

# 一种基于小生境策略的阴性选择算法

杨宁 王茜

(重庆大学计算机学院 重庆 400044)

**摘要** 阴性选择算法是计算机人工免疫系统的传统核心算法之一,并以此为基础产生了许多改进算法,但这些算法大多存在计算时间过长以及空间资源消耗过大等问题。针对这些问题,提出了一种基于小生境策略的阴性选择算法,算法引入了“小生镜”策略,增强了检测器生成的多样性,降低了算法的复杂度并减少了检测器的生成时间,提高了阴性选择算法的生成效率。

**关键词** 人工免疫,阴性选择算法,检测器,小生境策略

## Negative Selection Algorithm Based on Niche Strategy

YANG Ning WANG Qian

(College of Computer Science, Chongqing University, Chongqing 400044, China)

**Abstract** Negative selection algorithm is one of the traditional core algorithms of the artificial immune system, many improved algorithms are based on it. However, there are many problems about these algorithm such as the long-time computing and the excessive consumption of space resources. In order to reduce the complexity of the algorithm and the generation time of the detector, we presented a algorithm which was based on the niche strategy, it improve the efficiency of the generation of negative selection algorithm.

**Keywords** Artificial immune, Negative selection algorithm, Detector, Niche strategy

## 1 引言

作为生物系统中的信息处理系统,生物免疫系统的模式识别、记忆、学习、多样性产生、分布式检测、噪声耐受等功能对于解决目前许多问题很有启发,并形成了人工免疫系统(Artificial Immune System, AIS)这个研究领域<sup>[1,2]</sup>。

在生物免疫系统中,免疫识别是一项主要功能,其主要的本质是区分“自体”与“非自体”,免疫识别是通过淋巴细胞上的抗原识别体(receptor)与抗原的结合实现的,结合强度称为亲和力(affinity),淋巴细胞的产生是免疫系统一切功能的基础。美国 New Mexico 大学的 Forrest 教授模拟淋巴细胞的产生过程,并提出了阴性选择算法<sup>[3-5]</sup>,此后阴性选择成为人工免疫系统应用中异常分类器(成熟检测器)产生的必经过程。阴性选择算法是模仿淋巴细胞的阴性选择过程而提出的适合人工免疫系统的一个算法。但该算法存在计算时间过长的的问题,为了克服直接使用标准阴性选择算法所带来的计算时间长的问题,相继有多个检测器生成算法被提出,其中比较著明的是 Dhaeseleer 提出的改进阴性选择算法<sup>[6]</sup>;线性时间算法和贪婪算法。但该算法存在两点不足:空间消耗太大;它们只适用于连续位串匹配规则、用二进制编码表示特征串的情形。该算法的时间和空间复杂度与匹配阈值呈指数关系,分别为  $O((l-r) * 2^r) + O((l-r) * N_s)$  与  $O((l-r)^{2*2^r})$ 。其中  $l$  是一个匹配字符串整体的长度, $r$  是匹配阈值而  $N_s$  表示整个自体集中自体的个数。

作为人工免疫系统中的核心算法的阴性选择算法,其性能对整个系统具有重要的意义。以往的阴性选择算法检测率不高,于是一些研究人员展开了一系列的研究,如文献<sup>[7]</sup>提

出了一种改进的基于实值的阴性选择算法在异常检测中的应用,利用了一种可变参数生成检测器以及产生了相应的匹配规则。文献<sup>[8]</sup>提出了一种  $r$  可变阴性选择算法,同传统的阴性选择算法相比,该算法大大减少了不可避免的“黑洞”数量。文献<sup>[9]</sup>提出了一种自适应阴性选择方法在异常检测中的应用,引入了进化思想。文献<sup>[10]</sup>提出了一种基于人工免疫系统的模糊异常检测算法,该算法提高了异常检测的准确率。

本文在深入研究阴性选择算法的基础上,为了进一步减少传统的阴性选择算法运算时间过长以及空间资源消耗过大等问题,提出了一种基于小生境策略的阴性选择算法,该算法能够更好地保持检测器的多样性并提高检测效率,同时对这种算法下产生的成熟检测器的迭代次数与检测率进行了仿真分析。

