

文章编号: 1001-0920(2007)06-0608-05

分级变异的动态克隆选择算法

胡江强^a, 郭 晨^b, 尹建川^a, 李铁山^a

(大连海事大学 a. 航海学院, b. 自动化与电气工程学院, 辽宁 大连 116026)

摘 要: 基于浮点数编码, 提出一种分级变异的动态免疫克隆选择优化算法. 根据抗体的亲和力将种群分解为 3 个子种群, 分配以不同的搜索任务, 实施不同的变异策略. 在进化过程中动态改变种群规模、克隆规模和变异参数, 从而加快了全局搜索速度, 提高了局部搜索精度. 对 5 个复杂函数的优化仿真实验表明了该算法的有效性.

关键词: 浮点数编码; 克隆选择; 进化算法; 动态参数; 函数优化

中图分类号: TP18 **文献标识码:** A

Dynamic clonal selection algorithm with classified mutation

HU Jiang-qiang^a, GUO Chen^b, YIN Jian-chuan^a, LI Tie-shan^a

(a. College of Navigation, b. College of Automation and Electrical Engineering, Dalian Maritime University, Dalian 116026, China. Correspondent: HU Jiang-qiang, E-mail: hujiangqiang@126.com)

Abstract: A dynamic immune clonal selection algorithm with classified mutation is proposed based on floating point coding. To speed up the global search and improve the local convergence precision, the following two main strategies are introduced. According to the antibody affinity in relation to the antigen, the antibody population is decomposed into several subsets, and they are submitted to respective mutation processes for their different given tasks. Then, the population size, the clone size and the mutation parameters are dynamically changed with evolution processing. The proposed algorithm is used to optimize 5 complex functions for testing and the results show its effectiveness.

Key words: Floating point number code; Clonal selection; Evolutionary algorithm; Dynamic parameters; Function optimization

1 引 言

生物免疫系统是生物体内结构最为复杂、功能最为独特的系统. 研究生物免疫系统机理的免疫学, 成为当代生命科学中的前沿学科^[1]. 人工免疫系统(AIS)是模仿自然免疫系统功能的一种新的智能方法. 近年来, 人工免疫系统发展迅速, 它是继神经网络、模糊逻辑和进化计算后, 人工智能领域的又一研究热点^[2-4]. 免疫抗体克隆选择是免疫系统理论的重要学说, 受到了人工智能研究者的关注^[5-8]. 继Castro等提出二进制克隆选择算法(COLNALG)^[5]并成功用于模式识别和多峰函数的优化后, 一些基于克隆选择学说的新算法^[6-11]相继提出, 并应用于多目标函数优化^[9]、组合优化^[6,8,10]等问题.

克隆选择算法(CSA)是模拟免疫系统防御外来侵袭的一种学习进化过程. 与遗传算法的有性繁

殖(交叉、变异)不同, 免疫克隆选择算法是通过无性繁殖(克隆、变异)连续传代形成群体, 通过抗体克隆群的变异来提高免疫力(亲和度、适应值), 从而实现种群的进化.

克隆选择算法虽能克服遗传算法局部搜索能力差以及常出现的进化缓慢的问题, 但用较小的变异概率来提高局部搜索能力, 却降低了全局搜索能力; 而较大的变异概率虽能提高全局寻优速度, 却使收敛精度下降. 因此, 其全局搜索能力与局部搜索能力是矛盾的.

本文基于克隆选择学说, 采用 $[0, 1]$ 区间内的浮点数编码, 以搜索全局最优解为目标, 提出分级变异和动态进化参数的思想. 依据抗体与抗原的亲和力, 将种群划分为功能不同的子种群: 高亲和力抗体在较小邻域内精细搜索局部最优解, 中亲和力抗体在

收稿日期: 2006-02-24; 修回日期: 2006-06-05.

基金项目: 国家自然科学基金项目(60474014); 教育部高等学校博士学科点专项基金项目(20040151007); 交通部应用基础研究项目(200432922504).

