

◎ 理论研究 ◎

差分进化算法的交叉概率因子递增策略研究

邓泽喜^{1,2}, 刘晓冀²DENG Ze-xi^{1,2}, LIU Xiao-ji²

1. 毕节学院 数学系, 贵州 毕节 551700

2. 广西民族大学 数学与计算机科学学院, 南宁 530006

1. Department of Mathematics, Bijie College, Bijie, Guizhou 551700, China

2. College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China

E-mail: liuxiaoji.2003@yahoo.com.cn

DENG Ze-xi, LIU Xiao-ji. Study on strategy of increasing cross rate in differential evolution algorithm. Computer Engineering and Applications, 2008, 44(27): 33-36.

Abstract: To efficiently control the global and local search of Differential Evolution algorithm (DE), motivated by the idea of increasing Cross Rate (CR), three nonlinear strategies for CR, a parabola opening upwards, a parabola opening downwards and an exponential curve, based on the existing differential evolution algorithm with adaptive second mutation are proposed. Four classic Benchmarks functions are used to evaluate the strategies on the DE performance. The experimental results show that for most continuous optimization problems, the strategy of concave function gains an advantage over the linear strategy, while the linear strategy outperforms strategy of convex function with the identical initial and final weights.

Key words: Differential Evolution algorithm (DE); premature convergence; Cross Rate (CR)

摘要: 为了有效地控制差分进化算法的全局搜索和局部搜索, 基于递增交叉概率因子的基本思想, 在已有的自适应二次变异差分进化算法的基础上, 提出了开口向下抛物线、开口向上抛物线和指数曲线 3 种非线性的交叉概率因子递增策略, 并用 4 种经典函数对它进行测试。测试结果表明, 对于多数连续优化问题, 在初始参数相同的情况下, 凹函数递增策略优于线性策略, 而线性优于凸函数策略。凹函数递增策略能够在不影响收敛精度的情况下较大幅度地提高差分进化算法的收敛速度。

关键词: 差分进化; 早熟收敛; 交叉概率

DOI: 10.3778/j.issn.1002-8331.2008.27.011 文章编号: 1002-8331(2008)27-0033-04 文献标识码: A 中图分类号: TP391

差分进化算法 (Differential Evolution, DE) 是 Dainer Storn 和 Kenneth Price 于 1995 年共同提出的一种采用浮点矢量编码, 在连续空间中进行启发式随机搜索的优化算法^[1]。

近年来的研究已越来越多地表明, DE 对函数优化有强大的生命力, 特别是对一些规模大、维数高、非线性和不可微等特性的函数进行优化^[2]。与其它进化算法一样易陷入局部最优, 存在早熟收敛现象。本文在自适应二次变异差分进化算法^[3]的基础上, 提出了开口向下抛物线、开口向上抛物线和指数曲线 3 种非现象的交叉概率因子递增策略, 并用 4 种经典函数对它进行测试对比研究。

1 差分进化算法

DE 的基本操作包括变异、交叉、选择 3 种操作, 但与其它进化算法如遗传算法 (GA) 不同。DE 由 N_p (种群大小) 个 D 维

(变量个数) 的参数矢量 x_i^g ($i=1, 2, \dots, N_p$) 构成种群在搜索空间进行寻优, 其中 g 表示迭代数。首先由父代个体间的差分矢量构成变异算子; 接着按一定的概率, 父代个体与变异个体之间进行交叉操作, 生成一试验个体; 然后在父代个体与实验个体之间根据适应度大小进行选择操作, 选择适应度更优的个体作为子代。

1.1 变异操作

DE 最基本的变异成分是父代的差分矢量, 每个矢量对包括父代两个不同的个体 ($x_{r_1}^g, x_{r_2}^g$), 根据变异个体的生成方法不同, 形成不同的差分进化算法方案^[4]。本文选择从种群中随机选择 4 个不同个体生成差分矢量对每代最优个体进行变异操作的方案。这种方案既能提高算法的收敛速度, 又能在一定程度上保持较高的种群多样性。个体变异操作的公式为:

基金项目: 广西自然科学基金 (the Natural Science Foundation of Guangxi of China under Grant No.05775032, No.06400161); 广西民族大学研究生教育创新计划 (the Graduated Educate Innovation Program of Guangxi University for Nationalities under Grant No.GXUN-CHX0752)。

作者简介: 邓泽喜 (1982-), 男, 硕士, 主要研究领域为人工智能、广义逆理论; 刘晓冀 (1972-), 男, 教授, 主要研究领域广义逆理论及其应用。

收稿日期: 2008-04-02 修回日期: 2008-06-20

$$x_m = x_{gbest}^g + F[(x_a^g - x_b^g) + (x_c^g - x_d^g)] \quad (1)$$

式中: x_{gbest}^g 为种群中适应度最好的个体; x_a^g, x_b^g, x_c^g 和 x_d^g 为 x_{gbest}^g 与不同的 4 个互不相同的个体; F 为缩放因子, 其取值范围为 $(0, 1.2]$; 相当于 x_{gbest}^g 的一个噪音版本, F 越大, x_{gbest}^g 变异越多, 对 x_m 的影响越大。

1.2 交叉操作

DE 利用交叉操作以保持种群的多样性, 对于群体中第 i 个个体 x_i^g , 将与 x_m 进行交叉操作, 产生实验个体 x_T 。为保证个体 x_i^g 的进化, 首先通过随机选择, 使得 x_T 的 D 维变量中至少有一维由 x_m 贡献, 而对于其它维, 可利用一个交叉概率因子 CR 决定 x_T 中哪位由 x_m 贡献, 哪位由 x_i^g 贡献。

交叉操作的方程为:

$$x_{Tj} = \begin{cases} x_{mj}, & rand() \leq CR \\ x_{ij}^g, & rand() \geq CR \end{cases} \quad (2)$$

其中, $j=1, 2, \dots, d$ 式中: $rand()$ 为 $[0, 1]$ 之间的均匀随机数, j 表示第 j 个变量(基因), D 为变量的维数。由式(2)可知, CR 越大, x_m 对 x_T 的贡献越多, 当 $CR=1$ 时, $x_T=x_m$, 有利于局部搜索和加速收敛速率; CR 越小, x_i^g 对 x_T 的贡献越多, 当 $CR=0$ 时, $x_T=x_i^g$, 有利于保持种群的多样性和全局搜索。由此可见, 保持种群多样性和收敛速率是矛盾的。

1.3 选择操作

DE 采用“贪婪”的搜索策略, 经过变异和交叉操作后生成的试验个体 x_T 和进行竞争。只用当 x_T 的适应度 x_i^g 更优时才被选作子代; 否则直接将 x_i^g 作为子代, 选择操作的公式为:

$$x_i^{g+1} = \begin{cases} x_T, & f(x_T) < f(x_i^g) \\ x_i^g, & f(x_T) \geq f(x_i^g) \end{cases} \quad (3)$$

2 自适应二次变异差分进化算法

2.1 自适应二次变异的思想

为了加快收敛速率, 避免早熟收敛, 人们提出了多种方法, 其中文献[3]提出了自适应二次变异差分进化算法, 该文提出了一种时变交叉概率因子 CR 的方法, 设 CR_{min} 为最小交叉概率, CR_{max} 为最大交叉概率, g 为当前迭代次数, G 为最大迭代次数, 即:

$$CR = CR_{min} + \frac{g(CR_{max} - CR_{min})}{G} \quad (4)$$

使交叉概率因子 CR 随算法迭代的进行而线性变大。

初始阶段 x_i^g 对 x_T 贡献多, 提高全局搜索能力, 而在后期则 x_m 对 x_T 贡献多, 提高局部搜索能力。该方法加快了收敛速度, 提高了算法的性能。当求解问题很复杂时, 该方法使得 DE 在迭代后期全局搜索能力不足, 导致得不到最优解。

