

一种新的免疫算法及其在多模态函数优化中的应用

张著洪^{1,2}, 黄席樾¹

(1. 重庆大学 自动化学院, 重庆 400044; 2. 贵州大学 数学系, 贵州 贵阳 550025)

摘要: 提取免疫应答的部分简化机制并结合小生境技术, 提出一种用于多峰值或非连续函数优化的免疫算法. 该算法由记忆细胞获取、克隆选择、亲和突变及群体更新这四种算子模块构成, 这些算子的有机组合不仅为最优化问题的解决提供了实用新方法, 而且反映了抗体应答抗原的简化运行机制. 算法设计的重点是借鉴小生境共享实现方法的思想建立有助于增强群体多样性及保留优良抗体的记忆细胞获取算子, 以及利用亲和成熟机理设计抗体突变算子. 所获算法具有整体和局部搜索能力及并行搜索特点. 理论证明了其收敛性. 仿真事例比较表明此算法不仅是有效的, 而且能快速搜索到多个最优解(针对于多解最优化问题).

关键词: 免疫算法; 免疫应答; 小生境; 函数优化; 全局收敛

中图分类号: TP3111.51 **文献标识码:** A

Novel immune algorithm and its application to multi-modal function optimization

ZHANG Zhu-hong^{1,2}, HUANG Xi-yue¹

(1. Automatic College, Chongqing University, Chongqing 400044, China;

2. Department of Mathematics, Guizhou University, Guizhou Guiyang 550025, China)

Abstract: A novel immune algorithm, suitable for function optimization problems with either multimodal or noncontinuous objective functions, was proposed based on four immune operators of memory cell keeping, clone selection, somatic mutation and population updating from a niching technique and several immune mechanisms. The algorithm, of which had capability of global and local searching and characteristics of parallel searching, not only provided a practical method for solving optimization problems, but also reflected on some simple performance characteristics of antibodies to antigens. The key of its design was to construct a memory cell-keeping operator based on a niching method that could maintain diversity of population and keep good antibodies of current populations, and to design a mutation operator by utilizing the metaphor of affinity maturation. Further, its convergence was proved in theory. Simulation comparison showed that the algorithm can accomplish effectively optimization tasks, and specially search rapidly more than one of optimal solutions for certain optimization problems.

Key words: immune algorithm; immune response; niche; function optimization; global convergence

1 引言 (Introduction)

人工免疫系统及其应用已成为崭新的研究领域正受到计算智能学者的广泛关注, 它是继神经网络及进化计算之后的又一研究热点^[1,2]. 由于免疫系统能自适应地维持群体多样性及具有自我调节功能, 导致基于免疫机制的算法具有整体、局部搜索能力强的特点, 并且这类算法在函数优化、组合优化、模式识别、数据挖掘及机器学习等方面得到了有效应用. 探讨基于免疫机理的算法主要有两种途径, 一种是基于免疫应答原理的免疫算法 (IAs: immune algorithms)^[3,4]; 另一种是借鉴免疫系统中接种育苗或抗体的促进和抑制机制构建增强群体多样性的算子, 并结合遗传算法提出的免疫遗传算法 (IGAs: immune genetic algorithms)^[5,6]; 接种育苗操作极大地依赖于问题的特征信息; 抗体的促进和抑制操作在鼓励匹配度高的抗体的同时, 抑制浓度高的抗体, 增强群体的多样性, 但此操作对于匹配度和浓度都很高

的抗体将有很大机会被抑制, 从而影响算法的收敛性. 尽管这些 IGAs 各自有不同的应用限制, 但提供了探讨 IAs 的启示. 文献[3]从克隆选择原理出发利用二进制编码及贪婪搜索思想提出克隆选择算法, 此算法并行性高, 但群体多样性不理想, 进化群体有更多机会仅沿一个最优解所在区域转移, 搜索多个最优解的能力较弱, 同时仿真表明, 该算法对于复杂的函数优化问题易于陷入局部搜索. 为此, 通过分析以上算法的优缺点, 本文采用实数编码提出一种既能增强群体多样性和搜索多个全局最优解的能力, 又能维持全局最优解的免疫算法. 仿真结果表明, 该算法的优化效果具有明显的优越性.