作者简介: 胡江强(1970—), 男, 浙江天台人, 博士生, 从事进化算法、船舶运动控制的研究; 郭晨(1956—), 男, 江苏如东人, 教授, 博士生导师, 从事智能控制、船舶自动化系统等研究.

较大邻域内进行全局粗搜索,低亲和度抗体则死亡并随机再生,以保持种群的多样性,且在进化中动态调整参数.本文算法具有以下特点:1)定义了分级变异的概念;2)搜索空间等级不同;3)搜索空间随抗体亲和度不同;4)搜索空间动态改变;5)种群规模收敛程度逐渐缩小,克隆规模则随之增大;6)全局搜索与局部优化有机结合.

2 浮点数编码的分级变异动态克隆选择算法(FCMDCSA)

为了描述方便,设一类连续对象的全局极大值优化问题如下:

$$\max f(x_i), x_i \in (a_i, b_i), i = 1, 2, \dots, l. \quad (1)$$

其中 l 为变量个数.

2.1 浮点数编码

设第 g 代抗体群 $A_b(g) = \{a_{b_1}, a_{b_2}, \dots, a_{b_N}\}$ 为抗体 a_{b_j} 的 N 元组,抗体 $a_{b_j} = (z_1^j, z_2^j, \dots, z_l^j)$,第 j 个抗体的第 i 个基因 z_i^j 为 $[0, 1]$ 区间内的浮点数.对 z_i^j 和 x_i^j 建立如下映射:

$$x_i^j = a_i + z_i^j(b_i - a_i). \quad (2)$$

则称 z_i^j 为 x_i^j 的浮点数编码, x_i^j 为 z_i^j 的解码.

2.2 种群分解

将 $A_b(g)$ 依据抗体与抗原亲和度的大小降序排序后,按 1:6:3 的比例分解成 3 个子种群 $A_{b_{l_{mj}}}(g)$, $A_{b_{l_{nj}}}(g)$ 和 $A_{b_{l_{oj}}}(g)$,其中 $A_b(g) = A_{b_{l_{mj}}}(g) \cup A_{b_{l_{nj}}}(g) \cup A_{b_{l_{oj}}}(g)$, $A_b(g) \in S^N \times \mathbb{R}^L, m + n + r = N$.

$A_{b_{l_{mj}}}(g)$ 为亲和度最高的子种群, $A_{b_{l_{nj}}}(g)$ 为中等亲和度的子种群, $A_{b_{l_{oj}}}(g)$ 为亲和度最低子种群.对 3 个子种群分配不同的任务: $A_{b_{l_{mj}}}(g)$ 用于寻找局部最优解,将给予较小的变异; $A_{b_{l_{nj}}}(g)$ 用于开发更多的解空间,以便找到全局最优解区域,应给予较大的变异;而 $A_{b_{l_{oj}}}(g)$ 应死亡,并随机产生新个体予以补充,即接收器重组^[5,6],以保持种群的多样性,协助 $A_{b_{l_{oj}}}(g)$ 寻找全局最优解区域.

2.3 克隆

对于抗体 $a_{b_i} \in A_{b_{l_{mj}}}(g) \cup A_{b_{l_{nj}}}(g)$,按下式进行克隆:

$$C = \{C_i\}, C_i = I \times a_{b_i}, i = 1, 2, \dots, n_c. \quad (3)$$

其中: I 为 n_c 维单位行向量, n_c 为抗体 a_{b_i} 的克隆规模, $C_i = a_{b_i}$ 为 a_{b_i} 的一个克隆体.

