

抗独特型克隆选择算法^{*}

张立宁^{1,2+}, 公茂果^{1,2}, 焦李成^{1,2}, 马文萍^{1,2}

¹(西安电子科技大学 智能信息处理研究所, 陕西 西安 710071)

²(西安电子科技大学 智能感知与图像理解教育部重点实验室, 陕西 西安 710071)

Clonal Selection Algorithm Based on Anti-Idiotypic

ZHANG Li-Ning^{1,2+}, GONG Mao-Guo^{1,2}, JIAO Li-Cheng^{1,2}, MA Wen-Ping^{1,2}

¹(Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China)

²(Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University, Xi'an 710071, China)

+ Corresponding author: E-mail: liningzh@126.com

Zhang LN, Gong MG, Jiao LC, Ma WP. Clonal selection algorithm based on anti-idiotypic. Journal of Software, 2009,20(5):1269–1281. <http://www.jos.org.cn/1000-9825/3266.htm>

Abstract: Based on the antibody clonal selection theory of immunology, an artificial immune system algorithm, clonal selection algorithm based on anti-idiotypic (AICSA), is proposed to deal with complex multi-modal optimization problems by introducing the anti-idiotypic. This algorithm evolves and improves the antibody population through clonal proliferation, anti-idiotypic mutation, anti-idiotypic recombination and clonal selection operation, which can perform global search and local search in many directions rather than one direction around the identical antibody simultaneously. Theoretical analysis proves that AICSA can converge to the global optimum. By introducing the anti-idiotypic, AICSA can make the most of the structure information of antibodies, accelerate the convergence, and obtain the global optimization quickly. In experiments, AICSA is tested on four different types of functions and compared with the clonal selection algorithm and other optimization methods. Theoretical analysis and experimental results indicate that AICSA achieves a good performance, and is also an effective and robust technique for optimization.

Key words: clonal selection; anti-idiotypic; evolutionary algorithm; artificial immune system; numerical optimization

摘要: 基于免疫学中的抗体克隆选择学说,通过引入抗独特型结构,提出了一种用于求解复杂多峰函数优化问题人工免疫系统算法——抗独特型克隆选择算法.该算法通过克隆增殖操作、抗独特型变异操作、抗独特型重组操

* Supported by the National Natural Science Foundation of China under Grant No.60703107 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2009AA12Z210 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2006CB705700 (国家重点基础研究发展计划(973)); the Program for New Century Excellent Talents in University under Grant No.NCET-08-0811 (新世纪优秀人才支持计划); the Program for Cheung Kong Scholars and Innovative Research Team in University of China under Grant No.IRT0645 (长江学者和创新团队发展计划)

Received 2007-09-04; Accepted 2008-01-29

作和克隆选择操作这 4 个操作算子来实现抗体种群的进化,能够同时在同一抗体周围的多个方向进行全局搜索和局部搜索,具有较强的搜索能力.理论分析表明,抗独特型克隆选择算法具有全局收敛性.抗独特型结构的引入充分利用了优势抗体的结构信息,加快了抗体种群的收敛速度,从而以更快的速度获得全局最优解,同时降低了算法陷入局部极值点的几率.实验部分采用 4 组不同类型的函数对算法性能进行测试.理论分析及实验结果表明,与克隆选择算法等已有算法相比,该算法性能好,求解精度高,鲁棒性强.

关键词: 克隆选择;抗独特型;进化算法;人工免疫系统;数值优化

中图法分类号: TP18 **文献标识码:** A

从 20 世纪以来,优化问题吸引了越来越多不同背景研究人员的注意力,成为科学研究中的一个热点领域.同时,真实世界中处理的问题变得越来越复杂,常规算法已经不能很好地解决这些复杂的研究问题,这时就需要能够解决这些复杂问题的新算法.通常,非约束优化问题可以用以下 D 维最小化函数来表示:

$$\text{Min}f(x), x=[x_1, x_2, \dots, x_D] \quad (1)$$

其中, D 表示优化参数的维数.上面的表达式给出了常规优化问题的统一数学模型.

人工免疫系统借鉴脊椎动物免疫系统的作用机理,特别是高级脊椎动物(主要是人)免疫系统的信息处理模式,以免疫学术语和基本原理构造新的智能算法,为问题的求解提供了新颖的方法.人工免疫系统提供噪声忍耐、无教师学习、自组织、记忆等进化学习机理,结合分类器、神经网络和机器推理等系统的一些优点,其研究成果涉及到控制、数据处理、优化学习和故障诊断等许多领域,已成为继神经网络、模糊逻辑和进化计算后人工智能的又一研究热点^[1,2].在人工免疫系统中,克隆选择机理已经引起了研究者的兴趣,一些基于克隆选择学说的算法^[3-5]相继被提出来,但是这些研究大多集中在对生物免疫功能的简单模拟上,对于抗体细胞独特型结构^[6]并没有过多的考虑,因此,现有的人工免疫系统算法并没有完全模拟生物免疫系统的功能.

本文通过将抗独特型结构引入克隆选择算法,构造了一种人工免疫系统算法——抗独特型克隆选择算法(clonal selection algorithm based on anti-idiotypic,简称 AICSA).具体来说,该算法通过抗独特型变异和重组算子来模拟抗体的独特型结构,从而实现了对生物免疫功能的模拟.仿真实验结果表明,相对于 de Castro 等人的克隆选择算法及其他已有的优化算法,抗独特型克隆选择算法在处理复杂函数优化问题时具有更高的求解精度和更强的鲁棒性.

本文第 1 节介绍有关人工免疫系统的一些基本知识.第 2 节详细描述抗独特型克隆选择算法,并对其收敛性进行分析.第 3 节首先介绍测试函数的形式,并对抗独特型克隆选择算法的参数进行分析,同时给出对测试函数的仿真实验和分析.最后一节是本文的结论和进一步的研究工作.

1 人工免疫系统

1.1 克隆选择学说

1958 年, Burnet 等人提出了克隆选择学说^[7],其中心思想为:抗体是天然产物,以受体的形式存在于细胞表面,抗原可与之选择性地反应.抗原与相应抗体受体的反应可导致细胞克隆性增殖.该群体具有相同的抗体特异性,其中某些细胞克隆分化为抗体生成细胞,另一些形成免疫记忆细胞以参加以后的二次免疫反应.在此过程中,主要借助克隆使其激活、分化和增殖,以增加抗体的数量,通过进行免疫应答最终清除抗原.因此,克隆选择是生物免疫系统自适应抗原刺激的动态过程,在这一过程中所体现出的学习、记忆、抗体多样性等生物特性^[5],正是人工免疫系统所借鉴的.

1.2 克隆选择算法

根据 Burnet 的抗体克隆选择学说, de Castro 以及 Kim 等人从不同的角度模拟上述生物学抗体克隆选择机理,相继提出了不同的克隆选择算法^[3-5].本文重点描述文献[3]的克隆选择算法,并给出 de Castro 提出的克隆选择算法的流程.

克隆选择算法(clonal selection algorithm).

