ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.6, June 2007, pp.1361−1368
DOI: 10.1360/jos181361

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种检测器长度可变的非选择算法*

何 申+, 罗文坚, 王煦法

(中国科学技术大学 计算机科学技术系,安徽 合肥 230026)

A Negative Selection Algorithm with the Variable Length Detector

HE Shen+, LUO Wen-Jian, WANG Xu-Fa

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China)

+ Corresponding author: Phn: +86-10-88258511, Fax: +86-10-88258093, E-mail: scriptvirus@sina.com

He S, Luo WJ, Wang XF. A negative selection algorithm with the variable length detector. *Journal of Software*, 2007,18(6):1361−1368. http://www.jos.org.cn/1000-9825/18/1361.htm

Abstract: The detector generation is the key step of negative selection. Current detector generation algorithms have holes area and redundancy detector problems. A negative selection algorithm with the variable length detector is proposed in this paper. This algorithm can not only remove the holes, but also decrease redundancy detectors by the corresponding detector optimization algorithm. Therefore, both the detector generation efficiency and the detecting efficiency are improved well. This algorithm is analyzed in this paper and verified by experiments. The experimental results prove that this algorithm is better than the traditional negative selection algorithms and the negative selection algorithm with the *r*-adjustable detector.

Key words: artificial immune system; negative selection algorithm; detector generation; hole

摘 要: 检测器生成是非选择算法的关键步骤.已有检测器生成算法在生成检测器时存在“漏洞”区域和冗余检测器问题.提出了一种检测器长度可变的检测器生成算法,不仅可以消除“漏洞”区域,还可以通过相应的检测器优化算法减少冗余检测器,进而提高检测器生成效率和检测效率.对算法进行了分析和实验证明,结果表明,该算法比传统的非选择算法及 *r* 可变的非选择算法具有更好的性能.

关键词: 人工免疫系统;非选择算法;检测器生成;漏洞
中图法分类号: TP18 文献标识码: A

人工免疫系统(artificial immune system,简称 AIS)是一类基于生物免疫系统(biological immune system,简称 BIS)的功能、原理、基本特征以及相关理论免疫学说而建立的用于解决各种复杂问题的计算系统[1,2].生物免疫系统具有良好的分布式并行处理、自组织、自学习、自适应和鲁棒性等特点[2,3].基于生物免疫系统的人工免疫模型和算法研究已经成为计算智能领域中继人工神经网络、进化计算之后又一个新的研究热点.人工免疫算法大致分为基于群体的(population-based)人工免疫算法和基于网络的(network-based)人工免疫算法.二者的区别在于个体间是否相互作用:前者个体间无直接相互作用,其代表性的工作有非选择算法(negative selection algorithm,简称 NSA)[4]和克隆选择算法[5]等;后者强调构成系统的个体间存在相互作用,其代表性的工作有

* Supported by the National Natural Science Foundation of China under Grant No.60404004 (国家自然科学基金)
Received 2005-11-23; Accepted 2006-04-03

RLAIS[6]和 aiNet 算法[7]等等.

非选择算法是 Forrest 等人根据生物免疫系统中 T 细胞的产生与作用机制而提出的一种变化检测算法[4].目前,大多数的检测器生成算法在生成检测器时存在“漏洞”区域和冗余检测器问题.本文提出了一种检测器长度可变的非选择算法以及相应的检测器集优化算法,解决了“漏洞”区域问题,减少了检测器之间的冗余.

本文第 1 节介绍非选择算法及已有改进算法的优劣.第 2 节描述本文提出的检测器长度可变的非选择算法,并分析该算法的时间和空间复杂度.第 3 节是仿真实验.最后是结束语.

1 非选择算法及其分析

生物免疫系统最基本的特点就是可以通过区分自我和非我对病毒进行识别,进而分类清除.非选择算法是模拟 T 细胞成熟过程中的自我耐受过程而提出来的,在人工免疫系统研究领域具有很大影响,目前主要运用于网络安全、计算机病毒检测、入侵检测和异常检测等[2,4,8−11].

