

文章编号: 1001-0920(2006)08-0898-05

自适应二次变异差分进化算法

吴亮红^{1,2}, 王耀南¹, 袁小芳¹, 周少武²

(1. 湖南大学 电气与信息工程学院, 长沙 410082; 2. 湖南科技大学 信息与电气工程学院, 湖南 湘潭 411201)

摘 要: 提出一种基于群体适应度方差自适应二次变异的差分进化算法。该算法在运行过程中根据群体适应度方差的大小, 增加一种新的变异算子对最优个体和部分其他个体同时进行变异操作, 以提高种群多样性, 增强差分进化算法跳出局部最优解的能力。对几种典型 Benchmarks 函数进行了测试, 实验结果表明, 该方法能有效避免早熟收敛, 显著提高算法的全局搜索能力。

关键词: 差分进化; 自适应二次变异; 时变概率; 早熟收敛

中图分类号: TP391

文献标识码: A

Differential Evolution Algorithm with Adaptive Second Mutation

WU Liang-hong^{1,2}, WANG Yao-nan¹, YUAN Xiao-fang¹, ZHOU Shao-wu²

(1. School of Electric and Information Engineering, Hu'nan University, Changsha 410082, China; 2. School of Information and Electric Engineering, Hu'nan University of Science and Technology, Xiangtan 411201, China
Correspondent: WU Liang-hong, E-mail: lhwu@hnust.edu.cn)

Abstract: A new adaptive second mutation differential evolution algorithm (ASMDE) based on the variance of the population's fitness is presented. In order to improve the population's diversity and the ability of breaking away from the local optimum, according to the value of the variance of the population's fitness during the running time, a new mutation operator is adapted to mutate both the best individual and partial other individuals. Several classic Benchmarks functions are tested and the results show that the proposed algorithm can avoid the premature convergence and improves the global convergence ability remarkably.

Key words: Differential evolution; Adaptive second mutation; Time varying probability; Premature convergence

1 引言

差分进化(Differential evolution, DE)算法是一种采用浮点矢量编码在连续空间中进行随机搜索的优化算法^[1]。DE的原理简单, 受控参数少, 实施随机、并行、直接的全局搜索, 易于理解和实现。在日本召开的第一届国际进化优化计算竞赛(ICEO)中^[2], DE表现突出, 已成为进化算法(EA)的一个重要分支。近年来, DE在约束优化计算, 模糊控制器优化设计, 神经网络优化, 滤波器设计等方面得到了广泛的应用。

DE是根据父代个体间的差分矢量进行变异、交叉和选择操作, 与其他进化算法(如遗传算法)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, 很多学者提出了改进的方法。文献[3]在算法中加入了迁移算子和加速算子, 提高了算法的种群多样性和收敛速率, 但要计算适应度函数的梯度信息, 实现较为麻烦, 应用会受到限制。文献[4]针对差分矢量的缩放因子 F 和交叉概率 CR 两参数对算法的影响, 提出了一种模糊自适应差分进化算法, 但要选择模糊隶属函数, 实现较为困难。文献[5]将缩放因子 F 由固定数值设计为随机函数, 减少了需调整的参数, 实现了一个简化的DE版

收稿日期: 2005-06-11; 修回日期: 2005-09-21

基金项目: 国家自然科学基金项目(60375001); 高校博士点基金项目(20030532004)。

作者简介: 吴亮红(1978—), 男, 湖南宁乡人, 硕士生, 从事智能控制、计算智能等研究; 王耀南(1957—), 男, 昆明人, 教授, 博士生导师, 从事智能控制、智能信息处理、智能图像处理等研究。

本, 但并不能解决早熟收敛问题

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。为克服早熟现象, 跳出局部最优解, 本文提出了一种基于群体适应度方差自适应二次变异的差分进化算法 (ASMDE)。对几种典型的 Benchmarks 函数测试表明, 与遗传算法和进化算法相比, 本文提出的算法能有效地避免早熟收敛, 全局收敛能力得到了显著提高

2 差分进化算法及其早熟收敛问题

2.1 差分进化算法

DE 与其他进化算法如遗传算法 (GA), 进化规划 (EP), 进化策略 (ES) 及其它们的变种不同。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的大小进行选择操作, 选择适应度更优的个体作为子代

