第35卷 第11期 计 算 机 工 程 2009年6月
Vol.35 No.11 Computer Engineering June 2009

·人工智能及识别技术· 文章编号：1000—3428(2009)11—0216—02 文献标识码：A 中图分类号：TP18

一种新的混沌差分进化算法

谭 跃[1,2]，谭冠政[1]，涂 立[2]

(1. 中南大学信息科学与工程学院，长沙 410083；2. 湖南城市学院物理与电信工程系，益阳 413000)

摘 要：提出一种新的混沌差分进化(CDE)算法，在每一代中通过差分进化(DE)算法找到最佳个体，在最佳个体附近用混沌方法进行局部搜索，通过引入调节因子加强其搜索能力。6个基本测试函数的优化结果表明，当误差函数精度为10^{-14}时，与DE相比，CDE的寻优能力更强、收敛速度较快。

关键词：差分进化；混沌；局部搜索

Novel Chaos Differential Evolution Algorithm

TAN Yue[1,2], TAN Guan-zheng[1], TU Li[2]

(1. School of Information Science and Engineering, Central South University, Changsha 410083;
2. Department of Physics and Telecom Engineering, Hunan City University, Yiyang 413000)

【Abstract】This paper proposes a new Chaos Differential Evolution(CDE) algorithm. It uses Differential Evolution(DE) algorithm to find the best individual each generation, then chaos based local search is executed nearby the best individual. A scaling factor is introduced to enhance the searching ability of CDE. Experimental results on six benchmark functions show the error function value is 10^{-14}, both the ability of finding optimal solution and convergence speed using CDE are better than using DE.

【Key words】Differential Evolution(DE); chaos; local search

差分进化(Differential Evolution, DE)算法已被多次改进，其中，混沌方法是一种重要的改进方法。文献[1]将混沌差分进化用于障碍物环境中的移动机器人路径规划，它利用混沌方法改变DE的控制参数。文献[2]提出一种改进的差分进化算法，利用混沌方法产生初始群体与子群体。本文提出的算法与以往的混沌差分进化算法不同，它将进化算法和局部搜索相结合。

1 经典差分进化算法

DE算法是一种连续空间全局优化启发式算法。它与标准遗传算法相同的是包含选择、交叉和变异3个操作，与标准遗传算法不同的是它采用由变异到交叉，再到选择的操作顺序。

变异操作是从种群中随机选择一个个体作为基向量，以2个不同的个体作为差分向量，变异向量如下：

$$\boldsymbol{v}_i^{G+1}=\boldsymbol{x}_{r1}^G+F(\boldsymbol{x}_{r2}^G-\boldsymbol{x}_{r3}^G) \tag{1}$$

其中，$r_1, r_2, r_3\in\{1, 2, \cdots, N_P\}$，且$r_1\neq r_2\neq r_3$；$N_P$为种群的规模；$F$为缩放因子；$G$为当前种群的代数，$G$+1表示下一代。

交叉操作在变异产生的第i个个体v_i^{G+1}和种群中的第i个个体x_i^G之间进行，交叉操作的试验向量如下：

$$\boldsymbol{u}_{i,j}^{G+1}=\begin{cases}\boldsymbol{v}_{i,j}^{G+1} & \text{if}(randb(j)\leqslant CR)\text{or } j=rnbr(j)\\ \boldsymbol{x}_{i,j}^{G} & \text{if}(randb(j)>CR)\text{or } j\neq rnbr(j)\end{cases} \tag{2}$$

其中，$j\in\{1, 2, \cdots, D\}$，D为问题的维数；CR为交叉因子；$randb(j)$为第j次评价时的一个均匀分布的随机数，其范围是(0, 1)；$rnbr(j)$为$\{1, 2, \cdots, D\}$中随机选择的一个整数。

DE的选择是一对一的，选择操作是在试验向量$\boldsymbol{u}_i^{G+1}$与原种群的个体$\boldsymbol{x}_i^G$之间进行。选择原则是适应度较优的个体进入下一代，即

$$\boldsymbol{x}_i^{G+1}=\begin{cases}\boldsymbol{u}_i^{G+1} & \text{if } f(\boldsymbol{u}_i^{G+1})\leqslant f(\boldsymbol{x}_i^G)\\ \boldsymbol{x}_i^G & \text{else}\end{cases} \tag{3}$$

其中，$\boldsymbol{x}_i^{G+1}$为下一代的第i个个体；f为适应度函数。

2 混沌差分进化算法

2.1 混沌系统

遍历性是混沌系统的特征之一，即系统由某个初始状态开始按其自身的运动规律，在足够长的时间内，遍历相空间中所有状态。利用此特性在最优解附近进行k次遍历，即进行k次无重复的局部搜索。此搜索方式的最大优点是搜索步长按规律自动无重复地变化，这样能有效搜索到当前最优解附近的更优解，上述混沌和DE相结合的方法能增强DE的寻优能力。

