

# 差分进化算法综述

杨启文 蔡 亮 薛云灿

(河海大学常州校区 计算机与信息工程学院 常州 213022)

**摘 要** 差分进化算法是一类基于种群的启发式全局搜索技术,对于实值参数的优化具有较强的鲁棒性.为了提高差分进化算法的寻优速度、克服启发式算法常见的早熟收敛问题,许多学者对差分进化算法进行改进.本文综述差分进化的基本形式及其多种改进形式,讨论它们的优缺点,指出下一步的改进方向.

**关键词** 差分进化,启发式优化,遗传算法

**中图分类号** TP 181

## A Survey of Differential Evolution Algorithms

YANG Qi-Wen, CAI Liang, XUE Yun-Can

(College of Computer and Information Engineering, Hohai University, Changzhou 213022)

### ABSTRACT

Differential evolution (DE) is a heuristic global optimization technique based on population. It is robust for real parameter optimization. To speed up the optimization and overcome the premature convergence of the heuristic optimization technique, many modifications are made to DE. The basic version of DE and its modifications are presented, and their advantages and disadvantages are also discussed. Some issues for further research on DE are addressed.

**Key Words** Differential Evolution, Heuristic Optimization, Genetic Algorithm

## 1 引 言

最优化方法分为传统优化方法和启发式优化方法两大类.传统优化方法大多利用目标函数的梯度(或导数)信息实现单可行解的循序、确定性搜索;启发式优化方法以仿生算法为主,通过启发式搜索策略实现多可行解的并行、随机优化.启发式搜索算法不要求目标函数连续、可微等信息,具有较好的全

局寻优能力,因而成为最优化领域的一个研究热点<sup>[1]</sup>.

在众多启发式优化方法中,差分进化(Differential Evolution, DE)是一种基于群体差异的启发式随机搜索算法,该算法是 R. Storn 和 K. Price 为求解 Chebyshev 多项式而提出的<sup>[2-4]</sup>.差分进化算法因原理简单、受控参数少、鲁棒性强等特点,引起越来越多的学者关注<sup>[5-6]</sup>.近年来,DE 在约束优化计算<sup>[7]</sup>、

收稿日期:2007-07-10;修回日期:2007-12-07

**作者简介** 杨启文,男,1969 年生,副教授,博士,主要研究方向为进化计算及其应用、智能控制. E-mail: qwyang@webmail.hhu.edu.cn. 蔡亮,男,1982 年生,硕士研究生,主要研究方向为进化计算. 薛云灿,男,1965 年生,教授,博士后,主要研究方向为智能控制与优化、生产计划与调度.

聚类优化计算<sup>[8]</sup>、非线性优化控制<sup>[9]</sup>、神经网络优化<sup>[10]</sup>、滤波器设计<sup>[11]</sup>、阵列天线方向图综合<sup>[12-13]</sup>及其它方面<sup>[14-16]</sup>得到广泛应用。

## 2 基本差分进化算法

DE 采用实数编码方式,其算法原理与遗传算法十分相似<sup>[17]</sup>,进化流程与遗传算法相同:变异、交叉和选择。DE 算法中的选择策略通常为锦标赛选择,而交叉操作方式与遗传算法也大体相同,但在变异操作方面使用差分策略,即利用种群中个体间的差分向量对个体进行扰动,实现个体变异。DE 的变异方式,有效利用群体分布特性,提高算法的搜索能力,避免遗传算法中变异方式的不足<sup>[18]</sup>。

对于优化问题:

$$\min f(x_1, x_2, \dots, x_D)$$

$$s. t \quad x_j^L \leq x_j \leq x_j^U, j = 1, 2, \dots, D,$$

其中,  $D$  是解空间的维数,  $x_j^L$ 、 $x_j^U$  分别表示第  $j$  个分量  $x_j$  取值范围的上界和下界。DE 算法流程如下。

1) 初始化种群。初始种群  $\{x_i(0) | x_{j,i}^L \leq x_{j,i}(0) \leq x_{j,i}^U, i = 1, 2, \dots, NP; j = 1, 2, \dots, D\}$  随机产生:

$$x_{j,i}(0) = x_{j,i}^L + \text{rand}(0, 1) \cdot (x_{j,i}^U - x_{j,i}^L),$$

其中,  $x_i(0)$  表示种群中第 0 代的第  $i$  条“染色体”(或个体),  $x_{j,i}(0)$  表示第 0 代的第  $i$  条“染色体”的第  $j$  个“基因”。 $NP$  表示种群大小,  $\text{rand}(0, 1)$  表示在  $(0, 1)$  区间均匀分布的随机数。