2.1.1 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 (第 g 代) 群体中两个不同的个体 (x_a^g, x_b^g) 。差异矢量定义为

$$D_{ab} = x_a^g - x_b^g \tag{1}$$

根据变异个体的生成方法不同, 形成了多种不同的差分进化算法方案^[1,6]。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。个体变异操作的方程为

$$x_m = x_{gbest}^g + F[(x_a^g - x_b^g) + (x_c^g - x_d^g)] \tag{2}$$

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 为与 x_{gbest}^g 不同的 4 个互不相同的个体; F 为缩放因子, 其取值范围为 $(0, 1.2]$; x_m 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

2.1.2 交叉操作

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 至少有一位由 x_m 贡献, 而对于其他位, 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

$$x_{Tj} = \begin{cases} x_{mj}, & \text{rand}() < CR; \\ x_{ij}^g, & \text{rand}() > CR; \end{cases} \tag{3}$$

$$j = 1, 2, \dots, D.$$

式中: $\text{rand}()$ 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 (3) 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

2.1.3 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 与 x_i^g 进行竞争。只有当 x_T 的适应度较 x_i^g 更优时才被选作子代; 否则, 直接将 x_i^g 作为子代。选择操作的方程为

$$x_{i+1}^g = \begin{cases} x_T, & f(x_T) < f(x_i^g); \\ x_i^g, & f(x_T) \geq f(x_i^g). \end{cases} \tag{4}$$

2.2 差分进化算法的早熟收敛分析

由于 DE 采用贪婪的搜索策略, 可以加快收敛速度, 但会大大提高早熟收敛的概率。由式 (2) 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, 随着种群的不断进化, 个体之间的差异越来越小, 变异矢量 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, 出现了所谓的早熟收敛现象。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, 也同样有陷入该局部最优的可能, 而且收敛速度会变慢。因此, DE 与其他进化算法 (如 GA) 一样, 容易早熟收敛, 陷入局部最优

3 自适应二次变异差分进化算法设计

3.1 自适应二次变异思想

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, 形成了“聚集”现象。这表现在种群个体的适应度之间的差异越来越小。为定量描述种群的状态, 下面给出群体适应度方差的定义^[7]。

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 可定义为

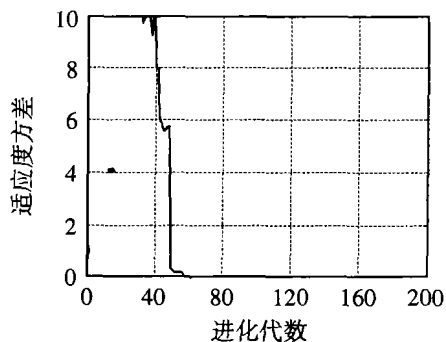
$$\delta^2 = \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \frac{f_i - f_{avg}}{f} \right|^2 \tag{5}$$

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

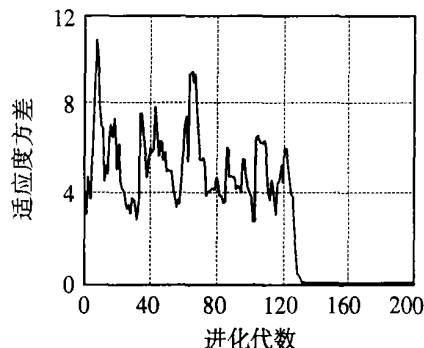
$$f = \begin{cases} \max\{|f_i - f_{avg}|\}, & \max\{|f_i - f_{avg}|\} > 1; \\ 1, & \text{others} \end{cases} \tag{6}$$

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

由2.2节分析可知, 当 $x_{g\text{best}}$ 为局部最优或全局最优点时, σ^2 很小, 其他个体都聚集在 $x_{g\text{best}}$ 的周围, 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 $x_{g\text{best}}$ (变异操作), 在其后的搜索过程中发现新的 $x_{g\text{best}}$, 如此循环, 算法就可以找到全局最优点. 但陷入局部最优点时, 由于 D_{ab} 趋于0, 算法本身的变异不能改变 $x_{g\text{best}}$, 必须加入新的变异算子. 同时, 由于产生了聚集现象, 种群多样性很小, 不利于算法的进一步搜索. 因此, 在对 $x_{g\text{best}}$ 进行变异的同时还应对部分其他个体进行变异, 以提高种群的多样性, 进而提高算法的全局搜索能力和稳定性. 根据以上分析, 当 σ^2 小于某一设定值 deta 时, 对 $x_{g\text{best}}$ 和部分随机选择的个体同时进行新的变异操作; 当 σ^2 大于 deta 时则不进行新的变异. 但若 $x_{g\text{best}}$ 为全局最优点, 即使 σ^2 小于 deta , 也不应进行新的变异, 进行一些无用的操作. 因此, 只有当最优个体的适应度大于某一设定的精度 ϵ (还没有达到全局最优) 且 σ^2 小于 deta 时才进行新的变异操作. 对于理论最优解未知的情况, 以最优个体的适应值连续多少次不变作为 ϵ 的设定值. 这就是本文提出的自适应二次变异的基本思想.



