交叉算子。它是解空间上的一种随机映射： $\Psi_c : \Omega \times S \rightarrow S^2$ ，根据文献[16]可定义为

$$P\{\omega | \Psi_c(\omega, \mathbf{x}) = \langle \mathbf{x}, \mathbf{v} \rangle\} = 1 - \left(1 - CR - \frac{1}{D}\right)^D \quad (13)$$

其中， $CR$  为交叉率。

**定义 3** 选择算子  $\Psi_s$  是标准 DE 算法选择算子。它是解空间上的映射： $\Psi_s : S^2 \rightarrow S$ ，对任意 2 个体  $\mathbf{x}$  和  $\mathbf{y}$ ，在  $\Psi_s$  的作用下

$$\Psi_s(\langle \mathbf{x}, \mathbf{y} \rangle) = \min\{f(\mathbf{x}), f(\mathbf{y})\} \quad (14)$$

设  $H(t)$  表示第  $t$  次迭代产生的种群，每迭代一次等价于进行一次映射，即

$$H(t+1) = \Psi(\omega, H(t)) = \Psi_s(\Psi_c(\Psi_m(\omega, H(t)))) \quad (15)$$

其中， $t = 0, 1, \dots, T-1$ ， $T$  为算法最大迭代次数。

**定理 1** SMGBDE 算法一次迭代所形成的随机映射  $\Psi$  是一个随机压缩算子。

**证明** 根据定义 1~定义 3 可知，SMGBDE 算法每次迭代产生的新种群总是不劣于其父代。根据文献[16]的方法容易证得定理 1 成立。

**定理 2** 设  $\Psi$  为 SMGBDE 算法形成的随机压缩算子，则  $\Psi$  具有唯一的随机不动点，即 SMGBDE 算法是渐进收敛的。

**证明** 根据定理 1，再按照文献[16]的方法，容易得证。

## 4 实验结果与分析

### 4.1 测试函数与参数设置

本文将 SMGBDE 算法与 BBDE 算法家族中的其他算法以及知名的 DE 改进算法，包括自适应算法 jDE<sup>[4]</sup>和基于反向搜索策略的 ODE 算法<sup>[17]</sup>在基准测试函数上进行性能对比实验。如表 1 所示，测试函数分为 2 组：组 1 为 13 个低维( $D = 30$ )、具

表 1 基准测试函数

类别	记号	测试函数	维数	搜索区间	最优解
单调函数	$f_1$	Sphere	30	[-100, +100]	0
	$f_2$	Schwefel 1.2	30	[-100, +100]	0
	$f_3$	Schwefel 2.21	30	[-100, +100]	0
	$f_4$	Schwefel 2.22	30	[-10, +10]	0
	$f_5$	Rosenbrock	30	[-30, +30]	0
	$f_6$	Step	30	[-100, +100]	0
	$f_7$	Quartic with noise	30	[-1.28, +1.28]	0
多峰函数	$f_8$	Schwefel 2.26	30	[-500, +500]	-12 569.5
	$f_9$	Rastrigin	30	[-5.12, +5.12]	0
	$f_{10}$	Ackley	30	[-32, +32]	0
	$f_{11}$	Generalized Griewank Funcion	30	[-600, +600]	0
	$f_{12}$	Generalized Penalized Funcion 1	30	[-50, +50]	0
	$f_{13}$	Generalized Penalized Funcion 2	30	[-50, +50]	0
高级带偏移函数	$f_{14}$	Shifted Sphere Function	50	[-100, +100]	-450
	$f_{15}$	Shifted Schwefel's Problem 1.2	50	[-100, +100]	-450
	$f_{16}$	Shifted Rosenbrock's Funcion	50	[-100, +100]	390
	$f_{17}$	Shifted Rotated Ackley's Funciton	50	[-32, +32]	-140
	$f_{18}$	Shifted Rastrigin's Funcion	50	[-5, +5]	-330

有多种特性的测试函数<sup>[18]</sup>, 其中,  $f_1 \sim f_7$  为单峰函数,  $f_8 \sim f_{13}$  为多峰函数, 其局部最优解个数会随问题维数的增加呈指数增长。组 2 为 5 个高维 ( $D = 50$ ) 带偏移的测试函数<sup>[19]</sup>。各函数均为最小优化问题, 数学表达式详见附录 A。为保证比较公平性, 所有算法的共同参数设置相同, 如表 2 所示, 其他参数取各算法的推荐值。各算法均独立运行 30 次。所有实验在硬件配置为 Intel Xeon(R) CPU E5-2640 2.5 GHz、32 GB 内存、64 bit Windows 7 操作系统的计算机上

运行, 程序采用 Matlab R2015b 编写。

### 4.2 收敛精度比较

表 3 给出了各算法独立运行 30 次寻到的最优解平均值 (每个函数的第一行) 及标准差 (每个函数的第二行), 各测试函数上的最优结果以黑体加粗标注, 评判准则是平均值越小越好, 在平均值相同的情况下, 选择标准差较小的一个。w/t/l 表示在 Wilcoxon 符号秩检验下 SMGBDE 算法优于、相当于和劣于相对应算法的次数, 其中, 显著性水平  $\alpha$  为 0.05。从