将自我个体和检测器的长度均设为 L.如图 1 所示,非选择算法的基本流程是[4]:(1) 首先生成长为 L 的预检测器,然后与自我集合按照匹配规则进行匹配,若匹配成功,则删除;否则,放入成熟检测器集合;(2) 重复以上过程,直至生成预定数量的成熟检测器;(3) 成熟检测器用于匹配待检测串,若匹配,则表明发生了异常变化.这里,匹配通常是部分匹配规则,比如 r 位联系匹配等.有必要指出,所谓“漏洞”问题是指依据检测器生成方法在非我空间无法产生检测器的区域[12].另外,一个检测器很可能可以匹配非我空间中的多个点或区域,因此能够被检测器匹配的点或区域称为被检测器所覆盖.

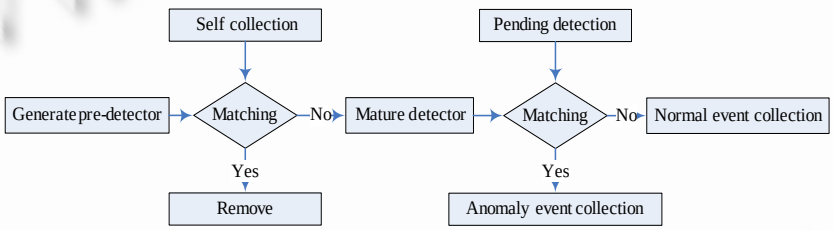

Fig.1 Negative selection algorithm

图 1 非选择算法

目前,主要的检测器生成算法有如下 4 种:

(1) 穷举生成算法[4].此方法是一种很耗时的算法,其时间复杂度和空间复杂度如下:

时间复杂度: $$O\left(\frac{-\ln(P_f)}{P_m\cdot(1-P_m)^{N_s}}\cdot N_s\right),$$

空间复杂度: $$O(l\cdot N_S).$$

其中:P_m 为匹配概率;P_f 为检测器失败概率;N_S 为自我集合大小[12].

该算法的明显不足是生成时间与受保护的自我集合成指数关系,并且会产生冗余的检测器.

(2) 线性检测器生成算法[12].该方法是对于 r-连续位(r-contiguous bits)进行匹配,即对于任意两个字符串 x,y,如果两个字符串对应的连续 r 位相同则 x,y 匹配.它可在匹配长度 l 和连续位长度 r 一定的情况下,使得时间复杂度与自我集合大小成线性关系.

时间复杂度: $$O((l-r)\cdot N_S)+O((l-r)\cdot 2^r)+O(l\cdot N_R),$$

空间复杂度: $$O((l-r)^2\cdot 2^r).$$

其中,N_R 为候选检测集合大小[4].

(3) 贪心检测器生成算法[12].贪心法消除了一些冗余,尽可能多地覆盖非己空间,此算法也是基于 r-连续位的.

时间复杂度: $$O((l-r)\cdot 2^r\cdot N_R),$$

空间复杂度: $O((l-r)^2\cdot 2^r)$.

方法(2)和方法(3)都无法解决“漏洞”问题.

(4) r 可变的检测器生成算法[13].该算法减少了“漏洞”数量和产生成熟检测器的迭代次数.文献[13]对此算法作了仿真分析.此算法的检测器生成效率有所提高,但由于预检测器随机生成,产生一个检测器需要的迭代次数为 $\left(\frac{1}{P_D}\right)^{N_S}$ (P_D 表示任意两个字符串以可变的 r 匹配概率)[13],与自我集合大小呈指数级增长,其时间复杂度仍然较高;而且当 $r_c<l$ 时也无法完全解决“漏洞”问题(r_c 表示最大匹配阈值).

综上所述,目前的检测器生成算法均不同程度地存在检测器生成效率低、“漏洞”问题或冗余检测器问题.此外,目前的方法多是采用同样长度的自我个体和检测器.本文从另外一个角度出发,对检测器长度不加限制,提出了一种既可彻底消灭“漏洞”,又能通过优化检测器进而减少冗余检测器的方法.下一节给出了此方法的详细说明.

2 检测器长度可变的非选择算法

2.1 相关定义和问题描述

首先给出相关的定义.

(1) 模式串:长为 l 的编码串,$l\geq 0$.当 $l=0$ 时,模式串=“”;当 $l>0$ 时,每一位有 h 种可能的编码.这里,h 表示字符集的大小.为便于讨论,h 个字符分别用{0,1,2,…,h–1}表示.