- Step 1. 生成初始种群(P),(P)由一组记忆单元(M)和保留种群(P_r)构成,即($P=P_r+M$).
- Step 2. 依据抗体-抗原亲和度,从中选择 n 个抗体细胞组成抗体种群(P_n).
- Step 3. 针对抗体种群(P_n)执行克隆操作,得到临时种群(C),克隆规模与抗体-抗原亲和度成正比.
- Step 4. 对克隆临时种群进行高频超变异,变异概率与亲和度成正比.变异操作之后获得的抗体种群为(C^*).
- Step 5. 从(C^*)重新选择改进的个体组成记忆单元 M , P 中的一些个体也被(C^*)其他一些改进个体替代.
- Step 6. 利用新产生的抗体代替 d 个旧抗体,亲和度低的抗体更容易被替代.

上述 de Castro 的克隆选择算法正是对抗体克隆选择机理的简单模拟.它是最早的对生物免疫系统的简单模拟,也是最典型的人工免疫系统算法,后续的许多人工免疫系统算法都是在该算法基础上加以改进而得到的.

1.3 抗独特型结构

通过生物免疫学机理,我们了解到抗体是 B 细胞识别抗原后克隆增殖分化为浆细胞所产生的一种蛋白质分子,也称为免疫球蛋白分子.抗体结合由外界入侵的感染微组织或者有毒素等抗原,然后依靠自己或者借助免疫系统其他元素(T 细胞、MHC 分子等)帮助破坏这些抗原,消除对人体的威胁.抗体具有两种截然不同的功能区:一部分是保持相对静态的稳定区(constant),称为稳定区,简称 C 区;而另一部分则是负责与不同的多种感染抗原结合的分子变化区(variable),称为可变区,简称 V 区.可变区基本上只关心抗原结合.稳定区与宿主组织上的受体相互作用.C 区变化有限,V 区的变化使得免疫系统在自适应处理中具有一定的能力和速度.对免疫应答期间产生的抗体多样性的研究表明,V 区的(细胞分子物质)体细胞(somatic)变异随时间增加,正是可变区为免疫系统提供了大部分鲁棒性和自适应能力.抗体的结构如图 1 所示.

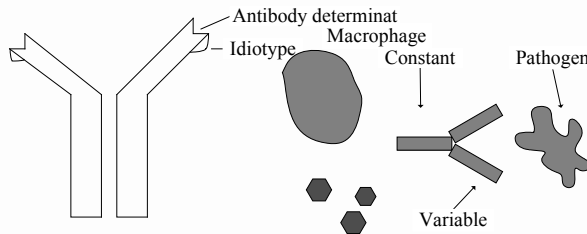


Fig.1 Structure of the antibody

图 1 抗体的结构

从图 1 可以看出,抗体部分是由抗体决定基和独特型(idiotypic)组成.由生物免疫学机理可知,抗体与抗原之间的匹配程度越接近,分子结合强度越大,识别效果越好.抗体也能与其他抗体结合,因为它们既含有抗体决定基,也含有抗原决定基,即独特型抗原决定基(简称独特型).不同的特异性抗体又具有不同的独特型抗原决定基,它不仅能够激发不同物种或者同种异体产生相应的抗体,也能在自身体内刺激其他克隆的细胞产生抗独特型抗体^[6].因此,抗体独特型在免疫系统中起着识别抗原、产生抗体、消灭抗原的作用,抗体独特型结构在生物免疫系统中的作用不容忽视.本文正是基于免疫学中的这一现象而设计出一种人工免疫系统算法.

2 抗独特型克隆选择算法

2.1 抗独特型克隆选择操作算子

本文根据 Burnet 的抗体克隆选择学说,并结合生物免疫学中抗体的独特型结构,提出了一种人工免疫算法——抗独特型克隆选择算法(简称 AICSA).它模拟了生物免疫系统的作用机理,其状态转移图形式如下:

$$\text{AICSA: } A(k) \xrightarrow{\text{克隆增殖}} A'(k) \xrightarrow{\text{抗独特型变异}} A''(k) \xrightarrow{\text{抗独特型重组}} A'''(k) \xrightarrow{\text{克隆选择}} A(k+1)$$

下面分别对这 4 个操作算子进行简单介绍.

(1) 克隆增殖算子 T_c^C

在人工免疫系统中,克隆增殖算子按如下方式进行定义:

$$T_c^C(A(k)) = [T_c^C(a_1(k)) T_c^C(a_2(k)) \dots T_c^C(a_{NP}(k))]^T \tag{2}$$

其中, $T_c^C(a_i(k)) = I_i \times a_i(k), i=1,2,\dots, NP, I_i$ 为元素为 1 的 q_i 维行向量,称为 a_i 的 q_i 克隆.

$$q_i(k) = g(N_c, f(a_i(k))) \tag{3}$$

一般取

$$q_i(k) = \text{Int} \left(N_c \times \frac{f(a_i(k))}{\sum_{j=1}^{NP} f(a_j(k))} \right), j=1,2,\dots, NP \tag{4}$$

$N_c > NP$ 是与克隆规模有关的设定值, $\text{Int}(x)$ 表示大于 x 的最小整数.由此可见,对于单一抗体而言,其克隆规模是根据抗体-抗原亲和度而自适应调整的.在本文后面的实验中,为简便起见,文章中选取固定的克隆规模,即 $q_i(k)$ 为一个固定整数 nc .克隆增殖过后,种群变为

$$A'(k) = \{a'_1(k), a'_2(k), \dots, a'_{NP}(k)\} \tag{5}$$

其中,

$$a'_i(k) = \{a_{i1}(k), a_{i2}(k), \dots, a_{inc}(k)\}, a_{ij}(k) = a_i(k), j=1,2,\dots, nc \tag{6}$$

(2) 抗独特型变异算子 T_m^C

在人工免疫系统中, B 细胞与抗原结合后会被激活,激活后即以极高的频率分裂,该过程为体细胞的高频变异,是克隆扩增期间重要的变异方式.根据抗体的独特型结构,抗体可变区为免疫系统提供了大部分的鲁棒性和自适应能力,当抗原决定基发现匹配时,它不仅可以激发个体产生相应的抗体,也能在自身体内刺激其他克隆的 B 细胞产生抗独特型抗体,从而消灭抗原.de Castro 提出的 CSA 中的变异算子通过超变异来实现^[3],这种变异方式通过在抗体空间中进行随机搜索来实现.显然,这种方式的效率是很低的.本文中的抗独特型变异操作结合了抗体自身独特型结构信息,有指导地进行变异.下面对抗独特型变异算子进行简单的介绍.

首先依据概率 P_m^i 对克隆增殖后的群体进行抗独特型变异操作, $A''(k) = T_m^C(A'(k))$.具体地,针对任一抗体 $a_{ij}(k), i=1,2,\dots, NP, j=1,2,\dots, nc$ 按如下操作进行变异(其中, NP 表示抗体种群规模, nc 表示抗体克隆规模):

$$a'_{ij}(k) = a_{ij}(k) + v \times (a_{r_1j}(k) - a_{r_2j}(k)) \tag{7}$$

其中, $i, r_1, r_2 \in \{1,2,\dots, NP\}$ 为互不相等的整数.因此,必须满足 $NP \geq 3$ 的条件. $v \in [0, 1]$ 为功能系数,表示抗独特型结构在产生新抗体时的作用.图 2 给出了一个二维向量个体 $a_{ij}(k)$ 抗独特型变异操作的示意图.

