

文章编号: 1001-0920(2011)03-0428-05

## 基于变异记忆矩阵的克隆选择算法

宋丹<sup>1,2</sup>, 赖旭芝<sup>1</sup>, 吴敏<sup>1</sup>, 张传科<sup>1</sup>

(1. 中南大学信息科学与工程学院, 长沙 410083; 2. 湖南财政经济学院信息管理学院, 长沙 410205)

**摘要:** 利用免疫系统的免疫记忆机制, 提出一种适于函数优化的基于变异记忆矩阵的克隆选择算法. 首先, 利用变异记忆矩阵保存进化中有用的变异信息, 以引导抗体的克隆和变异操作, 加强局部搜索能力; 然后, 利用当代种群的综合信息生成新抗体进入种群, 以加强全局搜索能力; 最后, 对最优抗体进行自学习, 以提高算法结果的精度. 标准函数仿真表明, 该算法适合求解复杂函数优化问题, 具有收敛速度快、全局收敛能力强、精度高、鲁棒性强的优点.

**关键词:** 免疫记忆; 克隆选择; 变异记忆矩阵; 全局优化

中图分类号: TP18

文献标识码: A

## Clonal selection algorithm based on mutation memory matrix

SONG Dan<sup>1,2</sup>, LAI Xu-zhi<sup>1</sup>, WU Min<sup>1</sup>, ZHANG Chuan-ke<sup>1</sup>

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China; 2. Department of Information Management, Hu'nan University of Finance and Economics, Changsha 410205, China. Correspondent: SONG Dan, E-mail: s1020d@126.com)

**Abstract:** Inspired from the immune memory mechanism in immune systems, a clonal selection algorithm based on mutation memory matrix, CSAB3M, is proposed in order to optimize complex functions. Firstly, an immune memorial matrix is developed to save mutation information, which guides operations of clone and mutation of antibodies, so that the local search ability can be improved. Then, new antibodies resulted from the comprehensive information of contemporary population are generated and introduced to the population in order to improve global search ability. Finally, the optimal antibody does self-learning operator to improve the precision of the algorithm. Simulation results on standard test functions show that the algorithm is well suitable for the complex function optimization, and has the characteristics of rapid convergence, powerful global search capability, high precision and good robustness.

**Key words:** immune memory; clonal selection; mutation memory matrix; global optimization

### 1 引言

生物免疫系统是一个蕴含学习、优化和记忆等信息处理机制的并行自适应系统. 人工免疫系统(AIS)是模仿生物免疫系统机理而构造的高性能、自组织、鲁棒性好的人工智能系统. Castro等人<sup>[1]</sup>构造了第1个克隆选择算法, 利用抗体克隆和变异进行搜索, 通过加入新抗体来保持种群的多样性, 并已成功用于模式识别、数值优化和组合优化问题.

实参数的函数优化问题在工程实际中有着广泛的应用, 针对高维、非凸且有多个局部极值点的函数优化问题, 其理想解往往很难获得. 戚玉涛等人<sup>[2]</sup>提出基于信息素模因的免疫克隆选择算法, 将学习进化经验以及用于指导进化的思想引入人工免疫算法, 在

函数优化中表现出较好的性能. 本文在克隆选择算法中引入免疫记忆机制, 设置变异记忆矩阵记忆变异信息, 并指导后续的进化过程, 以提高克隆选择算法的效率.

交叉和变异是进化计算中十分重要的2个算子, 文献[3]讨论了交叉点规模的动态调整策略对进化算法的影响; [4]针对高维优化, 论证了单基因变异较全基因变异更优; [5]讨论了不同变异算子对进化算法的影响; [6]采用一种令变异概率与亲和度成反比的变异算子, 提高了收敛的精度; [7-8]在进化计算中变步长自适应地变化, 使算法更有效; 高辉等人<sup>[9]</sup>提出实数编码量子进化算法, 采用互补双变异算子改进算法性能, 在求解复杂函数优化问题时, 取得了较好

收稿日期: 2010-01-12; 修回日期: 2010-03-17.

基金项目: 国家自然科学基金项目(60874042); 高等学校博士学科点专项科研基金新教师基金项目(20090162120068).

作者简介: 宋丹(1976-), 男, 讲师, 博士生, 从事人工智能、免疫算法的研究; 赖旭芝(1966-), 女, 教授, 博士生导师, 从事智能系统、机器人和非线性控制等研究.