2.4 分级变异操作

对于 $C_i \in C$,设 C_{ij} 为 C_i 中的第 j 个变量,按概率 P_m 进行如下变异操作:

$$C_{ij}^* = C_{ij} + a_m(a_b)(2^{-j} - 1), j = 1, 2, \dots, l. \quad (4)$$

其中: r 为 $[0, 1]$ 区间的随机数, a_m 为变异尺度变换因子,

$a_m(a_b)$ 为如下变异尺度函数:

$$a_m(a_b) = \exp(-F), F = \frac{F - \text{rank}(a_b)}{N}. \quad (5)$$

其中: F 为常数, F 为抗体亲和度, $\text{rank}(a_b)$ 为抗体 a_b 在种群中的排名, N 为种群规模.显然,抗体的亲和度越高, $a_m(a_b)$ 越小.

从式(4)可以看出,对抗体 a_b 的 n_c 个克隆体的变异操作,实质上是抗体 a_b 在其邻域 $a_b \pm a_m(a_b)$ 范围内进行 n_c 次搜索.其搜索范围即邻域大小一方面与抗体的亲和度相关,另一方面取决于变异尺度因子 $a_m(a_b)$ 的选取.

变异概率 P_m 值决定着抗体 a_b 进入一致变异操作的变量个数.显然, P_m 和 $a_m(a_b)$ 越大,对变异后克隆体的亲和度影响越大.

定义 1 设 $\forall a_b \in A_{b_{l_{mj}}}(g), \forall a_b \in A_{b_{l_{nj}}}(g)$ 的变异概率分别为 P_{m_1} 和 P_{m_2} , 尺度变换因子分别为 a_{m_1} 和 a_{m_2} .若 $P_{m_1} < P_{m_2}, a_{m_1} < a_{m_2}$, 则称其为分级变异.

分级变异的思想是让高亲和度抗体有较小的搜索空间,搜索局部最优解,而低亲和度抗体有较大的搜索空间,进行全局最优解的粗搜索,避免陷入局部最优.

2.5 克隆选择^[5,6,8]

对 a_b 抗体变异后的克隆群 $C^* = \{C_i^*\}$ 进行亲和度评估,得 $f(C^*) = \max_i f(C_i^*)$.若存在 $f(C_i^*) = \max f(C^*) > f(a_b)$, 则 $a_b := C_i^*, f(a_b) := f(C_i^*)$.从而更新抗体群,实现信息交换.

2.6 动态参数

种群中各抗体经过克隆、变异和克隆选择,从空间的不同初始点以不同的路径向全局最优点移动.随着种群的进化,种群中各抗体都集中于全局最优点附近.此时平均适应度的变化减小,可以适当减小种群规模,以去除不必要的低亲和度的搜索;同时增大抗体的克隆规模,使变异操作后亲和度提高的概率增大.

需要注意的是,平均适应值变化很小时,种群也可能陷入局部最优,因此需要调整变异参数.

当 $A_{b_{l_{mj}}}(g) \cup A_{b_{l_{nj}}}(g)$ 的平均适应值在连续若干代内变化很小时,本文动态调整如下参数:

1) 种群规模 N 按下式逐渐减小:

$$N(g+1) = \max\{\lfloor N(g) \rfloor, N_{\min}\}. \quad (6)$$

其中: $\lfloor \cdot \rfloor$ 表示上取整, α ($0, 1$) 为常数(本文取 0.8), N_{\min} 为最小种群规模.

2) 克隆规模 n_c 按下式逐渐增大:

$$n_c(g+1) = \min\{\lceil n_c(g) \rceil, n_{c\max}\}. \quad (7)$$

其中: $\lceil \cdot \rceil$ 表示上取整, $(0;1)$ 为常数(本文取0.8), n_{cmax} 为最大克隆规模.

3) 借鉴进化策略(ES)^[12]高斯随机数中的自适应方法,尺度变换因子 σ_1 和 σ_2 的自适应调整策略如下: σ_1 和 σ_2 分别随机地乘以或除以1.3.实验表明,这种自适应方法能有效避免陷入局部最优.跳出局部最优后,即平均适应值提高后, σ_1 和 σ_2 值分别回到初始设定值 σ_{10} 和 σ_{20} . $0 < \sigma_{10} < 0.2$, $0.2 < \sigma_{20} < 10$.由于采用了自适应策略,初值的选择对算法性能的影响不大.