(a) DE 早熟收敛的适应度方差进化曲线



(b) DE 全局收敛的适应度方差进化曲线

图1 适应度方差进化曲线

图1(a)为差分进化算法求解某函数最小化问题时早熟收敛的适应度方差进化曲线, 图1(b)则为全局收敛时的适应度方差进化曲线

对于 $x_{g\text{best}}$ 和其他部分个体变异的操作, 本文算法将采用增加随机扰动的方法. 设 $x_{g\text{best}, d}$ 为 $x_{g\text{best}}$ 的第 d 维取值, η 为服从 Gauss(0, 1) 分布的随机变量, 则

$$x_{g\text{best}, d} = x_{g\text{best}, d} (1 + 0.5\eta). \quad (7)$$

其他个体的变异操作与式(7)相同

3.2 时变交叉概率因子CR思想

由2.1.2节交叉操作分析可知, CR 越大, x_m 对 x_T 的贡献越多, 有利于局部搜索和加速收敛速率; 反之, 则 x_i 对 x_T 的贡献越多, 有利于保持种群的多样性和全局搜索. 良好的搜索策略应该是在搜索的初始阶段保持种群多样性, 进行全局搜索, 而在搜索的后期应加强局部搜索能力, 以提高算法的精度. 基于这种思想, 本文提出了一种时变交叉概率因子CR的方法, 即CR随迭代次数的增加而由小变大, 初始阶段 x_i 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. 设 CR_{\min} 为最小交叉概率, CR_{\max} 为最大交叉概率, Iterations 为当前迭代次数, Maxiterations 为最大迭代次数, 则

$$\text{CR} = \text{CR}_{\min} + \frac{\text{Iterations}(\text{CR}_{\max} - \text{CR}_{\min})}{\text{Maxiterations}}. \quad (8)$$

3.3 自适应二次差分进化算法实现步骤

综合以上对DE的改进, 提出如下自适应二次变异的时变交叉概率算法的流程:

Step1: 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 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:

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

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

Step2: 计算每个个体的适应度 Fitness, 求出最优适应度 Bestfitness 及最优个体 $x_{g\text{best}}$

Step3: 判断最优适应度 Bestfitness 是否达到精度要求或是否达到最大迭代次数. 若是则退出; 否则, 执行下一步.

Step4: 按式(5)和(6)计算适应度方差 σ^2 .

Step5: 若 $\sigma^2 < \text{deta}$ 且 $\text{Bestfitness} > \epsilon$, 则按式(7)对 $x_{g\text{best}}$ 及部分随机选择的个体进行变异操作; 否则不变异, 进行下面的操作.

Step6: 对 x_i^g ($i = 1, 2, \dots, N_p$) 执行 Step7 ~ Step9, 生成第 $g + 1$ 代种群

Step7: 在种群中随机选择4个不同的个体, 按

式(2) 进行变异操作, 生成变异个体 x_m .

Step 8: 按式(3) 进行交叉操作, 其中 CR 由式(8) 确定, 生成试验个体 x_r .

Step 9: 按式(4) 进行选择操作, 生成 $g + 1$ 代个体 x_i^{g+1} .

Step 10: $g = g + 1$, 返回 Step 2

4 自适应二次变异差分进化算法性能分析

为验证本文提出的算法的有效性, 下面通过 4 个典型的Benchmarks 函数进行测试, 同时与遗传算法和基本的DE 算法进行比较. 其中 f_1 为 Sphere 单峰二次函数; f_2 为Rosenbrock 函数, 是一个非凸、病态函数; f_3 为Rastrigin 函数, 多峰, 在 $S = \{x_i | (-5.12, +5.12), i = 1, 2, \dots, n\}$ 范围内大约存在 $10n$ 个局部极小点; f_4 为 Griewank 函数, 多峰, 存在大量局部极小点.