表 2 算法参数设置

种群规模 $PS$	缩放因子 $F$	交叉率 $CR$	最大函数评价次数 $F_{es}(D=30)$	最大函数评价次数 $F_{es}(D=50)$
100	0.5	0.9	200 000	500 000

表 3 各算法独立运行 30 次的最优解平均值和标准差及 Wilcoxon 符号秩检验结果

测试函数	维数	最优解平均值 (标准差)						SMGBDE
		BBDE	GBDE	MGBDE	tBBDE	jDE	ODE	
$f_1$	30	9.74×10 <sup>-91</sup>	1.41×10 <sup>-32</sup>	3.33×10 <sup>-68</sup>	1.79×10 <sup>-67</sup>	3.16×10 <sup>-22</sup>	1.70×10 <sup>-46</sup>	<b>1.36×10<sup>-97</sup></b>
		1.06×10 <sup>-90</sup>	3.28×10 <sup>-32</sup>	9.84×10 <sup>-68</sup>	5.62×10 <sup>-67</sup>	2.19×10 <sup>-22</sup>	6.76×10 <sup>-46</sup>	<b>4.37×10<sup>-97</sup></b>
$f_2$	30	<b>3.24×10<sup>-5</sup></b>	7.27×10 <sup>2</sup>	3.12	6.52	1.10×10 <sup>2</sup>	6.64×10 <sup>-1</sup>	3.28
		<b>3.39×10<sup>-5</sup></b>	5.79×10 <sup>2</sup>	2.91	5.33	4.29×10 <sup>1</sup>	1.36	2.76
$f_3$	30	2.53×10 <sup>-8</sup>	9.41×10 <sup>-4</sup>	9.65×10 <sup>-4</sup>	1.55×10 <sup>-8</sup>	7.05×10 <sup>-3</sup>	<b>1.49×10<sup>-25</sup></b>	6.66×10 <sup>-1</sup>
		1.30×10 <sup>-7</sup>	4.83×10 <sup>-4</sup>	2.48×10 <sup>-3</sup>	3.05×10 <sup>-8</sup>	2.27×10 <sup>-3</sup>	<b>8.14×10<sup>-25</sup></b>	3.71×10 <sup>-1</sup>
$f_4$	30	9.02×10 <sup>-45</sup>	1.20×10 <sup>-22</sup>	9.41×10 <sup>-38</sup>	1.81×10 <sup>-44</sup>	1.70×10 <sup>-13</sup>	2.87×10 <sup>-19</sup>	<b>9.92×10<sup>-54</sup></b>
		8.25×10 <sup>-45</sup>	1.19×10 <sup>-22</sup>	7.26×10 <sup>-38</sup>	2.36×10 <sup>-44</sup>	7.96×10 <sup>-14</sup>	1.20×10 <sup>-18</sup>	<b>1.95×10<sup>-53</sup></b>
$f_5$	30	1.68×10 <sup>1</sup>	5.02×10 <sup>1</sup>	1.86×10 <sup>1</sup>	5.80×10 <sup>1</sup>	1.80×10 <sup>1</sup>	1.75×10 <sup>1</sup>	<b>1.62×10<sup>1</sup></b>
		1.66×10 <sup>1</sup>	3.82×10 <sup>1</sup>	1.78×10 <sup>1</sup>	6.53×10 <sup>1</sup>	6.18×10 <sup>-1</sup>	3.50	<b>1.62×10<sup>1</sup></b>
$f_6$	30	<b>0</b>	<b>0</b>	6.67×10 <sup>-2</sup>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
		<b>0</b>	<b>0</b>	3.65×10 <sup>-1</sup>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
$f_7$	30	<b>2.22×10<sup>-3</sup></b>	8.39×10 <sup>-3</sup>	3.20×10 <sup>-3</sup>	3.11×10 <sup>-3</sup>	8.89×10 <sup>-3</sup>	6.61	3.27×10 <sup>-3</sup>
		<b>6.72×10<sup>-4</sup></b>	2.65×10 <sup>-3</sup>	1.44×10 <sup>-3</sup>	8.87×10 <sup>-4</sup>	2.22×10 <sup>-3</sup>	2.51	1.01×10 <sup>-3</sup>
$f_8$	30	-1.16×10 <sup>4</sup>	-1.24×10 <sup>4</sup>	-1.19×10 <sup>4</sup>	-1.17×10 <sup>4</sup>	<b>-1.26×10<sup>4</sup></b>	-1.11×10 <sup>4</sup>	-1.21×10 <sup>4</sup>
		3.31×10 <sup>2</sup>	8.07×10 <sup>1</sup>	2.68×10 <sup>2</sup>	2.93×10 <sup>2</sup>	<b>1.85×10<sup>-12</sup></b>	3.04×10 <sup>3</sup>	2.24×10 <sup>2</sup>
$f_9$	30	1.29×10 <sup>2</sup>	4.21	1.42×10 <sup>1</sup>	2.04×10 <sup>1</sup>	2.51×10 <sup>1</sup>	<b>2.34</b>	1.20×10 <sup>1</sup>