产生早熟收敛的根本原因是随迭代次数的增加和种群多样性的快速下降, 出现了“聚集”现象。为了定量描述种群的状态, 下面给出群体适应度方差的定义。

定义 1 设群体规模为 N_p , f_i 为第 i 个个体的适应度, f_{avg} 为

种群目前的平均适应度, 则 δ^2 可定义为:

$$\delta^2 = \sum_{i=0}^{N_p} \left| \frac{f_i - f_{avg}}{f} \right|^2 \quad (5)$$

其中 f 为归一化定标因子, 其作用是限制 δ^2 的大小。本文算法中, f 取值为:

$$f = \begin{cases} \max\{|f_i - f_{avg}|\}, \max\{|f_i - f_{avg}|\} > 1 \\ 1, \text{other} \end{cases} \quad (6)$$

定义 1 表明, 群体适应度方差 δ^2 反映的是种群中所以个体的“聚集”程度。 δ^2 越小, 则种群越聚集在一起; 反之, 种群则处于随机搜索阶段。

在差分进化算法运行过程中, 如果群体适应度方差等于零, 且此时得到的最优解不是理论最优解, 则种群陷入局部最优, 算法将出现早熟收敛。要使算法跳出局部极值点进入其他区域进行搜索, 则必须改变 x_{gbest}^g 因此必须加入新的变异算子, 才能改变 x_{gbest}^g 同时因为产生“聚集”现象使得种群多样性变小, 所以在对 x_{gbest}^g 进行变异的同时还需对部分其他个体进行变异, 以提高种群的多样性。综上所述, 当最优个体的适应度大于某一设定的精度 ε 且 δ^2 小于某一设定值 deta 时, 对 x_{gbest}^g 及部分随机选择的个体进行新的变异操作。对于理论最优解未知的情况, 以最优个体的适应度连续多少次不变作为 ε 的设定值。

对于 x_{gbest}^g 及部分随机选择的个体进行新的变异, 本文算法采用增加随机扰动的方法, 设 $x_{gbest,d}^g$ 为 x_{gbest}^g 的第 d 维取值, η 为服从 Gauss(0, 1) 分布的随机变量, 则:

$$x_{gbest,d}^{g+1} = x_{gbest,d}^g (1 + 0.5\eta) \quad (7)$$

其它个体的变异操作与式(1)相同。

2.2 非线性交叉概率因子递增策略

线性交叉概率因子递增策略简单, 且具有较好的寻优性能。因此能广泛应用于各类工程优化问题中。那么, 有没有比它更优的递增策略呢? 本文构造了 3 种交叉概率因子递增策略。试图通过实验计算找到问题的答案。

第 1 种策略为一条开口向下的抛物线, 即:

$$CR1 = (CR_{min} - CR_{max}) \left[\left(\frac{g}{G} \right)^2 - 2 \frac{g}{G} + 1 \right] + CR_{max} \quad (8)$$

采用第 1 种策略的 DE 算法简称为 CR1-DE。第 2 种递增策略是一条指数曲线, 即:

$$CR2 = CR_{max} \left(\frac{CR_{min}}{CR_{max}} \right)^{1/(1+g/G)} \quad (9)$$

采用第 2 种策略的 DE 算法简称为 CR2-DE。第 3 种策略为一条开口向下的抛物线, 即:

$$CR3 = (CR_{max} - CR_{min}) \left(\frac{g}{G} \right)^2 + CR_{min} \quad (10)$$