2) 变异操作。DE 通过差分策略实现个体变异,这也是区别于遗传算法的重要标志。

在 DE 中,常见的差分策略是随机选取种群中两个不同的个体,将其向量差缩放后与待变异个体进行向量合成,即

$$v_i(g+1) = x_{r_1}(g) + F \cdot (x_{r_2}(g) - x_{r_3}(g)),$$

$$i \neq r_1 \neq r_2 \neq r_3,$$

其中,  $F$  为缩放因子,  $x_i(g)$  表示第  $g$  代种群中第  $i$  个个体。

在进化过程中,为了保证解的有效性,必须判断“染色体”中各“基因”是否满足边界条件,如果不满足边界条件,则“基因”用随机方法重新生成(与初始种群的产生方法相同)。

第  $g$  代种群  $\{x_i(g) | x_{j,i}^L \leq x_{j,i}(g) \leq x_{j,i}^U, i = 1, 2, \dots, NP; j = 1, 2, \dots, D\}$  通过变异后,产生一个中间体  $\{v_i(g+1) | v_{j,i}^L \leq v_{j,i}(g+1) \leq v_{j,i}^U, i = 1, 2, \dots, NP; j = 1, 2, \dots, D\}$ 。

3) 交叉操作。对第  $g$  代种群  $\{x_i(g)\}$  及其变异的中间体  $\{v_i(g+1)\}$  进行个体间的交叉操作:

$$u_{j,i}(g+1) = \begin{cases} v_{j,i}(g+1), & \text{if } \text{rand}(0, 1) \leq CR \text{ or } j = j_{\text{rand}} \\ x_{j,i}(g), & \text{otherwise} \end{cases}$$

其中,  $CR$  为交叉概率,  $j_{\text{rand}}$  为  $[1, 2, \dots, D]$  的随机整数。

图 1 为 6 个基因位的“染色体”交叉操作示意图。为了确保变异中间体  $\{v_i(g+1)\}$  的每个“染色体”至少有一个“基因”遗传给下一代,第一个交叉操作的基因是随机取出  $v_i(g+1)$  中的第  $j_{\text{rand}}$  位“基因”作为交叉后“染色体” $u_i(g+1)$  第  $j_{\text{rand}}$  位等位“基因”。后续的交叉操作过程,则是通过交叉概率  $CR$  来选取  $x_i(g)$  还是  $v_i(g+1)$  的等位基因作为  $u_i(g+1)$  的等位基因。

4) 选择操作。DE 采用贪婪算法来选择进入下一代种群的个体:

$$x_i(g+1) = \begin{cases} u_i(g+1), & \text{if } f(u_i(g+1)) \leq f(x_i(g)) \\ x_i(g), & \text{otherwise} \end{cases}$$

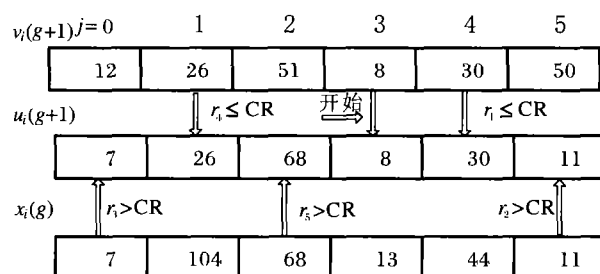


图 1 交叉操作过程

Fig. 1 Crossover operation process

## 3 差分进化算法的改进

由于选择作用的影响,随着进化代数的增加,个体间的差异会逐渐降低。个体差异性的减少又影响变异所带来的多样性,从而导致算法过早收敛到局部极值附近时,形成早熟收敛现象。

为了提高 DE 的寻优能力、加快收敛速度、克服启发式算法常见的早熟收敛现象,许多学者对 DE 算法进行改进。

### 3.1 控制参数的改进

DE 主要涉及种群规模  $NP$ 、缩放因子  $F$ 、交叉概率  $CR$  这 3 个控制参数。通常的做法是根据经验选取一组固定参数<sup>[2-3,19]</sup>: 种群规模  $NP \in [5D, 10D]$ , 缩放因子  $F \in [0.5, 1]$ , 交叉概率  $CR \in [0.8, 1]$ 。这样

既保证较高的寻优成功率,也保证较快的收敛速度。

经验参数反映的是统计学上的性能效果,指导 DE 算法的整个搜索过程。由于它们在寻优的过程中没有改变,无法较好地满足进化各阶段中算法性能对控制参数的特殊要求。因此,产生不少参数自适应的 DE 算法。

根据缩放因子  $F$ 、交叉概率  $CR$  在寻优过程中的作用,文献[20]提出一种线性的变化策略:

$$CR(g+1) = CR(g) - \frac{CR(0) - CR_{\min}}{G_{\max}},$$

$$F(g+1) = F(g) + \frac{F_{\max} - F(0)}{G_{\max}},$$

其中,  $CR(0)$  和  $F(0)$  分别代表变异概率和缩放因子的初始值,  $CR_{\min}$  表示进化过程中变异概率的最小值,  $F_{\max}$  表示进化过程中缩放因子的最大值,  $G_{\max}$  为最大进化代数。随着进化代数的增加,  $CR$  线性减小,  $F$  线性增加。这种参数自调整的目的,是希望 DE 算法在搜索初期能够保证较大的种群多样性,而到后期具有较快收敛速度。其不足之处是,如果在搜索后期还没有找到满意解的话,算法很容易早熟收敛。