		2.32×10 <sup>1</sup>	2.17	4.14	5.31	2.81	<b>1.27×10<sup>1</sup></b>	3.37
$f_{10}$	30	5.63×10 <sup>-15</sup>	1.64×10 <sup>-14</sup>	1.49×10 <sup>-14</sup>	6.69×10 <sup>-15</sup>	4.80×10 <sup>-12</sup>	<b>1.72×10<sup>-15</sup></b>	2.29×10 <sup>-14</sup>
		1.35×10 <sup>-15</sup>	4.91×10 <sup>-15</sup>	3.45×10 <sup>-15</sup>	1.80×10 <sup>-15</sup>	1.88×10 <sup>-12</sup>	<b>1.60×10<sup>-15</sup></b>	7.47×10 <sup>-15</sup>
$f_{11}$	30	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
		<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
$f_{12}$	30	3.46×10 <sup>-3</sup>	9.60×10 <sup>-32</sup>	1.04×10 <sup>-2</sup>	6.91×10 <sup>-3</sup>	2.88×10 <sup>-23</sup>	2.39×10 <sup>-31</sup>	<b>1.67×10<sup>-32</sup></b>
		1.89×10 <sup>-2</sup>	3.90×10 <sup>-31</sup>	3.16×10 <sup>-2</sup>	2.63×10 <sup>-2</sup>	1.84×10 <sup>-23</sup>	9.21×10 <sup>-31</sup>	<b>4.16×10<sup>-33</sup></b>
$f_{13}$	30	3.66×10 <sup>-4</sup>	8.91×10 <sup>-32</sup>	3.66×10 <sup>-4</sup>	1.46×10 <sup>-3</sup>	3.30×10 <sup>-22</sup>	3.62×10 <sup>-31</sup>	<b>5.16×10<sup>-32</sup></b>
		2.01×10 <sup>-3</sup>	1.49×10 <sup>-31</sup>	2.01×10 <sup>-3</sup>	3.80×10 <sup>-3</sup>	3.50×10 <sup>-22</sup>	1.85×10 <sup>-30</sup>	<b>1.86×10<sup>-31</sup></b>
$f_{14}$	50	<b>-4.50×10<sup>2</sup></b>	-4.50×10 <sup>2</sup>	-4.50×10 <sup>2</sup>	<b>-4.50×10<sup>2</sup></b>	-4.50×10 <sup>2</sup>	-4.50×10 <sup>2</sup>	-4.50×10 <sup>2</sup>
		<b>0</b>	1.19×10 <sup>-13</sup>	1.15×10 <sup>-13</sup>	<b>0</b>	2.59×10 <sup>-14</sup>	2.59×10 <sup>-14</sup>	2.63×10 <sup>-13</sup>
$f_{15}$	50	<b>-4.50×10<sup>2</sup></b>	1.30×10 <sup>4</sup>	-2.32×10 <sup>2</sup>	1.67×10 <sup>3</sup>	1.37×10 <sup>2</sup>	-1.21×10 <sup>2</sup>	-1.77×10 <sup>2</sup>
		<b>1.22×10<sup>-1</sup></b>	9.44×10 <sup>3</sup>	2.10×10 <sup>2</sup>	1.14×10 <sup>3</sup>	2.80×10 <sup>2</sup>	1.58×10 <sup>2</sup>	2.09×10 <sup>2</sup>
$f_{16}$	50	4.25×10 <sup>2</sup>	5.89×10 <sup>2</sup>	<b>4.20×10<sup>2</sup></b>	6.64×10 <sup>3</sup>	4.35×10 <sup>2</sup>	4.36×10 <sup>2</sup>	4.43×10 <sup>2</sup>
		1.51×10 <sup>1</sup>	2.49×10 <sup>2</sup>	<b>3.86×10<sup>1</sup></b>	1.46×10 <sup>4</sup>	2.85×10 <sup>1</sup>	2.44×10 <sup>1</sup>	1.15×10 <sup>2</sup>
$f_{17}$	50	<b>-1.19×10<sup>2</sup></b>	-1.19×10 <sup>2</sup>	-1.19×10 <sup>2</sup>	-1.19×10 <sup>2</sup>	-1.19×10 <sup>2</sup>	-1.19×10 <sup>2</sup>	-1.19×10 <sup>2</sup>
		<b>2.80×10<sup>-2</sup></b>	3.25	3.80	4.09	3.23×10 <sup>-2</sup>	5.88×10 <sup>-2</sup>	3.87×10 <sup>-2</sup>
$f_{18}$	50	-9.38×10 <sup>1</sup>	<b>-3.12×10<sup>2</sup></b>	-2.86×10 <sup>2</sup>	-2.90×10 <sup>2</sup>	-2.81×10 <sup>2</sup>	-2.93×10 <sup>1</sup>	-2.87×10 <sup>2</sup>
		8.70×10 <sup>1</sup>	<b>4.81</b>	1.13×10 <sup>1</sup>	8.74	4.66	3.39×10 <sup>1</sup>	9.88
w/t/l		8/4/6	9/4/5	9/3/6	10/4/4	11/4/3	9/4/5	—

