

基于克隆选择的免疫粒子群优化算法^{*})

李莉¹ 李洪奇¹ 谢绍龙² 李雄炎²

(中国石油大学计算机科学与技术系 昌平 102249)¹ (中国石油大学资源信息学院 昌平 102249)²

摘要 粒子群优化算法在进化中随种群多样性降低易出现早熟收敛等问题。针对这一问题,在粒子群算法中引入免疫克隆选择算法的思想,提出了基于克隆选择的免疫粒子群优化算法(Immune Particle Swarm Optimization, ImmunePSO),即在算法进化过程中,引入克隆复制算子、克隆高频变异算子、克隆选择算子。成比例克隆复制可以使优良个体得到保护,加快算法收敛;高频变异为新个体的产生提供了新的途径,可以增加种群的多样性;克隆选择算子从所有子代、父代中选择出最优个体,避免算法退化。最后通过对基本测试函数的仿真试验,验证了算法不仅可以增加种群的多样性,加快算法的收敛速度,而且提高了最优解的精度,有效地避免算法陷入到局部极值。

关键词 粒子群算法,种群多样性,克隆复制,克隆选择,高频变异,亲和力

Immune Particle Swarm Optimization Algorithms Based on Clone Selection

LI Li¹ LI Hong-qi¹ XIE Shao-long² LI Xiong-yan²

(Department of Computer Science and Technology, China University of Petroleum, Beijing 102249, China)¹

(School of Resource and Information, China University of Petroleum, Beijing 102249, China)²

Abstract There exists the disadvantages such as prematurity in particle swarm optimization because of the decrease of swarm diversity. In order to solve this problem, an immune particle swarm optimization (ImmunePSO) algorithm is proposed, which is combined with immune clone selection algorithm. Clone copy operator, clone hypermutation operator and clone selection operator are performed during the evolutionary. Proportion clone copy according to particles' affinity can protect eximious individuals and speed up convergence, clone hypermutation provides a new mechanism producing new ones and maintaining diversity, clone selection which select best individuals can avoid algorithm degenerate effectively. The typical benchmark functions are performed. The numerical simulation results show that the improved algorithm not only can maintain swarm's diversity, speed up convergence speed but also help the algorithm escape from local extremum.

Keywords Particle swarm optimization, Diversity of swarm, Clone copy, Clone selection, Hypermutation, Affinity

1 引言

粒子群算法^[1,2] (Particle Swarm Optimization, PSO)是一种模拟鸟群觅食行为而设计的群智能优化算法,由于其实现简单、易于理解,得到了广泛的应用。然而,标准 PSO(Standard PSO, SPSO)算法在进化过程中由于种群多样性的降低,容易导致算法出现早熟收敛等问题。为此, Silva 等提出了掠夺-被掠夺模式,保持种群的多样性^[3]; Zhang 等通过重新初始化粒子的速度维持种群多样性^[4]; Krink 等提出了一种冲突策略避免种群的拥挤^[5]; Løvbjerg 利用自组织策略(self-organized criticality)增加种群多样性^[6]; Riget 等根据种群多样性的测量,在进化过程中交替执行吸引与排斥粒子群算法^[7]。

免疫算法(Immune Algorithms, IA)^[8]是一种借鉴生物免疫系统特性而形成的启发式搜索算法,具有保持解群多样性的特性,避免算法陷入到局部最优解的优点。因此本文在粒子群算法中引入免疫算法的思想,提出了一种基于克隆选择的免疫粒子群优化算法(Immune Particle Swarm Optimization, ImmunePSO),增加种群多样性,改进 PSO 算法的搜索

能力,提高 PSO 算法的优化效果。

本文的组织结构如下:第 2 部分简单叙述 SPSO 算法的实现原理与生物免疫系统原理;第 3 部分详细阐述免疫粒子群优化算法(ImmunePSO)的实现过程;第 4 部分给出试验结果与分析;最后对全文工作进行总结。

2 标准粒子群优化算法与生物免疫系统原理

2.1 标准粒子群优化算法实现原理

标准粒子群优化(Standard Particle Swarm Optimization, SPSO)算法通过粒子群中个体的合作与竞争来实现优化问题的求解。种群中的每个个体称为粒子,每个粒子代表待优化问题的一个可能解。