## 2 基本概念

**概念 1(模式)** 由  $l$  个符号组成的符号串  $X = X_1 X_2 \dots X_l$ , 其中  $X_i (i=1, 2, 3, \dots, l) = 0$  or  $1$ , 这样模式就是长度为  $l$  的二进制串。

用  $U$  表示所有模式的集合,  $N$  表示所有非自体模式(nonself)的集合, 简称非自体集,  $S$  表示所有自我模式(self)的集合, 简称自体集。  $U = N \cup S$  成立。通过阴性选择算法生成的、可以检测出非自体模式集合的模式集合称为检测器集。

**概念 2(匹配)** 在一定的规则下, 两个模式串  $a$  和  $b$  的相似程度超过一个给定的阈值, 则称  $a$  和  $b$  匹配, 记为  $Match(a, b)$ 。

设有两个模式串  $a$  和  $b$ ,  $a = X_{a1} X_{a2} X_{a3} \dots X_{al}$ ,  $b = Y_{b1} Y_{b2} Y_{b3} \dots Y_{bl}$ 。在人工免疫系统中, 一般采用的匹配规则是  $rcb$  匹

配规则<sup>[11]</sup>。

**概念 3(rcb 匹配规则)** 对于两个模式串  $a$  和  $b$ , 当且仅当它们在  $r$  或者多于  $r$  个连续位置上有相同的字符时, 则它们在  $r$  连续位规则下匹配。

同样例如,  $a=1001011101, b=10011010101$ , 当  $r \leq 3$  的时候,  $\text{Match}(a, b)$ 。

两个随机的模式串  $a$  和  $b$  在连续规则下匹配的概率为:

$$P(\text{Match}(a, b)) = 2^{-r} \left( \frac{L-r}{2} + 1 \right)$$

本文采用的匹配规则为 rch 匹配规则。

**概念 4(rch 匹配规则<sup>[11]</sup>)** 定义的  $d'$  是窗口  $w$  上的长为  $r$  的位串, 并具有指定的窗口起始位置。  $d$  为模式集合  $U$  中包含  $d'$  的一个位串, 则匹配规则可以表述为:

$$\forall x, d \in U, \text{Match}(x, d) \Leftrightarrow x[w] = d'$$

例如:  $x:110101110; d':101$ (起始位置 2);  $d:010111100$ 。

当  $r=3$  时匹配成功, 其优势主要在于窗口起始位置的合理分配。

### 3 阴性选择算法

检测器生成算法包括两个部分: 初始未成熟检测器集合的产生和成熟检测器集合的生成。自从 Forrest 于 1994 年提出阴性选择思想之后, 阴性选择成为人工免疫系统应用中异常分类器(成熟检测器)产生的重要部分。阴性选择算法是模仿淋巴细胞的阴性选择过程而提出的适合人工免疫系统的算法。标准的阴性选择算法一般由 5 个步骤来完成:

- (1) 定义自体为长度为  $L$  的字符串的集合  $S$ ;
- (2) 随机产生长度为  $L$  的字符串  $a$ ;
- (3) 将字符串  $a$  依次与集合  $S$  中的字符串匹配;
- (4) 根据匹配规则, 如果  $a$  遇到与之匹配的字符串, 则结束匹配, 转到第(2)步;
- (5) 如果  $a$  不与  $S$  中任何字符串匹配, 则  $a$  成熟, 将  $a$  加入到成熟检测器集中。