(2) “+”操作:“模式串 a+模式串 b”表示模式串 b 连接在 a 结尾,如“13”+“654”=“13654”.特别地,任意模式串 a+“”=“”+模式串 a=模式串 a 本身.

(3) 子串:一个模式串中的部分连续的字符,称作这个模式串的子串.即如果存在模式串 a,b,c,d 且 $d=a+b+c$,则 a,b 和 c 均是模式串 d 的子串.

(4) 求子串集合$\Phi(S)$:设 S 是一个模式串集合,集合$\Phi(S)$表示 S 中所有模式串的所有子串的集合.

(5) 待选检测器:是指任意可能的模式串.

(6) 匹配规则:对于任意两个模式串 a 和 b,当且仅当 a 是 b 的子串时,称 b 被 a 匹配.

(7) Str(node):节点 node 所表示的模式串.这里,节点 node 是一个模式串.

根据以上定义,构造出如下的问题模型:字符集的大小为 h,自我个体长度为 l,自我集合为 S,有限全空间 I 表示在此字符集上全部长度不大于 l 的字符串.对于给定的覆盖率 P_c,要求找到检测器集合 D,D 是 I 的子集,且按照上述匹配规则, $P_c\leq \frac{Cover(D,\Phi(S))}{|I-\Phi(S)|}$,$Cover(D,\Phi(S))$表示非我空间 $I-\Phi(S)$中被 D 覆盖的区域.

根据以上问题模型,对漏报率 P_f,有 $P_f=1-P_c$.此外,全空间 I 的大小为

$$\sum_{i=0}^{l}h^i=\frac{1}{h-1}(h^{l+1}-1)<\frac{h}{h-1}h^l\leq 2h^l .$$

对于第 2 节中的穷举生成算法、线性检测器生成算法、贪心检测器生成算法,其全空间为 h^l.对于 r 可变的检测器生成算法,若 r 的变化范围为($r_{\max}-r_{\min}$),则其全搜索空间为($r_{\max}-r_{\min}$)$\cdot h^l$,又对于 r 可变的检测器生成算法应有($r_{\max}-r_{\min}$)>2,故其搜索空间大于本文方法所采用的问题模型.

2.2 检测器长度可变的非选择算法

依据上面给出的问题模型,本文构造了检测器长度可变的非选择算法.该方法的特点是提出了检测器长度变化的概念,通过检测器匹配长度的变化来伸缩检测器的覆盖范围,消除漏洞区域和漏洞点.

算法 **1**. 检测器长度可变的非选择算法.

(1) 如图 2 所示,将全空间看作一棵根为空串,深度为 l+1 的 h 叉满树,其中深度为 1 时表示空串,深度为 2 时表示长度为 1 的串的集合.以此类推,当深度为 l 时,表示长度为 l–1 的串集合.每一个节点包含一个字符,但每

个节点均代表一个模式串,这个模式串等于在其父亲节点表示的模式串后面连接其自身字符而构成的模式串.如第 3 层第 2 个节点包含的字符为 1,其父亲节点的模式串为 0,则该节点所代表的模式串为“0”+“1”=“01”.又如,对于该节点的 h 个孩子来说,它们所代表的模式串分别为“01”+“0”,“01”+“1”,…,“01”+“h–1”(这里,h–1 表示一个字符).

(2) 从根节点开始,从上到下,从左至右,对于每一个预检测器d(即结节所代表的模式串):

(2.1) 如果d是某自我个体的子串,则丢弃.

(2.2) 如果d不是任何自我个体的子串,则将d加入检测器集合D,并对全空间搜索树进行剪枝.由于d所在节点的子孙节点均可被d匹配,无须再进行搜索,则删除其所有子树节点.

(3) 直到满足算法终止条件.终止条件可以是找到足够数量的检测器,或满足一定的覆盖率,或搜索完毕.

(4) 利用检测器监视受保护的串.若待检测的串被某个检测器匹配,则表示发生了异常变化.检测时可使用 K.M.P.算法[14].