与线性变化不同, Mendes 和 Mohais 提出  $F$  和  $CR$  的随机选取原则<sup>[21-22]</sup>。文献[23]在此基础上进一步将  $F$  修正为  $(0.5, 1)$  间的均匀分布随机数,即  $F \sim U(0.5, 1)$ 。而文献[5]则将缩放因子  $F$  建议为  $F \sim U(0, 1)$ , 减少需调整的参数。随机选择在一定程度上避免控制参数所影响的早熟收敛。

在进化过程中,种群中各个体可共享一个控制参数值,也可独立使用它们<sup>[24-26]</sup>。

维持种群的多样性是克服算法早熟收敛的一个关键因素。但是,  $F$  和  $CR$  不论采取上述线性变化还是随机改变,都没有与种群的多样性有直接的联系。在文献[27]中,这种不足得到改善。

Liu<sup>[27]</sup>等人利用个体位置多样性:

$$PC = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} \sum_{j=1}^D (x_{j,i}(g) - x_{j,i}(g+1))^2}$$

及个体适应值多样性

$$FC = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} (f(x_i(g)) - f(x_i(g+1)))^2}$$

等信息,进行控制参数的模糊自适应调节(其中  $f(x_i(g))$  表示第  $g$  代种群中第  $i$  个个体的适应度)。当个体位置多样性  $PC$  和个体适应度值多样性  $FC$  都很小时,算法将缩小  $F$  和  $CR$  的值,提高局部搜索能力,加速算法收敛速度。反之,增加  $F$  和  $CR$  的值,提高算法的全局搜索能力。

相对于参数的线性改变或随机改变策略,文献

[27]有效利用种群的分布情况来针对性地调整控制参数,能够最大限度地提高对进化信息的利用率,增强对解空间的探测能力。因此,这是参数自适应的一个方向。但由于需要进行种群多样性评估,如何减少由此产生的计算负荷,也是一个需要注意的问题。

### 3.2 差分策略的改进

DE 算法以差分策略为主要特征。Price 和 Storn 一共提出十余种不同的差分策略来实现变异操作<sup>[21]</sup>。

表 1 以二项式交叉(bin)为例列举其中的 7 种差分策略,其中,  $DE/x/y/z$  含义为:  $x$  表示变异基向量,它既可以是种群中的随机(rand)向量  $x_{r1}$ ,也可以是当前种群中的最优(best)向量  $x_b$ ,还可以是当前(current)个体  $x_i$ ;  $y$  表示差分向量的个数;  $z$  代表交叉的模式,而  $\lambda$  为组合因子。

表 1 差分策略

Table 1 Differential strategies

差分策略	差分表达式
DE/rand/1/bin	$x_1 + F(x_2 - x_3)$
DE/rand/2/bin	$x_1 + F(x_2 + x_3 - x_4 - x_5)$
DE/best/1/bin	$x_b + F(x_2 - x_3)$
DE/best/2/bin	$x_b + F(x_2 + x_3 - x_4 - x_5)$
DE/rand - to - best/bin	$x_1 + \lambda(x_b - x_1) + F(x_2 - x_3)$
DE/current - to - rand/bin	$x_i + \lambda(x_1 - x_i) + F(x_2 - x_3)$
DE/current - to - best/bin	$x_i + \lambda(x_b - x_i) + F(x_1 - x_2)$

DE/rand/1/bin 和 DE/best/2/bin 是目前使用最广、应用最为成功差分策略。前一种策略有利于保持群体的多样性,后一种策略则更强调算法的收敛速度。

除此之外, Fan 等人<sup>[28]</sup>还提出一种三角形差分策略:

$$x_i(g) = \frac{1}{3} [x_{r1}(g) + x_{r2}(g) + x_{r3}(g)] +$$

$$(p_2 - p_1)(x_{r1}(g) - x_{r2}(g)) +$$

$$(p_3 - p_2)(x_{r2}(g) - x_{r3}(g)) +$$

$$(p_1 - p_3)(x_{r3}(g) - x_{r1}(g)),$$

其中

$$p_i = \frac{|f(x_{r_i}(g))|}{|f(x_{r_1}(g))| + |f(x_{r_2}(g))| + |f(x_{r_3}(g))|},$$

$$i \neq r_1 \neq r_2 \neq r_3.$$

### 3.3 选择策略的改进

基本差分进化算法采用的贪婪算法,有利于加快算法的收敛速度,但由于降低了种群多样性,增加了早熟收敛的风险。

文献[23]、[29]通过改变评价标准来解决有用个体可能被丢弃的问题,从而有效维持种群的多样性信息.