除了上述参数外,本文还对式(5)中参数 α 进行动态调整.根据式(5),当 F 值一定时, α 值越大, a_m 值越小,且对中等亲和度的影响较大.显然, α 取较小值,抗体有较大的搜索空间,全局搜索能力加强,但局部优化能力变差;相反, α 取较大值,抗体在较小的空间搜索增强了局部峰值的搜索能力.为解决在进化过程中全局进化与局部优化的矛盾,有必要动态改变 α 值.经过大量试验,按下式循环变化 α 的效果较好:

$$\alpha(g+1) = \begin{cases} \alpha(g), \text{rem}(g, g_D) = 0; \\ \alpha_0, \text{rem}(g, g_{Dmax}) = 0. \end{cases} \quad (8)$$

其中: α_0 为初始值(取 $\alpha_0 = 2$), $\alpha > 1$ 为调整因子(取 $\alpha = 1.1$), $\text{rem}(x, y)$ 为取 x/y 的余数, g_D 为间隔代数(取 $g_D = 3$), g_{Dmax} 为最大间隔代数(取 $g_{Dmax} = 20g_D$).

3 FCMDCSA 伪代码

初始化: $g = 0, T = 0$; 随机产生初始种群

$A_b(0)$, 计算目标函数值 $\{f(A_b(0))\} = \{f(a_{b_i}^0)\}$;

while 不满足停止条件 do

$A_b(g)$ 按 $f(A_b(0))$ 降序排序;

计算 $\bar{f}(g) = \frac{1}{m+n} \sum_{i=1}^{m+n} f(a_{b_i}^g)$;

if $\bar{f}(g) < \bar{f}(g-1)$ then

$T := T + 1$;

else $T := 0$; $\sigma_1 := \sigma_{10}$, $\sigma_2 := \sigma_{20}$; end

if $T = T_{max}$ then

动态调整参数 $N, n_c, \sigma_1, \sigma_2$; $T := 0$;

end

分解 $A_b(g)$ 为 $A_{b_{[m]}}(g)$, $A_{b_{[n]}}(g)$ 和 $A_{b_{[r]}}(g)$;

对 $A_{b_{[m]}}(g)$ 和 $A_{b_{[n]}}(g)$ 进行克隆、分级变异

和克隆选择;

$A_{b_{[r]}}(g)$ 死亡,由随机产生的新抗体替代;

$g := g + 1$;

动态调整参数 α ;

end

4 仿真结果及分析

分别用下面式(9)~(13)5个测试函数对本文算法(FCMDCSA)进行性能测试,并与CLONAL G^[5,6]和Share-GA^[13]进行比较. Share-GA是多种遗传算法经过实验后性能较好的算法.5个测试函数分别如下:

$$f_1(x, y) = x^2 + y^2 - 3\cos 3\pi x + 3\cos 3\pi y + 0.3. \quad (9)$$

式(9)是Bohachevsky测试函数3,如图1所示.全局最大值约为-0.240 034 4,有两个全局极点,分布在 $(0, -0.24)$ 和 $(0, 0.24)$ 上.

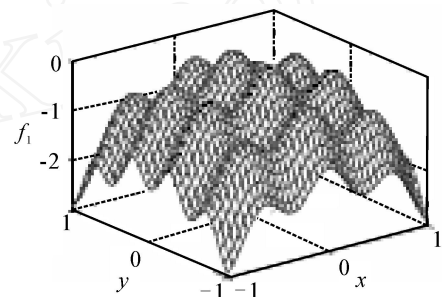


图1 $f_1(x, y)$ 特征

$$f_2(x, y) = 100(x^2 - y)^2 + (1 - x)^2. \quad (10)$$

式(10)是De Jong函数2,为一连续凹函数,又称马鞍函数,如图2所示.全局极值为0,分布在 $(1, 1)$ 上,有无数局部极值分布在两条脊上,采用普通算法很难求出其全局极值.