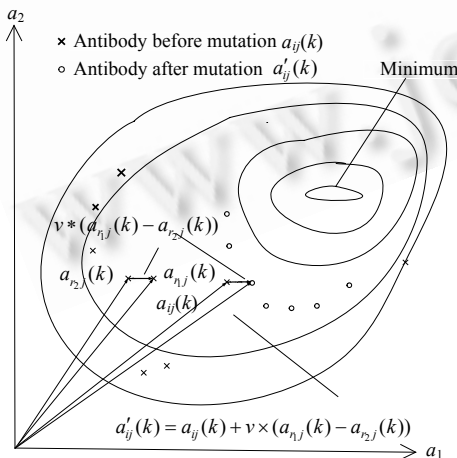


Fig.2 Illustration of the anti-idiotypic mutation operator
图 2 抗独特型变异算子示意图

图 2 采用二维函数等值线图来表示测试函数的函数值, x 和 o 表示了抗体在空间中的分布.抗体采用与原点的向量表示, 3 条实线向量表示变异前抗体, 点划线表示变异生成抗体.显然,抗独特型变异操作通过利用抗体信息,采用差值获得抗体的独特型结构信息,从而指导抗体的变异.图 2 中采用 x 表示当代抗体,采用 o 表示变异生成抗体.可以看出,抗独特型变异操作使变异生成的抗体更加靠近空间中的最优极值点.因此,抗体独特型结构信息的引入,为抗体的进化提供了有效的路径,从而实现了抗体种群的快速、有效变异.

(3) 抗独特型重组算子 T_x^C

de Castro 等人的克隆选择算法仅仅通过抗体变异操作来实现抗体亲和度成熟.与此不同,本文的抗

独特型克隆选择算法引入了抗独特型重组算子 T_x^C ,这一重组算子的引入可以增加抗体种群的多样性.抗独特型重组操作作用在原始抗体 $a_i(k)$ 和抗独特型变异抗体 $a'_{ij}(k)$ 之上.给定重组概率 CR ,重组操作为 $A''(k) = T_x^C(A'(k))$.通过抗独特型重组操作,新生成的抗体可以更好地遗传父代抗体的模式,从而保证了抗体优势模式的保留.具体地,图 3 给出了一个 8 维向量抗独特型重组操作示意图.

图 3 表示抗体 $a_i(k)$ 与变异抗体 $a'_{ij}(k)$ 抗独特型重组操作,其中 $Mask$ 表示一维 0,1 数组 $M(j)$,且

$$M(j) = \begin{cases} 0, & \text{if } rand(j) < CR \text{ or } j = d(i) \\ 1, & \text{if } rand(j) > CR \text{ and } j \neq d(i), j = 1, 2, \dots, D \end{cases} \quad (8)$$

其中, D 表示测试问题的维数, $rand(j)$ 表示 $[0,1]$ 之间的随机数, CR 为重组概率, $d(i)$ 为 $[1,D]$ 之间的随机整数.

从图 3 中可以看出,本文提出的抗独特型重组操作与遗传操作中的一致交叉(uniform crossover)^[8]从形式上很相似.但是,本文的操作又有所不同:一致交叉作用在两个父代个体上,生成两个子代个体,而抗独特型重组操作作用在父代抗体和对应的变异个体上.因此,本文的抗独特型重组操作更加有利于保持抗体自身的优势模式,保证抗体种群的多样性.

(4) 克隆选择操作 T_s^C

de Castro 提出的 CSA 在进行选择操作时采用了 (μ, λ) 选择策略^[9,10],即在生成的子代种群中选择与父代个体数目相同的个体组成下一代种群.本文中的抗独特型克隆选择算法采取了 $(\mu+\lambda)$ 选择策略^[9,10],这是由于抗独特型克隆选择算法在变异和重组操作上与 CSA 存在很大的差别,高频率的变异和重组容易导致优势模式的丢失.为使优势模式在子代抗体基因中得以保留,我们采用了 $(\mu+\lambda)$ 选择策略,以使优秀个体得以生存.具体来说,算法在生成的子代抗体种群和父代抗体种群中,根据适应度大小,选择与父代个体数目相同的个体组成下一代抗体种群,可以表示为 $A(k+1) = T_s^C(A(k) + A''(k))$.

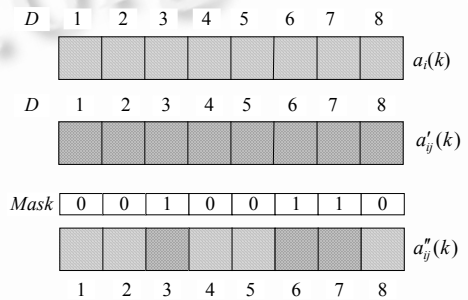


Fig.3 Illustration of the anti-idiotypic recombination operation

图 3 抗独特型重组操作示意图

2.2 抗独特型克隆选择算法(AICSA)

在人工免疫系统算法中,抗体种群的进化都是通过各种各样的操作算子来实现的.通过上述对算法的描述我们可以看出,抗独特型克隆选择算法通过抗体克隆增殖操作、抗独特型变异操作、抗独特型重组操作和克隆选择操作这 4 个操作算子完成对种群的更新和进化.

针对 AICSA 测试问题及问题的所有约束条件记作抗原 F^* ,初始抗体种群随机生成.在本文的算法中,我们采用了实数编码方式,抗体种群记作 $A = \{a_1, a_2, \dots, a_{NP}\}$.抗独特型克隆选择算法的流程如下:

抗独特型克隆选择算法(clonal selection algorithm based on anti-idiotypic).

- Step 1. 初始化算法参数:抗体种群规模 NP ,抗体克隆规模 nc ,抗独特型变异概率 MR ,抗独特型重组概率 CR ,功能系数 v ;随机产生初始抗体种群 $A(0) = \{a_1(0), a_2(0), \dots, a_{NP}(0)\}$;设定亲和度成熟条件;设定当前迭代次数 $k=0$.
- Step 2. 若抗体种群 $A(k)$ 满足亲和度成熟条件,则输出 $A(k)$ 中抗体-抗原亲和度最高的抗体,算法停止;否则,转 Step 3.
- Step 3. 对抗体种群 $A(k)$ 执行克隆增殖操作: $T_c^C(A(k))$, 获得抗体种群 $A'(k)$.
- Step 4. 对抗体种群 $A'(k)$ 执行抗独特型变异操作: $T_m^C(A'(k))$, 获得抗体种群 $A''(k)$.
- Step 5. 对抗体种群 $A''(k)$ 执行抗独特型重组操作: $T_x^C(A''(k))$, 获得抗体种群 $A''(k)$.
- Step 6. 计算抗体 $A''(k)$ 的亲和度: $A''(k) : \{\Phi(A''(k))\}$.
- Step 7. 根据亲和度大小,针对抗体种群 $A''(k)$ 和抗体种群 $A(k)$ 执行克隆选择操作: $T_s^C(A(k) + A''(k))$, 生成下

一代抗体种群 $A(k+1); k:=k+1$, 转 Step 2.

2.3 算法的收敛性分析

定义 1. 函数最小化问题的最优抗体集合为

$$A^* \triangleq \{a^* : \neg \exists a \neq a^*, f(a) > f(a^*)\} \tag{9}$$

对于抗体种群 A , 令 $\mathcal{G}(A) \triangleq |A \cap A^*|$ 表示抗体种群 A 中包含最优抗体的个数.