的效果. 人工免疫系统中的变异操作较一般进化算法更为重要, 为此本文利用祖辈的变异信息和变异策略一起指导后代的抗体生成, 以加强算法的收敛能力.

针对复杂的函数优化问题, 一个好的算法应具有如下特征: 具有较快的局部收敛速度, 能快速收敛到局部极值点; 具有较强的全局收敛能力, 能在有限迭代中找到全局最优解; 具有较好的稳定性, 在多次重复测试中表现出好的性能.

本文提出一种基于变异记忆矩阵的克隆选择算法(CSAB3M). 首先, 在人工免疫算法进化过程中, 利用变异记忆矩阵记忆祖辈变异信息, 加以学习并指导后续的进化过程; 然后, 利用当代种群的综合信息生成新抗体进入种群, 以加强全局搜索能力; 最后, 对最优抗体进行自学习, 以提高算法结果的精度. 实验结果表明, 算法的免疫记忆机制和变异策略能够进行有效的搜索, 较好地避免盲目搜索和重复搜索. 免疫记忆机制和变异策略的引入改进了克隆选择算法的性能, 具有收敛速度快、全局收敛能力强、精度高和鲁棒性强的优点.

## 2 算法描述

对于函数优化问题

$$\min f(x) = f(x_1, x_2, \dots, x_n), \quad (1)$$

其中  $f(x)$  为  $n$  维解空间  $S$  上的连续实函数. 本文中的抗体  $x_i$  需从函数的定义域映射到  $[0,1]$  区间, 在计算函数值时相应地进行解码.

CSAB3M 是在克隆选择算法的基础上引入变异记忆矩阵 Array ( $m \times n \times 13$ ), 以加强进化信息的记录和利用. 该矩阵用来记忆祖辈在相应维上的变异信息.

$$\text{Array} = \{A_{pqr}\}, p = 1, 2, \dots, m, \\ q = 1, 2, \dots, n, r = 1, 2, \dots, 13. \quad (2)$$

其中:  $m$  表示种群内抗体的个数, 如  $\{A_{pq1}, A_{pq2}, \dots, A_{pq13}\}$  表示记录种群中第  $p$  个抗体的祖辈在第  $q$  维上的变异信息.

### 2.1 算法流程

算法伪码如下:

初始化抗体群  $P$  和变异记忆矩阵  $A$ ;

计算初始抗体亲和度;

While (终止条件不满足)

据亲和度排名并根据排名对抗体进行克隆;

对克隆后的抗体进行变异, 对变异信息进行记忆和更新;

利用种群综合信息生成新抗体, 计算亲和度, 选择亲和度高的  $N$  个抗体;

对最优抗体进行自学习;

If (交叉条件满足)

Then 随机选取一个非最优抗体, 与最优抗体交叉;

End If

End While

### 2.2 算法组成

#### 2.2.1 初始化

算法采用实数编码, 第 0 代抗体群  $G_0$  对应于问题的初始候选解, 初始化过程在搜索空间中利用混沌序列生成  $m$  个  $n$  维向量, 由此构成  $G_0$ .  $G_0 = \{a_{mn}\}$ , 如  $a_{12}$  表示第 1 个抗体的第 2 维的值. 先随机生成第 1 个抗体各维的值 ( $a_{11}, a_{12}, \dots, a_{1n}$ ), 再据 Logistic 映射生成其他抗体:

$$a_{(i+1)j} = \mu a_{ij}(1 - a_{ij}), \\ i = 1, 2, \dots, m - 1, j = 1, 2, \dots, n. \quad (3)$$

初始化变异记忆矩阵 Array ( $N \times d \times 13$ ), 令其元素皆为 0.

#### 2.2.2 克隆

据亲和度对抗体进行排序, 再据排名进行克隆. 亲和度高的抗体克隆的概率更高.

#### 2.2.3 变异和记忆

每个抗体如果克隆后代, 则克隆规模都为 2 的整数倍. 对克隆后的所有抗体进行变异, 变异采取单维变异, 随机选取其中的一维进行变异. 单次克隆产生 2 个后代, 这 2 个后代在该维上的变异方向相反.