2 免疫应答原理及免疫算法的提出 (Immune response principle and proposed immune algorithm)

免疫应答是免疫系统与外来物质(抗原)发生作用的关键环节. 当抗原入侵机体时, 免疫系统产生

的淋巴细胞数量少,机体抵御抗原的能力较弱,抗原清除也较慢.由于免疫系统具有学习、识别、记忆、自保护、多样性和自适应等特点,其经过一段时间后,淋巴细胞(B细胞)数目增加较快,这些细胞被分化和繁殖,经由亲和突变达到亲和成熟,提高识别能力.此时,抗原被清除较快.剩余的B细胞中,一部分应答能力弱或产生自反应,被有效清除,其余的作为记忆细胞存于免疫系统中,防御同一抗原再次入侵.这种应答属于免疫应答.在这应答过程中,克隆选择、细胞克隆、亲和成熟及募集新成员(指随机产生的新细胞)等四种机理发挥重要作用.粗略地讲,这四种机理的运行机制是:当抗原入侵机体时,机体中选择应答能力(即亲和力:B细胞携带的抗体识别抗原的能力)强的B细胞进行克隆,对克隆细胞经由亲和突变提高亲和力(即亲和成熟),然后随机产生新抗体更新群体中低亲和力抗体,这种机制不断重复,最终清除入侵的抗原.此过程为一进化过程,模拟其运行机制可构建免疫算法解决如下单目标函数优化问题

$$P) \quad x^* = \arg \max_{x \in X_C} f(x).$$

其中

$$X = \{x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^p \mid a_i \leq x_i \leq b_i, i = 1, 2, \dots, p\},$$

$f(x)$ 为 X 上的目标函数.

由于每一B细胞仅携带一种类型的抗体,因此将B细胞视为抗体.在本文中,抗原被视为问题P),抗体对应问题P)的候选解,抗原与抗体的亲和力视为问题P)中候选解(抗体)的目标函数值,抗体采用十进制编码,不妨设其染色体长度皆为 l ,于是抗体表示为由 p 个染色体构成的有序十进制数组,群体规模指定为 N .免疫算子包括免疫选择、亲和突变、募集新成员;另外基于“物以类聚,人以群分”以及“资源共享”的思想引入小生境实现方法^[7](简记为小生境法)获取记忆细胞群,目的是增强群体多样性及保存优良个体,提高搜索性能.利用此4个算子可构建免疫算法(aiA: artificial immune algorithm).该算法的主要思想可简要描述为,首先记忆当前抗体群并利用小生境法获记忆细胞群;然后选出当前抗体群中较高亲和力的抗体进行突变;最后将记忆细胞与已突变的抗体组合在一起,并随机产生新抗体更新低亲和力的抗体,获新抗体群.为了便于叙述本文免疫算法,在此利用小生境法的思想给出具体产生记忆细胞群的算法,即

算法 1

步骤 1 设抗体群为 $P = \{Ab_1, Ab_2, \dots, Ab_N\}$, 计算适应度(即指亲和力) $f(Ab)$, $Ab \in P$.

步骤 2 按下述步骤确定小生境及优良抗体.

步骤 2.1 置 $i = 1$.

步骤 2.2 计算抗体间的距离

$$d_{ik} = \|Ab_i - Ab_k\|, k = i, i+1, \dots, N. \quad (1)$$

步骤 2.3 根据 $d_{ik} < \sigma_0, k = i, i+1, \dots, N$, 确定小生境子群 M_{p_i} , p_i 为 M_{p_i} 的元素个数,其中 σ_0 依据问题 P) 而确定.

步骤 2.4 若 M_{p_i} 中包含适应度最高的抗体,则保存其中一个最高适应度抗体的适应度,其余抗体按步骤 2.5 更新其适应度;否则,按步骤 2.5 更新 M_{p_i} 中抗体的适应度.