定义 2. 如果对于任意的初始状态 A_0 , 均有

$$\lim_{k \rightarrow \infty} p\{A(k) \cap A^* \neq \emptyset \mid A(0) = A_0\} = \lim_{k \rightarrow \infty} p\{A(k) \in I^* \mid A(0) = A_0\} = 1 \tag{10}$$

也就是说,

$$\lim_{k \rightarrow \infty} p\{\mathcal{G}(A(k)) \geq 1 \mid A(0) = A_0\} = 1 \tag{11}$$

则称算法以概率 1 收敛到最优种群集.

定理 1. 抗独特型克隆选择算法以概率 1 收敛.

证明: 记 $P_0(k) = P\{\mathcal{G}(A(k)) = 0\} = P\{A(k) \cap A^* = \emptyset\}$, 由贝叶斯条件概率公式^[11]有

$$P_0(k+1) = P\{\mathcal{G}(A(k+1)) = 0\} = P\{\mathcal{G}(A(k+1)) = 0 \mid \mathcal{G}(A(k)) \neq 0\} \times P\{\mathcal{G}(A(k)) \neq 0\} + P\{\mathcal{G}(A(k+1)) = 0 \mid \mathcal{G}(A(k)) = 0\} \times P\{\mathcal{G}(A(k)) = 0\} \tag{12}$$

由克隆选择的性质可知, 克隆选择获得的最优解至少比克隆前要好, 因此

$$P\{\mathcal{G}(A(k+1)) = 0 \mid \mathcal{G}(A(k)) \neq 0\} = 0 \tag{13}$$

所以

$$P_0(k+1) = P\{\mathcal{G}(A(k+1)) = 0 \mid \mathcal{G}(A(k)) = 0\} \times P_0(k) \tag{14}$$

又

$$P\{\mathcal{G}(A(k+1)) = 1 \mid \mathcal{G}(A(k)) = 0\}_{\min} > 0 \tag{15}$$

记 $\zeta = \min_k P\{\mathcal{G}(A(k+1)) = 1 \mid \mathcal{G}(A(k)) = 0\}_{\min}, k = 0, 1, 2, \dots$, 因此

$$P\{\mathcal{G}(A(k+1)) = 1 \mid \mathcal{G}(A(k)) = 0\} \geq \zeta > 0 \tag{16}$$

所以

$$\begin{aligned} P\{\mathcal{G}(A(k+1)) = 0 \mid \mathcal{G}(A(k)) = 0\} &= 1 - P\{\mathcal{G}(A(k+1)) \neq 0 \mid \mathcal{G}(A(k)) = 0\} \\ &= 1 - P\{\mathcal{G}(A(k+1)) \geq 1 \mid \mathcal{G}(A(k)) = 0\} \\ &\leq 1 - P\{\mathcal{G}(A(k+1)) = 1 \mid \mathcal{G}(A(k)) = 0\} \\ &\leq 1 - \zeta < 1 \end{aligned} \tag{17}$$

因此

$$0 \leq P_0(k+1) \leq (1-\zeta) \times P_0(k) \leq (1-\zeta)^2 \times P_0(k-1) \leq \dots \leq (1-\zeta)^{k+1} \times P_0(0) \tag{18}$$

因为

$$\lim_{k \rightarrow \infty} (1-\zeta)^{k+1} = 0, 1 \geq P_0(0) \geq 0 \tag{19}$$

所以

$$0 \leq \lim_{k \rightarrow \infty} P_0(k) \leq \lim_{k \rightarrow \infty} (1-\zeta)^{k+1} P_0(0) = 0 \tag{20}$$

故

$$\lim_{k \rightarrow \infty} P_0(k) = 0 \tag{21}$$

因此

$$\lim_{k \rightarrow \infty} P\{A(k) \cap A^* \neq \emptyset \mid A(0) = A_0\} = 1 - \lim_{k \rightarrow \infty} P_0(k) = 1 \tag{22}$$

即 $\lim_{k \rightarrow \infty} p\{\mathcal{G}(A(k)) \geq 1 \mid A(0) = A_0\} = 1$.

于是可以证明抗独特型克隆选择算法是以概率 1 收敛的. □

3 实验分析

为了评价本文提出的抗独特型克隆选择算法(AICSA)在对函数优化时的求解性能,将该算法与克隆选择算法(CSA)^[3]和微分进化算法(DEA)^[12]以及广泛学习粒子群算法(CLPSO)^[13]进行了对比.

3.1 实验测试函数

在实验部分,本文首先采用了文献[13]中所使用的 16 个不同类型的问题对算法性能进行测试.这 16 个测试问题按其性质分为 4 组,分别为单峰测试函数、多峰测试函数、旋转多峰测试函数及复合函数.

表 1~表 4 给出了 4 组测试函数的具体表达形式.有关函数的构造方式和函数的特性参见文献[13].为了更好地测试 AICSA 解决复杂函数的特性,本文还针对近期 Liang 等人提出来的几种其他形式的复合函数^[14]进行了测试.实验中,我们取维数 $D=10$ (记作 10- D)进行测试,表 4 中的两个测试函数正是其中的 CF1 和 CF5.

复合函数是 Liang,Suganthan 和 Deb 等人于 2005 年提出的一组新的复杂多峰测试函数^[14].由于进化算法在求解已有的基准测试函数,如 Sphere,Weierstrass 和 Rastrigin's 等函数时,往往通过利用这些基准函数已知的特性来设计与之具有针对性的操作算子,例如,已有的基准函数的局部极值点多沿着平行的坐标轴,而全局最优点多很多维的坐标值都相同,或最优点位于空间中原点位置等等,这些特点往往会被学者用来设计算法,因此已有的基准函数就往往不能很好地检验算法在未知空间的寻优能力,而现实世界问题往往是未知结构的问题.正是基于以上原因,Liang 等人通过对一组已有的基准测试函数进行平移、旋转、组合等操作构造了一组复合测试函数.这些复合函数形状复杂,局部极值点极多,而且位置不固定,最优值点位置随机分布于空间中,常规算法都难以很好地解决这类复杂问题.基于以上考虑,本文采用了这组测试函数来检验 AICSA 解决复杂多峰函数优化问题的能力.有关函数的构造方法、表达形式及具体特性均可从 <http://www.ntu.edu.sg/home/EPNSugan/>处获得.

Table 1 Unimodal and simple multimodal functions

表 1 单峰测试函数及简单多峰测试函数

Groups	f	Test functions	Optimization
Group A	f_1	$f_1(x) = \sum_{i=1}^D x_i^2$	0
	f_2	$f_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	0

Table 2 Unrotated multimodal functions

表 2 未旋转多峰测试函数

Groups	f	Test functions	Optimization
Group B	f_3	$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	0
	f_4	$f_4(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0
	f_5	$f_5(x) = \sum_{i=1}^D \left[\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))] \right]$	0
	f_6	$f_6(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0
	f_7	$f_7(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi x_i) + 10), y_i = \begin{cases} x_i, & x_i < 1/2 \\ \frac{\text{round}(2x_i)}{2}, & x_i \geq 1/2 \end{cases}, i = 1, 2, \dots, D$	0
	f_8	$f_8(x) = 418.9829 \times D - \sum_{i=1}^D x \sin(x_i ^{1/2})$	0