利用变异记忆矩阵 Array 记忆上二代的变异信息.  $\{A_{pq1}, A_{pq2}, \dots, A_{pq13}\}$  表示记录种群中第  $p$  个第  $q$  维的祖辈在该维上的变异信息. 具体地,  $\{A_{pq1}, A_{pq2}, \dots, A_{pq6}\}$  记录的是祖代的二次变异的信息,  $\{A_{pq7}, A_{pq8}, \dots, A_{pq12}\}$  记录的是父代的二次变异的信息. 每一次变异的信息用 3 个元素记录: 比如  $\{A_{pq1}, A_{pq2}, A_{pq3}\}$  记录变异的尺度、成败、维值.  $A_{pq13}$  分别记录种群中第  $p$  个抗体在第  $q$  维上变异的次数. 以  $\{A_{pq1}, A_{pq2}, A_{pq3}\}$  为例, 先记录  $A_{pq1}, A_{pq3}$ , 其中  $A_{pq2}$  需计算抗体亲和度与原有抗体比较优劣, 若比原抗体优, 则记入 1, 否则记入 -1. 在记录的前提下, CSAB3M 执行下面的变异策略对当前情况进行判断, 然后再引导当前抗体的变异操作.

变异策略的伪码如下 (假定第  $p$  个抗体在第  $q$  维上发生变异):

If ( $A_{pq13} = 0$ ) // 第 1 次变异

随机生成新值, 变异尺度设为正值;

Else

If ( $A_{pq13} = 1$ ) // 第 2 次变异

If (上一代二次变异都失败)

```

    变异尺度设为上一代变异尺度的二分之一;
    Else
    变异尺度设为上一次成功变异的尺度的 2 倍
    和负二分之一;
    End If
    Else // 第 3 次及以后的变异
    If ( $A_{pq13} > q_2$ ) // 如果变异次数超过阈值, 则
    重新开始
     $A_{pq13} = 0$ 
    End If
    If (上一代二次变异都失败)
    变异尺度设为上一代变异尺度的二分之一;
    Else
    If (上一代变异尺度是上二代成功变异的
    的 2 倍)
    变异尺度设为上一次变异的尺度的 2 倍;
    Else
    变异尺度设为上一次成功变异的尺度的
    二分之一和负二分之一;
    End If
    End If
    End If
    End If

```

从以上变异策略可以看出, 首先保存上二代的变异信息, 然后根据这些进化信息有效地引导下代的抗体生成. 这些策略不仅考虑了上代成功的探索信息, 也吸取了失败的探索信息, 使下代的变异更有效, 可有效地加强局部搜索速度. 从变异尺度的取值规则可以看出, 变异尺度的值为原有变异尺度值的二分之一或 2 倍, 使变异步长调整快速高效, 在一定程度上避免了重复搜索和低效搜索. 此外, 变异采取双向变异, 可以避免算法在错误的单方向上不停地搜索. 最后, 针对函数复杂性的不同, 设定了变异深度阈值  $q_2$ , 当变异次数超过  $q_2$  时, 重新随机生成新的变异尺度, 并更新变异记忆矩阵.

### 2.2.4 补 充

为提高整个群体的多样性和群体的亲和度, 算法利用种群信息综合生成新抗体.  $a_i$  表示种群中第  $i$  个抗体,  $a_{ij}$  表示第  $i$  个抗体的第  $j$  维的值. 给每一个  $a_{ij}$  一个小的邻域半径  $r_1$  ( $[a_{ij} - r_1, a_{ij} + r_1]$ ). 在第  $j$  维上生成新抗体的值时, 采用轮盘赌的方法确定在第  $i$  个抗体的领域内随机生成. 具体地, 在  $a_{ij}$  的领域内生成新值的概率为

$$P_{aij} = (f(a_i))^{-1} / \sum_{i=1}^N (f(a_i))^{-1}, \quad (4)$$

其中  $f(a_i)$  表示抗体  $a_i$  的函数值.

由式 (4) 可以看出, 抗体亲和度越高, 在其邻域内生成新抗体的概率越高. 此外, 如果有 2 个以上的抗体 (如  $a_{ij}$  和  $a_{kj}$ ) 存在领域重叠现象, 则重叠区域内生成新抗体的概率  $p = p_{aij} + p_{akj}$ , 在优胜抗体的重叠区域产生新抗体的概率更高. 这样, 与传统的交叉和变异算子不同, CSAB3M 补充机制综合利用了群体中所有抗体的基因信息, 并兼顾了每个抗体的亲和度, 产生的新抗体具有较大的概率表现较优. 最后, 在种群和新抗体中取亲和度高的  $N$  个抗体进入下一步.