SPSO 算法首先初始化一群随机粒子(初始解),然后进化(迭代)找到最优解。每个粒子通过跟踪两个“极值”来更新自己:一个极值是粒子本身找到的最优解,这个位置称为个体极值 *pbest*;另一个极值是邻居粒子所找到的最优位置,通常称作 *gbest*。SPSO 算法的数学描述为:

设在一个 *n* 维的搜索空间中,由 *m* 个粒子组成的种群 $X = \{x_1, x_2, \dots, x_m\}$, 其中第 *i* 个粒子位置为 $x_i = \{x_{i1}, x_{i2},$

^{*})基金项目:国家“十五”科技攻关课题(2001BA605A09)。李莉 在读博士研究生;李洪奇 教授,博士,博士生导师;谢绍龙 在读硕士研究生;李雄炎 在读硕士研究生。

$\dots, x_{in}\}^T$, 速度为 $v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}^T$ 。该粒子迄今为止搜索到的最好位置记为 $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}^T$, 其所在邻域内的所有粒子迄今为止搜索到的最好位置为 $p_g = \{p_{g1}, p_{g2}, \dots, p_{gn}\}^T$, 粒子 x_i 的速度和位置更新公式分别为

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 R_1 (p_{id}^t - x_{id}^t) + c_2 R_2 (p_{gd}^t - x_{id}^t) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

式中, $d=1, 2, \dots, n, i=1, 2, \dots, m, n$ 为搜索空间维数(待优化的变量个数); m 为种群规模; t 为当前进化代数; c_1, c_2 表示正的加速常数(acceleration constants); R_1, R_2 表示 $[0, 1]$ 之间的均匀分布随机数。 ω 为惯性权重, 它决定了粒子先前速度对当前速度的影响程度。为了避免粒子飞出搜索空间, 速度 v_i 通常被限制在区域 $[-v_{max}, v_{max}]$ 中, $v_{max} = k \times x_{max}$, 其中 $0.1 \leq k \leq 1.0$ 。

如果粒子的邻域包含整个群体, 则上述算法为全局模式的粒子群算法 g-PSO, 否则为局部模式的粒子群算法 l-PSO。

2.2 生物免疫系统原理

Burnet 于 1959 年提出了克隆选择学说^[9]。该学说认为, 只有那些能完全匹配或部分匹配抗原(Ag)、具备较高亲和力的 B 细胞, 才能被免疫系统选中并对其进行克隆复制, 产生大量的后代, 后代再经过体细胞高频变异、受体编辑等过程实现亲和力的成熟。而那些亲和力低、无法识别抗原的 B 细胞, 将无法获得克隆的机会。这种机制被称为克隆选择和扩增。如果 B 细胞经过体细胞高频变异和受体编辑后出现了退化, 亲和力反而下降了, 则会被免疫系统删除, 这种机制称为克隆删除。然而, 免疫系统中已有的抗体也不一定能识别所有的抗原, 这可能会出现严重的问题。因此, 免疫系统通过动态更新解决该问题, 每天产生 10^7 个新抗体, 不断更新抗体群。虽然大部分新增的抗体没有进入克隆阶段就凋亡了, 但是极少数亲和力高的抗体得到了发展, 这种抗体补充机制提高了免疫系统的抗原识别概率。亲和力成熟后的抗体在完成了对抗原的消灭之后, 克隆规模将受到抑制, 极少数高亲和力的抗体转化为记忆细胞。当免疫系统受到相同或类似的抗原入侵时, 将会再次被激活, 进行克隆扩增, 进入亲和力成熟过程, 此过程称为免疫记忆应答。如此周而复始, 使得生物系统具备了高度的自适应性。

免疫算法(Immune Algorithm, IA)就是受到免疫系统启发而发展起来的一类仿生算法, 该类算法主要是模拟自然界生物免疫系统的机理和功能而实现的。作为免疫算法中基于克隆选择学说(Clonal Selection Theory)而产生的克隆选择算法^[10](Clonal Selection Algorithm, CSA)是模拟生物免疫系统内部的一种微演化(Microevolution)过程的免疫优化算法, 以个体对函数的适应值以及个体本身的浓度的综合作为评价标准, 维持系统的平衡; 采用高频变异维持群体的多样性, 实现多峰值搜索; 利用克隆选择避免算法退化。