步骤 2.5 计算 M_{p_i} 中抗体的更新适应度

$$f'_j = f_j / \sum_{s=1}^{i+p_i-1} shd_{js}, j = i, i+1, \dots, i+p_i-1. \quad (2)$$

其中

$$Shd_{js} = \begin{cases} 1, & d_{js} = 0, \\ 1 - (d_{js}/\sigma)^\lambda, & d_{js} < \sigma, \\ 0, & \text{否则.} \end{cases}$$

步骤 2.6 利用适应度 f'_j 及处罚函数 Penalty 对该子群体中低适应度的抗体进行处罚,即当

$$Ab_j, Ab_k \in M_{p_i}, \|Ab_j - Ab_k\| < L, L < \sigma$$

时,比较两个抗体的适应度,并对其中适应度较低的抗体进行处罚,

$$\begin{cases} f_{\min}(Ab_j, Ab_k) = \text{Penalty}, \\ j, k = i, i+1, \dots, i+p_i-1. \end{cases} \quad (3)$$

步骤 2.7 当 $i+p_i < N$ 时,置 $i \leftarrow i+p_i$, 并返回步骤 2.2, 否则,进入下一步.

步骤 3 确定记忆抗体集.按从大到小的次序排列各抗体的适应度,选择前 M 个抗体构成记忆抗体集 P_M .

以上算法通过利用共享适应度调整抗体的适应度,抑制浓度较高的抗体;但其中适应度最高的一个抗体不受抑制的影响.设计此操作的目的在于增强群体的多样性,同时保存当前群体中最好的抗体.于是利用此算法及免疫应答的几种机制可构建如下免疫算法(aiA):

算法 2

步骤 1 确定进化代数 $n \leftarrow 1$, 随机产生 N 个初始抗体构成初始群体 A_n .

步骤2 记忆细胞获取. 将算法1作用于 A_n , 获 M 个记忆细胞构成群体 M_n .

步骤3 克隆选择. 按选择率 σ , 选择 N_0 个较高亲和力的抗体构成抗体群 B_n , 其中 $N_0 \equiv \text{int}(\sigma \cdot N)$, $\text{int}(\cdot)$ 表示取整函数, σ 满足 $\text{int}(\sigma \cdot N) + M = N$.

步骤4 亲和突变. B_n 中抗体 Ab_i 被突变的概率为 α_i , 在突变的情形下, 抗体的各个染色体(即可行解的分量)分别进行均匀突变, 从而获抗体群 C_n , 其中

$$\begin{cases} \alpha_n = 1 - \exp\left(-\frac{\mu_n + \max_n f(Ab_i)}{\mu_n + \max_n f(Ab_i) - \min_n f(Ab_i)}\right), \\ \mu_n = \mu + \frac{T_0}{n}, \\ \max_n (\min_n) = \max_{Ab \in B_n} (\min_{Ab \in B_n} (f(Ab))). \end{cases} \quad (4)$$

这里 μ, T_0 为可调节参数.

步骤5 抗体组合及更新. 随机产生 d 个新抗体取代 $C_n \cup M_n$ 中 d 个低亲和力抗体, 获新抗体群 A_{n+1} , 其中 $d \equiv \text{int}(\mu \cdot N)$.

步骤6 终止条件判断. 若不满足终止条件, 则继续返回步骤2, 否则, 输出结果.

注 此算法中并未引入细胞克隆操作, 目的在于提高寻优速度. 步骤4在于提高抗体的亲和力, 其主要基于抗体的突变率与其亲和力成反比的机理而设计; 步骤5产生 d 个新抗体主要是防止算法陷局部搜索, μ 一般应在 0.05 至 0.08 之间^[3]. aiA 与 GA 的主要区别在于: 1) GA 通过交叉算子增强群体多样性, 并使群体中个体突变的可能性较小, 而 aiA 主要通过记忆细胞、随机产生新抗体实现群体多样性, 具有记忆细胞保存和再利用功能, 抗体突变受其亲和力制约; 2) GA 按概率随机选择, 而 aiA 的选择为确定性操作; 3) aiA 对初始群体的分布无敏感性.