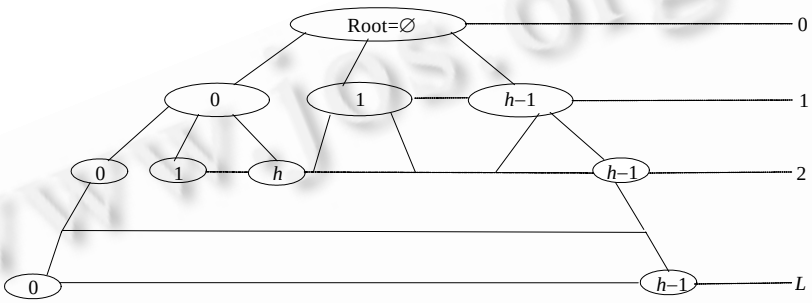

Fig.2 Whole-Space tree

图 2 全空间树

算法 **2**. 检测器优化算法.

(1) 计算不同长度的检测器的数量.根据生成过程,由于搜索过程是从根节点开始,从上到下,从左至右,因此检测器的长度应该是递增的.

(2) 按检测器长度从小到大的顺序,如果检测器 1 是检测器 2 的子串,则删除检测器 2.对于长度相同的检测器,无须执行相互匹配操作.

算法 2 借助了变长模式串匹配的特点,通过找到模式串覆盖范围的真包含关系,减少了模式串的个数.但是有必要指出:对于一个节点来说,即使其所有孩子节点所代表的模式串都是检测器,该节点所代表的模式串也可能不是检测器.

例如:字符集为{0,1},自我集为{0000,1101},检测器为{011,010}.尽管“01”的所有孩子节点都是检测器,但“01”不是检测器,不能执行合并操作(即用“01”代替检测器“011”和“010”是不合适的).

算法 1 是通过宽度优先遍历来生成预检测器,通过剪枝来减少搜索空间.实际上,可以采用利用折半遍历的思想来加速算法.具体步骤见算法 3.

算法 **3**. 自底向上查找的检测器生成算法.

定义当前搜索的层次为 i,从 $i=L$ 层开始从左到右搜索,对于每个节点:

(1) 如果该节点是检测器,则其父亲节点可能是检测器,对 $i=\lfloor i/2\rfloor$层祖先节点进行搜索.如果其祖先节点是检测器,则丢弃当前预检测器,并将其祖先节点设为当前预检测器,令 $i=\lfloor i/4\rfloor$继续进行搜索.如果其祖先节点不是检测器,令 $i=\lfloor i/2\rfloor+\lfloor i/4\rfloor$继续进行搜索.如此类推即可.最后,将找到的检测器加入检测器集并执行剪枝操作,即将找到的检测器的所有子树节点删除.

(2) 若该节点不是检测器,则其父亲节点必然不是检测器而无须进行搜索,对同一层的其余节点进行搜索.

(3) 如果找到的检测器数量已足够,或 $i=L$ 层已无可选节点,则算法终止.

因为该方法利用了折半查找的思想,算法速度更快,下面对算法时间复杂度的分析以此方法为基础.

最后,将算法 2 合并到算法 1 或算法 3,合并后为:对于每一个预检测器 d,如果它被已有检测器匹配,则直接丢弃 d 并删除其所有子树节点.

2.3 算法分析

2.3.1 时间复杂度

下面计算本算法的时间复杂度.从第 2.2 节可知,其基本操作为:(1) 产生检测器;(2) 检测器的优化;(3) 用检测器检测待检测串.

动作(1)和动作(2)是串行操作,两个动作完成了整个算法产生检测器的过程.因此,检测器生成的时间复杂度为两部分的时间总和.第(3)个动作是运用产生的优化后的检测器去进行检测待检测串的时间复杂度.

(1) 检测器生成

检测器生成由两部分操作完成,这两部分操作是乘的关系.

(a) 查找节点

采取折半查找的方法先从左向右搜索待检测器(节点的模式串),对于每个节点应执行的操作数为$\lfloor \log L \rfloor+1$次,其中,$\lfloor \log L \rfloor$表示下取整.对于最底层的节点,它若为检测器的概率是$1-\frac{N_S}{2^L}$,若需产生 N_D 个检测器,需要遍历的最底层的检测器数目最多为 $\frac{2^L}{2^L-N_S}N_R$,则查找节点的时间复杂度为 $O\left(\frac{2^L N_R}{2^L-N_S}(\lfloor \log L \rfloor+1)\right)$.实际上,按照算法 3,当找到检测器时,通过剪枝可以减少需要遍历的最低层的节点.