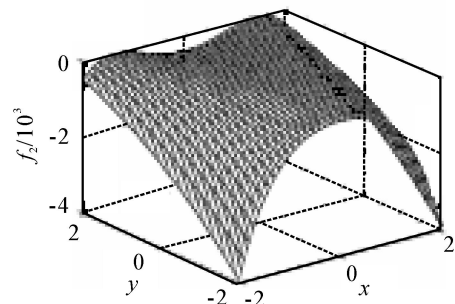


图2 $f_2(x, y)$ 特征

$$f_3(x, y) = \exp(-x^2 - y^2) - 1. \quad (11)$$

式(11)只有一个峰值为0,分布在 $(0, 0)$ 上,如图3所示.

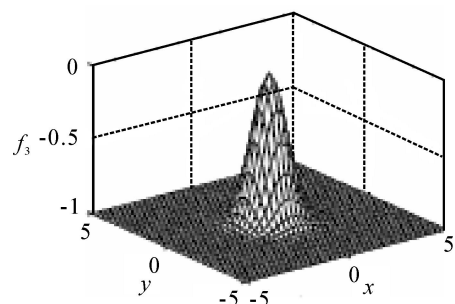


图3 $f_3(x, y)$ 特征

$$f_4(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{1 + 0.001(x^2 + y^2)} \quad (12)$$

式(12)是“大海捞针函数”,如图4所示。全局最大值为1,分布在(0,0)上,其周围有一个圈脊为0.990283,很容易陷入此局部极大点。

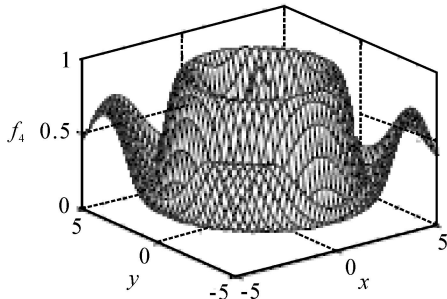


图4 $f_4(x, y)$ 特征

$$f_5(x, y) = -(x^2 + y^2)^{0.25} [\sin^2(50(x^2 + y^2)^{0.1}) + 1] \quad (13)$$

式(13)全局最大值为0,分布在(0,0)上,如图5所示。其四周分布着很多由外向里越来越密的局部最大值,是极难优化函数。

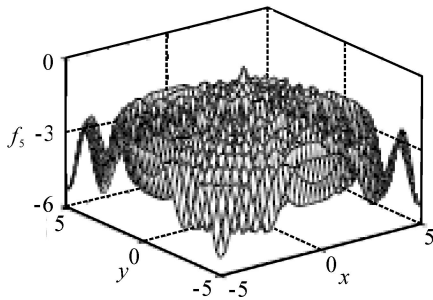


图5 $f_5(x, y)$ 特征

CLONAL G和 Share- GA 算法采用文献[14]的代码,种群规模为100,每一变量二进制编码长度为22位。CLONAL G算法中各抗体的克隆规模为10,对于 f_1, f_4 和 f_5 , 变异概率 $P_m = 0.05$;对于 f_2 , $P_m = 0.1$;对于 f_3 , $P_m = 0.07$ 。Share- GA 算法中交叉概率 P_c 和变异概率 P_m 为:对于 f_1 和 f_2 , $P_c = 0.8$, $P_m = 0.1$;对于 f_3 , $P_c = 0.9$, $P_m = 0.1$;对于 f_4 和 f_5 , $P_c = 0.9$, $P_m = 0.05$ 。FCMDCSA 算法参数如下:初始种群规模 $N_0 = 100$, 初始克隆数目 $n_{c0} = 10$, 对于 f_1, f_3 和 f_4 , 初始变异尺度变换因子 $\tau_{10} = 0.01$, $\tau_{20} = 1.4$;对于 f_2 和 f_5 , $\tau_{10} = 0.1$, $\tau_{20} = 0.4$; 分级变异概率均为 $P_{m1} = 1/l$, $P_{m2} = 1$ 。