文献[23]的选择策略为

$$\text{If } f(u_i(g+1)) - f(x_i(g)) \leq k \cdot \sigma^2(g)$$

$$\text{Then } x_i(g+1) = u_i(g+1),$$

其中,  $\sigma^2(g)$  是一个关于进化代数的函数,  $k$  为指定的常数. 在进化过程中, 每代的选择标准有所不同. 这种方法, 采取与文献[20]相似的思想, 根据不同进化阶段对选择策略的要求, 构造一个以进化代数为函数的评价函数, 算法简单且具有较好的物理可解释性. 但这种方法忽略进化过程中种群的具体信息, 因此, 算法性能受到最大进化代数以及求解问题的复杂性的影响.

此外, 文献[23]还提出一个随机选择策略. 用  $f(x_p)$  表示父代个体的适应度函数,  $f(x_o)$  为其子代个体的适应度函数, 子代取代父代的概率为

$$p_{\text{selection}} = \min(1, \frac{f(x_p)}{f(x_o)}).$$

这意味着, 如果  $f(x_o) < f(x_p)$ , 子代被选中进入下一代种群, 如果  $f(x_o) \geq f(x_p)$ , 则按照  $f(x_p)/f(x_o)$  的概率在  $x_o$  和  $x_p$  中选取一个个体进入下一代. 这种方法可以看作是锦标赛策略的一种广义形式. 不同之处在于, 在锦标赛策略中, 如果  $f(x_o) \geq f(x_p)$ , 则  $f(x_o)$  被选中的概率为 0. 因此, 随机选择相对增加了子代的存活率.

不同于文献[23]的选择策略, 文献[29]提出采用如下方式对个体进行选择.

初始化  $G, T$

While  $G < G_{\max}$

产生个体  $u_i(g+1)$

If  $f(u_i(g+1)) \leq (1+T) \cdot f(x_i(g))$

Then  $x_i(g+1) = u_i(g+1), G = G_{\max}$

Else  $T = a^{-1} \exp(-\sin(\frac{G+G}{g_{\max}+G_{\max}}))$

$G = G+1$

Endif

Endwhile

其中,  $G$  表示当前个体在进化时间  $T$  内的更新次数,  $G_{\max}$  表示在进化时间  $T$  内所允许的最大更新次数,  $g$  表示算法当前的进化代数,  $g_{\max}$  表示算法最大的进化代数,  $a$  是与问题有关的常数. 这种选择方式的特点是对个体适合度进行加权比较, 然后择优录取. 在进化的初始阶段, 权系数  $(1+T)$  的变化较大, 因而适合度差的个体也存在一定的生存概率, 这有利于维持种群的多样性. 到了进化的后期, 权系数趋近于

1, 个体的生存就直接依赖于它们的适合值, 这样就加速了种群的收敛.

### 3.4 种群重构

种群重构是将已收敛的种群通过外部作用在一定有效范围内重新分布个体的位置, 增加种群的多样性.

Blackwell 和 Branke 根据量子力学中电子的量子状态, 提出在最优个体附近随机产生重构个体<sup>[30]</sup>:

$$x_i = x_b + \frac{r \cdot p_i}{\sqrt{\sum_{i=1}^{NP} p_i}},$$

其中,  $p_i \sim N(0, 1)$ ,  $r \sim U(0, r_{\text{cloud}})$ ,  $r_{\text{cloud}}$  为指定的粒子云半径.

另一种简单方法是在最优个体附近应用高斯分布函数产生重构个体<sup>[21]</sup>:  $x_i = x_b + N(0, \sigma)$ , 其中  $x_b$  为当前种群中最优个体.

根据种群信息来确定种群重构时机是防止种群早熟收敛的一种有效方法.

Y. C. Lin<sup>[31]</sup>提出当种群多样性指数

$$u = \frac{\sum_{i=1}^{NP} \sum_{j=1}^D \eta_{j,i}}{D(NP-1)} < \varepsilon_1,$$

即超过临界指数时, 执行迁移操作来进行种群重构:

$$x_{j,i}(g+1) =$$

$$\begin{cases} x_{j,b}(g) + \text{rand}(0, 1) \cdot (x_j^L - x_{j,b}(g)), \\ \quad \text{if } \text{rand}(0, 1) < \frac{x_{j,b}(g) - x_j^L}{x_j^U - x_j^L} \\ x_{j,b}(g) + \text{rand}(0, 1) \cdot (x_j^U - x_{j,b}(g)), \\ \quad \text{otherwise} \end{cases}$$

其中,  $x_{j,b}(g)$  表示到第  $g$  代种群为止种群最优个体的第  $j$  维分量,  $\varepsilon_1 \in (0, 1)$  为预定义的聚集程度指标常数,  $\varepsilon_2 \in (0, 1)$  表示基因多样性程度指标常数,  $\eta_{j,i}$  为第  $i$  个个体与种群最优个体在第  $j$  维分量上的相似度:

$$\eta_{j,i} = \begin{cases} 1, & \text{if } | \frac{x_{j,i} - x_{j,b}}{x_{j,b}} | > \varepsilon_2 \\ 0, & \text{otherwise} \end{cases}$$

吴亮红等人<sup>[14]</sup>则根据群体适应度方差

$$\delta^2 = \sum_{i=1}^{NP} \left| \frac{f_i - f_{\text{avg}}}{f} \right|^2$$

来进行种群重构(式中,  $f_i$  为第  $i$  个个体的适应度,  $f_{\text{avg}}$  为种群目前的平均适应度). 当种群适应度方差小于预先制定的阈值时, 种群重构算法如下:

$$\begin{aligned}x_{j,b} &= x_{j,b} \cdot (1 + 0.5\eta), \\x_{j,i} &= x_{j,i} \cdot (1 + 0.5\eta),\end{aligned}$$

其中  $\eta$  为服从  $Gauss(0,1)$  分布的随机变量.