(b) 模式串匹配

模式串匹配可以使用 K.M.P 算法,其算法的时间复杂度与自我串的长度 L 和检测器的长度有关,为 $O(L+l)$[14].又因为 $l \le L$,则时间复杂度为 $O(L)$.设自我集合大小为 N_S,因此,匹配部分的时间复杂度为 $O(L \cdot N_R)$.

综上所述,检测器生成算法的时间复杂度是

$$O\left(\frac{2^L N_R}{2^L-N_S}(\lfloor \log L \rfloor+1)\cdot L \cdot N_S\right).$$

(2) 检测器的优化

检测器优化时,只需将检测器相互之间匹配一次,但同样长的模式串之间不需要匹配,因此时间复杂度应为需要匹配的次数乘以匹配操作的时间复杂度.最坏情况下,其时间复杂度为 $O\left(\frac{N_R \cdot (N_R-1)}{2}(L+L)\right)$,即 $O(N_R^2 L)$.

因此,产生检测器的时间复杂度为

$$O\left(\frac{2^L \cdot N_R \cdot L \cdot N_S}{2^L-N_S}(\lfloor \log L \rfloor+1)+N_R^2 L\right).$$

(3) 用检测器检测待检测串的检测时间

用检测器检测带检测串时的时间复杂度是用所有的检测器串匹配待检测串,因此,最坏情况下,检测的时间复杂度为 $O(L \cdot N_R)$.

2.3.2 空间复杂度

对于算法 3,在检测器生成过程中,只需记录当前预检测器(对应于节点模式串)和下一个待搜索的最低层的节点的位置.因此,其空间复杂度为 $O(1)$.另外,还要有 $O(N_S+N_R)$的空间存放自我集合和检测器集合.

3 仿真实验

本节对算法的漏洞数、检测率及优化后检测器个数的变化进行了模拟.当连续匹配位数 r 固定时,穷举生成算法、线性检测器生成算法和贪心检测器生成算法的漏洞个体和数量是完全一致的.因此,本文以下实验主要

与传统非选择算法(采取穷举生成算法检测器)、r 可变非选择算法[13]进行结果对比和分析.

实验 **1**. 漏报率 P_f=0 实验.

设定 L 为 16,使用随机生成的自我集合,使用本文方法计算 P_f=0 时所需的检测器个数,并实际测试是否存在“漏洞”,得到的结果如图 3 所示.

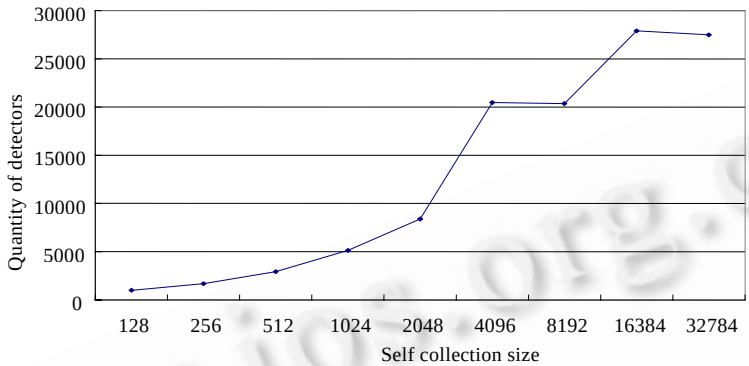

Fig.3 The relationship between the self collection size and the number of detectors when P_f=0

图 3 漏报率 P_f=0 时自我集合大小与检测器数量关系图

为了证明本实验中 P_f=0,本实验对全部非我字符串进行了遍历,通过实际检测证明,任意非我串都可被检测到.从图 3 可以看出,给出了 P_f=0 时需要的检测数目.

实验 **2**. 检测器数目的变化带来的覆盖区域变化比较.

固定 L=16,自我大小为 2 048,现通过检测器数目的变化来比较 3 种非选择算法的覆盖范围.其中,覆盖范围定义为“可检测到的集合大小占非我集合的比例”.从图 4 可以明显地看出:本算法可以通过调整检测器的长度,可以优先生成检测器覆盖面积较大的检测器,使得在相同条件检测器的效率最大化,覆盖率明显高于传统非选择算法和 r 可变非选择算法.