实验参数设置如下: 函数维数为 30, 种群规模 N_p 都设为 60, 最大进化代数为 600, DE 的收缩因子 $F = 0.5$, 交叉概率 $CR = 0.6$; 遗传算法^[8] 采用实数编码, 算术交叉和非一致变异算子, 其交叉概率为 0.8, 变异概率为 0.05; ASMDE 算法中, 交叉概率 $CR_{min} = 0.3, CR_{max} = 0.9$, 适应度方差阈值 $deta = 0.001$, 适应值变异控制精度 $\epsilon = 0.001$.

$$f_1 = \sum_{i=1}^{30} x_i^2, -100 \leq x_i \leq 100; \tag{10}$$

$$f_2 = \sum_{i=1}^{30} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], -30 \leq x_i \leq 30; \tag{11}$$

$$f_3 = \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i) + 10), -5.12 \leq x_i \leq 5.12; \tag{12}$$

$$f_4 = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, -50 \leq x_i \leq 50 \tag{13}$$

ASMDE 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是实际应用时必须面对的一个问题. 个体数目太小, 可能对算法没有多大的贡献; 而个体数目太大, 又会增加计算量, 进一步的改善作用不大. 本文用 f_2 函数对不同个体数目进行了实验研究, 对每一确定数目

个体, 独立运行 20 次, 每次运算最优解的统计结果如表 1 所示. 由表 1 可知, 变异个体数目较小时, 算法不太稳定, 而随着变异个体数目的增加, 标准差减小, 算法越来越稳定, 但当个体数目超过 25 以后, 算法改善效果不明显. 实验表明, 部分变异个体的数目选择群体规模的 1/4 ~ 1/3 比较合适. 在本文后面的实验中, 变异个体数目设为 15.

表 1 变异个体数目对算法性能的影响分析 ($N_p = 60$)

变异个体数目	最小值	最大值	平均值	标准差
0	23.519 3	116.830 8	43.030 6	6.829 1
5	19.746 6	89.546 7	36.120 2	4.410 9
10	16.701 3	94.184 7	32.683 9	3.650 5
15	18.371 8	37.821 9	26.553 9	0.844 0
20	19.201 8	40.742 3	27.266 1	0.992 4
25	24.462 8	29.041 5	27.073 6	0.286 3
30	22.768 5	29.022 7	26.887 8	0.405 8

表 2 列出了以上 3 种算法求解上述优化问题运行 20 次后得到的函数最优解平均值. 由表 2 可知, 本文算法的优化结果明显优于其他两种算法, 其中对于函数 f_1, f_3 和 f_4 每次都能得到精度很高的最优值. 同时由标准差可知, 本文提出的算法很稳定. 而对于高维多峰函数, GA 基本上都不能收敛到理论最优解, 容易产生早熟收敛. 由标准差可知, GA 对于 f_2 还很不稳定. DE 的效果也不理想, 除对 f_4 的精度较好外, 对其他 3 个函数的优化结果都不能令人满意. 由标准差可知, DE 对于 f_2 也不稳定.

图 2 ~ 图 5 是上述 4 个函数采用 3 种算法运行 20 次后求得平均最优适应度进化曲线. 为方便比较, 图中的纵坐标都采用适应度的对数表示. 由图 2, 图 4 和图 5 可以看出, ASMDE 对于 f_1, f_3 和 f_4 都能很快收敛到全局最优解; 而 GA 都较早出现了早熟收敛现象, 虽然 f_1 是单峰函数, 但维数较高, GA 也难以找到全局最优点; DE 的效果虽然比 GA 好一些, 但也很容易早熟收敛, 不能随迭代次数的增加进一步优化. 由图 3 可知, 对于高维非凸、病态函数 f_2 , 3 种算法都不能收敛到全局最优, 但 ASMDE 的精度相对其他两种算法要高得多, 而且算法很稳定. 从图 3 可以看出, 在搜索的前期, DE 与 ASMDE 的种群多样性较高, ASMDE 没有二次变异, 实际上就是 DE, 因而有相似的收敛特性. 但随着种群多样性的

表 2 3 种算法运行 20 次的函数平均最优解与标准差

函数	理论最优	SGA		DE		ASMDE	
		平均最优解	标准差	平均最优解	标准差	标准差	平均最优解
$f_1(x)$	0	13.285	0.7838	1.177	0.698 1	1.199e-08	1.024e-08
$f_2(x)$	0	1.809e+05	1.508e+04	1832.3	1.465e+03	26.861	0.349
$f_3(x)$	0	50.536	4.516 9	11.502	0.768 3	7.516e-08	1.799e-08
$f_4(x)$	0	0.128 75	0.013 2	0.009	0.001 9	2.877e-10	1.207e-10

