一种新的差分进化算法

邓泽喜[1,2],曹敦虔[2],刘晓冀[2],李 娜[3]
DENG Ze-xi[1,2],CAO Dun-qian[2],LIU Xiao-ji[2],LI Na[3]

1.毕节学院 数学系,贵州 毕节 557100
2.广西民族大学 数学与计算机科学学院,南宁 530006
3.德州学院 数学系,山东 德州 253023
1.Department of Mathematics,Bijie College,Bijie,Guizhou 557100,China
2.College of Mathematics and Computer Science,Guangxi University for Nationalities,Nanning 530006,China
3.Department of Mathematics,Dezhou College,Dezhou,Shandong 253023,China
E-mail:liuxiaoji.2003@yahoo.com.cn

DENG Ze-xi,CAO Dun-qian,LIU Xiao-ji,et al.New differential evolution algorithm.Computer Engineering and Applications, 2008,44(24):40-42.

Abstract: For complex functions with high dimensions,a New Differential Evolution algorithm (NDE) based on the variance of the population's fitness is presented.In order to balance global and local search ability,fasten convergence speed,avoid premature,the cross rate is automatically updated according to the generation.The experimental results show that the new algorithm not only has great advantage of convergence, but also can avoid the premature convergence problem effectively.
Key words: differential evolution;premature convergence;cross rate

摘 要:针对高维复杂函数的优化问题,提出了一种新的差分进化算法(NDE)。该算法在运行中根据迭代次数自动地调整交叉概率因子,从而在搜索的初始阶段提高种群多样性,而在搜索后期加强局部搜索能力。对几种经典函数的测试表明,新算法不仅具有很强的全局搜索能力,而且能有效避免早熟收敛问题。
关键词:差分进化;早熟收敛;交叉概率
DOI:10.3778/j.issn.1002-8331.2008.24.010 **文章编号:**1002-8331(2008)24-0040-03 **文献标识码:**A **中图分类号:**TP391

1 引言

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,差分进化算法作为一种性能卓越的优化算法正受到日益关注,其应用领域也越来越广。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[2-3],与其他进化算法一样易陷入局部最优,存在早熟收敛现象。目前的解决方法主要是增加种群的规模,但这样会明显增加算法的运算量,而且也不能从根本上克服早熟收敛的问题。为提高 DE 的性能,许多学者提出了改进的办法。本文以几种典型的 Benchmarks 函数为测试算例, 提出了一种重构交叉概率因子 CR 差分进化算法,协调好全局搜索能力与局部搜索能力,快速收敛,高效避免早熟收敛问题,实验结果表明,解的质量好,算法高效可行。

2 差分进化算法及其早熟收敛问题

2.1 差分进化算法

DE 的基本操作包括变异、交叉、选择三种操作,但与其他进化算法如遗传算法(GA)不同。DE 由 Np(种群大小)个 D 维(变量个数)的参数矢量 $\boldsymbol{x}_i^g$(i=1,2,…,Np)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;然后在父代个体与试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。

2.2 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_{r1}^g,x_{r2}^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[2]。本文采用从种群中随机选

基金项目:广西省自然科学基金(the Natural Science Foundation of Guangxi Province of China under Grant No.06400161,No.0832084);广西民族大学青年科研基金项目资助课题(the Youth Science Research Foundation of Guangxi University for Nationalities under Grant No.0509QN031);广西民族大学研究生教育创新计划(the Graduated Education Innovation Program of Guangxi University for Nationalities under Grant No.gxun-chx0752)。
作者简介:邓泽喜(1982-),男,在读硕士,主要研究领域为人工智能,广义逆理论;曹敦虔(1978-),男,讲师,主要研究领域为计算智能及多元样条函数;刘晓冀(1972-),通讯作者,男,教授,主要研究领域为广义逆理论及其应用。
收稿日期:2007-10-29 **修回日期:**2008-01-02

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:

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

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。

2.3 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,产生试验个体 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,而对于其他维,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:

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

其中,$j=1,2,\cdots,d$,式中:$rand()$ 为一[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。

2.4 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 x_i^g 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^t 作为子代,选择操作的公式为:

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

2.5 差分进化算法的早熟收敛分析

由式(1)可知,x_{gbest}^g 为当前最优解,随着进化的进行,其他个体将迅速向其靠拢。如果 x_{gbest}^g 为一局部最优点,随着种群不断进化,个体之间的差异越来越小,变异矢量 $\boldsymbol{D}_{ab}$ 趋于 0,交叉和选择操作不能改变种群的多样性,最后所有个体都将趋向于 x_{gbest}^g,种群便无法在解空间内重新搜索。因此,算法陷入局部最优,出现了早熟收敛现象。由于差分进化算法是一种随机搜索策略,如果问题是高维多峰函数,则存在多个局部最优点,算法很容易陷入局部最优,难以找到全局最优。即使进行变异的个体不采用 x_{gbest}^g,而是种群中任一其他个体,若某一个体是局部最优解,也同样有陷入局部最优的可能,收敛速度也会变慢。因此,DE 与其他算法一样,容易早熟收敛,陷入局部最优。

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[4]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。

3 一种新的差分进化算法

3.1 自适应二次变异的思想

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 Np,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:

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

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 反映的是种群中所有个体的"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

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

其他个体的变异操作与式(1)相同。

3.2 重构交叉概率因子算法的基本思想

在式(4)中,交叉概率因子 CR 越大,局部搜索能力越强;当 CR 越小全局搜索能力越强。CR 先是最小,后是由小变大,最后为最大。显然 CR 最小最大所经历的时间较短,全局搜索和精细的局部搜索时间不够,用式(4)做 DE 的交叉概率因子去优化复杂的高维函数找不到要求的最优解。

根据上面的分析,越是难优化的函数,越需要加强全局搜索能力,一旦定位到最优解的大概位置,越需要加强局部搜索能力,即加强精细的局部搜索。因此,本文重新构造一个交叉概

率因子,它由三段构成:CR 先保持最小,然后由小变大过渡,最后保持接近 1 的最大,这样,协调好何时进行全局搜索,何时进行局部搜索,以及搜索时的强度,使全局搜索能力与局部搜索能力良好平衡,快速收敛得到最优解。这个重构的交叉概率因子为:

$$CR=CR_{\min}+(CR_{\max}-CR_{\min})\times e^{-30\times(1-\frac{g}{G})^{s}} \tag{8}$$

其中 s 为大于 1 的整数,本文 s 取值为 3,g 为当前迭代次数,G 为最大迭代次数,参数为 $CR_{\min}$=0.25,$CR_{\max}$=0.85。

对图 1 中交叉概率因子曲线而言,式(4)的交叉概率因子直线为式(8)交叉概率因子曲线的特例。因此,用式(8)交叉概率因子曲线来优化高维复杂函数会得到较好的效果。

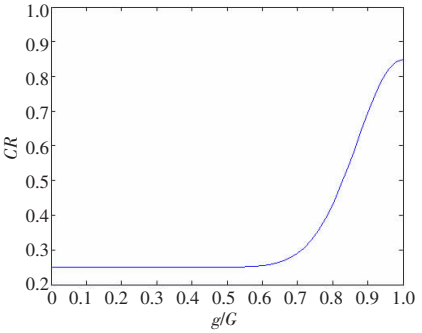

图 1 式(8)的交叉概率曲线

3.3 算法步骤

1.初始化种群规模 Np,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 deta,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:

$$x_{ij}=rand[0,1](x_j^u-x_j^l)+x_j^l \tag{9}$$

式中:i=1,2,…,Np;j=1,2,…,D;x_j^u 为第 j 维变量的上限;x_j^l 为下限。

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。

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。

4.按式(5)和(6)计算适应度方差 δ^2。

5.若 δ^2<deta 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操作;否则不变异,转向 6。

6.对 x_i^g(i=1,2,…,Np)执行 7-9,生成第 g+1 代种群。

7.种群中的个体按式(1)进行变异操作,生成变异个体 x_m。

8.按式(2)进行交叉操作,其中 CR 由式(8)确定,生成试验个体 x_T。

9.按式(3)进行选择操作,生成 g+1 代个体 x_i^{g+1}。

10.$g=g+1$,返回 2。

4 性能分析

下面将通过四个典型函数优化问题来测试本文提出算法的性能,同时与基本的 DE 算法、自适应二次变异差分进化算法(ASMDE)[4]进行比较。其中 $f_1(x)$为 Sphere 单峰二次函数;$f_2(x)$为 Rastrigin 函数,是一个多峰函数,在 $S=\{x_i\in(-5.12,5.12),i=1,2,\cdots,n\}$范围内大约存在 $10n$ 个局部极小点;$f_3(x)$为 Griewank 函数,多峰,存在大量局部极小点;$f_4(x)$是具有强烈震荡的多峰函数,一般算法难以得到最优解。

实验参数设置如下:函数维数为 10,种群规模 Np 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。NDE 算法中,交叉概率 $CR_{\min}$=0.25,$CR_{\max}$=0.85,适应度方差阈值 deta=0.001,适应度变异控制精度 ε=0.001。

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

$$f_2(x)=\sum_{i=1}^{10}(x_i^2-10\cos(2\pi x_i)+10),\ -5.12\leqslant x_i\leqslant 5.12$$

$$f_3(x)=\frac{1}{4\,000}\sum_{i=1}^{10}x_i^2-\prod_{i=1}^{10}\cos\left|\frac{x_i}{\sqrt{i}}\right|+1,\ -30\leqslant x_i\leqslant 30$$

$$f_4(x)=0.5+\frac{\sin\sqrt{x_1^2+x_2^2}-0.5}{[1+0.1\times(x_1^2+x_2^2)]^2},\ -10\leqslant x_i\leqslant 10$$

NDE 算法中对最优个体进行变异的同时还对部分其他个体进行变异,部分个体数目的选择需要由实验确定。个体数目太小,可能对算法没有多大的贡献;个体数目太大时,又会增加计算量。在本文后面的实验中,变异个体数目设为 15。表 1 列出了以上 4 种算法求解上述优化问题运行 20 次得到的函数最优解平均值。由表 1 可知,本文算法的结果明显优于其他两种算法,对于函数 $f_1(x)$、$f_2(x)$、$f_3(x)$、$f_4(x)$每次都能得到精度很高的最优值。同时由标准差可知,本文提出的算法比较稳定。

表 1 三种算法运行 20 次的函数平均最优解

函数	理论最优解	DE	ASMDE	NDE
$f_1(x)$	0	4.531 0E-3	1.078 3E-6	0
$f_2(x)$	0	0.801 6	0.451 9	2.620 0E-9
$f_3(x)$	0	0.324 0	0.086 4	2.768 4E-7
$f_4(x)$	0	0.719 0	0.453 7	7.367 0E-6

图 2~图 5 是上述 3 个函数采用 3 种算法运行 20 次后求得的平均最优适应度进化曲线。图中横坐标表示进化代数,纵坐标表示适应度对数值。由图 2~图 5 可以看出,NDE 对于 $f_1(x)$、$f_2(x)$、$f_3(x)$、$f_4(x)$都能很快收敛到全局最优解;而 DE 都较早出现早熟收敛现象。NDE 在搜索后期结果比 ASMDE 更优,因为 NDE 在后期的局部搜索更精细。

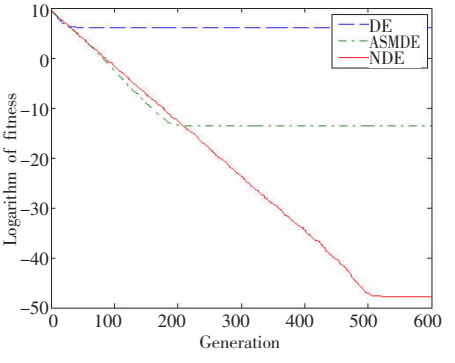

图 2 $f_1(x)$20 次平均最佳适应度进化曲线

5 结束语

本文针对高维复杂函数的优化问题,提出了一种新的差分进化算法。实验表明,NDE 算法不仅具有很强的全局搜索能力,

(下转 52 页)