Table 3 Rotated multimodal functions

表 3 旋转多峰测试函数

Groups	f	Test functions	Optimization
Group C	f_9	$f_9(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D y_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi y_i)\right) + 20 + e, y = M \times x$	0
	f_{10}	$f_{10}(x) = \sum_{i=1}^D \frac{y_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1, y = M \times x$	0
	f_{11}	$f_{11}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (y_i + 0.5))]\right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)], a = 0.5, b = 3, k_{\max} = 20, y = M \times x$	0
	f_{12}	$f_{12}(x) = \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10), y = M \times x$	0
	f_{13}	$f_{13}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10), z_i = \begin{cases} y_i, & y_i < 1/2 \\ \frac{\text{round}(2y_i)}{2}, & y_i \geq 1/2 \end{cases}, i = 1, 2, \dots, D, y = M \times x$	0
	f_{14}	$f_{14}(x) = 418.9829 \times D - \sum_{i=1}^D z_i, z_i = \begin{cases} y_i \sin(y_i ^{1/2}), & y_i \leq 500 \\ 0.001(y_i - 500)^2, & y_i > 500 \end{cases}, y = y + 420.96, y' = M \times (x - 420.96)$	0

Table 4 Composition functions

表 4 复合测试函数

Groups	f	Test functions	Optimization
Group D	f_{15}	CF1	0
	f_{16}	CF5	0

3.2 实验对比算法

针对本文提出的抗独特型克隆选择算法,我们通过与以下几种算法进行对比来验证算法的有效性:

- (1) 克隆选择算法(CSA)^[3].de Castro 的 CSA 是基于克隆选择机理最早的人工免疫系统算法,也是最经典的人工免疫系统算法;
- (2) 微分进化算法(DEA)^[12].DEA 是一种有效的解决全局优化问题的启发式算法,近些年引起了广泛的关注;
- (3) 广泛学习粒子群算法(CLPSO)^[13].CLPSO 是一种新提出来的粒子群算法,文献[13]将其与其他几种经典的粒子群算法进行了对比,并且取得了良好的效果.因此,本文将 AICSA 与 CLPSO 进行了比较,CLPSO 的结果引自文献[13].

3.3 仿真实验结果

3.3.1 参数敏感性分析

本节我们选择两个有代表性的复杂复合多峰函数 f_{15} 和 f_{16} 来分析参数对抗独特型克隆选择算法所产生的影响.为使经过克隆增殖后生成的抗体均能利用抗体独特型结构信息,算法设置抗独特型变异概率 $MR=1$.为获得最优参数设置,本节针对抗独特型克隆选择算法参数 NP, nc, CR 和 v 进行了敏感性分析,分析策略如下:

(1) 抗体种群规模 NP 与克隆规模 nc 的敏感性分析

对参数 NP 在区间[5,50]以步长 5 采样,参数 nc 在区间[2,20]以步长 2 采样,于是获得了 $10 \times 10 = 100$ 组参数.实验停止条件为 $|f_{best}| < \epsilon$,针对每组参数进行 30 次独立实验,然后计算该参数设置条件下函数值平均计算次数.图 4 给出了实验结果,算法的其他参数设置为 $CR=0.8, v=0.8$,最大函数值计算次数设定为 50 000 次.

如图 4 所示,两个函数的函数值计算次数随 NP 和 nc 变化趋势相似.当 NP 和 nc 较小或 NP 和 nc 都很大时,函数 f_{15}, f_{16} 计算次数已达到 50 000 的最大限制条件,而算法却陷入了局部极值点,未能收敛到指定精度.而抗体种群规模的增加可以增加抗体密度,抗体克隆规模的增加可以扩大搜索空间.从图 4 可以看出,当 NP 和 nc 合适时,可以在较少的计算次数条件下获得指定精度.通过对参数的实验数据分析我们认识到,当 NP 取值为 25~35

和 nc 取值为 3~6 时,算法在获得同样计算精度的情况下所需的计算次数较小.因此,实验中设置 $NP=30,nc=5$.

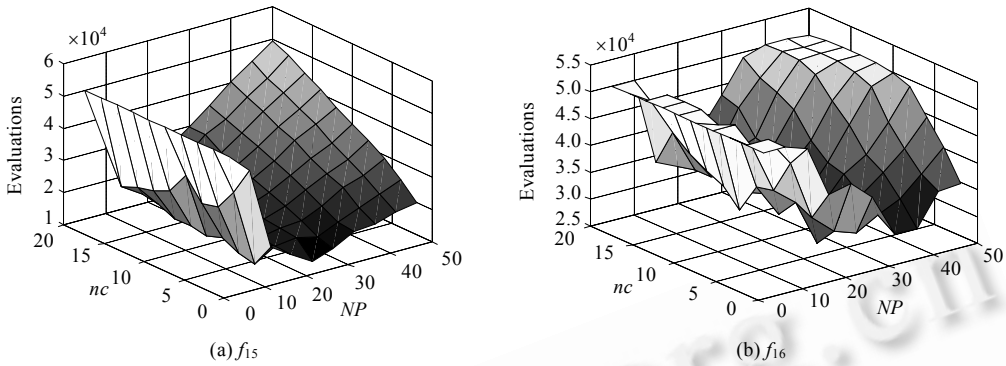


Fig.4 AICSA sensitivity in relation to NP and nc

图 4 AICSA 对 NP 和 nc 的敏感性

(2) 抗体重组概率 CR 和功能系数 v 的敏感性分析

与上面的测试方法相同,参数 CR 在 $[0.1,1]$ 以步长 0.1 采样,参数 v 在 $[0.1,1]$ 以步长 0.1 采样,于是同样获得了 $10 \times 10 = 100$ 组参数.实验停止条件为 $|f_{best}| < \epsilon$,其余参数设置 $NP=30,nc=5$,最大计算次数设定为 50 000 次,图 5 给出的是针对每组参数 30 次随机实验的统计平均结果.

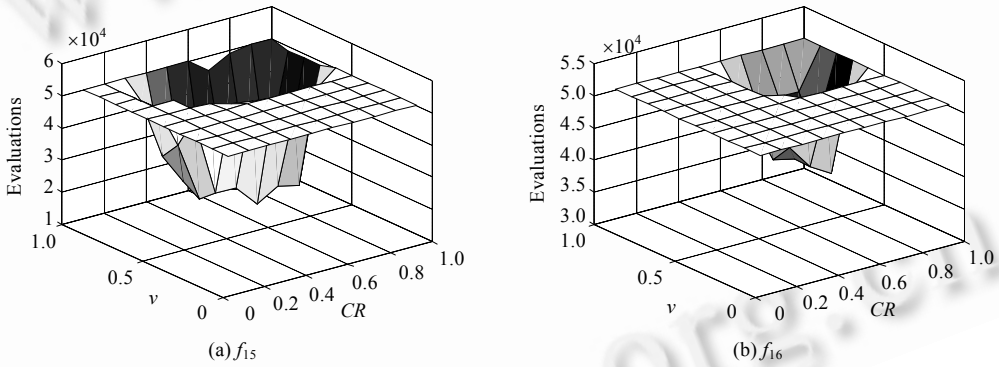


Fig.5 AICSA sensitivity in relation to CR and v

图 5 AICSA 对 CR 和 v 的敏感性分析

与参数 NP 和 nc 分析相似,从图 5 的结果可以看出,重组概率 CR 和功能系数 v 对于算法的性能也有很大影响,当 CR 和 v 取值较小时,算法陷入了局部极值点,在规定计算次数内不能收敛到指定的精度.当 CR 和 v 同时增大时,算法的性能获得了改善.同样, CR 和 v 也存在最优的参数设置,这时,算法可以在获得同样精度的结果条件下,函数值计算次数最小.通过对参数的实验数据分析我们认为, CR 取值在 0.7~0.9, v 取值在 0.7~0.9 时,算法可以在较小代价条件下获得同等结果.下面的实验中,我们取 $CR=0.8$ 和 $v=0.8$.