Wilcoxon 符号秩检验的结果来看, SMGBDE 算法整体优于其他算法。

对于单峰问题 ( $f_1 \sim f_7$ ), SMGBDE 算法除了对  $f_3$  寻优不够理想外, 其他整体表现较好, 其中, 在  $f_1$ 、 $f_4$  和  $f_5$  上优势明显, 在  $f_2$  上仅劣于 BBDE 算法和 ODE 算法, 在  $f_7$  上劣于 BBDE 算法和 tBBDE 算法, 其他函数上均不劣于其他算法。BBDE 算法由于将全局最优和个体最优信息引入变异策略, 体现了贪婪机制, 有利于单峰问题的寻优。对于阶跃函数  $f_6$ , 除 MGBDE 算法外, 其他算法每次均能寻找最优解, MGBDE 算法在某几次运行中, 在寻到最优解附近时出现停滞。这是因为在间断附近区域内的所有点适应值相同, 需跳出该区域寻找更优解。这反映了 MGBDE 算法在后期搜索能力大幅减弱, 而 SMGBDE 算法改进了变异策略的选择并加入停滞扰动机制, 较好地解决这一问题。

对于多峰问题  $f_8 \sim f_{13}$ , SMGBDE 算法表现也较为良好, 仅在  $f_8$  上较弱于 GBDE 算法和 jDE 算法, 在  $f_9$  上较弱于 GBDE 算法和 ODE 算法, 在  $f_{10}$  上弱于 BBDE 算法、tBBDE 算法和 ODE 算法。在  $f_{12}$  和  $f_{13}$  上, SMGBDE 算法明显优于 BBDE 算法、MGBDE 算法和 tBBDE 算法, 这是因为这 3 种算法在寻优过程中均陷入了局部最优。虽然 tBBDE 算法也加入了停滞扰动策略, 但其在种群更新比例为 0 时再随机选择若干个体进行初始化以对种群扰动, 而 SMGBDE 算法是在种群尚未完全失去多样性时, 通过调节 SA, 改变所有个体的进化方向, 扰动效果优于 tBBDE 算法。

对于高维偏移问题  $f_{14} \sim f_{18}$ , SMGBDE 算法虽未有独特优势, 除在  $f_{15}$  上明显弱于 BBDE 算法, 对于其他函数, 与最优算法和次优算法差距不大。具体来看, 在  $f_{16}$  上 SMGBDE 算法明显优于 GBDE 算法和 tBBDE 算法, 在  $f_{18}$  上 SMGBDE 算法对 BBDE 算法和 ODE 算法占绝对优势, 且优于 MGBDE 算法和 jDE 算法。结合 SMGBDE 算法在低维问题上的表现, 可以看出 SMGBDE 算法具有较好的顽健性, 适用于低维问题和高维问题求解。

为进一步直观得出各算法的优劣, 采用 Friedman 检验对算法的寻优结果进行排名。Friedman 检验是利用秩分析多个总体分布是否存在显著差异的非参数检验方法, 在检验时以行为单位将数据按升序排序, 得出各变量值在各行的秩, 然后计算各样本的秩总和与秩均值。在本实验中秩

均值越小越好, 检验结果如表 4 所示。

表 4 各算法的 Friedman 检验排名

算法	秩均值
BBDE	3.39
GBDE	4.19
MGBDE	4.08
tBBDE	4.42
jDE	4.81
ODE	3.86
SMGBDE	3.25

综合以上实验结果可以发现, SMGBDE 算法通过引入变异策略选择因子 SA, 为个体指定最为适合的变异策略, 使种群向有利于进化的方向演变。同时, 停滞扰动策略的设计使算法能较好地避免陷入局部最优。结合对实验结果的统计学分析, 认为 SMGBDE 算法的寻优能力要优于现有的 BBDE 算法家族以及知名的自适应 DE 算法 jDE 和反向 DE 算法 ODE。

### 4.3 收敛速度比较

图 1 描绘了各算法的收敛曲线, 所用数据为各算法独立运行 30 次在 12 个固定观测点适应值的平均值。观测点定义为当前函数评价次数  $F_{ES}$ ,  $F_{ES}$  的取值情况如表 5 所示。由于 ODE 算法在每次迭代时会以一定概率执行 jump 操作,  $F_{ES}$  的取值难以精准定位。为确保比较公平性, 未将 ODE 算法作为比较对象。本文从单峰函数、多峰函数和高维偏移函数中各选择了 2 个有代表性的函数的收敛曲线进行比较分析, 其他各函数的收敛曲线见附录 B。

表 5 12 个观测点的  $F_{ES}$  取值情况

观测点	$F_{ES}(D=30)$	$F_{ES}(D=50)$
1	$1.00 \times 10^2$	$1.00 \times 10^2$
2	$1.00 \times 10^3$	$1.00 \times 10^3$
3	$2.00 \times 10^3$	$5.00 \times 10^3$
4	$4.00 \times 10^3$	$1.00 \times 10^4$
5	$8.00 \times 10^3$	$2.00 \times 10^4$
6	$1.60 \times 10^4$	$4.00 \times 10^4$
7	$2.00 \times 10^4$	$8.00 \times 10^4$
8	$4.00 \times 10^4$	$1.60 \times 10^5$
9	$8.00 \times 10^4$	$2.40 \times 10^5$
10	$1.20 \times 10^5$	$3.20 \times 10^5$
11	$1.60 \times 10^5$	$4.00 \times 10^5$
12	$2.00 \times 10^5$	$5.00 \times 10^5$