采用第 3 种策略的 DE 算法简称为 CR3-DE。

为了方便与 CR0-DE 算法作比较, CR_{max} 和 CR_{min} 取为 0.9 和 0.3, G 取 600。当 $g=0$ 时, 式(8)~式(10)都等于初始值 CR_{min} ; 当 $g=G$ 时, 式(8)~式(10)恰好为 CR_{max} 。需要指出的是, 式(9)中

c 取不同的值时会得到不同的效果。本文中取 $c=20$ 。图 1 画出了 4 个递增策略函数随迭代次数的变化曲线。

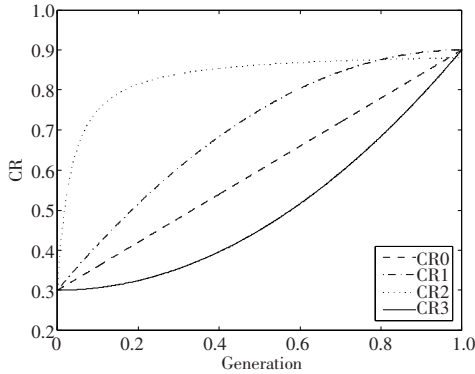


图 1 式(8)的交叉概率曲线

2.3 算法步骤

步骤 1 初始化种群规模 N_p , 差分变异矢量收缩因子 F , 最小交叉概率 CR_{min} , 最大交叉概率 CR_{max} , 适应度方差设定值 δ , 变异适应度设定精度 ε , 令 $g=1$, 按如下公式随机初始化每一个个体:

$$x_{ij} = rand[0, 1](x_j^u - x_j^l) + x_j^l \quad (11)$$

式中: $i=1, 2, \dots, N_p$; $j=1, 2, \dots, D$; x_j^u 为第 j 维变量的上限; x_j^l 为下限。

步骤 2 计算每个个体的适应度 Fitness, 求出最优适应度 Bestfitness 及最优个体 x_{gbest}^g 。

步骤 3 判断最优适应度 Bestfitness 是否达到精度要求或是否达到最优迭代次数。若是则退出; 否则, 执行下一步。

步骤 4 按式(5)和式(6)计算适应度方差 δ^2 。

步骤 5 若 $\delta^2 < \delta$ 且 Bestfitness $> \varepsilon$, 则按式(7)对 x_{gbest}^g 及部分随机选择的个体进行变异; 否则不变异, 转向步骤 6。

步骤 6 对 $x_i^g (i=1, 2, \dots, N_p)$ 执行步骤 7~步骤 9, 生成第 $g+1$ 代种群。

步骤 7 种群中的个体按式(1)进行变异操作, 生成变异个体 x_m 。

步骤 8 按式(2)进行交叉操作, 其中 CR 由式(8)~式(10)确定, 生成实验个体 x_T 。

步骤 9 按式(3)进行选择操作, 生成 $g+1$ 代个体 x_i^{g+1} 。

步骤 10 $g=g+1$, 返回步骤 2。

3 实验分析

下面将通过 4 个典型函数优化问题来测试 CR0-DE、CR1-DE、CR2-DE 和 CR3-DE 算法的寻优性能。其中 $f_1(x)$ 为 Sphere 单峰二次函数; $f_2(x)$ 为 Rosenbrock 函数, 是一个非凸, 病态函数; $f_3(x)$ 为 Griewank 函数, 多峰, 存在大量局部极小点。 $f_4(x)$ 为 Rastrigrin 函数, 是具有强烈震荡的多峰函数, 一般算法难以得到最优解。这 4 个测试函数的全局极小值均为 0。

实验参数设置如下: 函数维数为 10, 种群规模 N_p 设为 60,

最大进化代数为 600, DE 的收缩因子 $F=0.5$, 交叉概率 $CR=0.6$, 适应度方差阈值 $\delta=0.001$, 适应度变异控制精度 $\varepsilon=0.001$ 。

$$f_1(x) = \sum_{i=1}^{10} x_i^2, -100 \leq x_i \leq 100$$

$$f_2(x) = \sum_{i=1}^{10} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], -100 \leq x_i \leq 100$$

$$f_3(x) = \frac{1}{4000} \sum_{i=1}^{10} x_i^2 - \prod_{i=1}^{10} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, -100 \leq x_i \leq 100$$

$$f_4(x) = \sum_{i=1}^{10} (x_i^2 - 10\cos(2\pi x_i) + 10), -10 \leq x_i \leq 10$$