3.3.2 实验结果与分析

通过上面对算法参数的实验分析,我们获得了最佳参数模型,即 $NP=30,nc=5,CR=0.8$ 及 $v=0.8$.在下面的实验中,我们按照上面获得的参数设置对测试函数进行实验仿真.实验分为 3 个部分:首先针对 16 个测试函数的 10-D 问题进行实验仿真,同时对比了 4 种算法的 30 次实验的均值、方差;接下来针对 16 个测试问题的 30-D 实验函数进行仿真,同时给出对于结果的分析;实验最后部分给出了算法针对 6 个 10-D 复杂复合函数的测试结果.

(1) 10-D 测试问题实验结果.实验中针对 10-D 测试问题,设定停止条件为最大函数值计算次数达到 30 000 次.表 5 给出了 4 种算法针对 4 组测试问题的 30 次随机独立实验的结果.4 种算法中最优的结果加粗显示.

Table 5 Statistical results for 10-D functions

表 5 10-D 测试函数统计结果

Groups	CSA		DEA		CLPSO		AICSA		
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
A	f_1	3.54e+000	1.53e+000	9.55e-013	1.32e-012	5.15e-29	2.16e-028	1.50e-024	2.67e-024
	f_2	1.69e+000	0.63e+000	1.02e-002	8.60e-003	2.46e+000	1.70e+000	4.86e+000	1.64e+000
B	f_3	1.83e+000	0.36e+000	4.80e-007	3.58e-007	4.32e-014	2.55e-014	3.33e-013	3.68e-013
	f_4	0.91e+000	0.10e+000	4.30e-001	7.19e-002	4.56e-003	4.81e-003	2.72e-002	3.90e-002
	f_5	1.17e+000	0.13e+000	7.31e-004	2.87e-004	0	0	0	0
	f_6	1.92e+000	0.55e+000	2.20e+001	4.60e+000	0	0	0	0
	f_7	1.90e+000	0.61e+000	1.37e+001	2.08e+000	0	0	0	0
	f_8	7.66e+000	3.82e+000	1.28e-002	4.45e-002	0	0	0	0
C	f_9	2.45e+000	0.36e+000	5.98e-007	3.66e-007	3.56e-005	1.57e-004	4.85e-013	5.35e-013
	f_{10}	0.90e+000	0.11e+000	5.37e-001	9.71e-002	4.50e-002	3.08e-002	1.93e-001	1.22e-001
	f_{11}	2.71e+000	0.57e+000	1.10e-002	4.50e-003	3.72e-010	4.40e-010	3.09e-013	1.05e-013
	f_{12}	1.61e+001	3.66e+001	3.62e+001	5.49e+001	5.97e+000	2.88e+000	1.41e+001	6.43e+000
	f_{13}	1.05e+001	2.99e+000	2.10e+001	4.24e+001	5.44e+000	1.39e+000	11.0e+001	4.53e+000
	f_{14}	4.81e+002	1.43e+002	9.00e+002	4.47e+002	1.14e+002	1.28e+002	5.52e+002	2.44e+002
D	f_{15}	6.05e+000	3.51e+000	1.73e-012	2.55e-012	1.64e+001	3.63e+001	7.63e-021	1.22e-020
	f_{16}	3.20e+000	9.64e+000	1.20e+001	2.99e+001	1.98e+001	2.93e+001	2.26e+000	1.39e+000

从上面的测试结果可以看出,针对 Group A 中的单峰测试函数,AICSA 并不是 4 种算法中最优的测试结果.这是由于 AICSA 引入了抗独特型变异算子,使得算法扩大了解的搜索空间,但同时也导致了算法针对简单单峰测试函数在相同计算次数条件下并没有取得最好的结果.针对其余 3 组多峰测试函数,AICSA 则在函数 $f_9, f_{11}, f_{15}, f_{16}$ 上取得了最佳结果.针对函数 f_9 算法所得解的平均值为 $4.85e-013$,与其他 3 种算法相比,无论从解的精度函数稳定性上都有很大的提升.同样,针对函数 f_{15} ,算法解的平均值为 $7.63e-021$,而性能一直很好的 CLPSO 为 $1.64e+001$,因此,AICSA 在解决更加复杂的函数时,比 CLPSO 具有更好的特性.同样,在解决函数 $f_5 \sim f_8$ 等时,AICSA 和 CLPSO 取得了同样最优的结果.而 CLPSO 算法在解决函数 $f_3, f_4, f_{10}, f_{12} \sim f_{14}$ 上面取得了最优的结果,但与 AICSA 相比,它们在解的精度和稳定性上很相近,可以认为它们的结果是统计相同的.由上面的分析可以看出,CLPSO 在解决更加复杂的函数时却容易陷入局部最优值,函数 f_{15} 恰好说明了这一点.因此,AICSA 在解决低维多峰函数、旋转多峰函数和复杂函数等问题时,与 CSA,DEA 和 CLPSO 相比均表现出了良好的性能,尤其是针对复杂函数,AICSA 更是可以获得高质量的解,同时表现出了更强的鲁棒性.

(2) 30-D 测试问题实验结果.实验中针对 30-D 测试问题,设定停止条件为最大函数值计算次数达到 200 000 次.表 6 给出了 4 种算法针对 4 组测试问题的 30 次随机统计独立实验结果.4 种算法中最优的结果加粗显示.

Table 6 Statistical results for 30-D functions

表 6 30-D 测试问题统计结果

Groups	CSA		DEA		CLPSO		AICSA		
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
A	f_1	3.71e+000	0.72e+000	1.46e-009	2.40e-009	1.16e-113	2.92e-113	5.90e-035	3.23e-034
	f_2	4.97e+000	1.91e+000	1.21e+001	0.93e-001	2.10e+001	2.98e+000	2.32e+001	2.01e+000
B	f_3	1.04e+000	0.12e+000	1.16e-005	5.71e-006	0	0	4.44e-015	4.44e-015
	f_4	1.03e+000	0.01e+000	1.10e-003	3.40e-003	3.14e-010	4.64e-010	0	0
	f_5	2.36e+000	0.15e+000	1.12e-002	3.80e-003	7.82e-015	8.50e-015	0	0
	f_6	2.08e+000	0.41e+000	1.03e+002	2.44e+001	0	0	0	0
	f_7	2.29e+000	0.54e+000	8.88e+001	2.98e+001	4.36e-010	2.44e-010	0	0
	f_8	1.11e+000	2.34e+000	5.13e+003	1.25e+003	1.27e-012	8.79e-013	0	0
C	f_9	2.21e+000	0.22e+000	1.54e-005	9.49e-006	3.43e-004	1.91e-004	4.44e-015	1.25e-015
	f_{10}	1.03e+000	0.01e+000	1.30e-003	4.10e-003	7.04e-010	1.25e-011	1.24e-009	1.58e-009
	f_{11}	1.26e+001	1.31e+000	1.54e-001	6.05e-002	3.07e+000	1.61e+000	3.81e+000	2.06e+000
	f_{12}	6.68e+001	1.17e+001	2.17e+002	1.13e+001	3.46e+001	4.59e+000	4.44e+001	1.20e+001
	f_{13}	4.00e+001	7.45e+000	1.93e+002	1.51e+001	3.77e+001	5.56e+000	3.72e+001	1.06e+001
	f_{14}	2.38e+003	3.96e+002	7.66e+003	2.99e+002	1.70e+003	1.86e+002	2.73e+003	6.09e+002
D	f_{15}	0.69e-001	0.19e-001	3.80e-010	6.03e-010	7.50e-005	1.85e-004	1.29e-030	9.78e-031
	f_{16}	5.34e+001	0.66e-001	8.76e+001	5.70e+000	7.86e+000	3.64e+000	3.86e+000	1.07e+000