作为预防早熟收敛的一种手段,种群重构时机对算法性能的影响要大于重构方式对算法的影响.基于此,此方面的改进工作应加强种群重构条件的研究,即加强对种群多样性的评估,减少重构方式的复杂性.

### 3.5 多种群差分进化算法

当多个种群共同执行进化搜索时,即使单个种群出现多样性的丧失,由于种群间存在差异,通过种群间的信息共享和交换,依然可以保证整个算法继续进化.

Ali<sup>[32]</sup> 采用双种群  $P_1$ 、 $P_2$  进行差分进化算法.每个种群包含  $NP$  个个体,  $P_2$  作为辅助种群,用于存放  $P_1$  在选择时被舍弃的个体.  $P_2$  中的个体虽然较“差”,但  $P_2$  却具有较好的种群多样性.

种群  $P_1$  通过种群间的交叉操作

$$u_{j,i}(g+1) = \begin{cases} a_{j,i}(g), & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand} \\ v_{j,i}(g), & \text{otherwise} \end{cases}$$

实现与  $P_2$  的交换( $a_{j,i} \in P_2$ ). 这种措施,以较少的计算量来增加种群  $P_1$  的多样性.

Tasgetiren<sup>[33]</sup> 则提出基于多个子种群的多种群差分进化模式.子种群之间每隔  $R$  代进行重组:

$$v_{j,i,k}(g+1) = x_{j,i,a}(g) + F \cdot (x_{j,i,b}(g) - x_{j,i,c}(g)),$$

其中,  $k \neq a \neq b \neq c$  表示不同的子种群,  $i$  为个体编号,  $j$  为个体中的分量编号.

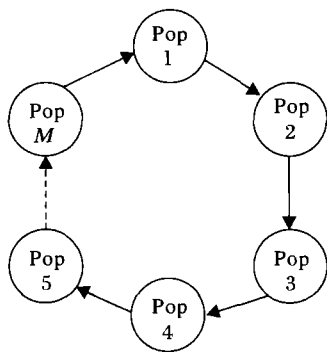


图2 多种群进化的圆环结构

Fig.2 Circle structure for multi-population evolution

图2是多种群进化的一种拓扑结构——圆环结构<sup>[34]</sup>.各子种群间的最优个体通过一个循环路径实现信息共享和交换,子种群从上一级子种群继承当前的最优个体,用来指导本种群的进化过程.

在多种群进化算法中,各子种群规模既可固定,也可动态改变<sup>[35]</sup>.

## 4 相关混合算法

在最优化技术中,除差分进化外,还有许多启发式搜索方法.结合不同搜索思想的混合差分进化算法,可提高寻优性能.

混合差分进化算法的一个切入点是如何通过父代种群的个体信息智能确定下一代个体在解空间的位置<sup>[36]</sup>.

### 4.1 与传统最优化技术相结合

Ahuja 和 Orlin 认为,好的进化操作应兼具随机搜索与方向性搜索的优点,能够组合父代的优秀性状而产生优于父代的子代个体<sup>[36]</sup>.因此,为了能够在差分进化中实现确定性搜索,最直接的方法就是利用传统的最优化技术.

1) 利用梯度信息.当种群适应度不再下降时,利用当前最优点的梯度信息<sup>[31]</sup>,加速种群向全局最优点的移动:

$$x_b(g+1) = \begin{cases} u_b(g+1), & \text{if } f(u_b(g+1)) < f(x_b(g+1)) \\ x_b(g) - \alpha \nabla f(x_b(g)), & \text{otherwise} \end{cases}$$

当前种群最优个体  $x_b(g)$  的梯度信息可利用  $[f(x_b(g)) - f(x_b(g-1))][x_b(g) - x_b(g-1)]^{-1}$  近似估计,步长  $\alpha \in (0,1)$ .如果

$$f(x_b(g)) - \alpha \nabla f(x_b(g)) \leq f(x_b(g)),$$

则加速过程结束;否则减少  $\alpha$  的值继续搜索.