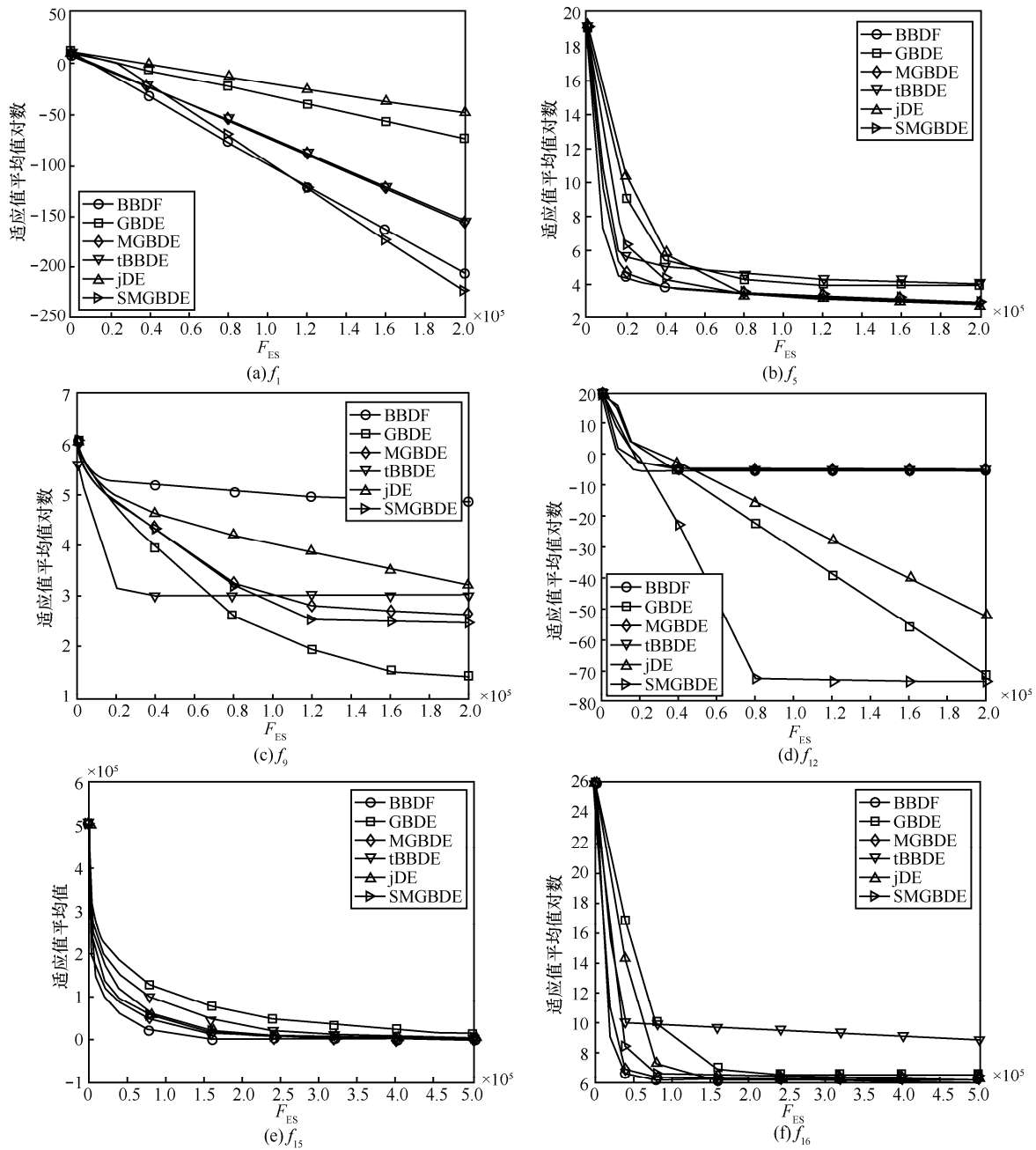


图 1 各算法在函数  $f_1$ 、 $f_5$ 、 $f_6$ 、 $f_{12}$ 、 $f_{15}$  和  $f_{16}$  上的收敛曲线

从图 1 可以看出, SMGBDE 算法具有较强的收敛能力。对于部分函数, 虽在搜索初期收敛效果弱于某些算法, 但随着进化过程的推进, SMGBDE 算法能够追上甚至超过大多数算法的收敛速度, 并在进化结束时, 收敛效果最佳或近似最佳, 如函数  $f_1$ 、 $f_5$  和  $f_{16}$ 。值得注意的是, 在  $f_{12}$  上, SMGBDE 算法能以最快速度收敛于最优值, 与所有被比较算法相比, 有绝对优势。由此可见, SMGBDE 算法在进化前期注重对解空间的探索, 发现最优解可能产

生的区域, 因而收敛速度较慢。在进化后期聚焦于优质解邻域的开发, 在 DE/best/1 策略的作用下, 加快了收敛速度。

### 5 结束语

本文针对经典的基于高斯变异的骨架差分进化算法 MGBDE 的不足, 提出一种改进算法。新算法能够根据个体的进化信息自适应地选择更合适的变异策略, 改善种群进化的趋利性; 加入停滞扰