混沌系统有多种形式，文献[2]中有如下一种形式：

$$y_{k+1}=\mu y_k(1-y_k) \tag{4}$$

其中，k为迭代次数，k=1, 2, $\cdots$, K，K为最大的迭代次数；μ为控制参数，当μ=4时，$y_1\in(0, 1)$，且$y_1\neq\{0.25, 0.5, 0.75\}$时，式(4)是一个混沌系统。

2.2 混沌差分进化算法的描述

混沌差分进化(Chaos Differential Evolution, CDE)算法中混沌搜索的基本原理是在每一代中用DE搜索到最佳个体X_{best}，在X_{best}的附近再进行k次混沌搜索，得到k个个体，在k个个体中找出适应度值最佳的个体$X_{k\text{best}}$，若$X_{k\text{best}}$适应

基金项目：国家自然科学基金资助项目(50275150)

作者简介：谭 跃(1974－)，男，博士研究生，主研方向：进化计算，机器人；谭冠政，教授、博士生导师；涂 立，讲师

收稿日期：2008-12-19 **E-mail：**tanyue7409@163.com

度值比 X_{best} 好，则用 X_{kbest} 的适应度值取代 X_{best} 的适应度值，并将 X_{kbest} 随机取代种群中的某个个体后返回；若 X_{best} 适应度值比 X_{kbest} 好，则直接返回。在 X_{best} 附近进行混沌搜索，即

$$X_k = X_{best} + \alpha y_k \tag{5}$$

其中，y_k 是式(4)的解；α为调节因子。由式(4)可知，当初值 y_1 为(0, 1)时，式(4)的解全为正解。引入调节因子α可使 X_k 朝正、反 2 个方向变化，α的取值如下：

$$\alpha = \begin{cases} 1 & \text{if } r \geqslant 0.5 \\ -1 & \text{else} \end{cases} \tag{6}$$

其中，r 为(0, 1)中一个均匀分布的随机数。

CDE 算法的流程如下：

(1)初始化种群规模 N_P、缩放因子 F、交叉因子 CR、种群最大进化代数 G_{max}、输入变量的上界 $H_{i,j}$ 和下界 $L_{i,j}$，$i\in\{1, 2, \cdots, N_P\}, j\in\{1, 2, \cdots, D\}$、混沌控制参数$\mu$=4、混沌搜索次数 k。

(2)根据式(7)产生初始种群 P^0

$$P_{i,j}^0 = L_{i,j} + rand(1)(H_{i,j} - L_{i,j}) \tag{7}$$

其中，$i\in\{1, 2, \cdots, N_P\}; j\in\{1, 2, \cdots, D\}$；$rand(1)$为(0, 1)中一个均匀分布的随机数。

(3)对初始种群进行评价。找出适应度值最佳的个体 X_{best}，记录 X_{best} 的适应度值 F_{best} 及其在种群中的索引 $index(X_{best})$，置进化代数 G=1。

(4)由式(1)得变异向量 $\boldsymbol{v}_i^{G+1}$。

(5)由式(2)得试验向量 $\boldsymbol{u}_i^{G+1}$。

(6)由式(3)得下一代个体 x_i^{G+1}。

(7)比较 $f(\boldsymbol{u}_i^{G+1})$ 和 F_{best}，若 $f(\boldsymbol{u}_i^{G+1})$ 比 F_{best} 好，则 $X_{best} = \boldsymbol{u}_i^{G+1}$，$F_{best} = f(\boldsymbol{u}_i^{G+1})$，$index(X_{best})=i$，进入(8)；否则继续。

(8)重复执行(4)~(7)N_P 次。

(9)$y_1=rand$，$rand$ 为(0, 1)中一个均匀分布的随机数，且 $y_1\neq\{0.25, 0.5, 0.75\}$。

(10)由式(4)~式(6)计算得 X_k。

(11)重复执行(10)k 次。

(12)找出 X_k 中适应度值最好的个体 X_{kbest}，若 $f(X_{kbest})$比 F_{best} 好，则 $X_{best}=X_{kbest}$，$F_{best}=f(X_{kbest})$，并产生一个随机整数 $j\in[1, N_P]$，$index(X_{best})=j$，$x_j^{G+1} = X_{kbest}$，进入(13)；否则继续。

(13)$G=G+1$，若 $G<G_{max}$，则进入(4)；否则继续。

(14)输出 X_{best} 和 F_{best}。

3 仿真实验

从文献[3]中选择 6 个基本的测试函数作为测试对象，它们包括单峰独立、单峰非独立、多峰独立和多峰非独立，具体描述如下：

(1) $f_1(x) = \sum_{i=1}^{30} x_i^2$，$|x_i| \leqslant 100, \min(f_1) = f_1(0,0,\cdots, 0) = 0$

(2) $f_2(x) = \sum_{i=1}^{30} (\sum_j^i x_j)^2, |x_i| \leqslant 100, \min(f_2) = f_2(0,0,\cdots,0) = 0$

(3) $f_3(x) = \sum_{i=1}^{29} \left|100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right|$,