3 免疫粒子群优化算法(ImmunePSO)的实现原理

粒子群优化算法在运行过程中, 如果某粒子发现了一个当前最优位置, 其他粒子将迅速向其靠拢, 出现“聚集”现象, 导致种群多样性的降低。如果当前所发现的最优位置是局部最优, 粒子群就无法在解空间内重新搜索, 算法陷入局部最优, 出现早熟收敛现象。本文将免疫克隆选择算法的实现思想引入到粒子群算法, 克服粒子群算法的不足。ImmunePSO 的实现流程图如图 1 所示。从图中可以看到, 在 ImmunePSO 中, 首先通过初始化产生粒子, 然后计算每个粒子的适应度

值, 并根据粒子适应度值的好坏挑选出个体极值 $pbest$ 和全局极值 $gbest$, 然后执行免疫克隆选择算法, 将粒子视作抗体, 计算抗体亲和力, 然后依次执行克隆复制算子、克隆高频变异算子、克隆选择算子, 最后对经过克隆选择之后的粒子执行粒子群算法中速度和位置的更新。如果算法满足结束条件, 则输出结果, 算法停止运行, 否则重复执行。

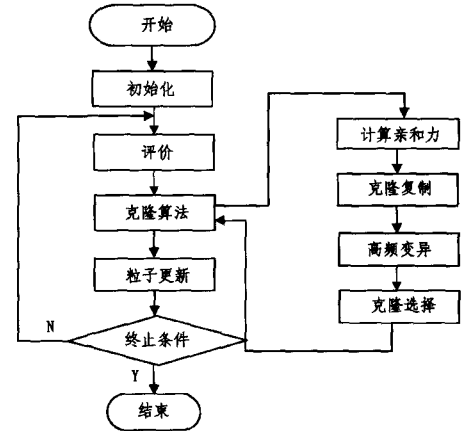


图 1 ImmunePSO 实现流程图

3.1 粒子亲和力的计算

抗体亲和力是种群中抗体优良的度量标准。在 ImmunePSO 中, 将粒子视为抗体, 因此抗体亲和力的计算就是粒子亲和力的计算。综合粒子本身适应度值大小及其所处的位置, 本文将第 i 个粒子的亲和力定义如下:

$$affinity_i = \frac{fitness_i}{dis_i + 1} \quad (3)$$

其中 dis_i 是第 i 个粒子与全局最优粒子 $gbest$ 的距离:

$$dis_i = \sqrt{\sum_{j=1}^{DIM} (p_{ij} - gbest_j)^2} \quad (4)$$

其中, p_{ij} 与 $gbest_j$ 分别是第 i 个粒子与全局最优粒子在第 j 维上的位置。从式(4)中可以看到, 粒子的适应度值越大, 与最优位置越近, 则其亲和力越大, 反之则越小。

3.2 克隆复制算子

根据个体亲和力的大小对种群中的每个个体执行独立的按比例克隆操作。第 i 个粒子被克隆的数目为

$$num_i = \left\lfloor \frac{affinity_i}{\sum_{i=1}^n affinity_i} \cdot n \right\rfloor \quad (5)$$

其中 n 是种群规模。个体的亲和力越大, 个体越优良, 就会克隆出更多的子个体, 保护优良基因, 加快算法收敛速度。

3.3 克隆高频变异算子

对复制的每个子个体, 依概率大小判断是否执行克隆高频变异操作。在粒子群优化算法中, 如果一个粒子发现了“最好的”解, 则其他粒子将飞向它, 聚集在这个粒子周围, 出现“聚集”现象。随着种群进化, 种群的多样性将迅速降低。如果这个“最好的”解是局部极值, 将导致算法陷入到局部极值。在粒子进化中加入克隆高频变异, 提供了粒子飞出局部极值的可能。

高频变异, 即高斯变异(Gaussian Mutation, GM)与柯西变异(Cauchy Mutation, CM), 已经成功地应用在遗传算法中^[12]。高频变异算子提供了产生新解的方法, 使得新解的产生不受其它粒子的影响, 提高了种群的多样性。

在粒子进化过程中, 本文同时采用高斯变异与柯西变异这两种策略。利用高斯变异策略对个体执行小步长变异, 利

用柯西变异对个体执行大的变异。根据个体的好坏,选择执行哪个变异,即对优良个体执行小步长变异——高斯变异,提高解的精度;对适应度值较低的个体执行大步长变异,即柯西变异,帮助逃离局部极值。