图 1 表示了这一过程。

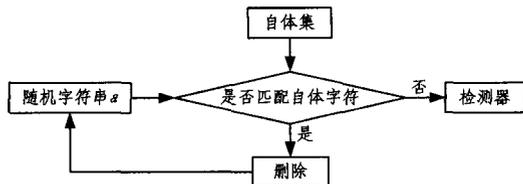


图 1 阴性选择算法流程

由阴性选择算法产生的检测器是成熟的, 可以参加实际的检测活动。该算法表现出许多优点, 已经被成功地应用在病毒检测、异常检测等领域, 而且 Dasgupta 对该算法进行了深入的研究。实验表明: 阴性选择算法被应用于网络安全保障, 解决了许多过去在同类问题中难以解决的难题, 且成效显著。但当 Kim 等人在 2001 年将这种算法应用于网络入侵检测系统中时, 却发现该算法在处理真实的网络流量数据中存在着严重的伸缩性 (scaling) 问题, 其原因是初始未成熟检测器采用随机的方式产生。该算法初始检测器集合的大小  $N_{D0}$  与自体集合  $N_S$  的大小呈指数关系, 如下所示。

$$N_{D0} = \frac{-\ln(p_f)}{p_m \cdot (1-p_m)^{N_S}}$$

式中,  $P_m$ (匹配概率) 为随机选择的一个“非自体”特征串和特定的检测器两者匹配的概率。

$P_f$ (失败概率) 为随机选择的一个“非自体”特征串不与任何一个检测器匹配的概率。

而且该算法的时间复杂性和自体集的大小也呈指数关系。所以在网络入侵检测中直接采用阴性选择算法会造成时间和空间的巨大浪费, 是不可行的。

### 4 基于小生镜的阴性选择算法

为了克服直接使用阴性选择算法所带来的计算时间长的问题, 相继有多个检测器生成算法被提出。其中, Kim 和 Bentley<sup>[12]</sup> 认为人体免疫系统采用了“抗体”向“抗原”进化的策略, 即通过小生镜策略以维持抗体的一般性和多样性。向人工免疫系统引入“小生镜”策略以提高阴性选择算法的效率。

**概念 5(小生镜)** 生物学上, 小生境是指特定环境中的一种组织功能。引进小生境的概念, 有助于算法实现找到全部最优解的目的<sup>[13]</sup>。小生境技术就是将每一代个体划分为若干类, 每类中选出若干适应度较大的个体作为该类的优秀代表, 组成一个种群, 再在种群中以及不同种群之间通过交叉和变异产生新一代个体群, 同时采用预选择机制、排挤机制或分享机制完成选择操作。基于这种小生境技术的遗传算法 (Niche Genetic Algorithms, NGA), 可以更好地保持解的多样性, 同时具有很高的全局寻优能力和收敛速度, 特别适合于复杂的多峰值函数的优化问题。

本文提出一种基于共享函数的小生境阴性选择算法, 用来产生多样性的检测器。

首先给出如下定义。

**定义 1(检测器距离  $d(i, j)$ )**  $d(i, j)$  为检测器之间的距离,  $l$  为长度, 其计算公式为:

$$d(i, j) = \sum_{k=1}^l |\text{bit}(i, k) - \text{bit}(j, k)|$$

式中,  $\text{bit}(i, k)$  表示第  $i$  个检测器第  $k$  位的值。

**定义 2(共享函数  $sh(d(i, j))$ )** 共享函数是表示检测器集合中两个检测器之间密切关系程度的一个函数。检测器之间的密切程度主要体现为检测器所具有的检测覆盖面之间的相似程度。当个体之间比较相似时, 其共享函数值就比较大; 反之, 当个体之间不太相似时, 其共享函数值比较小。

检测器  $i$  和  $j$  之间的共享函数为:

$$sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i, j)}{\delta}\right)^\alpha, & d(i, j) < \delta \\ 0, & d(i, j) > \delta \end{cases}$$

式中,  $\delta$  为事先指定的峰半径 (即判断两个检测器共享条件是否成立的相隔的最大距离)。  $\alpha$  为控制共享函数形状的参数, 通常  $\alpha=1$ , 即为线性共享函数, 如果  $\alpha > 1$ , 则为凹函数, 如果  $\alpha < 1$  为则凸函数。这里  $\alpha$  取值为 1。