### 2.2.5 自 学 习

自然界的智能生物具有从局部环境进行学习的能力. 为提高算法的精度, 对最优抗体实施局部的 CSAB3M 算法. 给定阈值  $q_3$ , 即对最优抗体实施  $q_3$  次变异, 变异策略与前述相同. 同时, 记忆和更新 Array 矩阵. 通过自学习算子, 可以在最优抗体的较小邻域范围内实现较高精度的搜索. 相应地, 阈值  $q_3$  取值越大, 其精度越高, 算法的计算成本也相应提高.

### 2.2.6 交 叉

交叉算子具有两面性: 一方面, 它的运用能有效避免算法陷入局部最优解; 另一方面, 过于频繁地使用这一算子, 会对祖辈传承的优良基因模式造成破坏. 所以在进化算法中应慎重使用这一算子. CSAB3M 设定阈值  $q_4$ , 若间隔代数大于  $q_4$ , 则进行交叉. 交叉的对象只选取种群中的 2 个抗体, 即对最优抗体和随机选择的 1 个非最优抗体进行交叉. 此外, 算法实施的是单维交叉, 交叉时只选取其中的一维, 用非最优抗体的该维替换最优抗体的该维. 如果新抗体优于最优抗体, 则取代最优抗体, 并更新相应的 Array 矩阵.

## 3 数值实验

采用以下 5 种标准测试函数:

$$\min F_1 = \sum_{i=1}^{30} x_i^2, \quad -100 \leq x_i \leq 100;$$

$$\min F_2 = \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i|, \quad -10 \leq x_i \leq 10;$$

$$\min F_3 = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \\ -600 \leq x_i \leq 600;$$

$$\min F_4 = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}\right) - \\ \exp\left(\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)\right) + 20 + e, \\ -32 \leq x_i \leq 32;$$

$$\min F_5 = 10 \times 30 + \sum_{i=1}^{30} (x_i^2 - 10 \cos(2\pi x_i)),$$

$$-5.12 \leq x_i \leq 5.12.$$

函数  $F_1$  和  $F_2$  是相对简单的单峰函数, 但随着维数的增高, 其全局最优解变得难以获取. 函数  $F_3, F_4$  和  $F_5$  是具有许多局部极小点的多峰值函数, 在高维情况下极难获得全局最优解. 这些标准测试函数已广泛用于数值优化实验中, 在  $x = (0, \dots, 0)$  处存在全局最小值 0. 本文算法进行仿真实验时维数取为 30.

CSAB3M 对上述函数均独立运行 50 次进行数据统计. 具体参数设置如表 1 所示. 其中:  $m$  表示种群规模;  $g$  表示进化最大代数;  $q_1$  表示全局收敛阈值, 即当最优抗体的函数值与全局理论最优值之间的差值小于  $q_1$  时, 认为算法达到最优, 统计其评价次数.  $q_1$  的值均设为  $1E-10$ , 表明对算法的全局收敛条件的设定是较严格的. 对于不同函数, 参数  $m$  和  $q_2$  的设定较为关键, 前者表示对种群规模的要求, 后者表示在局部的搜索深度.  $F_3$  对种群规模要求最高,  $F_3, F_4$  和  $F_5$  则要求较快地结束局部深度搜索, 避免算法在局部最优解附近作过多的无效搜索.

表 1 CSAB3M 参数列表

	$m$	$g$	$q_1$	$q_2$	$q_3$	$q_4$	$r_1$
$F_1$	10	1000	1E-10	50	30	10	0.015
$F_2$	10	1000	1E-10	50	30	10	0.015
$F_3$	30	1000	1E-10	10	30	10	0.005
$F_4$	3	1000	1E-10	12	50	10	0.05
$F_5$	3	1000	1E-10	12	30	10	0.05

本文用算法对上述标准测试函数进行测试, 并将 CSAB3M 与 QIEA, StGA, FEP, FES 和 PSO 共 5 种算法进行比较. StGA, FEP, FES 和 PSO 算法的数据来源于文献 [10], QIEA 的数据来源于 [11], 这些算法的参数设定可参阅 [10-11]. 表 2~表 4 分别比较了各种算法独立运行 50 次的平均评价次数、平均最优函数值和标准差.

从实验数据可以看出, FEP, FES 和 PSO 算法的结果较差, 其平均评价次数多, 寻优结果不理想. CSAB3M, QIEA 和 StG 算法的结果较优, 用较少的函数评价次数便搜索到全局最优点. 从表 2 可以看出, CSAB3M 在  $F_1, F_2, F_3$  和  $F_5$  函数上平均评价次数均少于 QIEA 和 StGA, 表明变异记忆矩阵的引入和变异策略的使用有效地提高了算法的收敛速度. 从表 3 可以看出, CSAB3M 在 5 个函数的全局最优值上均取得了较 QIEA 和 StGA 精度更高的解. 表 4 的数据显示, CSAB3M 的标准差更小, 具有更稳定的搜索性能.

