

改进的自适应混沌差分进化算法^①

王涛, 王焕

(辽宁工程技术大学, 电气与控制工程学院, 葫芦岛 125105)

摘要: 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, 提出一种基于帐篷映射(Tent)的自适应混沌嵌入式差分进化算法(CLSDE). 算法采用 Tent 映射生成的混沌序列来取代基本 DE 算法选择操作中的随机数, 充分利用了混沌运动的随机性、遍历性和规律性. 通过与混沌 PSO 算法与普通的 DE 算法比较, 测试函数仿真结果表明, 该算法具有良好的全局搜索能力, 寻优精度较高, 收敛速度快, 鲁棒性好.

关键词: 混沌; 自适应; 差分进化; 帐篷映射; 适应度函数

Improved Adaptive Chaotic Differential Evolution Algorithm

WANG Tao, WANG Huan

(College of Electrical and Control Engineering, Liaoning Technical University, Huludao 125105, China)

Abstract: In order to improve the differential evolution algorithm for optimum speed and overcome the heuristic algorithm common premature convergence problem is proposed based on a Tent mapping (Tent) of adaptive chaotic embedded differential evolution algorithm (CLSDE). Tent mapping algorithm using of the generation of chaotic sequence to replace basic DE algorithm to select the operation of the random, and make full use of the chaotic motions of the randomness, ergodicity and regularity, so it can smooth out the problem of local precocious and rapid speed up its global convergence speed. Through the and chaos PSO algorithm and the ordinary DE comparison algorithm, the simulation results show that the new algorithm in the solution precision, stability and convergence borrows and good performance.

Key words: chaos; adaptive; differential evolution; tent mapping; fitness function

差分进化算法(Differential Evolution, DE)是属于遗传算法的一个分支, 它是由 Storn 等人于 1995 年提出的, 最初的设想是用于解决切比雪夫多项式问题, 后来发现 DE 也是解决复杂优化问题的有效技术. 但是差分进化算法的优化性能受控制参数取值和差分进化类型的影响较大, 算法容易早熟收敛的问题, 随迭代次数的增加种群多样性会快速的下降容易陷入局部最优.

许多研究者对基于混沌搜索的进化算法进行了研究, 文献[1]提出了一种混沌粒子群优化算法, 而在文献[2]中将混沌变量的搜索机制引入到遗传算法、粒子群算法、差分进化算法等等. 文献[3]中提出了一种自

适应的差分进化算法(DE), 引入差分算法的交叉因子和变异因子对目标函数的自适应取值, 算法收敛度较高. 本文基于混沌映射的优点, 将帐篷映射嵌入到差分进化算法中, 提出了一种改进的嵌入混沌搜索的自适应差分进化. 克服了差分算法早熟陷入局部最优的问题, 加快了差分算法的寻优速度, 该算法能够有效跳出局部极小, 并且其搜索结果的好坏不依赖于初始点的选择, 而且经过仿真实验算法收敛度高.

1 算法基本原理

1.1 差分进化基本原理差分进化基本原理

一般标准的 DE 算法的流程如下:

^① 基金项目:辽宁省高校优秀人才项(2008RC25); 辽宁省创新团队项目(LT2010047)

收稿时间:2012-08-04;收到修改稿时间:2012-09-23

Step1. 首先按照公式(1)随机生成初始种群.

$$X_{ij}(0) = X_{\min,j}(0) + \text{rand} \cdot (X_{\max,j} - X_{\min,j}) \quad (1)$$

Step2. 变异操作. 产生变异中间个体公式如式(2)所示:

$$V_{ij}(t+1) = X_{r1,j}(t) + F \cdot (X_{r2,j}(t) - X_{r3,j}(t)) \quad (2)$$

其中, $r1, r2, r3 \in [1, 2, \dots, N]$ 为正整数, 且满足 $r1 \neq r2 \neq r3 \neq i$, 变异因子 $F \in [0, 1]$ 用来控制差分向量 $X_{r2,j}(t) - X_{r3,j}(t)$ 的幅度.

Step3 交叉操作. 将中间变异个体 V_{ij} 和目标个体 X_{ij} 按(3)进行交叉, 生成规则如下:

$$u_{ij}(t+1) = \begin{cases} v_{ij}(t+1), & \text{if } \text{rand} < p_c, \text{ or } j = j_{\text{rand}} \\ X_{ij}(t) & \text{else} \end{cases} \quad (3)$$

其中, rand 是 $[0, 1]$ 间均匀分布的随机数, P_c 为交叉因子, J_{rand} 是 $[1, D]$ 间的随机整数. X_{ij} , V_{ij} , U_{ij} , 分别是第 i 目标个体, 变异个体和试验个体的第 j 维分量.