不失公平性,3种算法的停止准则均选为 $|f^* - f^{\text{best}}| < \epsilon$ 。其中 f^* 为目标函数最优值, f^{best} 为当前最优解, ϵ 为设定精度。附加停止准则设为目标函数评估最大次数 221 200, 相当于 CLONAL G 运行 200 代。各算法独立随机运行 100 次, 实验统计结果如表 1 ~ 表 5 所示。算法性能评价准则采用计算量

和阻滞次数^[15],表中计算量采用100次目标函数评估平均次数及其标准偏差,阻滞次数是上述设定条件下种群受困于局部极值点的次数。表中“*”表示仅对未陷入局部最优的情况进行统计,“—”表示100次实验均不符合条件无法进行统计。

表1 $f_1(x, y)$ 实验结果($= 10^{-6}$)

变量范围	算 法	目标函数评估次数	阻滞次数
	Share- GA	-	100
[- 1,1]	CLONAL G	13 894 ±3 318	0
	FCMDCSA	9 028 ±2 594	0

表2 $f_2(x, y)$ 实验结果($= 10^{-6}$)

变量范围	算 法	目标函数评估次数	阻滞次数
	Share- GA	88 347 * ±57 676 *	68
[- 10,10]	CLONAL G	147 500 * ±37 527 *	85
	FCMDCSA	27 076 ±4 955	0

表3 $f_3(x, y)$ 实验结果($= 10^{-7}$)

变量范围	算 法	目标函数评估次数	阻滞次数
	Share- GA	3 522 ±1 396	0
[- 10,10]	CLONAL G	20 538 ±3 321	0
	FCMDCSA	11 854 ±2 884	0
	Share- GA	23 873 ±10 594	0
[- 300,300]	CLONAL G	97 304 * ±42 652 *	2
	FCMDCSA	24 106 ±5 005	0

表4 $f_4(x, y)$ 实验结果($= 10^{-7}$)

变量范围	算 法	目标函数评估次数	阻滞次数
	Share- GA	7 399 ±6 380	0
[- 5.12,5.12]	CLONAL G	23 838 ±6 089	0
	FCMDCSA	12 596 ±2 152	0

表5 $f_5(x, y)$ 实验结果($= 10^{-3}$)

变量范围	算 法	目标函数评估次数	阻滞次数
	Share- GA	25 966 * ±43 444 *	2
[- 1,1]	CLONAL G	51 360 ±13 038	0
	FCMDCSA	31 494 ±4 709	0
	Share- GA	-	100
[- 10,10]	CLONAL G	-	100
	FCMDCSA	47 773 ±22 359	0

从阻滞次数的统计结果看,CLONAL G 和 Share- GA 对较为复杂、优化难度较大的 f_2 和搜索范围增大后的 f_5 两个函数,优化容易陷入局部最优解;Share- GA 很难实现对诸如 f_1 函数这种具有多个全局极点的优化;FCMDCSA 对5个函数均能收敛到设定精度。

从目标函数评估次数的统计结果看,

FCMDCSA 的平均计算量明显小于 CLONAL G 算法,而且标准偏差明显减小,表明 FCMDCSA 算法稳定,并能有效地改进克隆选择算法. Share-GA 对 f_3 函数在搜索范围较小时显示出良好的性能,而对优化难度适当增大后的 f_4 函数、搜索范围较大的 f_3 函数和搜索范围较小的 f_5 函数,虽然比 FCMDCSA 的平均计算量要小,但其标准偏差明显偏大,这表明 Share-GA 的稳定性不如 FCMDCSA.