2) 借鉴单纯形方法. Kaelo 借鉴单纯形搜索方法,在差分进化算法中引入反射和压缩算子来提高局部搜索效率<sup>[37]</sup>,具体方法如下.

step 1 判断父代个体  $x_i$ , 子代个体  $u_i$ , 种群最优个体  $x_b$  的适应度大小.

step 2 如果  $f(x_i) < f(u_i) < f(x_b)$  且  $rand(0,1) < \omega$ , 执行 step 3; 否则转 step 5.

step 3 反射操作产生反射点  $r_i$ . 如果  $f(r_i) < f(u_i)$ ,  $x_i = r_i$ , 转到 step 5; 否则, 继续执行 step 4.

step 4 压缩操作产生压缩点  $c_i$ . 如果  $f(c_i) < f(u_i)$ ,  $x_i = c_i$ .

step 5  $x_i = u_i$ ,  $i = i + 1$ , 转到 step 1.

其中,  $\omega$  是小于 1 的常数, 控制着反射和压缩操作发生的概率.

传统的最优化技术具有较高的局部搜索效率, 它与差分进化算法相结合, 可明显改善差分进化算

法的寻优速度,同时也能提高寻优精度.

#### 4.2 与模拟退火算法相结合

模拟退火算法也是一种启发式寻优算法,杨静宇等人<sup>[38]</sup>采用模拟退火的选择方式来确定选择概率  $p(g)$ :

$$p(g+1) = \frac{p(g)}{\log(10 + g \times AS)},$$

其中  $AS$  为退火速度. 选择概率随进化代数的增加而减少,与种群的分布信息无关. 因此,这种方式与其它任何以进化代数为函数的改进措施相似,不能为算法性能带来本质上的提高.

#### 4.3 与 PSO 算法相结合

PSO 算法是目前启发式搜索算法的一个分支. Chakraborty 采用 PSO 算法形式实现变异操作<sup>[39]</sup>:

$$L_i = x_i + \lambda'(x_{\text{best}} - x_i) + \kappa'(x_p - x_q),$$

$$G_i = x_i + \lambda(x_{\text{best}} - x_i) + \kappa(x_r - x_s),$$

$$u_i = \omega G_i + (1 - \omega)L_i,$$

其中,  $L_i$  和  $G_i$  分别为局部搜索向量和全局搜索向量,  $r, s \in (1, NP)$ ,  $p, q \in (i - k, i + k)$ .  $x_p$  和  $x_q$  为  $x_i$  的  $k$  邻域:  $\{x_{i-k}, \dots, x_i, \dots, x_{i+k}\}$ .

在进化初期,  $\omega$  取值较大,算法侧重全局搜索;进化后期,  $\omega$  取值较小,算法侧重于局部搜索.

#### 4.4 与蚁群算法相结合

J. P. Chiou<sup>[40]</sup>将蚁群算法的思想引入到 DE 中用于对差分策略进行最优选择. 每代由  $NP$  个蚂蚁通过空间的信息素和个体差异信息来寻找最优的差分策略. 个体信息素的更新公式如下:

$$\tau_i^{\text{New}} = \rho \cdot \tau_i^{\text{Old}} + \Delta \tau_i,$$

$$\Delta \tau_i = \begin{cases} Q \frac{f(x_b(g)) - f(u_i(g+1))}{f(u_i(g+1))}, & \text{if } f(u_i(g+1)) < f(x_i(g)) \\ 0, & \text{otherwise} \end{cases}$$

其中,  $\rho$  是信息素挥发因子,  $Q$  为常数. 相应差分策略的选择概率为

$$p_i = \frac{\tau_i^\alpha \eta_i^\beta}{\sum_{i=1}^{NP} \tau_i^\alpha \eta_i^\beta},$$

其中,个体差异  $\eta_i$  定义为

$$\eta_i = \left( \sum_{j=1}^D \left( \frac{x_{j,i}(g+1) - x_{j,b}(g)}{x_{j,i}(g+1)} \right)^2 \right)^{1/2}.$$

这种混合算法的复杂性以及计算负荷均较高.

#### 4.5 与遗传算法相结合

受遗传算法选择压力的启发, Bergey<sup>[41]</sup>根据父代个体适应度的大小将种群重新排序,然后按照独

立伯努力实验,在选择压力下生成变异的基向量.

## 5 结束语

DE 的卓越性能已初露锋芒,吸引了众多研究者的关注,并提出许多有效的改进方法. 但是还有很多方面有待完善,需要加强并进行深入研究.

1) 加强 DE 算法理论基础和系统分析方法的研究. 目前 DE 以及各种改进方法均缺乏成熟的理论和系统的分析方法作指导,如何避免局部极值以及如何在多样性和收敛性之间进行平衡缺少更好的指导原则.

2) 加强 DE 各种改进方法的综合研究. 作为一种优化工具,它自身也存在许多待优化问题. 研究 DE 算法各要素之间的相互关系,并以此指导各控制参数的优化配置,对于提高 DE 算法的性能具有重要的意义.

3) 加强 DE 与其他算法的结合. 结合不同算法的寻优思想对 DE 进行改进,使 DE 在全局寻优能力和局部寻优能力都得到改进也是以后的一个研究方向.

4) 加强 DE 与应用的结合. 根据不同的工程求解问题,进一步改进和完善 DE 算法,使其在应用领域发挥更大的作用.