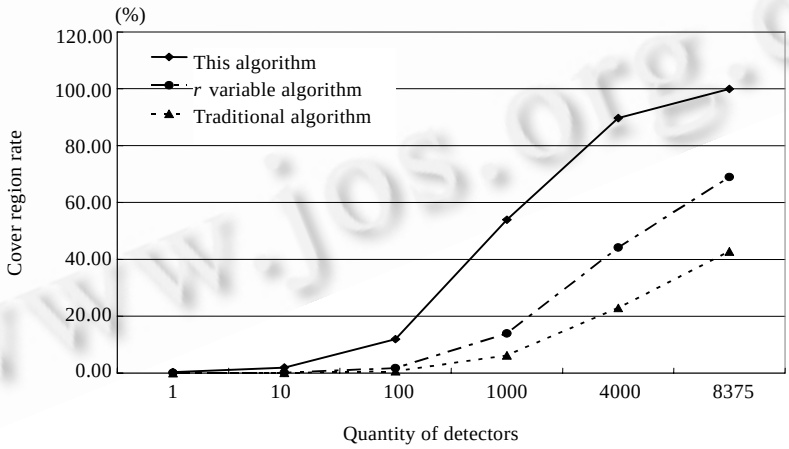

Fig.4 The relationship between the covering region and the number of detectors

图 4 检测器数量变化带来的覆盖区域变化比较图

实验 **3**. 随自我递增相同检测器数目覆盖范围变化比较.

在不同的自我集合大小的条件下,使用本算法生成检测器并保证漏报率 P_f=0,记录检测器的生成数目,使得传统非选择算法和 r 可变非选择算法产生与之相同的检测器数目(除非不能再产生新的检测器为止),检验以上 3 种非选择算法的覆盖范围,如图 5 所示.

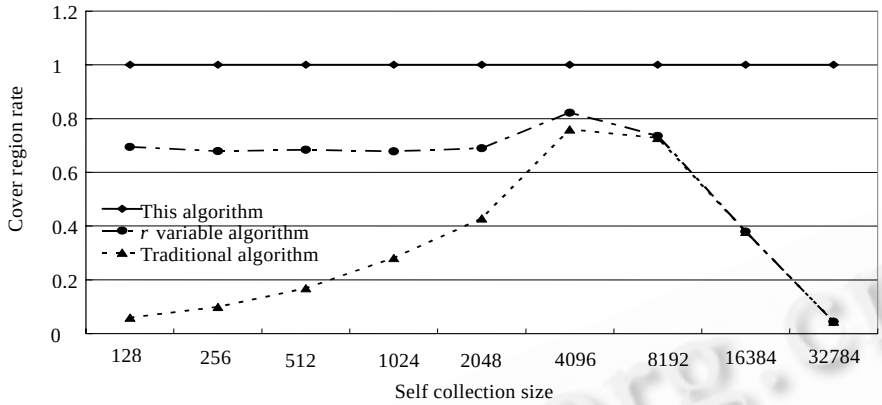

Fig.5 The relationship between the number of detectors and the self collection size

图 5 随自我递减相同检测器数目覆盖范围变化图

在自我集较小时,传统非选择算法和 r 可变非选择算法均可生成相当数量的检测器,但是其对非我空间的覆盖率明显不如本文算法;而在自我集较大时,传统非选择算法和 r 可变非选择算法能够生成的检测器数目逐步减少(这是由于“漏洞”问题),其覆盖率急剧下降.因此,本文算法生成的检测器在数目和质量上都优于传统非选择算法和 r 可变非选择算法.

4 结束语

非选择算法是人工免疫系统领域中的主要算法之一.但现有的非选择算法的检测器生成算法不能满足无“漏洞”精确检测要求,生成的检测器有冗余且数量较多.本文提出了检测器长度可变的非选择算法,解决了“漏洞”区域问题,使得生成的检测器无“漏洞”.另外,通过检测器优化算法,使得检测器数目较少,有利于实时检测.文中对算法进行了具体分析,实验证明了本文算法比已有算法具有更好的性能.

References:

[1] Hofmeyr S, Forrest S. Immunity by design: An artificial immune system. In: Wolfgan B, Jason M. D, eds. Proc. of the Genetic and Evolutionary Computation Conf. San Francisco: Morgan Kaufman Publishers, 1999. 1289−1296.