5 结 语

本文基于浮点数编码,对克隆选择算法采用了分级变异和动态参数,使得该算法在种群内部变异和种群进化过程中,能有机地结合全局搜索和局部搜索. 仿真实验表明,该算法有效地改善了克隆选择算法的性能,收敛速度快且精度高,不易陷入局部最优. 该算法为复杂连续函数的求解提供了一条有效的途径.

参考文献(References)

- [1] 左兴权, 李士勇, 李远贵. 人工免疫系统研究的新进展[J]. 计算机测量与控制, 2002, 10(11):701-705.
(Zuo X Q, Li S Y, Li Y G. New progresses in artificial immune system [J]. Computer Measurement and Control, 2002, 10(11):701-705.)
- [2] Dipankar Dasgupta, Stephanie Forrest. Artificial immune systems in industrial applications[C]. Proc of the 2nd Int Conf on Intelligent Processing and Manufacturing of Materials. Honolulu: IEEE Press, 1999: 257-267.
- [3] Gasper A, Collard P. From GAs to artificial immune systems: Improving adaptation in time dependent optimization[C]. Proc of the Congress on Evolutionary Computation. Washington: IEEE Press, 1999: 1859-1866.
- [4] 丁永生, 任立红. 人工免疫系统:理论与应用[J]. 模式识别与人工智能, 2000, 13(1): 52-59.
(Ding Y S, Ren L H. Artificial immune systems: Theory and applications [J]. Pattern Recognition and Artificial Intelligence, 2000, 13(1): 52-59.)
- [5] Leandro N de Castro, Fernando J Von Zuben. The clonal selection algorithm with engineering applications [C]. Workshop Proc of GECCO '00 Workshop on Artificial Immune Systems and Their Applications. Las Vegas, 2000: 36-37.
- [6] Leandro N de Castro, Fernando J Von Zuben. Learning and optimization using the clonal selection principle[J]. IEEE Trans on Evolutionary Computation, 2002, 6(3): 239-251.
- [7] Simon M Garrett. Parameter-free, adaptive clonal selection[C]. Congress on Evolutionary Computation. Portland, 2004, 1: 1052-1058.
- [8] 杜海峰, 焦李成, 刘若辰. 免疫优势克隆算法[J]. 电子与信息学报, 2004, 6(12):1918-1924.
(Du H F, Jiao L C, Liu R C. Immunodominance clone algorithms [J]. J of Electronics and Information Technology, 2004, 6(12):1918-1924.)
- [9] Nareli Cruze Cortes, Carlos A Coello Coello. Multi-objective optimization using ideas from the clonal selection principle [C]. GECCO2003. Chicago, 2003: 158-170.
- [10] 王磊, 潘进, 焦李成. 基于免疫策略的进化算法[J]. 自然科学进展, 2000, 10(5):450-455.
(Wang L, Pan J, Jiao L C. Evolutionary algorithm based on immune strategy [J]. Progress in Natural Science, 2000, 10(5): 450-455.)
- [11] Leandro N de Castro, Jon Timmis. An artificial immune network for multimodel function optimization [C]. Proc of IEEE Congress on Evolutionary Computation. Hawaii, 2002: 699-704.
- [12] Back T, Hoffmeister F, Schwefel H P. A survey of evolution strategies[C]. Proc of the 4th Int Conf on Genetic Algorithms. San Diego, 1991: 2-9.
- [13] Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization[C]. Proc 2nd Int Conf on Genetic Algorithms. Lawrence, 1987: 41-49.
- [14] Leandro N de Castro. Matlab code for CLONAL G is on his webpage. <http://www.dca.fee.unicamp.br/lnunes/immune.html>. 2001.
- [15] 罗印升, 李人厚. 基于免疫原理的进化算法[J]. 控制与决策, 2004, 19(2): 219-222.
(Luo Y S, Li R H. Evolutionary algorithm based on immune theory [J]. Control and Decision, 2004, 19(2): 219-222.)