动策略，降低了出现早熟收敛和停滞搜索的风险；采用随机泛函理论分析了算法的渐近收敛性。实验结果表明，本文提出的改进算法在收敛精度、收敛速度和顽健性上整体优于现有 BBDE 家族算法和一些知名的 DE 算法。

### 附录 A 基准测试函数表达式

SMGBDE 算法性能测试所用基准测试函数表达式如下。

1) Sphere

$$f_1(\mathbf{x}) = \sum_{i=1}^D x_i^2$$

2) Schwefel 1.2

$$f_2(\mathbf{x}) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)$$

3) Schwefel 2.21

$$f_3(\mathbf{x}) = \max_i \{|x_i|, 1 \leq i \leq D\}$$

4) Schwefel 2.22

$$f_4(\mathbf{x}) = \sum_{i=1}^D |x_i| + \prod_{i=1}^D |x_i|$$

5) Rosenbrock

$$f_5(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

6) Step

$$f_6(\mathbf{x}) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$$

7) Quartic with noise

$$f_7(\mathbf{x}) = \sum_{i=1}^D ix_i^4 + \text{random}[0,1]$$

其中，random 为随机数生成器。

8) Schwefel 2.26

$$f_8(\mathbf{x}) = \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|})$$

9) Rastrigin

$$f_9(\mathbf{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

10) Ackley

$$f_{10}(\mathbf{x}) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i \right) + 20 + e$$

11) Generalized Griewank Function

$$f_{11}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$$

12) 和 13) Generalized Penalized Function

$$f_{12}(\mathbf{x}) = \frac{\pi}{D} \left\{ \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin(\pi y_{i+1})] + (y_D - 1)^2 + 10 \sin^2(\pi y_1) \right\} +$$

$$\sum_{i=1}^D u(x_i, 10, 100, 4)$$

$$f_{13}(\mathbf{x}) = \frac{1}{10} \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \right.$$

$$\left. (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$$

其中，

$$y_i = 1 + \frac{1}{4}(x_i + 1)$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

14) Shifted Sphere Function

$$f_{14}(\mathbf{x}) = \sum_{i=1}^D z_i^2 + f_{-bias_1}, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$$

15) Shifted Schwefel's Problem 1.2

$$f_{15}(\mathbf{x}) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_{-bias_2}, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$$

16) Shifted Rosenbrock's Function

$$f_{16}(\mathbf{x}) = \sum_{i=1}^D \left( 100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right) + f_{-bias_6}, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$$

17) Shifted Rotated Ackley's Function

$$f_{17}(\mathbf{x}) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i) \right) + 20 + e + f_{-bias_8},$$

$$\mathbf{z} = (\mathbf{x} - \mathbf{o}) \mathbf{M}$$

18) Shifted Rastrigin's Function

$$f_{18}(\mathbf{x}) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{-bias_9}, \quad \mathbf{z} = \mathbf{x} - \mathbf{o}$$

### 附录 B 各算法收敛曲线比较

BBDE 算法、GBDE 算法、MGBDE 算法、tBBDE 算法、jDE 算法和 SMGBDE 算法在函数  $f_2 \sim f_4$ 、 $f_6 \sim f_8$ 、 $f_{10}$ 、 $f_{11}$ 、 $f_{13}$ 、 $f_{14}$ 、 $f_{17}$ 、 $f_{18}$  上的收敛曲线如图 2 所示。