[2] de Castro LN, Timmis JI. Artificial Immune Systems: A New Computational Intelligence Approach. London: Springer-Verlag, 2002.

[3] Qi AS, Du CY. Nonlinear Models in Immunity. Shanghai: Shanghai Scientific and Technological Education Publishing House, 1998 (in Chinese).

[4] Forrest S, Perelson AS, Allen L, Cherukuri R. Self-Nonself discrimination in a computer. In: Rushby J, Meadows C, eds. Proc. of the IEEE Symp. on Research in Security and Privacy. Oakland: IEEE Computer Society,1994. 202−212.

[5] de Castro LN, von Zuben FJ. The clonal selection algorithm with engineering applications. In: Whitley LD, Goldberg DE, eds. Proc. of the GECCO 2000. San Fransisco: Morgan Kaufman Publishers, 2000. 36−37.

[6] Timmis J, Neal M. A resource limited artificial immune system for data analysis. Knowledge Based Systems, 2001,14(2-4): 121−120.

[7] de Castro LN, von Zuben FJ. An evolutionary immune network for data clustering. In: Fran FMG, Ribeiro CHC, eds. Proc. of the IEEE SBRN 2000 (Brazilian Simposium on Artificial Neural Networks). Oakland: IEEE Computer Society, 2000. 84−89.

[8] Aickelin U, Greensmith J, Twycross J. Immune system approaches to intrusion detection—A review. In: Nicosia G, *et al*., eds. Proc. of the 3rd Int'l Conf. on Artificial Immune Systems. LNCS 3239, Heidelberg: Springer-Verlag, 2004. 316−329.

[9] Dasgupta D, Majumdar NS. Anomaly detection in multidimensional data using negative selection algorithm. In: Fogel DB, El-Sharkawi MA, Yao X, Greenwood G, Iba H, Marrow P, Shackleton M, eds. Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002). IEEE Press, 2002. 1039−1044.

[10] Gonzfilez F, Dasgupta D, Kozma R. Combining negative selection and classification techniques for anomaly detection. In: Fogel DB, El-Sharkawi MA, Yao X, Greenwood G, Iba H, Marrow P, Shackleton M, eds. Proc. of the 2002 Congress on Evolutionary Computation (CEC 2002). IEEE Press, 2002. 705−710.

[11] Singh S. Anomaly detection using negative selection based on the *r*-contiguous matching rule. In: Timmis J, Bentley PJ, eds. Proc. of the 1st Int'l Conf. on Artificial Immune Systems (ICARIS). Canterbury: University of Kent at Canterbury, 2002. 99−106.

[12] D'haeseleer P, Forrest S. An immunological approach to change detection: Algorithm, analysis and implication. In: Proc. of the IEEE Symp. on Research in Security and Privacy. Oakland: IEEE Computer Society Press, 1996. 110−119.

[13] Zhang H, Wu LF, Zhang YS, Zeng QK. An algorithm of *r*-adjustable negative selection algorithm and its simulation analysis. Chinese Journal of Computers, 2005,28(10):1614−1619 (in Chinese with English abstract).

[14] He LT, Fang BX, Yu XZ. A time optimal exact string matching algorithm. Journal of Software, 2005,16(5):676−683 (in Chinese with English abstract). http://www. jos.org.cn/1000-9825/16/676.htm

附中文参考文献:

[3] 漆安慎,杜婵英.免疫的非线性模型.上海:上海科技教育出版社,1998.

[13] 张衡,吴礼发,张毓森,曾庆凯.一种 *r* 可变阴性选择算法及其仿真分析.计算机学报,2005,28(10):1614−1619.

[14] 贺龙涛,方滨兴,余翔湛.一种时间复杂度最优的精确串匹配算法.软件学报,2005,16(5):676−683. http://www.jos.org.cn/1000-9825/16/676.htm

何申(1980－),男,北京人,博士生,主要研究领域为人工免疫系统,信息安全.

王煦法(1948－),男,教授,博士生导师,CCF高级会员,主要研究领域为智能信息处理.

罗文坚(1974－),男,副教授,主要研究领域为人工免疫系统,硬件进化.