3 全局收敛性(Global convergence)

鉴于 aiA 中每一抗体 Ab 的染色体长度皆为 l 获知, 所有抗体构成的抗体空间 S 为有限集. 分别设群体规模为 N, N_0, M 的抗体群构成群体空间 S^N, S^{N_0} 及 S^M , 每一空间中元素称为状态. 设 D_n^d 表示 n 时刻产生的 d 个新抗体构成的状态. 由于 aiA 中 A_{n+1} 所处状态仅与 A_n, M_n 及 D_n^d 有关, 与过去状态无关, 故由链

$$A_n \rightarrow B_n \cup M_n \rightarrow C_n \cup M_n \cup D_n^d \rightarrow A_{n+1}$$

描述的随机搜索过程为马氏链, 但由于亲和突变率与时间 n 有关, 故该链为非齐次马氏链. 用 A_n^i 表示 n 时刻 A_n 处于状态 s_i , $P(A_n^i)$ 表示 A_n 处于状态 s_i 的概率, $P(A_{n+1}^j | A_n^i)$ 指在 A_n^i 下, A_{n+1} 处于状态 s_j 的概率. 用 \max 及 \min 分别表示 $f(x)$ 在 S 中所取的最大

值和最小值, $P(T^d(S))$ 表示随机产生 d 个新抗体构成状态 s_j 的概率, $P(A_n \cap M^c \neq \emptyset)$ 表示 n 时刻 A_n 中含 M^c 中元素的概率, 其中

$$M^c = \{x^* \in S \mid f(x^*) = \max_{x \in S} f(x)\}. \quad (5)$$

于是引入如下定理, 其证明见附录.

定理 算法2对于任意初始分布, 均以概率1弱收敛, 即

$$\lim_{n \rightarrow \infty} P(A_n \cap M^c \neq \emptyset) = 1.$$

4 仿真结果(Simulation results)

为了验证 aiA 的有效性, 将其和 CLONALG^[3] 及 REGA^[8] 比较. 此3种算法用于求解下列具有代表性^[7]的多模态函数(式(6)~(9))的最大值, 以及求非连续函数(式(10))的最小值

$$F_2: f(x, y) = 100(x^2 - y)^2 + (1 - x)^2, \quad -2.048 \leq x, y \leq 2.048; \quad (6)$$

$$F_6: f(x, y) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1.0 + 0.001(x^2 + y^2)]^2}, \quad -100 \leq x, y \leq 100; \quad (7)$$

$$F_9: f(x, y) = \sum_{i=1}^5 i \cos [(i+1)x + i] \cdot \sum_{i=1}^5 i \cos [(i+1)y + i], \quad -10 \leq x, y \leq 10; \quad (8)$$

$$F_{10}: f(x, y) = (4 - 2.1x^2 + \frac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2, \quad -3 \leq x \leq 3, -2 \leq y \leq 2; \quad (9)$$

$$F_{11}: f(x_1, x_2, \dots, x_5) = \sum_{i=1}^5 \text{integer}(x_i), \quad -5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, 5. \quad (10)$$

为了公平比较, 指定此3种算法的群体规模为80, 迭代次数为100(对于函数 F_9 需200代). 其余各参数选定为使各算法获最佳效果时的参数, 即 REGA 的交叉概率为0.6, 突变概率为0.001, CLONALG 的每一代随机产生的新抗体数为6; aiA 的参数选定为 $M = 20, \sigma = 0.8, \mu = 0.06$. F_2, F_6, F_9 及 F_{10} 的最大值分别为3905.93, 0.99, 210.482及162.9, F_{11} 的最小值为-30. 将上3种算法用于上5个函数求最值, 通过各自对测试问题独立运行10次获知, aiA 对 F_{10} 及 F_{11} 每次皆获多个最优解, 而 CLONALG 及 REGA 却较困难. 另外, 用下式刻划各算法的群体多样性, 即对同一问题, 群体多样性的平均变化率为