下降, 适应度方差的变小, DE 产生早熟, 而ASMDE 由于二次变异, 能有效避免早熟

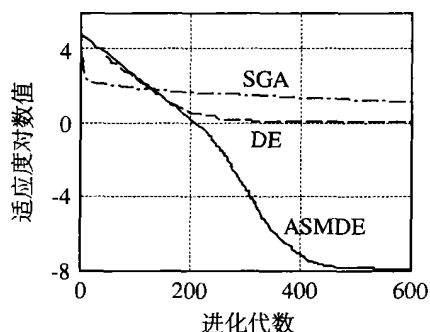


图2 $f_1(x)$ 20次平均最优适应度进化曲线

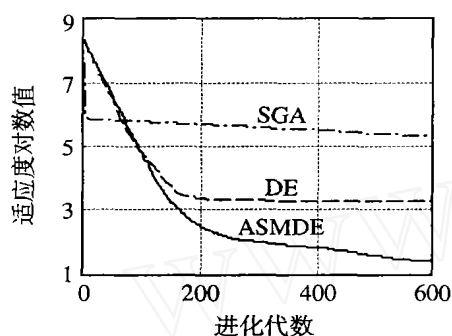


图3 $f_2(x)$ 20次平均最优适应度进化曲线

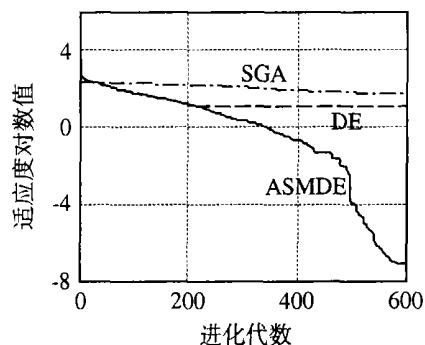


图4 $f_3(x)$ 20次平均最优适应度进化曲线

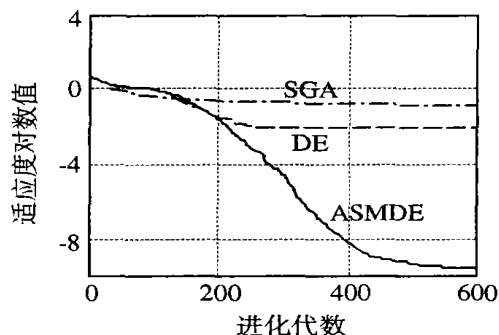


图5 $f_4(x)$ 20次平均最优适应度进化曲线

5 结 论

本文对差分进化算法的早熟收敛问题进行了分析, 导致早熟收敛的根本原因是随进化的进行, 种群多样性快速下降. 为提高种群多样性, 使算法跳出局部最优点, 提高全局搜索能力, 提出了一种基于群体适应度方差自适应二次变异和时变交叉概率机制的差分进化算法. 实验表明, 该算法有很强的全局搜索能力, 能有效避免早熟收敛, 而且算法的收敛速度快, 稳定性好. 算法中引进的自适应二次变异操作, 与DE相比运算量有所增加, 但与遗传算法相比, 运算量小, 是一种非常实用的优化算法.

参考文献(References)

- [1] Price K. Differential Evolution: A Fast and Simple Numerical Optimizer [A]. 1996 Biennial Conf of the North American Fuzzy Information Processing Society [C]. New York, 1996: 524-527.
- [2] Price K. Differential Evolution vs the Functions of the 2nd CEO [A]. IEEE Int Conf on Evolutionary Computation [C]. Indianapolis, 1997: 153-157.
- [3] Ji-Pyng Chiu, Feng-Sheng Wang. A Hybrid Method of Differential Evolution with Application to Optimal Control Problems of a Bioprocess System [A]. IEEE Int Conf on Evolutionary Computation Proceedings [C]. New York, 1998: 627-632.
- [4] Junhong Liu, Jouni Lampinen. A Fuzzy Adaptive Differential Evolution Algorithm [A]. IEEE Region 10 Conf on Computers, Communications, Control and Power Engineering [C]. Beijing, 2002: 606-611.
- [5] 谢晓峰, 张文俊, 张国瑞, 等. 差异演化的实验研究[J]. 控制与决策, 2004, 19(1): 49-52.
(Xie X F, Zhang W J, Zhang G R, et al. Empirical Study of Differential Evolution [J]. Control and Decision, 2004, 19(1): 49-52.)
- [6] Rainer S, Price K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces [J]. J of Global Optimization, 1997, 11(4): 341-359.
- [7] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420.
(Lv Z S, Hou Z R. Particle Swarm Optimization with Adaptive Mutation [J]. J of Electronics, 2004, 32(3): 416-420.)
- [8] 王耀南. 智能控制系统[M]. 长沙: 湖南大学出版社, 1996: 282-290.
(Wang Y N. Intelligent Control System [M]. Changsha: Publish House of Hunan University, 1996: 282-290.)