(1)高斯变异(Gaussian Mutation, GM)

经过高斯变异,粒子 p_i 将被 p_i' 代替:

$$p_i' = p_i + N(\mu, \sigma) \quad (6)$$

其中, $N(\mu, \sigma)$ 是由高斯方程^[12]产生的一个随机数:

$$f_{\text{Gaussian}}(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left[-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2\right] \quad (7)$$

为简化计算,本文分别取 μ 与 σ 为 0 与 1。因此,高斯变异为

$$p_i' = p_i + N(0, 1) \quad (8)$$

(2)柯西变异(Cauchy Mutation, CM)

柯西密度函数为:

$$f_{\text{Cauchy}}(x) = \frac{t}{\pi(t^2 + x^2)} \quad (9)$$

其中 t 是尺度参数,并且 $t > 0$ 。经过柯西变异后,粒子 p_i 变成 p_i' :

$$p_i' = p_i + \eta \delta_k \quad (10)$$

式中, δ_k 是由柯西方程产生的随机数, η 是修正步长。

3.4 克隆选择算子

粒子经过成比例克隆复制、克隆高频变异后,从父代个体与子代个体中,选择一个适应度最高的最优个体作为下一代个体。父代个体与子代个体的混合,避免了算法退化,其实现过程如图 2 所示。

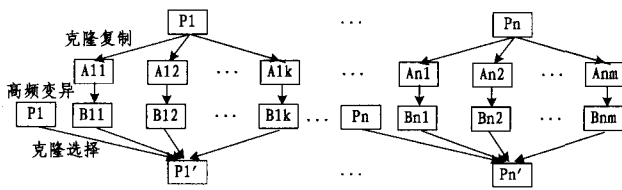


图 2 免疫克隆选择算法的主要操作

图 2 中, p_1, \dots, p_n 是父代个体, p_1 与 p_n 分别被克隆复制了 k 与 m 个,对克隆复制后的个体执行高频变异, p_1', \dots, p_n' 是经过克隆选择后的子代个体。

4 实验验证

测试中,参数设置如下:初始群体规模为 30,最大进化代数取 500 次,惯性权值为 $w = 0.729$,加速常数为 $c_1 = c_2 = 1.49445$ 。为验证算法的准确性,取 30 次独立试验的平均值作为最后的实验结果。

4.1 测试函数

为验证算法是否可以搜寻到所有山峰,本文选用下面几个经典函数进行测试:

(1)Generalized Rosenbrock 函数

$$f(x) = \sum_{n=1}^N [100(x_{n+1} - x_n^2)^2 + (1 - x_n)^2] \quad (11)$$

(2)Generalized Griewank 函数

$$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \frac{x_i}{\sqrt{i}} + 1 \quad (12)$$

(3)Schaffers 函数

$$f(x) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (13)$$

(4)Sphere 函数

$$f(x) = \sum_{n=1}^N x_n^2 \quad (14)$$

(5)Quadric 函数

$$f(x) = \sum_{i=1}^N \left(\sum_{j=1}^i x_j\right)^2 \quad (15)$$

这几个函数的性质如表 1 所示。

表 1 测试函数的性质

函数	定义域	x_{\min}	$f(x_{\min})$
Rosenbrock	$[-10, 10]$	$[0, \dots, 0]$	0
Griewank	$[-600, 600]$	$[0, \dots, 0]$	0
Schaffers	$[-100, 100]$	$[0, 0]$	0
Sphere	$[-100, 100]$	$[0, \dots, 0]$	0
Quadric	$[-100, 100]$	$[0, \dots, 0]$	0