下面 4 个函数优化的迭代曲线图横坐标代表迭代次数, 纵坐标代表适应度的对数(图 2)。

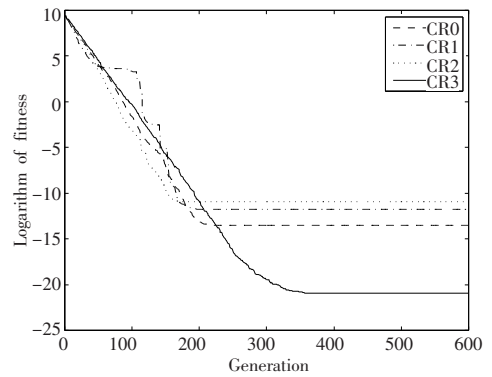


图 2 对 10 维 Sphere 函数优化的迭代曲线

3.1 Sphere 函数算法步骤

采用 4 种(CR0~CR3)递增策略的 DE 算法对 10 维的 Sphere 函数进行 30 次实验。表 1 给出了 30 次实验所得最优解的均值和方差。图 2 画出了 30 次实验平均最优解随迭代次数的变化过程。对于 10 维 Sphere 函数, CR3-DE 寻优性能最强。在迭代 600 次后达到的精度最高。与 CR3-DE 相比, CR1-DE 和 CR2-DE 的优化性能较差, CR2-DE 早期的收敛速度比 CR0-DE 快, 而后期的收敛精度最差, CR1-DE 较 CR0-DE 的精度低。

表 1 对 10 维 Sphere 函数的计算结果

统计项	CR0	CR1	CR2	CR3
均值	3.421×10^{-10}	7.023×10^{-5}	0.009	6.846×10^{-11}
方差	1.517×10^{-10}	3.871×10^{-5}	0.024	5.697×10^{-11}

3.2 Rosenbrock 函数算法步骤

采用 4 种(CR0~CR3)递增策略的 DE 算法对 10 维的 Rosenbrock 函数进行 30 次实验。表 2 给出了 30 次实验所得最优解的均值和方差。图 3 画出了 30 次实验平均最优解随迭代次数的变化过程。由表 2 和图 3 中的结果可知, CR3-DE 的寻优性能最强。在迭代了 600 次后达到的精度最高, CR0-DE 次之, CR1-DE 比 CR0-DE 略差, 而 CR2-DE 最差。

3.3 Griewank 函数

采用 4 种(CR0~CR3)递增策略的 DE 算法对 10 维的 Griewank 函数进行 30 次实验。表 3 给出了 30 次实验所得最优解的均值和方差。图 4 画出了 30 次实验平均最优解随迭代次

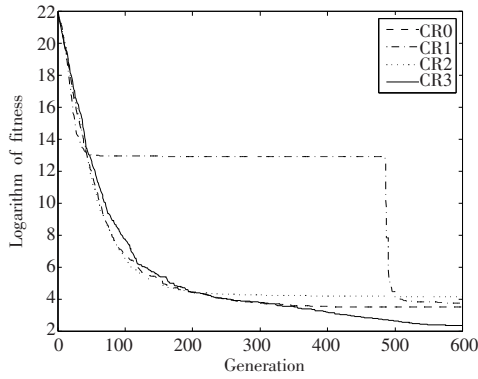


图3 对10维 Rosenbrock 函数优化的迭代曲线

表2 对10维 Rosenbrock 函数的计算结果

统计项	CR0	CR1	CR2	CR3
均值	56.279	30.251	48.028	8.907
方差	174.942	83.143	103.712	27.655