Step4. 选择操作. 利用适应度函数检验交叉变异产生的新的实验个体 U_{ij} , 然后根据式(4)决定是否在下一代中使用试验个体替换当前目标个体.

$$u_{ij}(t+1) = \begin{cases} u_i(t+1), & \text{if } f(u_i(t+1)) < f(x_i(t)) \\ X_i(t) & \text{else} \end{cases} \quad (4)$$

Step5. 如果拼 gen 或者满足系统误差要求, 则输出 g_{best} , 及其目标值并停止计算, 否则 $t = t + 1$, 返回到步骤 2.

本文提到的基本的 DE 算法可表示为 DE/rand/bin, 即采用随机选取的基点向量, 拥有 1 个差分向量, 并使用 bin 交叉方式的 DE 优化策略, 混沌 DE 算法都是采用这种方式. 同理根据不同变异操作策略也存在着多种的变异方式.

2.1 混沌算法的基本原理

帐篷映射的具体步骤和公式如下:

首先在将混沌空间映射到优化问题的解空间中, 表达式方程为:

$$x^{k+1} = \mu \times x^k (1-x^k), k=0,1,2,\dots,n \quad 0 \leq x^k \leq 1 \quad (5)$$

式中, μ 为一个可以决定混沌变量 x 稳定与否的控制参数.

其次在利用帐篷映射的随机分布特性对优化问题解空间的搜索中, 其数学表达式为:

$$x^{k+1} = \begin{cases} 2x^k, & 0 \leq x^k \leq 0.5 \\ 2(1-x^k), & 0.5 < x^k \leq 1 \end{cases} \quad (6)$$

其中, k 为迭代步数, $k=0,1,2,\dots,n$.

混沌局部搜索的步骤:

1. 将初始决策变量 X_{jk} 按照式(7)映射到 $0 \sim 1$ 之间的混沌变量 cx_j^k

$$cx_j^k = \frac{x_j^k - x_{\min,j}}{x_{\max,j} - x_{\min,j}}, \quad j = 1, 2, \dots, n \quad (7)$$

式中 $x_{\max,j}$ 和 $x_{\min,j}$ 分别为第 j 维变量的搜索上下界.

2. 根据式(6)计算得到下步迭代混沌变量 cx_j^{k+1} , 并将其按式(8)转化为新的决策变量 x_j^{k+1}

$$x_j^{k+1} = x_{\min,j} + cx_j^{k+1} (x_{\max,j} - x_{\min,j}) \quad (8)$$

3. 根据决策变量 x_j^{k+1} , 计算新解的适应度值.

4. 若新解达到混沌搜索最大步数 c_{\max} , 则将新解作为 CLS 的搜索结果, 否则返回步 2.

2 算法改进

本文对文献[3]中提出的 ACDE(adaptive chaotic differential evolution algorithm)算法进行改进, 采用了 $[0,1]$ 区间分布更为均匀的 Tent 映射, 改进了 DE 算法中交叉因子及变异因子的产生过程, 下面将混沌自适应 DE 算法的步骤及流程给出:

步骤 1. 初始化 DE 种群, 设初始进化代数 $t = 0$, 最大迭代代数 T_{\max} . 采用公式(1-1)随机生成初始种群.

步骤 2. 变异操作, 采用公式(2)生成变异中间个体.

步骤 3. 交叉操作, 采用公式(3)生成交叉试验个体.

本文采用文献[3]中提出的自适应交叉因子策略, 将交叉因子 P_c 与最大迭代代数以及当代代数 t 相关, 以增强种群的多样性. 自适应交叉因子 P_c 的表达式如下:

$$P_c = P_{cl} \times 2^{\left(1 - \frac{T_{\max}}{t+1}\right)} \quad (9)$$

式中, P_{cl} 是初始化交叉因子, 取值同样满足 $[0, 1]$ 间的概率约束.