得到个体共享值之后, 可以由下面公式计算检测器的小生镜数。

$$m_i = \sum_{j=1}^N sh(d(i, j)) \quad i=1, 2, 3, 4, \dots, N$$

式中,  $N$  为种群检测器的数目。显然, 检测器的小生镜数越大, 聚集在该检测器周围的检测器就越多。然后, 计算共享后检测器的适应度。

$$f_i' = f_i / m_i \quad i=1, 2, 3, \dots, N$$

式中,  $f_i$  为第  $i$  个检测器共享前的适应度。

然后,根据  $ave(f_i)$  计算  $end$ 。

$$end = \frac{|ave(f_i^{t+1}) - ave(f_i^t)|}{ave(f_i^t)}$$

式中,  $ave(f_i^t)$  为第  $k$  代的平均适应度,  $ave(f_i^{t+1})$  为第  $k+1$  代平均适应度。

算法结束的条件是:

$$end < \omega$$

式中,  $\omega$  为给定的判断标准, 可以是 0.001~0.1 之间的一个常数。

算法具体描述如下(其中  $|D|=N$ ):

(1) 采用基因表达和随机的方式构造初始检测器群  $D$ , 所有检测器适应度取值为 0;

(2) 进化代数  $Generation\ number=0$ ;

(3) 在  $D$  中随机抽取一个子集  $D'$ ;

(4) 从自体集  $N_s$  中随机抽取一个自体特征串;

(5)  $D'$  中所有检测器与该自体特征串进行匹配;

(6)  $D'$  中与自体特征串相似性最差的那个检测器的适应度值增加, 增加值为自体特征串未被匹配的位数, 其余检测器的适应度值不变;

(7) 重复(3)~(6), 重复次数为  $3 * |D'|$ ;

(8) 计算检测器之间的距离  $d(i, j)$ ;

(9) 计算检测器之间的共享函数值  $sh(d(i, j))$ ;

(10) 计算每个检测器的小生镜数  $m_i$ ;

(11) 计算每个检测器的适应度;

(12) 选择  $D$  中适应度最高的  $P_b\%$  个检测器作为进化群体, 采用交叉和变异遗传操作对群体进行进化;

(13)  $D$  中适应度最差的  $P_w\%$  个检测器被丢弃, 丢弃的个数等于步骤(12)新生成的检测器个数;

(14) 由经过选择的父代和新生成的下一代作为新的检测器群体;

(15)  $Generation\ number++$ ;

(16) 计算  $end$ ;

(17) 重复(3)~(16), 直到满足给出的条件, 最后得到的检测器集合为成熟检测器集合。

## 5 算法性能

### 5.1 实验数据

实验使用的数据源来源于 KDD CUP 1999 DATA 经过修正整理的数据。数据包括标准“自体”和“非自体”数据库, 共有 5 个不同的数据集合。其中一个是在没有网络入侵时获取的 TCP 数据包头, 即正常模式(“自体”); 另外四个是有网络入侵时获取的 TCP 数据包头, 即异常模式(“非自体”)。入侵方法分别是: IP 欺骗攻击(IP spoofing attack), 猜测密码(guessing rlogin or ftp passwords)、扫描攻击(scanning attack)和网络忙碌攻击(network hopping attack)。采用 Visual C++ 为编程工具。每个检测器  $d$  包含 9 个基因(含义见表 1), 因此检测器和抗原的长度为 27。

表 1 基因编码及其含义

序号	基因名	长度(bit)	含义
1	通信类型	3	001: SYN
			010: PUSH
			011: DNS
			100: UDP
			101: PIN

序号	基因名	长度(bit)	含义
2	源端口	3	001: 25
			010: 53
			011: 80
			100: 113
			101: [0, 1023]
110: [1024, -]			
3	目的端口	3	同上
			000: 0
4	握手信号	3	001: 1
			010: 2
			011: 3
			100: 4
			101: [5, -]
5	主叫方发送数据包数	3	000: [0, 10]
			001: [11, 50]
			010: [51, 100]
			011: [100, 500]
			100: [500, 999]
101: [1000, 4999]			
110: [5000, -]			
6	被叫方发送数据包数	3	同上
			000: [0, 10]
7	主叫方发送数据流量	3	001: [11, 99]
			010: [100, 999]
			011: [1000, 3000]
8	被叫方发送数据流量	3	同上
			000: [0, 9]
9	数据包序号出错次数	3	001: [10, 49]
			010: [50, 99]
			011: [100, 300]
			100: [300, -]