$$P = \frac{10 \text{ 次独立运行到指定的迭代数时不同的抗体数之和}}{10 \times N} \quad (11)$$

上述3种算法用于以上5个函数优化所获得最优或次优解及群体多样性比较结果如表1所示。

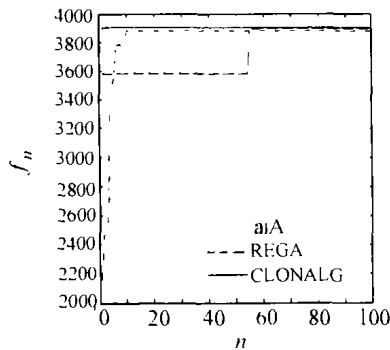
通过表1可知,从所获最优或次优解方面看, aiA 及 CLONALG 明显优于 REGA, 同时 aiA 优于 CLONALG, 一般 REGA 至少需 1000 代才获最优或次优解; 另外, 从群体多样性看, aiA 比 CLONALG 的群体多样性好, 且整体上也优于 REGA 的多样性. 由此获知, aiA 在处理多模态函数及非连续的函数优化问题方面优于 CLONALG 和 REGA, 这也说明

从免疫学角度开发的算法比遗传算法有更多的优越性. 另外, 为了说明搜索最优解的速度差异, 对以上每一问题, 将此3算法分别独立运行10次, 并分别随机选取一次运行结果进行比较, 获下列各图. 由图1(a)~(e)知, 对 F_2, F_9 及 F_{10} , aiA 比 CLONALG 较快地搜索到理想解; 对 F_6 及 F_9 , aiA 比 CLONALG 所获解好; 对 F_{11} , CLONALG 比 aiA 搜索到最优解的速度快. 同时获知, aiA 及 CLONALG 在迭代次数为100代范围内可获最优解或次最优解, 而 REGA 在迭代次数为100范围至多获次优解, 这也说明了本文的算法是行之有效的新智能计算方法.

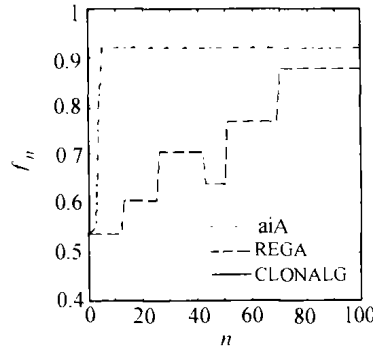
表1 3算法用于上述5函数优化, 所获最优或次优解及群体多样性比较

Table 1 Comparison of population diversity and acquired optimal solutions or sub-optimal ones with the three algorithms based on the above five problems

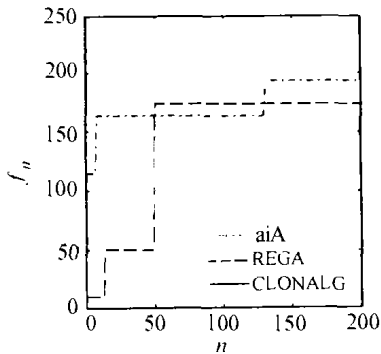
| 函数 | 算法 aiA | | | 算法 CLONALG | | | 算法 REGA | | |
|----------|------------------|----------|---------|----------------------|---------|---------|---------------------|----------|---------|
| | 最优或次优解 | 函数值 | P 值 | 最优或次优解 | 函数值 | P 值 | 最优或次优解 | 函数值 | P 值 |
| F_2 | (-2.048, -2.048) | 3905.93 | 0.935 | (-2.048, -2.048) | 3905.93 | 0.18 | (-2.03599, -2.044) | 3839.89 | 0.89375 |
| F_6 | (-4, -2) | 0.926164 | 0.94175 | (-4.10577, -2.63937) | 0.95022 | 0.0625 | (-6.74487, 4.00782) | 0.943635 | 0.8625 |
| F_9 | (-0.8, -0.8) | 210.482 | 0.837 | (-7, -0.8) | 193.602 | 0.462 | (5.38612, 5.54252) | 179.606 | 0.525 |
| F_{10} | (-3, -2), (3, 2) | 162.9 | 0.93625 | (-3, -2) | 162.9 | 0.29375 | (-3, -1.99218) | 162.006 | 0.89125 |
| F_{11} | 每次运行至少12个最优解 | -30 | 0.92125 | 每次运行至少2个最优解 | -30 | 0.91125 | 每次运行最多2个次优解 | -28或-29 | 0.9625 |