表 6 给出了 AICSA 针对 16 个测试函数的 30-D 问题的测试结果.与 10-D 的测试结果相似,算法针对函数 $f_4, f_5, f_7 \sim f_9, f_{15}$ 的结果无论在解的精度还是稳定性上与其他几种算法相比都有了很大提高,而针对函数 f_4, f_5, f_7 和 f_8 在同样的计算代价下,30 次随机实验均获得了函数的最优结果 0,体现出了算法极强的寻优能力.同样与上面 10-D 的结果相似,针对函数 f_9 ,算法获得了 $4.44e-015$ 的最优结果.与其他几种算法相比,从解的精度提升了 10 个数量级不等.同样,针对更加复杂的复合函数 f_{15} ,与其他几种算法相比,AICSA 同样在解的精度和稳定性上取得了质的提升.但是我们依然可以看出,并不是对于所有的函数 AICSA 均具有最好的特性,针对函数 f_1, f_3 和 f_{10} ,CLPSO 取得了最好的结果,并且精度和稳定性都要优于 AICSA.对于函数 $f_6, f_{12} \sim f_{14}$ 和 f_{16} ,AICSA 和 CLPSO 解的精度和稳定性相近,函数 f_2, f_{11} 最优结果分别为 CSA 和 DEA 取得.

从上面两组的结果可以看出,虽然 AICSA 并不是在所有问题上都有最优的结果,但总体而言,对于多峰复杂函数,AICSA 表现出了良好的性能,一方面能够找到高质量的解,另一方面方差小、算法性能稳定,具有很强的鲁棒性.

(3) 复合函数测试结果

为了进一步验证算法解决复杂多峰函数的有效性,本文针对其余几个多峰复合函数进行了测试.由文献[14]可知,CF1~CF6 为 10-D 测试函数,最优值均为 0.表 7 给出了 7 种算法的实验测试结果,其中最优的结果加粗显示.表中 AICSA 的结果是 100 次随机实验的平均,其余几种算法的结果均是 20 次随机独立实验的平均.在本次实验中,我们增大了抗体种群规模,参数设置为 $NP=40, nc=5, CR=0.8, v=0.8$,实验停止条件为函数值最大计算次数为 50 000.对比算法的实验结果取自文献[14].

Table 7 Comparison results achieved by seven algorithms on six composition functions

表 7 6 个复合函数的 7 种算法对比结果

	CF1		CF2		CF3	
	Mean	Std	Mean	Std	Mean	Std
PSO	1.00e+002	8.17e+001	1.56e+002	1.31e+002	1.7203e+002	3.2869e+001
CPSO	1.56e+002	1.34e+002	2.42e+002	1.49e+002	3.6264e+002	1.9631e+002
CLPSO	5.73e-008	1.04e-007	1.92e+001	1.48e+001	1.33e+002	2.00e+001
CMA-ES	1.00e+002	1.89e+002	1.62e+002	1.51e+002	2.14e+002	7.41e+001
G3-PCX	6.00e+001	6.99e+001	9.27e+001	9.91e+001	3.20e+002	1.25e+002
DEA	6.75e-002	1.11e-001	2.88e+001	8.63e+000	1.44e+002	1.94e+001
AICSA	3.12e-019	4.11e-019	5.21e+000	6.29e+000	1.32e+002	4.48e+001
	CF4		CF5		CF6	
	Mean	Std	Mean	Std	Mean	Std
PSO	3.14e+002	2.01e+001	8.34e+001	1.01e+002	8.61e+002	1.26e+002
CPSO	5.22e+002	1.22e+002	2.56e+002	1.76e+002	8.53e+002	1.28e+002
CLPSO	3.22e+002	2.75e+001	5.37e+000	2.61e+000	5.01e+002	7.78e-001
CMA-ES	6.16e+002	6.71e+002	3.59e+002	1.68e+002	9.00e+002	8.32e-002
G3-PCX	4.93e+002	1.42e+002	2.60e+001	4.16e+001	7.72e+002	1.89e+002
DEA	3.25e+002	1.48e+001	1.08e+001	2.60e+000	4.91e+002	3.95e+001
AICSA	2.85e+002	4.01e+001	0.16e-001	0.54e-001	5.11e+002	7.67e+001

为了更清楚地描述 AICSA 的结果,图 6 给出了 6 个函数 100 次实验所得解的分布图,其中横坐标为函数值,纵坐标为实验次数.

从表 7 及图 6 的结果可以看出,AICSA 在解决复合函数时,与其他几种算法相比表现出了良好的性能,尤其是针对 CF1,AICSA 在 100 次随机独立实验中均获得了 10^{-018} 数量级的解.同时,针对函数 CF5,AICSA 在 100 次随机独立实验中有 90 次解,均为函数的全局最优解,而只有 10 次算法求解结果大于 1,表现出了算法很好的求解特性.同时,针对函数 CF2~CF4,几种算法都未获得函数的最优解,这是由于函数结构非常复杂,表 7 中所列 7 种算法在多次运行中均未能寻找到最优解区域,同时也体现出了复合函数空间结构的复杂和对进化算法性能测试的效果.但是,AICSA 获得的结果依然是 7 种算法中最优的结果.当然,AICSA 对于 CF6 的求解结果与其余几种算法相似,均陷入了局部最优值,未能取得很好的效果.但是,上述实验结果同样表明了 AICSA 很好的求解特性和良好的鲁棒性.

需要说明的是,在前面的实验中,Group D 中的 CF1 和 CF5 由于算法参数和评价次数设置不同,测试结果也

有所不同.在本次实验中,我们增大了抗体种群规模和最大函数值的计算次数,这样同样会导致两个方面的作用:一方面,增大抗体规模会增加每一代种群函数值的评价次数,从而导致算法在同样评价次数条件下算法未能收敛到最优结果;但另一方面,这样会增加抗体种群的密度,减少陷入局部极值区域的几率.因此,CF1 的结果质量相对于上面略有降低,而 CF5 的结果则相对于上面有所提高.

总而言之,通过以上分析可知,AICSA 算法是一种受参数影响很大的人工免疫系统算法,通过设置合适的实验参数,AICSA 在解决复杂函数优化问题时可以获得高质量的解,同时,算法的性能较为稳定,鲁棒性较强.