4.2 实验结果

由于篇幅所限,本文只给出部分实验结果,分别从种群多样性、算法收敛速度以及最优解精度三方面来说明算法的效果。

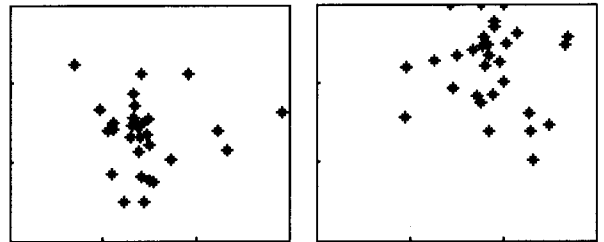


图 3 初始化时 ImmunePSO 粒子分布 图 4 初始化时 SPSO 粒子分布

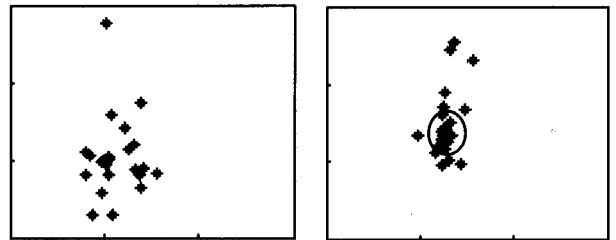


图 5 进化到 50 代时 ImmunePSO 粒子分布 图 6 进化到 50 代时 SPSO 粒子分布

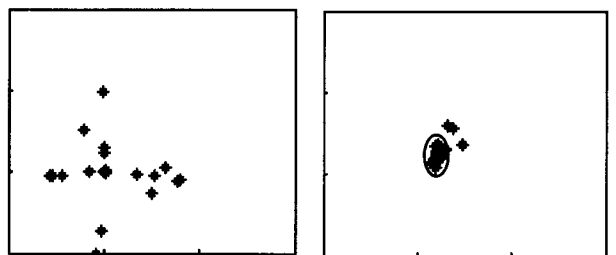


图 7 进化到 100 代时 ImmunePSO 粒子分布 图 8 进化到 100 代时 SPSO 粒子分布

(1)种群多样性分析

以函数 Rosenbrock 为例,图 3-8 给出了函数 Rosenbrock 在二维情况下 SPSO 算法和 ImmunePSO 算法在不同进化代数下的粒子分布图。从图中可以看出:初始化时,SPSO 与 ImmunePSO 算法中的粒子分布都比较均匀;进化到 50 代时,

(下转第 278 页)

业业务逻辑可以迅速地建构在 IT 基础平台上,弥合了企业与 IT 基础设施之间的鸿沟,实现 IT 企业向业务看齐,从而支持企业能快速灵活地变迁其业务流程,保持企业竞争力。

参考文献

[1] 韩光臣,王军强,孙书栋. 企业 CRM 与 ERP 系统信息集成. 航空制造技术,2004,8:57-59
 [2] 毛新生. SOA:原理·方法·实践,第 1 部分:SOA 的基本概念. <http://www.ibm.com/developerworks/cn/WebServices/>

0708_xinsheng/index1.html,2007,9,18

[3] 毛新生. SOA:原理·方法·实践,第 3 部分:业务流程简述. http://www.ibm.com/developerworks/cn/WebServices/0708_xinsheng/index3.html,2008,9,18
 [4] 卢致杰,覃正,韩景侗,等. SOA 体系设计方法与研究. 工业工程,2004,7(6):14-19
 [5] Zimmermann O,Krogdahl P,Gee C. 面向服务的分析与设计原理. <http://www.ibm.com/developerworks/cn/WebServices/ws-soad1/index.html>,2004,06,01

(上接第 255 页)

从图 6 可以看出 SPSO 中,粒子开始“聚集”,种群的多样性比图 5 差很多;进化到 100 代,SPSO 算法中粒子完全“聚集”在某个区域,已经趋于“收敛”,而 ImmunePSO 仍然保持着比较好的群体多样性,算法可以继续探索更多的搜索空间,寻找可能的极值。

(2) 算法收敛速度分析

图 9—10 给出了 Sphere 函数与 Quadric 函数 30 维的运行结果,从图中可以看出:ImmunePSO 的收敛速度远远快于 SPSO。

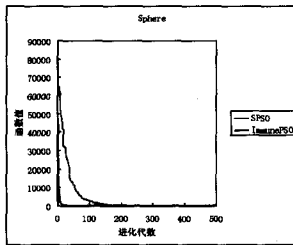


图 9 Sphere 函数 30 维的运行结果

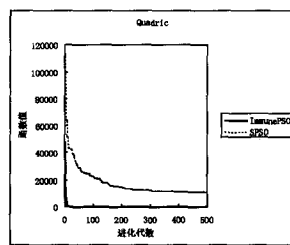


图 10 Quadric 函数 30 维的运行结果

(3) 最优解的精度

表 2,3 分别给出了 SPSO 与 ImmunePSO 的运行结果。表中第一列给出测试函数的维数,2—6 列分别给出算法计算的各个测试函数在不同维数下的全局最小值。可以看到,ImmunePSO 的运算精度比 SPSO 高很多,尤其是在高维情况下,寻优变得更加困难,SPSO 已无法找到全局最优解,而 ImmunePSO 仍然取得了较好的精度。例如在 30 维的情况下,SPSO 找到的 Rosenbrock 的最小值为 115862.84,ImmunePSO 找到的最小值为 0.94E-4。