## 参 考 文 献

- [1] Lopez C I L, van Willigenburg L G, van Straten G. Efficient Differential Evolution Algorithms for Multimodal Optimal Control Problems. *Applied Soft Computing*, 2003, 3(2): 97-122
- [2] Storn R, Price K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 1997, 11(4): 341-359
- [3] Storn R, Price K. Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report, TR-95-012, Berkeley, USA: University of California. International Computer Science Institute, 1995
- [4] Liu Mingguang. Differential Evolution Algorithms and Modification. *Systems Engineering*, 2005, 23(2): 108-111 (in Chinese)  
(刘明广. 差异演化算法及其改进. *系统工程*, 2005, 23(2): 108-111)
- [5] Xie Xiaofeng, Zhang Wenjun, Zhang Guorui, et al. Empirical Study of Differential Evolution. *Control and Decision*, 2004, 19(1): 49-52, 56 (in Chinese)  
(谢晓峰, 张文俊, 张国瑞, 等. 差异演化的实验研究. *控制与决策*, 2004, 19(1): 49-52, 56)
- [6] Vesterstrom J, Thomsen R. A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithms on

- Numerical Benchmark Problems // Proc of the IEEE Congress on Evolutionary Computation. Piscataway, USA, 2004, II: 1980 – 1987
- [7] Kim H K, Chong J K, Park K Y, *et al.* Differential Evolution Strategy for Constrained Global Optimization and Application to Practical Engineering Problems. IEEE Trans on Magnetics, 2007, 43(4): 1565 – 1568
- [8] Omran M G H, Engelbrecht A P. Self-Adaptive Differential Evolution Methods for Unsupervised Image Classification // Proc of the IEEE Conference on Cybernetics and Intelligent Systems. Bangkok, Thailand, 2006: 1 – 6
- [9] Zhang Renqian, Ding Jianxun. Non-Linear Optimal Control of Manufacturing System Based on Modified Differential Evolution // Proc of the IMACS Multiconference on Computational Engineering in Systems Applications. Beijing, China, 2006: 1797 – 1803
- [10] Dhahri H, Alimi A M. The Modified Differential Evolution and the RBF (MDE-RBF) Neural Network for Time Series Prediction // Proc of the International Joint Conference on Neural Networks. Vancouver, USA, 2006: 2938 – 2943
- [11] Yang Shiwen, Gan Y B, Qing Anyong. Sideband Suppression in Time-Modulated Linear Arrays by the Differential Evolution Algorithm. IEEE Trans on Antennas and Wireless Propagation Letters, 2002, 1(1): 173 – 175
- [12] Massa A, Pastorino M, Randazzo A. Optimization of the Directivity of a Monopulse Antenna with a Subarray Weighting by a Hybrid Differential Evolution Method. IEEE Trans on Antennas and Wireless Propagation Letters, 2006, 5(1): 155 – 158
- [13] Fan Yu, Jin Ronghong, Geng Junping, *et al.* A Hybrid Optimized Algorithm Based on Differential Evolution and Genetic Algorithm and Its Applications in Pattern Synthesis of Antenna Arrays. Acta Electronica Sinica, 2004, 32(12): 1997 – 2000 (in Chinese)  
(范瑜, 金荣洪, 耿军平, 等. 基于差分进化算法和遗传算法的混合优化算法及其在阵列天线方向图综合中的应用. 电子学报, 2004, 32(12): 1997 – 2000)
- [14] Wu Lianghong, Wang Yaonan, Yuan Xiaofang, *et al.* Differential Evolution Algorithm with Adaptive Second Mutation. Control and Decision, 2006, 21(8): 898 – 902 (in Chinese)  
(吴亮红, 王耀南, 袁小芳, 等. 自适应二次变异差分进化算法. 控制与决策, 2006, 21(8): 898 – 902)
- [15] Su C T, Lee C S. Network Reconfiguration of Distribution Systems Using Improved Mixed-Integer Hybrid Differential Evolution. IEEE Trans on Power Delivery Review, 2002, 22(12): 60 – 66
- [16] Zhai Jie, Wang Chunfeng, Li Guangquan. A Portfolio Management Model Based on Differential Evolution Algorithm. Journal of Tianjin University: Science and Technology, 2002, 35(3): 304 – 308 (in Chinese)  
(翟捷, 王春峰, 李光泉. 基于差分进化方法的投资组合管理模型. 天津大学学报: 自然科学与工程技术版, 2002, 35(3): 304 – 308)
- [17] Onwubolu G C, Babu B V. New Optimization Techniques in Engineering. Berlin, Germany: Springer-Verlag, 2004
- [18] Yang Qiwen, Jiang Jingping, Qu Zhaoxia, *et al.* Improving Genetic Algorithms by Using Logic Operation. Control and Decision, 2002, 15(4): 510 – 512 (in Chinese)  
(杨启文, 蒋静坪, 曲朝霞, 等. 应用逻辑操作改善遗传算法性能. 控制与决策, 2000, 15(4): 510 – 512)
- [19] Storn R. On the Usage of Differential Evolution for Function Optimization // Proc of the Biennial Conference of the North American on Fuzzy Information Processing Society. Berkeley, USA, 1996: 519 – 523
- [20] Chang C S, Xu D Y. Differential Evolution Based Tuning of Fuzzy Automatic Train Operation for Mass Rapid Transit System. IEE Proc—Electric Power Applications, 2000, 147(3): 206 – 212
- [21] Mendes R, Mohais A S. DynDE: A Differential Evolution for Dynamic Optimization Problems // Proc of the Congress on Evolutionary Computation. Edinburgh, UK, 2005, III: 2808 – 2815
- [22] Tomislav Š. Sensitivity of Differential Evolution Algorithm to Values of Control Parameters // Proc of the International Conference on Artificial Intelligence. Las Vegas, USA, 2002: 1087 – 1093
- [23] Das S, Konar A, Chakraborty U K. Improved Differential Evolution Algorithms for Handling Noisy Optimization Problems // Proc of the Congress on Evolutionary Computation. Edinburgh, UK, 2005, II: 1691 – 1698
- [24] Brest J, Zumer V, Maucec M S. Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization // Proc of the Congress on Evolutionary Computation. Vancouver, USA, 2006: 215 – 222
- [25] Nobakhti A, Wang Hong. Co-Evolutionary Self-Adaptive Differential Evolution with a Uniform-Distribution Update Rule // Proc of the IEEE International Symposium on Intelligent Control. Munich, Germany, 2006: 1264 – 1269
- [26] Nobakhti A, Wang Hong. A Self-Adaptive Differential Evolution with Application on the ALSTOM Gasifier // Proc of the American Control Conference. Minneapolis, USA, 2006: 4489 – 4494
- [27] Liu Junhong, Lampinen J. A Fuzzy Adaptive Differential Evolution Algorithm. Soft Computing: A Fusion of Foundations, Methodologies and Applications, 2005, 9(6): 448 – 462
- [28] Fan Huiyuan, Lampinen J. A Trigonometric Mutation Operation to Differential Evolution. Journal of Global Optimization, 2003, 27(1): 105 – 129
- [29] Schmidt H, Thierauf G. A Combined Heuristic Optimization Technique. Advances in Engineering Software, 2005, 36(1): 11 – 19
- [30] Blackwell T, Branke J. Multi-Swarm Optimization in Dynamic Environments // Raidl G R, Cagnoni S, Branke J, *et al.*, eds. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2004, 3005: 489 – 500
- [31] Lin Y C, Wang Fengsheng, Hwang K S. A Hybrid Method of Evolutionary Algorithms for Mixed-Integer Nonlinear Optimization Problems // Proc of Congress on Evolutionary Computation. Washington, USA, 1999, III: 2159 – 2166
- [32] Ali M M. Differential Evolution with Preferential Crossover. European Journal of Operational Research, 2007, 181(3): 1137 – 1147
- [33] Tasgetiren M F, Suganthan P N. A Multi-Populated Differential Evolution Algorithm for Solving Constrained Optimization Problem // Proc of the IEEE Congress on Evolutionary Computation. Vancouver, USA, 2006: 33 – 40