步骤 4. 选择操作, 为进一步加快 DE 算法的全局收敛速度, 引入混沌随机序列, 在每代个体和当前最

优个体 X_{best} 之间的解空间内进行混沌搜索, 可以发现适应度值优于个体 X_i , 却不如最优个体的 X_{best} 新个体 r_i 和 c_i , 第 i 个目标个体的第 j 维新试验个体 $r_{i,j}$ 和 $c_{i,j}$, 并通过公式(10)、(11)产生。

$$r_{i,j}(t) = \eta_j(t)x_{i,j}(t) + (1 - \eta_j(t))x_{best,j}(t) \quad (10)$$

$$c_{i,j}(t) = \eta_j(t)x_{best,j}(t) + (1 - \eta_j(t))x_{i,j}(t) \quad (11)$$

式中, η_j 为 [0, 1] 间均匀分布的 tent 混沌映射, 由公式(5)求得。

步骤 4.1. 对经过交叉变异操作后产生的新的试验个体的适应度函数值 $f(u_i(t))$ 进行计算比较, 如果 $f(u_i(t+1)) < f(x_i(t))$, 则 $x_i(t+1) = u_i(t+1)$, 并转到步骤 5; 如果 $f(u_i(t+1)) > f(x_{best}(t))$, 则转入到步骤 4.2; 若 $f(u_i(t+1)) > f(x_{best}(t))$, 则转入到步骤 4.4 继续。

步骤 4.2. 计算新个体 $f(r_i(t))$ 的值, 如果 $f(u_i(t+1)) > f(x_{best}(t))$, 则令 $x_i(t+1) = u_i(t)$ 转到步骤 5, 反之转到步骤 4.3。

步骤 4.3. 计算新个体 $f(c_i(t))$ 的值, 如果 $f(u_i(t+1)) < f(c_i(t))$, 则令 $x_i(t+1) = c_i(t)$ 转到步骤 5 反之转到步骤 4.4。

步骤 4.4. 令 $x_i(t+1) = x_i(t)$, 进入步骤 5。

步骤 5. 若终止准则满足 $t = T_{max}$, 则输出最优个体及其目标值并停止算法, 否则 $t = t + 1$ 并转到步骤 2 继续。

3 仿真实验

为了测试基于帐篷映射的混沌差分进化算法 (CLSDE) 的有效性, 将其与常规的差分进化算法 (DE)、混沌粒子群优化算法 (PSO)

进行比较下面将用三个标准测试函数来测试上述两个算法的性能, 这些标准函数如下:

(1) Sphere 函数是一个简单的单峰的、自变量之间相互独立的函数,

$$f_1(x) = \sum_{i=1}^{30} x_i^2, |x_i| \leq 100$$

(2) Rosenbrock 函数是一个单峰的、自变量之间互相影响的函数, 比 Sphere 增加了寻有复杂度。

$$f_2(x) = \sum_{i=0}^{29} |100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2|, |x_i| \leq 30$$

(3) Rastrigrin 函数是多峰的、自变量之间相互独立的函数, 且他的局部最优解随着正弦波动。

$$f_3 = \sum_{i=1}^{30} [x_i^2 - 10\cos(2\pi x_i) + 10], |x_i| \leq 5.12$$

这些标准函数理论最小值都是 $\min f(x) = 0$, 在优化计算的过程中, 我们主要通过收敛速度, 收敛率, 平均值, 解的稳定性, 对优化算法的性能进行评价。

实验的环境为 MATLABR2010a 开发环境, 种群的个体总数设置为 50, 最大迭代次数为 1000, 最大的函数求值次数设置为 30000. 对优化算法 DE/best/1/bin(简称为 DE)和文献[3]提出的混沌粒子群嵌入优化算法(ACEPSO)和 CLSDE 进行比较, 其中缩放因子 $F=0.5$,交叉因子 $P_c=0.9$,混沌搜索次数 $M=30$. 以 25 次实验的收敛速度, 收敛率, 解的平均值, 解的稳定性作为比较的评价标准,得到相关函数的寻优实验结果如下。

实验结果及分析:

由三图结果显示, 与普通的 DE 算法和 ACEPSO 相比, 基于帐篷映射的混沌差分进化算法的最优解的精度最高, 收敛速度快, 而且稳定性也最好, 其次是 ACEPSO 算法, 再次是常规的 DE 算法收敛曲线:

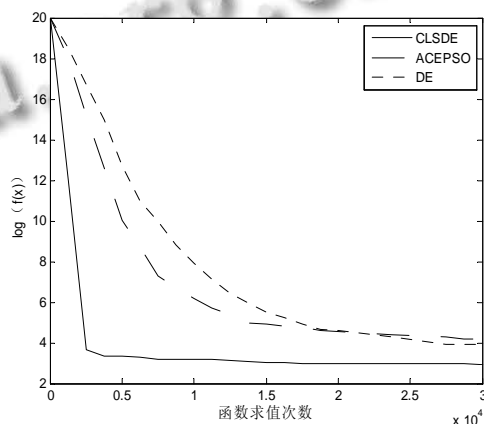


图 1 sphere 迭代寻优曲线

由以上三个函数迭代寻优曲线可以看出, CLSDE 算法在寻优过程中在避免陷入局部最优的前体下, 能够迅速正确的搜索到最优解, 且具有较高的精确度, 较快的收敛速度. 因此改进的基于帐篷映射的自适应

差分进化算法在全局寻优能力,收敛速度,获得解的精度,和解的稳定性等方面的综合结果比其他两种算法有明显优势.

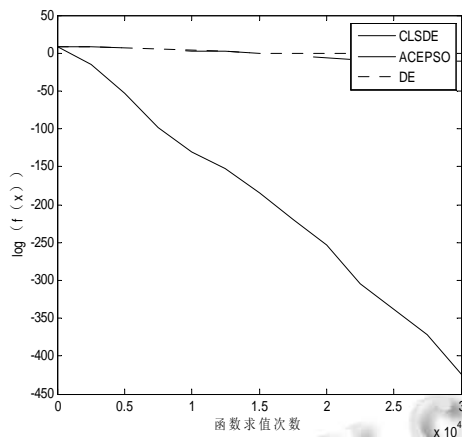


图 2 Rosenbrock 函数迭代寻优曲线:

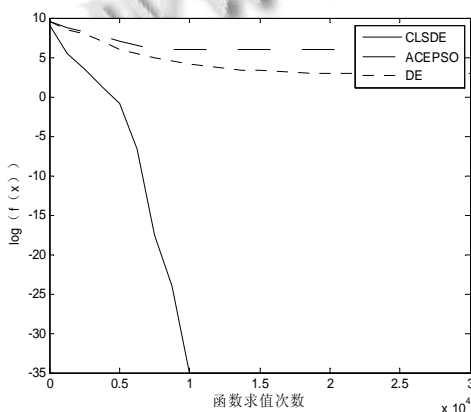


图 3 Rastrigrin 函数的迭代寻优曲线:

5 结论

差分进化算法作为遗传算法的分支,它集中了遗传算法和粒子群算法的优点,具有算法简单和较强的全局收敛能力和鲁棒性,本文又结合了帐篷映射对其

选择操作过程进行了改进,由实验证明,改进的算法确实增加了其收敛速度和稳定性.

参考文献

- 1 魏玉琴,戴永寿,等.基于 Tent 映射的自适应混沌嵌入式粒子群算法.计算机工程与应用,2012.
- 2 贾东立.改进的差分进化算法及其在通信信号处理中的应用研究.上海:上海大学.
- 3 Lu YL, Zhou JZ, Qin H, et al. An adaptive chaotic differential evolution for the short-term hydrothermal generation scheduling problem. Energy Conversion and Management, 2010, 51: 1481-1490.
- 4 单梁,强浩,李军,王执铨.基于 Tent 映射的混沌优化算法.控制与决策,2005,20(2):179-182.
- 5 Noman N, Iba H. Accelerating Differential Evolution Using an Adaptive Local Search. IEEE Trans. on Evolutionary Computation, 2008,12(1):107-125.
- 6 杨俊杰,周建中,喻菁,等.基于混沌搜索的粒子群优化算法.计算机工程与应用,2005,41(16):69-71.
- 7 贺徽,周建中,等.基于帐篷映射的混沌自适应粒子群优化算法在同步发电机励磁控制中的应用.电网技术,2011,35(6):45-49.
- 8 常俊林,李亚朋,等.基于改进差分进化算法的 PID 优化设计.控制工程,2010,17(6):807-810.
- 9 谭跃,谭冠正,等.一种新的混沌差分进化算法.计算机工程,2009,35(11):216-220.
- 10 银建霞,孟红云.具有混沌差分进化搜索的人工蜂群算法.计算机工程与应用,2011,47(29):27-30.
- 11 Price K, Storn R, LamPinen J. Differential Evolution A Practical Approach to Global Optimization, Berlin: Springer, 2005.