表 2 中 CSAB3M 在  $F_4$  上的平均评价次数略高于

表 2 各种算法运行 50 次的平均评价次数

	CSAB3M	QIEA	StGA	FEP	FES	PSO
$F_1$	2066	30000	30000	150000	/	250000
$F_2$	2888	17600	17600	200000	/	/
$F_3$	37718	52500	52500	200000	200030	250000
$F_4$	12811	10000	10000	150000	150030	/
$F_5$	21519	/	28500	500000	500030	250000

表 3 各种算法运行 50 次的平均最优结果

	CSAB3M	QIEA	StGA	FEP	FES	PSO
$F_1$	0	7.8E-17	2.45E-15	5.7E-4	/	11.175
$F_2$	0	3.0E-10	2.03E-7	8.1E-3	/	/
$F_3$	0	0	2.44E-17	1.6E-2	3.7E-2	0.4498
$F_4$	6.7E-14	1.36E-8	3.52E-8	1.8E-2	1.2E-2	/
$F_5$	0	/	4.4E-13	4.6E-2	0.16	47.1354

表 4 各种算法运行 50 次的标准差

	CSAB3M	QIEA	StGA	FEP	FES	PSO
$F_1$	0	1.0E-16	5.25E-16	1.3E-4	/	1.3208
$F_2$	0	2.3E-10	2.95E-8	7.7E-4	/	/
$F_3$	0	1.4E-17	4.54E-17	2.2E-2	5.0E-2	0.0566
$F_4$	9.6E-15	4.6E-8	3.51E-9	2.1E-3	1.8E-3	/
$F_5$	0	/	1.4E-13	1.2E-2	0.33	1.782

QIEA 和 StGA, 但结合表 3 中最优结果, CSAB3M 的  $6.7E-14$  较 QIEA 的  $1.36E-8$  和 StGA 的  $3.52E-8$  的结果更优, 因此 CSAB3M 在  $F_4$  上的搜索表现是可接受的.

综合表 2~表 4, 由于 CSAB3M 利用免疫记忆机制和变异策略进行了更有效的搜索, 在与 QIEA, StGA, FEP, FES 和 PSO 算法进行比较时表现出收敛速度快、全局收敛能力强、精度高和鲁棒性强的优点.

量子计算在寻优问题上表现出较优的性能. 文献 [9] 将量子计算的原理引入进化算法, 并在求解复杂函数优化问题时对 QEA, M-ES 和 RCQEA 算法进行了比较. 各种算法独立运行 30 次, 取 30 次中最终寻优结果的最优值和最劣值, 并统计其平均值和方差以比较各算法的性能. 本文对 CSAB3M 独立运行 30 次, 并统计结果进行比较.

QEA, M-ES 和 RCQEA10 算法的数据来源于文献 [9] (只对  $F_1, F_3, F_4$  和  $F_5$  进行了测试), 以其中效果最好的 RCQEA10 为例: 群体规模  $m = 10$ , 经求精、求泛和交叉等运算, 最大终止代数  $g = 5000$ . 为更好地对比, 对 CSAB3M 的参数作如下改动: 函数的群体规模  $m = 10$  ( $F_3$  取 30),  $r_1 = 0.015$  ( $F_3$  取 0.005). 4 种算法求解复杂函数优化问题的统计结果见表 5.

在种群规模相同的情况下, CSAB3M 每一代的函数计算次数与 RCQEA10 大致相当. 这样,  $F_1, F_4$  和  $F_5$  函数只运行了 1000 代便取得了较 RCQEA10 运行了 5000 代更好的结果.  $F_3$  函数相当于运行了 3000 代