$|x_i| \leqslant 30, \min(f_4) = f_4(1,1,\cdots,1) = 0$

(4) $f_4(x) = \sum_{i=1}^{30} [x_i^2 - 10\cos(2\pi x_i) + 10]$,

$|x_i| \leqslant 5.12, \min(f_6) = f_6(0,0,\cdots,0) = 0$

(5) $f_5(x) = -20\exp(-0.2\sqrt{1/30\sum_i^{30} x_i^2}) - \exp(1/30\sum_{i=1}^{30}\cos(2\pi x_i)) + 20 + e$,

$|x_i| \leqslant 32, \min(f_7) = f_7(0,0,\cdots,0) = 0$

(6) $f_6(x) = 1/4\ 000\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30}\cos(x_i/\sqrt{i}) + 1$,

$|x_i| \leqslant 600, \min(f_8) = f_8(0,0,\cdots,0) = 0$

为使 DE 达到更好的优化，选择 3 组有代表性的参数作为缩放因子和交叉因子，每个测试函数的进化代数根据实验结果动态调整，最大进化代数不超过 1 500 代，记录 DE 在不同参数下的最优值。选择最优的一组 F 和 CR 值作为 DE 和 CDE 的最终实验参数，最大进化代数作为 DE 和 CDE 的停止条件，以多次实验的最大值、最小值、平均值、收敛速度作为评价标准比较 CDE 和 DE 的优化效果。所有实验均采用 DE/*rand*/1/*bin* 的形式。

从现有的文献中，选择 3 组数据 F_1=0.2, CR_1=0.1; F_2=0.5, CR_2=0.5; F_3=0.5, CR_3=0.9 作为缩放因子和交叉因子，种群规模为 60。在实验过程中，进化代数动态调整，得到 f_1 的最大进化代数为 1 000 代，其他函数为 1 500 代。每个测试函数在 DE 作用下单独优化 5 次，得到最优值的平均结果如表 1 所示。

表 1 DE 对 6 个测试函数优化 5 次的最优解平均值

函数	F_1=0.2 CR_1=0.1	F_2=0.5 CR_2=0.5	F_3=0.5 CR_3=0.9
f_1	6.987 9e-19	3.953 3e-11	9.239 4e-14
f_2	5.714 0e+3	1.392 0e+4	0.113 0
f_3	42.085 6	23.359 9	22.463 0
f_4	0.204 7	129.928 7	152.220 6
f_5	4.085 6e-15	2.863 6e-10	2.660 1e-12
f_6	0.360 9	0.951 0	0.993 3

由表 1 可知，f_1, f_4, f_5, f_6 的 F=0.2, CR=0.1; f_2, f_3 的 F=0.5, CR=0.9，每一代混沌搜索 30 次。根据上述参数，用 DE 和 CDE 分别对每个测试函数优化 25 次，得到结果如表 2 所示，其中，粗体为最好结果，由此可见，一般情况下 CDE 比 DE 好。

表 2 DE 和 CDE 对 6 个测试函数优化 25 次的结果

函数	算法	最小值	最大值	平均值
f_1	DE	1.555 3e-19	7.437 0e-19	4.007 1e-19
	CDE	**2.529 6e-28**	**3.063 2e-26**	**8.202 1e-27**
f_2	DE	0.035 5	0.290 6	0.136 3
	CDE	**1.414 7e-21**	**2.881 9e-18**	**3.200 5e-19**
f_3	DE	**19.354 6**	24.619 8	22.243 1
	CDE	20.309 6	**23.596 9**	**21.903 4**
f_4	DE	**0**	0.995 0	0.199 0
	CDE	**0**	**0**	**0**
f_5	DE	**2.664 5e-15**	6.217 2e-15	4.440 9e-15
	CDE	**2.664 5e-15**	**2.664 5e-15**	**2.664 5e-15**
f_6	DE	0.316 3	0.407 3	0.366 6
	CDE	**0**	**0**	**0**

为比较 CDE 和 DE 的收敛速度，对 25 次测试结果进行排序，选择第 13 次的测试结果作为收敛结果[3]，其收敛曲线如图 1~图 6 所示。

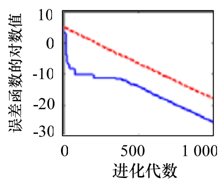

图 1 f_1 收敛曲线

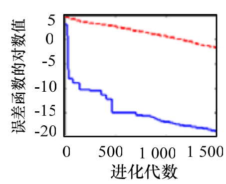

图 2 f_2 收敛曲线

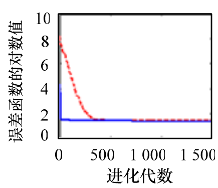

图 3 f_3 收敛曲线

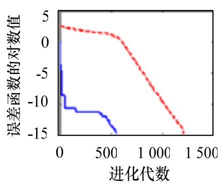

图 4 f_4 收敛曲线

其中，纵坐标为误差函数 $f(x)$- $f(x^*)$的对数值；$f(x^*)$为全局最优值；$f(x)$为算法找到的最优值；虚线和实线分别表示 DE 和 CDE 的优化曲线。在图中一些曲线在达到最大代数前

（下转第 220 页）