### 5.2 检测器多样性测试

首先, 对该算法生成的检测器的多样性进行分析, 检测器的多样性保证了检测异常数据的能力, 多样性的提高能有效防止误检和漏检的情况。我们采用两种多样性的度量方法: 基于基因的度量方法和基于种群的度量方法<sup>[14]</sup>。

(1) 基于基因的度量方法

设第  $t$  代种群的检测器  $D_c(d) = ((d1)_t, (d2)_t, \dots, (dN)_t)$ ,  $N$  个检测器组成的矩阵为:

$$P_{N \times L} = \begin{bmatrix} d_i(11), d_i(12), \dots, d_i(1L) \\ d_i(21), d_i(22), \dots, d_i(2L) \\ \vdots \\ d_i(N1), d_i(N2), \dots, d_i(NL) \end{bmatrix}$$

如果上述矩阵每列中 0 和 1 各占一半, 在交叉过程中一方面可以避免基因缺失, 另一方面可以以较大的概率产生新的个体, 这样若在矩阵  $P_{N \times L}$  中每列的 0 和 1 趋向于  $N/2$ , 则种群的多样性就越好。

根据上述分析, 种群多样性度量函数  $d(p)$  表示如下:

$$d(p) = \frac{1}{N \times L} \sum_{j=1}^L \max\left\{ \sum_{i=1}^N (1 - d_{ij}), \sum_{i=1}^N d_{ij} \right\}$$

式中,  $d_{ij}$  表示种群中第  $i$  个检测器第  $j$  位的值。  $d(p) \in [0.5, 1]$ , 表示每个等位基因位置上居多数量的二进制位的平均百分比率。如果每列中 0 和 1 各占一半, 为  $N/2$ ,  $d(p) = 0.5$ , 表明种群的多样性就好; 如果每列中二进制位都相同, 则  $d(p) = 1$ , 表明种群的多样性最差。显然,  $d(p)$  越小, 种群的多样性越好。

我们设计了仿真实验对  $d(p)$  的变化进行了分析。算法的参数为:  $N_s = 128$ ,  $|D| = 50$ ,  $|D'| = 25$ ,  $\delta = 10$ ,  $P_b = 0.8$ ,  $P_w = 0.5$ , 开始选择较大的交叉和变异率, 之后逐渐降低。在不

考虑进化结束条件的情况下,取进化代数为 200,实验结果如图 2 所示。

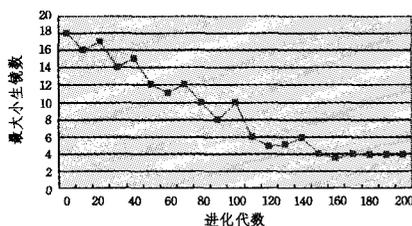


图 2

由图可知,开始进化后, $d(p)$ 明显降低,随后维持比较稳定的趋势。符合预期的效果。

### (2) 基于种群的度量方法

基于基因的多样性度量方法根据基因的内部结构评价种群的多样性,没有考虑种群检测器之间的关系,下面采用基于种群的度量方式来进行评价。

基于前面所述,种群中每个检测器的小生镜数  $m_i$  越大,表明和该检测器相似的检测器数量就越多。因此,可以根据种群中最大小生镜数来度量种群的多样性。参数设置同上,实验结果如图 3 所示。