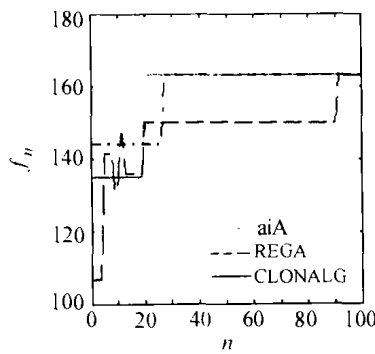
(a) 对 F_2 的收敛速度比较



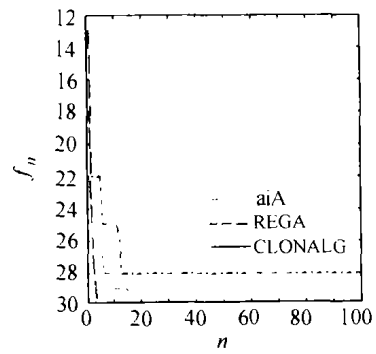
(b) 对 F_6 的收敛速度比较



(c) 对 F_9 的收敛速度比较



(d) 对 F_{10} 的收敛速度比较



(e) 对 F_{11} 的收敛速度比较

图1 对 $F_2, F_6, F_9, F_{10}, F_{11}$ 的收敛速度比较, 符号 n, f_n 分别表示迭代数及函数值

Fig. 1 Comparisons of convergence speed on F_2, F_6, F_9, F_{10} and F_{11} , respectively;

n and f_n mean iterative number and function value, respectively

5 结论(Conclusion)

aiA 算法是依据免疫应答原理及小生境概念提出的一种新免疫算法,该算法具有较强搜索多个全局最优解能力,群体搜索性能优于 CLONALG 及 REGA.理论证明了该算法的收敛性,仿真表明该算法是有效的,但此算法用于其它工程领域有待进一步研究.

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, Special Issue on Artificial Immune Systems*, 2002, 6(3): 239 - 251.
- [2] 韦魏,张宏国.人工免疫系统及其在控制系统中的应用[J].*控制理论与应用*,2002,19(2):157 - 160.
(WEI Wei, ZHANG Guohong. Artificial immune systems and their applications to control systems [J]. *Control Theory & Applications*, 2002, 19(2): 157 - 160.
- [3] de CASTRO L N. The Clonal Selection Algorithm with Engineering Applications [C]// *In Workshop Proc of GECC'00, Workshop on Artificial Immune Systems and Their Applications*, [s.l.]: [s.n.], 2000:36 - 37.
- [4] ZHANG Z H, HUANG X Y, MA X X. A Novel Fuzzy Immune Control System and Its Application to Multi-modal Function Optimization [C]// *Proc of the 2002 Int Conf on Control and Automation*, [s.l.]: [s.n.], 2002:777 - 780.
- [5] JIAO C, WANG L. A novel genetic algorithm based on immunity [J]. *IEEE Trans on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 2000, 30(5):552 - 561.
- [6] 王熙法,张显俊,曹先彬,等.一种基于免疫原理的遗传算法[J].*小型微型计算机系统*,1999,20(2):117 - 120.
(WANG Xifa, ZHANG Xianjun, CAO Xianbin, et al. A genetic algorithm based on immune metaphors [J]. *Min-micro Systems*, 1999, 20(2):117 - 120.)
- [7] 周明,孙树栋.遗传算法及应用[M].北京:国防工业出版社,2000.
(ZHOU Ming, SUN Shudong. *Genetic Algorithms: Theory and Applications* [M]. Beijing: National Defense Industry Press, 2000.)
- [8] 张著洪,黄席樾.采用重复交叉操作及最优保留策略的遗传算法[J].*重庆大学学报*,2002,25(7):23 - 25.
(ZHANG Zhuhong, HUANG Xiyue. A genetic algorithm based on repeating crossover and optimum maintaining strategy [J]. *J of Chongqing University*, 2002, 25(7):23 - 25.)