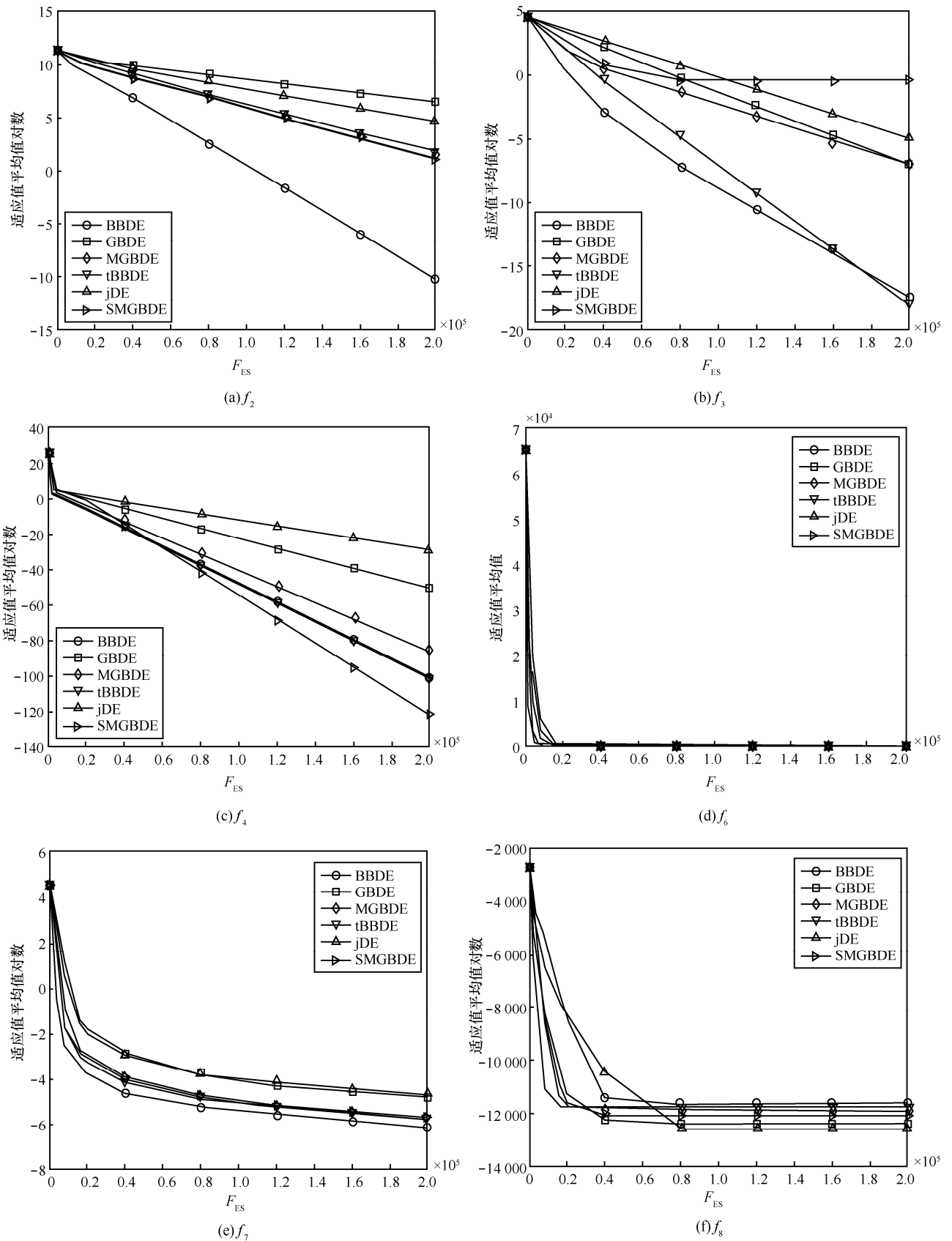


图 2 各算法在函数  $f_2 \sim f_4$ 、 $f_6 \sim f_8$ 、 $f_{10}$ 、 $f_{11}$ 、 $f_{13}$ 、 $f_{14}$ 、 $f_{17}$ 、 $f_{18}$  上的收敛曲线

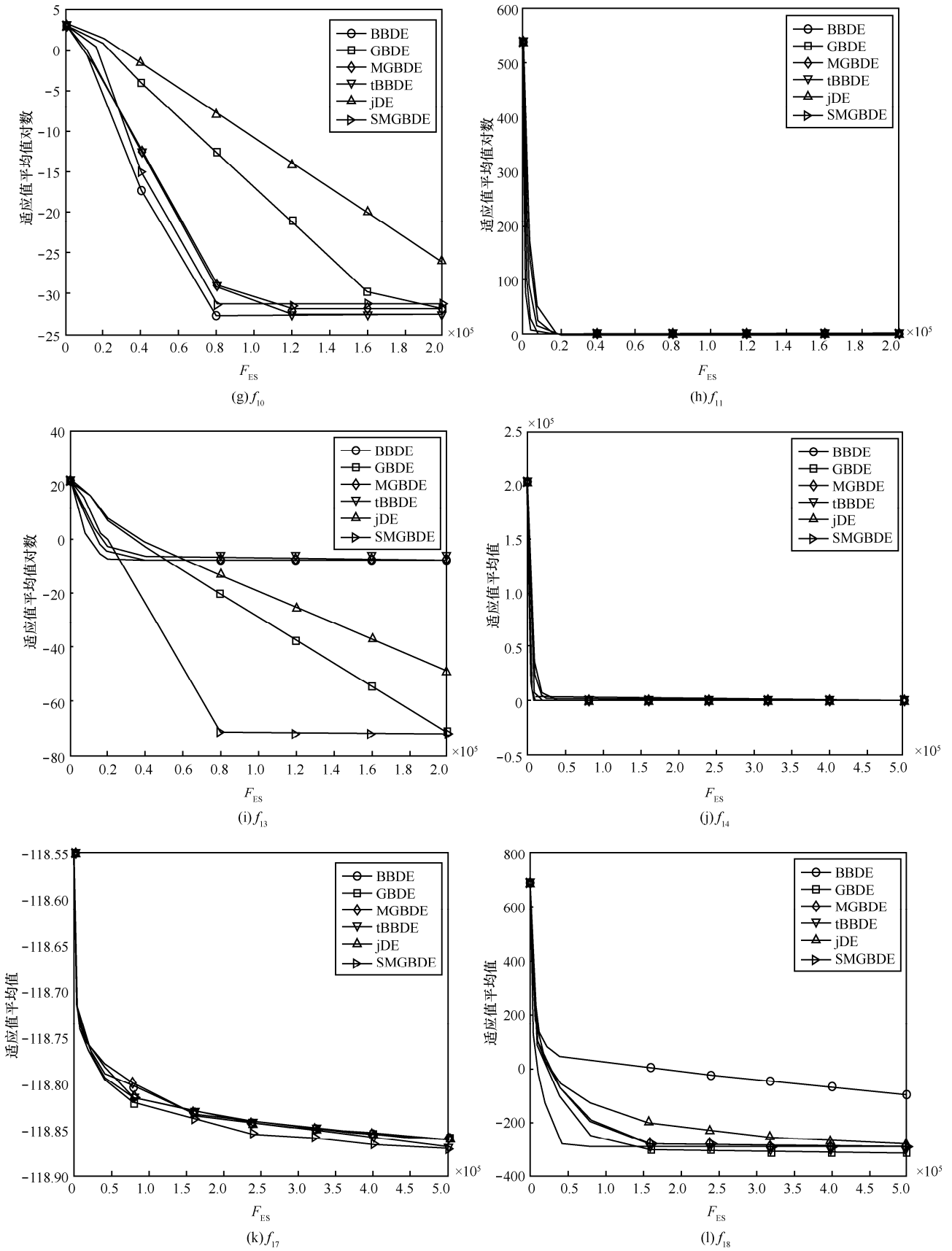
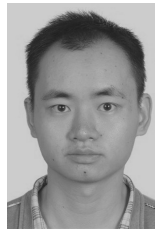