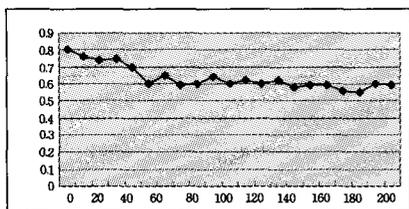


图 3

由图所示,开始进化后,最大小生镜数明显降低(多样性提高),随后维持比较稳定的趋势。

### 5.3 检测效率分析

其次,我们针对算法生成的检测器的检测率进行了实验分析。我们在取固定数量训练数据和测试数据的情况下,对该算法与传统阴性选择算法的检测率进行了比较。

在规定  $r=18$ ,其他参数不变,进化代数仍取为 200 时,得到的实验结果如图 4 所示。

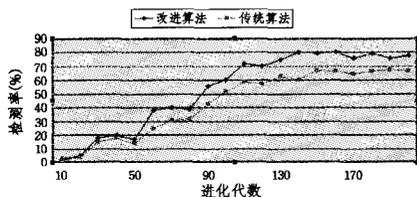


图 4

对上图所示结果进行分析,从 0 代进化到 50 代,改进算法和传统算法的检测率与检测率增加的幅度相差不大,无明显变化。从第 50 代以后检测率大幅度增加,并且和传统阴性选择算法相比,其检测效率以及增长的幅度都有比较明显的提高。

由上图可以发现,该算法生成的检测器具有较高的检测

效率,因此该算法是一个比较合理的检测器生成算法。

**结束语** 作为人工免疫系统中的核心算法之一的阴性选择算法,其性能对整个系统具有重要意义。为了克服直接使用标准阴性选择算法所带来的计算时间长的问题,本文提出了一种基于小生镜的阴性选择算法,算法的核心是通过加入一个共享函数来提高检测器的效率并同时保持检测器生成的多样性与稳定性。通过仿真分析表明,该算法达到了预定的要求,符合预期效果。

### 参考文献

- [1] Forrest S, Hofmeyr S, Somayaji A. A Computer Immunology [J]. Communications of the ACM, 1997, 40(10): 361-387
- [2] Forrest S, Perelson A S, Allen L, et al. Self-nonsel self discrimination in a computer [C] // Proc. of the IEEE Symposium on Research in Security and Privacy. 1994: 202-212
- [3] Hofmeyr S, Forrest S, Somayaji A. Intrusion Detection Using Sequences of System Calls [J]. Journal of Computer Security, 1998, 6: 151-180
- [4] Warrender C, Forrest S, Pearlmuter B. Detecting intrusions using system calls; Alternatedata models [C] // Proc. of the 1999 IEEE Symposium on security and Privacy. 1999: 133-145
- [5] D'haeseleer P, Forrest A S, Allen L. An immunological approach to change detection algorithms analysis and implications [C] // Proceedings of the 1996 IEEE Symposium on Security and Privacy Los Alamos, CA, 1996: 110-119
- [6] D'haeseleer P. Further efficient algorithms for generating antibody string [R]. The university of New Mexico, Albuquerque, NM: CS95-03, 1995
- [7] Hu Zheng-bing, Zhou Ji, Ma Ping. A Novel Anomaly Detection Algorithm Based on Real-Valued Negative Selection System [J]. Workshop on Knowledge Discovery and Data Mining, 2008
- [8] 张衡, 吴礼发, 张毓森, 等. 一种  $r$  可变阴性选择算法及其仿真分析 [J]. 计算机学报, 28(10): 1614-1619
- [9] Luis J, Gonzalez Z, Cannady J. A Self-Adaptive Negative Selection Approach for Anomaly Detection [J]. IEEE, 2004
- [10] Hu Zheng-bing, Zhou Ji, Ma Ping. A Novel Fuzzy Anomaly Detection Algorithm Based on Artificial Immune System [C] // Proceedings of the Eighth International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA'05). IEEE, 2005
- [11] Forrest S, et al. Using Genetic Algorithms to Explore Pattern Recognition in the Immune System, Evolutionary Computation [J], 1993, 1(3): 191-211
- [12] Kim J, Bentley P. Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection [C] // Proc. of GECCO'99. 1999: 149-158
- [13] 王小平, 曹立明. 遗传算法—理论、应用与软件实现 [M]. 西安: 西安交通大学出版, 2002
- [14] 凌军, 曹阳, 等. 基于小生境技术的多样性抗体生成算法 [J]. 电子学报, 2003(8)