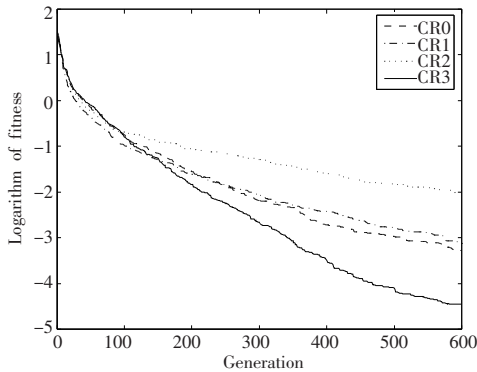


图4 对10维 Griewank 函数优化的迭代曲线

表3 对10维 Griewank 函数的计算结果

统计项	CR0	CR1	CR2	CR3
均值	0.050 6	0.094 1	0.826 0	0.046 9
方差	0.026 4	0.053 3	0.367 0	0.027 2

数的变化过程。由表3和图4的结果可知,CR3-DE经过600次迭代后的精度最高,CR0-DE和CR1-DE迭代600次后的最优解相差不大。与CR3-DE和CR0-DE相比,CR1-DE略差,CR2-DE最差。

3.4 Rastrigrin 函数

采用4种(CR0~CR3)递增策略的DE算法对10维的Griewank函数进行30次实验。表4给出了30次实验所得最优解的均值和方差。图5画出了30次实验平均最优解随迭代次数的变化过程。由表4和图5的结果可知,CR3-DE的收敛速度最快,收敛精度最高。其它3种算法得到的结果接近。CR1-DE的最后得到的精度比CR3-DE差,CR0-DE比CR1-DE略差,CR2-DE最差。

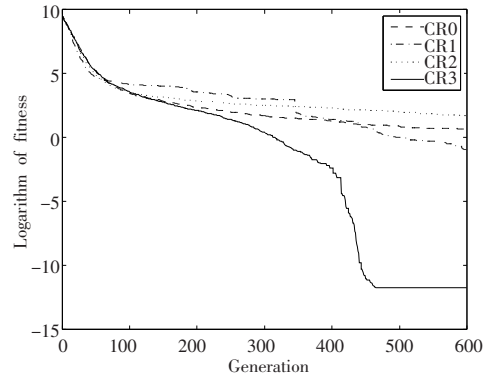


图5 对10维 Rastrigrin 函数优化的迭代曲线

表4 对10维 Rastrigrin 函数的计算结果

统计项	CR0	CR1	CR2	CR3
均值	0.007	0.013	0.015	6.317×10^{-4}
方差	0.004	0.007	0.010	4.028×10^{-4}

4 结束语

本文构造了3种不同于CR0-DE算法的非线性交叉概率因子递增策略。其中,CR1-DE和CR2-DE算法策略是在算法早期通过加速交叉概率因子的递增速度来让算法较快地进入局部搜索,但结果不大理想。而CR3-DE算法是放慢交叉概率因子的递增速度,加强了算法的全局寻优性能,因此算法的性能相对CR0-DE有了不同程度的提高。对于这一点,本文认为DE算法的早期收敛比较慢,所以应让算法进行较长时间的全局搜索,才能获得更好的求解效率。

综上所述,对于多数连续优化问题,凹函数递增策略优于线性策略,而线性优于凸函数策略。

参考文献:

- [1] Price K. Differential evolution: a fast and simple numerical optimizer[C]//1996 Biennial Conf of the North American Fuzzy Information Processing Society, New York, 1996: 524-527.
- [2] Price K V. Differential Evolution vs. the Functions of the second ICEO[C]//IEEE Int Conf on Evolutionary Computation, 1997: 153-157.
- [3] 吴亮红,王耀南,袁小芳,等.自适应二次变异差分进化算法[J].控制与决策,2006,21(8):898-902.
- [4] 陈贵敏,贾建援,韩琪.粒子群优化算法的惯性权重递减策略研究[J].西安交通大学学报,2006,40(1):53-56.
- [5] 何庆元,韩传久.带有扰动项的改进粒子群算法[J].计算机工程与应用,2007,43(7):84-86.
- [6] 李洪亮,侯朝楨,周绍生.一种高效的改进粒子群优化算法[J].计算机工程与应用,2008,44(1):14-16.