表 2 SPSO 实验结果

维数	Rosenbrock	Griewank	Schaffer	Sphere	Quadric
2	0.00	0.00	0.00	0.00	0.00
5	0.07	0.012		0.00	0.00
10	2502.22	0.17		0.00	0.00
20	55523.24	0.074		0.12E-3	5003.61
30	115862.84	0.2E-3		1.47	10700.30

表 3 ImmunePSO 实验结果

维数	Rosenbrock	Griewank	Schaffer	Sphere	Quadric
2	0.00	0.00	0.00	0.00	0.00
5	0.30E-6	0.30E-6		0.30E-6	0.30E-6
10	0.66E-5	0.66E-5		0.66E-5	0.66E-5
20	0.00	0.00		0.00	0.00
30	0.94E-4	0.94E-4		0.94E-4	0.94E-4

结束语 粒子群算法在优化过程中,随着粒子的进化粒

子向着最优粒子靠近,出现“群聚”现象,使得种群多样性降低,最终导致粒子群优化算法早熟收敛。免疫算法是受到免疫系统启发而发展起来的仿生算法,是模拟自然界生物免疫系统的机理和功能而设计的算法,具有保持种群多样性的特点。因此,本文在粒子群算法中融合免疫克隆选择算法的思想,首先根据粒子的亲和力对粒子执行按比例克隆复制,然后对克隆后的个体进行克隆高频变异,最后进行克隆选择,即从父代个体与子代个体中选择出最佳个体。成比例克隆复制可以使优良个体得到保护,加快算法收敛;高频变异为新粒子的产生提供了新的方法,维持种群多样性;克隆选择操作有效地避免了算法退化。最后通过对基本测试函数的仿真试验,验证了算法可以增加种群的多样性,加快算法收敛速度,提高最优解的精度。

参考文献

[1] Kennedy J,Eberhart R C. Particle swarm optimization [A]// Proceedings of the 1995 IEEE International Conference on Neural Networks [C]. New York, USA; IEEE,1995;1942-1948
 [2] Shi Y,Eberhart R C. A modified particle swarm optimizer [A] //Proceedings of the 1998 IEEE International Conference on Evolutionary Computation [C]. Piscataway, USA; IEEE,1998; 67-73
 [3] Silva A,Neves A,Costa E. An empirical comparison of particle swarm and predator prey optimization// Lecture Notes in Computer Science, vol. 2464. Berlin; Springer,2002;103-110
 [4] Zhang W J,Xie X F,Yang Z L. Hybrid particle swarm optimizer with mass extinction//International Conference on Communication,Circuits and Systems. 2002;1170-1173
 [5] Krink T,Vesterstrøm J S,Riget J. Particle swarm optimization with spatial particle extension//Proceedings of the Congress on Evolutionary Computation. 2002;1474-1479
 [6] Løvbjerg M, Krink T. Extending particle swarm optimisers with self-organized criticality// Proceedings of the Congress on Evolutionary Computation. 2002;1588-1593
 [7] Riget J, Vesterstrøm J S. A diversity-guided particle swarm optimizer—the ARPSO. Technical Report 2002-02, EVALife. Department of Computer Science, University of Aarhus,2002
 [8] Dasgupta D. Artificial Immune System and Their Applications [M]. Springer-Verlag,1999
 [9] Burnet F M. The Clonal Selection Theory of Acquired Immunity [M]. Cambridge University Press,1959
 [10] DeCastro L N,Zuben V. Learning and optimization using the clonal selection. issue on Artificial Immune System (AIS), 2002,6(3):239-351
 [11] Tiwari M K,Prakash K A,Mileham A R. Determination of an optimal assembly sequence using the PsychoCA. Journal of Engineering Manufacture,2005,219:137-149
 [12] Yao X,Liu Y. Evolutionary programming made faster. IEEE Transactions on Computation,3(2):82-102