图 2 各算法在函数  $f_2 \sim f_4$ 、 $f_6 \sim f_8$ 、 $f_{10}$ 、 $f_{11}$ 、 $f_{13}$ 、 $f_{14}$ 、 $f_{17}$ 、 $f_{18}$  上的收敛曲线 (续)

## 参考文献:

- [1] STORN R, PRICE K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces[M]. Berkeley: ICSI, 1995.
- [2] DAS S, SUGANTHAN P N. Differential evolution: a survey of the state-of-the-art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.
- [3] PRICE K, STORN R, LAMPINEN J A. Differential evolution a practical approach to global optimization[M]. Germany: Springer-Verlag Berlin Heidelberg, 2005.
- [4] JANEZ B, GREINER S 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.
- [5] QIN A K, HUANG V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398-417.
- [6] ZHANG J Q, SANDERSON A. JADE: adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.
- [7] GUO H X, LI Y N, LI J L, et al. Differential evolution improved with self-adaptive control parameters based on simulated annealing[J]. Swarm and Evolutionary Computation, 2014, 19(1): 52-67.
- [8] YI W C, ZHOU Y Z, GAO L, et al. An improved adaptive differential evolution algorithm for continuous optimization[J]. Expert Systems with Applications, 2016, 44(1): 1-12.
- [9] TRIVEDI A, SRINIVASAN D, BISWAS S, et al. Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem[J]. Swarm and Evolutionary Computation, 2015, 23(1): 50-64.
- [10] EZGI Z, ZEL S A. A hybrid approach of differential evolution and artificial bee colony for feature selection[J]. Expert Systems with Applications, 2016, 62(15): 91-103.
- [11] 孟庆昕, 杨士莪, 于盛齐. 基于波形结构特征和支持向量机的水面目标识别[J]. 电子与信息学报, 2015, 37(9): 2117-2123.  
MENG Q X, YANG S E, YU S Q. Recognition of marine acoustic target signals based on wave structure and support vector machine[J]. Journal of Electronics & Information Technology, 2015, 37(9): 2117-2123.
- [12] OMRAN M G H, ENGELBRECHT A P, SALMAN A. Bare bones differential evolution[J]. European Journal of Operational Research, 2009, 196(1): 128-139.
- [13] KENNEDY J. Bare bones particle swarm[C]//The 2003 IEEE Swarm Intelligence Symposium. 2003: 80-87.
- [14] WANG H, RAHNAMEYAN S, SUN H, et al. Gaussian bare-bones differential evolution[J]. IEEE Transactions on Cybernetics, 2013, 43(2): 634-647.
- [15] 彭虎, 吴志健, 周新宇, 等. 基于三角的骨架差分进化算法[J]. 计算机研究与发展, 2015, 52(12): 2776-2788.  
PENG H, WU Z J, ZHOU X Y, et al. Bare-bones differential evolution algorithm based on trigonometry[J]. Journal of Computer Research and Development, 2015, 52(12): 2776-2788.
- [16] 贺毅朝, 王熙照, 刘坤起, 等. 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.  
HE Y C, WANG X Z, LIU K Q, et al. Convergent analysis and algorithmic improvement of differential evolution[J]. Journal of Software, 2010, 21(5): 875-885.
- [17] RAHNAMEYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64-79.
- [18] YAO X, LIU Y, LIN G M. Evolutionary programming made faster[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82-102.
- [19] 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]. Singapore: Nanyang Technological University, 2005.

## 作者简介:



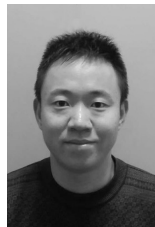
刘会宇 (1987-), 男, 湖北咸宁人, 解放军信息工程大学硕士生, 主要研究方向为人工智能。



韩继红 (1966-), 女, 山西定襄人, 博士, 解放军信息工程大学教授、博士生导师, 主要研究方向为网络与信息安全、安全协议形式化分析与自动化验证。



袁霖 (1981-), 男, 河南商丘人, 博士, 解放军信息工程大学副教授, 主要研究方向为安全协议形式化分析与自动化验证、软件可信性分析。



于波 (1991-), 男, 重庆人, 解放军信息工程大学硕士生, 主要研究方向为云计算和数据中心。