附录(Appendix)

定理的证明.为了叙述方便,简记

$$a_n = P(A_n \cap M^* = \emptyset),$$

$$a_n(M^*, i) = P(A_n \cap M^* = \emptyset | A_n^i).$$

由于算法 1 及克隆选择皆为确定性操作,因此不妨设 A_n^i 分别经此两操作所获状态依次为 s_{n_1} 及 s_{n_2} , 从而

$$P(M_n^i | A_n^i) = P(B_n^{i2} | A_n^i) = 1, s_{n_1} \in S^y, s_{n_2} \in S^0. \quad (A1)$$

由步骤 4 知,对于任意 $s_k, s_n \in S^0$,若用 $P(x_{nk} \rightarrow y_{n_1})$ 表示 n 时刻状态 s_k 中第 l 个分量 x_{nk_l} 变为状态 s_n 中第 l 个分量 y_{n_1} 的概率,则有

$$P(C_n^k | B_n^i) = \prod_{l=1}^{y_n} P(x_{nk_l} \rightarrow y_{n_1}) \geq \prod_{l=1}^{y_n} \left(\left(\frac{9}{100} \right)^{\mu^l} \left(1 - \exp \left(- \frac{\mu_n + \max_n - f(x_{nk_l})}{\mu_n + \max_n - \min_n} \right) \right) \right). \quad (A2)$$

令 $P_{nl} = \frac{\mu_n + \max_n - f(x_{nk_l})}{\mu_n + \max_n - \min_n}$, 由泰勒展开式定理知

$$1 - \exp(-P_{nl}) \geq \frac{\mu}{2(\mu + \max - \min)}. \quad (A3)$$

进而由式(A2)及式(A3)获得

$$P(C_n^k | B_n^i) \geq \left(\frac{9\mu}{2 \times 100^{\mu}(\mu + \max + \min)} \right)^{y_n}. \quad (A4)$$

现考虑 A_n^i 转移为 A_{n+1}^i 的概率.由步骤 5 知, A_{n+1}^i 由三部分构成,一部分由 A_n^i 经步骤 3 至步骤 4 转移所获状态 s_{j_1} , 一部分为 s_i 经算法 1 所获状态 s_{j_2} , 其余部分为随机产生的 d 个新抗体构成的状态 s_{j_3} , 于是有

$$P(A_{n+1}^i | A_n^i) = \sum_{s_k \in S^0} P(C_n^k | B_n^i) P(A_{n+1}^i | C_n^k) \geq \left(\frac{9\mu}{2 \times 100^{\mu}(\mu + \max - \min)} \right)^{y_n} \equiv \delta_1. \quad (A5)$$

若 $s_{j_3} \cap M^* \neq \emptyset$, 则必有

$$P(T^d(S)) = 1 - \left(\frac{|S| - |M^*|}{|S|} \right)^d \equiv \delta_2. \quad (A6)$$

取 $\delta = \min(\delta_1, \delta_2)$, 进而

$$\sum_{s_i \cap M^* \neq \emptyset} a_{n+1}(M^*, i) P(A_n^i) \leq (1 - \delta) a_n. \quad (A7)$$

又由算法 1 及步骤 5 知, A_n 所处状态中最优抗体被保存到了 A_{n+1} 所处状态中,因此

$$\sum_{s_i \cap M^* \neq \emptyset} a_{n+1}(M^*, i) P(A_n^i) = 0. \quad (A8)$$

从而由式(A7)、式(A8)及全概率公式得

$$a_{n+1} \leq (1 - \delta) a_n.$$

通过归纳得 $\lim_{n \rightarrow \infty} a_n = 0$, 此表明该定理的结论成立.

作者简介:

张著洪 (1966—),男,重庆大学自动化学院博士研究生.研究领域为智能控制,进化算法及免疫算法. E-mail: zhangzhuhong@263.net;

黄席樾 (1944—),男,教授,重庆大学博士生导师.研究领域为人工智能与知识工程,智能控制及智能故障诊断.