- [34] Parsopoulos K E, Tasoulis D K, Pavlidis N G, *et al.* Vector Evaluated Differential Evolution for Multi-Objective Optimization // Proc of the IEEE Congress on Evolutionary Computation. Portland, USA, 2004, 1: 204 – 211
- [35] Wang Yongjun, Zhang Jianshe, Zhang Gaiying. A Dynamic Clustering Based Differential Evolution Algorithm for Global Optimization. European Journal of Operational Research, 2007, 183(1): 56 – 73
- [36] Ahuja R K, Orlin J B. Developing Fitter Genetic Algorithms. Journal of Computing, 1997, 9(3): 251 – 253
- [37] Kaelo P, Ali M M. A Numerical Study of Some Modified Differential Evolution Algorithms. European Journal of Operational Research, 2006, 169(3): 1176 – 1184
- [38] Yan Jingyu, Ling Qing, Sun Demin. A Differential Evolution with Simulated Annealing Updating Method // Proc of the International Conference on Machine Learning and Cybernetics. Dalian, China, 2006: 2103 – 2106
- [39] Chakraborty U K, Das S, Konar A. Differential Evolution with Local Neighborhood // Proc of the IEEE Congress on Evolutionary Computation. Vancouver, USA, 2006: 2042 – 2049
- [40] Chiou J P, Chang C F, Su C T. Ant Direction Hybrid Differential Evolution for Solving Large Capacitor Placement Problems. IEEE Trans on Power System, 2004, 19(4): 1794 – 1800
- [41] Bergey P K, Ragsdale C. Modified Differential Evolution: A Greedy Random Strategy for Genetic Recombination. Omega—International Journal of Management Science, 2005, 33(3): 255 – 265

www.cnki.net