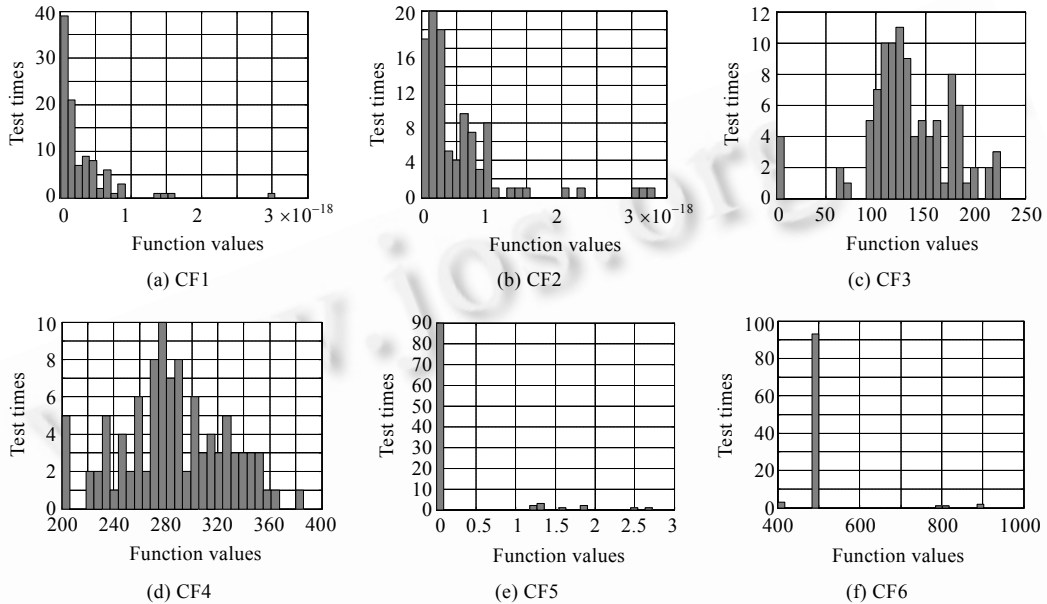


Fig.6 Distribution of the results of the six composition functions

图 6 6 个复合函数解的分布情况

4 结论与展望

基于免疫学中的抗体克隆选择学说,通过引入抗独特型结构,本文提出了一种用于解决复杂多峰函数优化问题的人工免疫系统算法——抗独特型克隆选择算法.AICSA 通过克隆增殖操作,使得算法可以在同一父代抗体周围的多个方向进行全局和局部搜索.抗独特型变异算子的引入充分利用了抗体结构信息,指导了算法有效变异,加快了收敛速度;抗独特型重组算子则使得优势抗体模式信息得以保留,降低了抗体进化后期的早熟现象;克隆选择操作则使生成的优势种群得以保留,保证了算法以概率 1 收敛于最优解.

理论分析和实验结果表明,与 de Castro 等人的 CSA,Storn 等人的 DEA 及 Liang 等人提出的 CLPSO 相比,该算法从求解的精度和稳定性上都显示出了良好的特性.尤其是针对复杂的复合函数,与其他算法相比,该算法更是从精度上有了更加明显的提升,体现了 AICSA 在解决复杂函数优化问题时的优势.而针对 6 个复合函数的求解结果则更显示了 AICSA 很好的寻优特性和良好的鲁棒性.

本文的结果表明,抗独特型克隆选择算法在数值优化问题上取得了很好的效果,说明该算法是非常有潜力的.因此,下一步我们将用该算法解决其他优化问题,如多目标优化、组合优化等.

References:

- [1] Dasgupta D, Forrest S. Artificial immune systems in industrial applications. In: Meech JA, Veiga MM, Smith MH, LeClair SR, eds. Proc. of the 2nd Int'l Conf. on Intelligent Processing and Manufacturing of Materials. Piscataway: IEEE Press, 1999. 257-267.

- [2] Jiao LC, Du HF. Development and prospect of the artificial immune system. *Acta Electronica Sinica*, 2003,31(10):1540–1548 (in Chinese with English abstract).
- [3] de Castro LN, von Zuben FJ. The clonal selection algorithm with engineering applications. In: Langdon WB, Cantu-Paz E, Mathias KE, Roy R, Davis D, eds. *Proc. of the 2002 GECCO, Workshop on Artificial Immune Systems and Their Applications*. San Francisco: Morgan Kaufmann Publishers, 2000. 36–37.
- [4] Kim J, Bentley PJ. Towards an artificial immune system for network intrusion detection: An investigation of clonal selection with a negative selection operator. In: Kim JH, Zhang BT, Fogel G, Kuscus I, eds. *Proc. of the 2001 Congress on Evolutionary Computation*. Piscataway: IEEE Press, 2001. 1244–1252.
- [5] Jiao LC, Du HF, Liu F, Gong MG. *Immunological Computation for Optimization, Learning and Recognition*. Beijing: Science Press, 2006 (in Chinese).
- [6] Mo HW. *The Principles and Applications of Artificial Immune System*. Harbin: Harbin Institute of Technology Press, 2003 (in Chinese).
- [7] Gong MG, Du HF, Jiao LC. Optimal approximation of linear systems by artificial immune response. *Science in China (Series F—Information Sciences)*, 2006,49(1):63–79.
- [8] Chen GL, Wang XF, Zhuang ZQ, Wang DS. *Genetic Algorithm and its Applications*. Beijing: Posts & Telecom Press, 1996 (in Chinese).
- [9] Yao X, Liu Y, Lin GM. Evolutionary programming made faster. *IEEE Trans. on Evolutionary Computation*, 1999,3(2):82–102.
- [10] Beyer HG, Schwefel HP. Evolution strategies—A comprehensive introduction. *Natural Computing*, 2002,1(2):3–52.
- [11] Sheng Q, Xie SQ, Pan CY. *Probability Theory and Mathematical Statistics*. 2nd ed., Beijing: Higher Education Press, 2003 (in Chinese).
- [12] Storn R, Price K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997,11(4):341–359.
- [13] Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. on Evolutionary Computation*, 2006,10(2):281–295.
- [14] Liang JJ, Suganthan PN, Deb K. Novel composition test functions for numerical global optimization. In: Arabshahi P, Martinoli A, eds. *Proc. of the 2005 Swarm Intelligence Symp*. Piscataway: IEEE Press, 2005. 68–75.

附中文参考文献:

- [2] 焦李成,杜海峰.人工免疫系统进展与展望. *电子学报*,2003,31(10):1540–1548.
- [5] 焦李成,杜海峰,刘芳,公茂果. *免疫优化计算、学习与识别*.北京:科学出版社,2006.
- [6] 莫宏伟. *人工免疫系统原理及应用*.哈尔滨:哈尔滨工业大学出版社,2003.
- [8] 陈国良,王煦法,庄镇泉,王东生. *遗传算法及其应用*.北京:人民邮电出版社,1996.
- [11] 盛骤,谢式千,潘承毅. *概率论与数理统计*.第2版,北京:高等教育出版社,2003.



张立宁(1981—),男,河北晋州人,硕士生,主要研究领域为自然计算,人工免疫系统,数据挖掘.



公茂果(1979—),男,博士,副教授,主要研究领域为人工免疫系统,进化计算,数据挖掘,多目标优化.



焦李成(1959—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为自然计算,数据挖掘,图像处理,智能信息处理.



马文萍(1981—),女,博士,讲师,主要研究领域为进化计算,图像处理.