表5 各种算法求解复杂函数优化问题的统计结果

函数	算法	平均值	最优值	最劣值	方差值
$F_1$	QEA	2.948 9	1.809 4	4.183 9	0.886 1
	M-ES	7.4E-12	3.5E-13	3.3E-11	8.2E-12
	RCQEA10	4.4E-21	7.0E-24	4.9E-20	9.6E-21
	CSAB3M	0	0	0	0
$F_3$	QEA	0.985 2	0.921 6	1.045 9	0.051 9
	M-ES	0.157 2	0.056 3	0.370 9	0.087 5
	RCQEA10	0	0	0	0
	CSAB3M	0	0	0	0
$F_4$	QEA	0.853 2	0.335 5	1.251 6	0.340 3
	M-ES	1.3E-6	1.3E-7	3.0E-6	6.7E-7
	RCQEA10	9.5E-12	1.5E-12	2.5E-12	6.3E-13
	CSAB3M	7.4E-16	0	1.1E-14	7.6E-30
$F_5$	QEA	45.51	28.02	58.30	11.7352
	M-ES	0.094 3	1.0E-12	0.995 0	0.286 2
	RCQEA10	2.8E-14	0	5.6E-14	2.8E-14
	CSAB3M	0	0	0	0

便取得了较RCQEA10运行5000代更好的结果。CSAB3M的性能表现更优,原因在于对上代变异信息的记忆和学习,用以指导后代抗体的产生,加快了收敛速度,提高了解的精度。表5的数据分析表明,与QEA, M-ES和RCQEA10算法相比,CSAB3M求解复杂函数优化时表现出更快的收敛速度、更强的全局搜索能力、更好的稳定性和鲁棒性。

## 4 结 论

本文提出的基于变异记忆矩阵的克隆选择算法,其核心是在随机进化过程中保留了祖辈探索的信息并用以指导子代的进化,类似于人类社会中的学习和教育。与传统的进化算子只侧重在个体的当代基因表现不同,算法保留并有效地利用了祖辈的探索过程和探索中获取的经验。仿真结果表明,CSAB3M在与其他多种进化算法的比较中表现出了更优的性能。如何进一步提高对进化经验的学习和利用,以及在其他进化算法中运用这一原理是尚待研究和解决的问题。

## 参考文献(References)

- [1] De Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. IEEE Trans on Evolutionary Computation, 2002, 6(3): 239-251.
- [2] 戚玉涛, 刘芳, 焦李成. 基于信息素模因的免疫克隆选择函数优化[J]. 计算机研究与发展, 2008, 45(6): 991-997. (Qi Y T, Liu F, Jiao L C. A pheromone meme based immune clonal selection algorithm for function

- optimization[J]. J of Computer Research and Development, 2008, 45(6): 991-997.)
- [3] 陈皓, 崔杜武, 李雪, 等. 交叉点规模的优化与交叉算子性能的改进[J]. 软件学报, 2009, 20(4): 890-901. (Chen H, Cui D W, Li X, et al. Optimization crossover scale for improving performance of crossover operator[J]. J of Software, 2009, 20(4): 890-901.)
- [4] 王湘中, 喻寿益. 适用于高维优化问题的改进进化策略[J]. 控制理论与应用, 2006, 23(1): 148-151. (Wang X Z, Yu S Y. Improved evolution strategies for high-dimensional optimization[J]. Control Theory & Applications, 2006, 23(1): 148-151.)
- [5] Fabricio Olivetti de Franca, Fernando J Von Zuben, Leandro Nunes de Castro. An artificial immune network for multimodal function optimization on dynamic environments[C]. Proc of the 2005 Genetic and Evolutionary Computation Conf. Washington, 2005: 289-296.
- [6] Vincenzo Cutello, Giuseppe Nicosia, Mario Pavone. Real coded clonal selection algorithm for unconstrained global optimization using a hybrid inversely proportional hypermutation operator[C]. Proc of the 2006 ACM Symposium on Applied Computing. France: ACM Press, 2006: 950-954.
- [7] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies[J]. Evolutionary Computation, 2001, 9(2): 159-195.
- [8] Petr Posik. Real-parameter optimization using mutation step co-evolution[C]. IEEE Congress on Evolutionary Computation. Edinburgh: IEEE, 2005: 872-879.
- [9] 高辉, 徐光辉, 张锐, 等. 实数编码量子进化算法[J]. 控制与决策, 2008, 23(1): 87-90. (Gao H, Xu G H, Zhang R, et al. Real-coded quantum evolutionary algorithm[J]. Control and Decision, 2008, 23(1): 87-90.)
- [10] Zhenguo Tu, Yong Lu. A robust stochastic genetic algorithm(StGA) for global numerical optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(5): 456-470.
- [11] André V Abs da Cruz, Marley M B R Vellasco, Marco Aurélio C Pacheco. Quantum-inspired evolutionary algorithm for numerical optimization[C]. IEEE Congress on Evolutionary Computation. Vancouver: IEEE, 2006: 2630-2637.