一个自体变异免疫检测器生成算法

陈 喆 周雁舟 吕志国
(解放军信息工程大学电子技术学院 郑州 450004)

摘 要 人工免疫系统作为一种计算智能方法,具备强大的信息处理和问题求解能力,检测器集的生成是构造人工免疫系统的关键技术,也是智能计算研究的热点之一。分析了传统免疫检测器生成算法,引入自体变异机制,结合空位模板技术,提出了一个自体变异的检测器生成算法。介绍了算法原理,描述了算法模板定义和实现步骤,分析了算法的性能和复杂性。理论分析与试验结果表明,该算法可以有效降低检测器集规模,提高检测器集的检测概率。

关键词 人工免疫系统,检测器,检测器生成算法,自体变异
中图法分类号 TP393.08 **文献标识码** A

Immune Detector Generating Algorithm Based on Self-mutation

CHEN Zhe ZHOU Yan-zhou LU Zhi-guo
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

Abstract As a novel branch of computational intelligence, artificial immune system has strong capabilities of information processing and problem-solving paradigm. The detector generation is the key technology in constructing artificial immune system and has been the research hotspot in computational intelligence. This paper analysed the traditional detector generating algorithm. Based on self-mutation mechanism and bland template technology, a self-mutation detector generating algorithm (SMDGA) was proposed. The principle, template definition and implementation steps of the SMDGA were described, and the performance and complexity of the SMDGA were analyzed. Both mathematical analysis and experiments show that SMDGA has the advantages in reducing the size of detector set and improving detecting efficiency.

Keywords Artificial immune system, Detector, Detector generating algorithm, Self mutation

生物体的免疫系统负责抵御外部病原体的入侵。作为一个信息处理系统,免疫系统具有增强学习、免疫记忆、分布式结构等特征[1]。研究人员将免疫系统的这些特性用于解决实际问题,形成了人工免疫系统这个新的研究领域。计算机免疫系统(Computer Immune System, CIS)是人工免疫、计算机科学的一个分支,是继神经网络、模糊系统、进化计算等研究之后的又一个热点。计算机免疫对计算机及网络安全性、可信性等领域的研究起到了推动作用,特别是在病毒防治、网络入侵和风险评估上,引入免疫概念后取得了满意的成果[2]。

美国 Forrest 等在自然免疫学中自体/非自体集(Self/Nonself)划分的基础上,提出了否定选择模型[3]用于解决异常变化的检测问题。否定选择模型首先随机地产生候选检测器集,然后在否定选择过程中清除那些对自身敏感的检测器(Detector),得到的检测器就能够而且只能够对 Nonself 进行免疫反应。可以看出,产生高性能的检测器是保障识别非自体的基础,因此,在否定选择模型中,初始检测器生成是一个重要的步骤。如何快速有效地生成尽可能多覆盖 Nonself 空间的候选初始检测器集,从而减少否定选择过程中被清除的候选检测器的数量,对于提高免疫系统性能至关重要。

本文依据否定选择原理,借鉴线性检测器生成算法产生检测器模板的方法,结合自体变异机制,在穷举检测器生成算法的基础上,提出一个新的自体变异的检测器生成算法(Self-Mutation Detector Generating Algorithm, SMDGA)。SMDGA 算法通过保留一定数量空位的变异自体来生成合格的检测器,可有效地提高检测器的生成效率,降低检测器集规模与检测漏报概率。该算法不仅可以应用在主机免疫系统中提高检测器的生成效率,而且可以推广到其它的人工免疫系统应用领域中,具有较高的理论和实用价值。

1 相关算法研究

在计算机免疫系统模型中,检测器用来识别和区分自体与非自体,检测器的性能直接决定着模型的检测性能。检测器生成算法的目的就是找出一个检测器集 R,它在不与自体集 S 中元素匹配的前提下,能尽可能多地匹配非自体集 N 中的元素。Forrest 等人[4]应用否定选择原理生成检测器,提出了穷举检测器生成算法(Exhaustive Detector Generating Algorithm, EDGA),该算法生成检测器存在冗余。Haeseleer 等人[5]提出了线性检测器生成算法(Linear Time Detector Ge-

到稿日期:2009-05-12 返修日期:2009-07-27 本文受国防预研基金项目(9140A16040206JB5203)资助。
陈 喆(1970—),男,博士生,副教授,CCF 会员,主要研究方向为系统工程与信息安全,E-mail:chenzhe197012@yahoo.com.cn;周雁舟(1971—),男,博士生,副研究员,主要研究方向为可信计算与信息安全;吕志国(1985—),男,硕士生,主要研究方向为信息安全。

nerating Algorithm,LTDGA)和贪婪检测器生成算法(Greedy Detector Generating Algorithm,GDGA)。线性生成检测器所耗时间与自体集合大小和检测器集合大小呈线性关系,该算法仍存在冗余,贪婪检测器生成算法虽然可消除冗余,但不能达到检测器生成时间最优化。Ayar 等[6]提出了带变异的检测器生成算法(Mutation Detector Generating Algorithm,MDGA),该算法通过对匹配自体的候选检测器进行变异,使之远离自体生成检测器。张衡[7]提出了一种 r 可变否定选择算法,用该算法思想来生成检测器,动态调整匹配参数,有效减少了检测盲区。罗文坚[8]提出了检测器自适应生成算法,该算法能够依据实际情况不断调整当前检测器集合,在使得仅用较小的检测器集就能够快速检测到大规模非自体空间中的异常变化的同时,也保证了算法的普适性。相关算法研究简要概述如下:

(1)穷举检测器生成算法(EDGA)。该算法在否定选择算法中最早得到应用。EDGA 算法的具体实现步骤是:首先定义自体集 S;其次随机生成一个候选检测器集合;最后把每个候选检测器和自体集中的元素做匹配比较,如果匹配成功,则放弃该检测器,否则,把它加入合格检测器集中。

(2)线性检测器生成算法(LTDGA)。在穷举线性检测器生成算法中,大多数候选检测器都被抛弃,效率并不高,但是它适合任意一种匹配规则。对于 r-连续位匹配规则,采用线性检测器生成算法比穷举法效率更高,它能在参数(字符串长度 l 和连续位长度 r)一定的前提下,使运行时间与系统输入成线性关系。LTDGA 算法分为两个阶段。第一阶段通过进行一个有限的递归运算来得到一定数量的不与自体匹配的字符串。第二阶段采用枚举法从候选检测器中随机选出检测器。

(3)其它检测器生成算法。贪婪检测器生成算法(GDGA)也是建立在 r-连续位匹配规则的基础上,通过消除冗余的检测器改进了线性检测器生成算法,提高了算法的效率,同时保证了生成的检测器能够尽可能多地覆盖非自体空间。算法分为两个阶段,第一个阶段取自于线性算法中的处理阶段,第二个阶段是检测器生成阶段。

2 匹配规则与自体变异机制

免疫检测器生成算法包括两个部分:候选检测器集的生成和合格检测器集的生成,将候选检测器与自体集的所有元素进行比较,按照检测器生成匹配规则,符合匹配条件的候选检测器将被清除,不符合匹配条件的候选检测器将成为合格检测器,免疫系统通过重复以上过程,直到产生所需要的合格检测器数目为止。

2.1 汉明距匹配规则

在检测器生成算法中最基本的操作是两个二进制串的比较,需要一个衡量两个串相似程度的尺度,即匹配规则。汉明距匹配规则通过设定阈值来确定两个字符串是否匹配。例如设阈值为 r,对于两个长度为 l 的二进制串 $X=x_1x_2\cdots x_l$ 和 $Y=y_1y_2\cdots y_l$,X 和 Y 的汉明距离为 $H(X,Y)$ 满足式(1)。若两个二进制串之间的汉明距离 $H(X,Y)$ 大于等于阈值 r,则这两个串匹配,反之则不匹配。

$$H(X,Y)=\sum_{i=1}^{l}\overline{(x_i\oplus y_i)} \tag{1}$$

2.2 自体变异机制

受免疫系统中 B 细胞对抗原的亲和力变异的启发,Ayar 等[6]提出了带变异的检测器生成算法,算法把每个候选检测器与自体数据作比较,如果匹配成功,则对候选检测器进行有导向性的变异,以使它远离自体。这个有导向性的变异是在候选检测器与自体元素匹配成功的基础上进行的,并且它是自适应的,能够根据候选检测器与自体的亲合力大小进行调节,亲合力越大,变异的几率就越大。免疫细胞在克隆时要经历变异,变异是为了维持和完善免疫系统的多样性,同时增大检测器和非自体亲和力。在汉明距离空间中,二进制串中的位置可以被随机地选择,并且它的元素可以在字母表{0,1}范围内变化。本文提出的自体二进制串变异的位置可以随机地选择,自体变异采用自体二进制串的部分位取反方法。在变异位上,可以在{0,1}中任意地选择属性来替换以前的属性。显然,与随机生成的二进制串相比,变异的自体串更有针对性地反映了非自体串。

2.3 算法环境参数

检测器生成算法中的数据采用二进制表示,即自体、非自体和检测器都用二进制串表示。为方便描述,本文应用的算法环境参数定义如表 1 所列。

表 1 算法环境参数定义表

参数	参数说明	参数	参数说明
l	二进制长度	N_S	自体集的规模,即自体集中自体字符串的个数
r	串匹配阈值	N_T	测试集的规模,即测试集合中非自体字符串的个数
S	免疫自体集	N_R	检测器集规模,即集合中有效检测器的个数
T	生成测试集	N_{R_0}	候选测试集规模,即候选检测器的个数
R	免疫检测器集	P_m	表示匹配概率,即随机字符串与检测器匹配概率
R_0	候选检测器集	P_f	检测器检测失败概率,即漏检率
f	任意一个随机字符串与自体集中的 N_S 字符串都不匹配的概率		

3 自体变异检测器生成算法

在 EDGA 和 LTDGA 算法中,前者随机产生候选检测器的数量为 2^l-1,远远大于有限的自体集;后者的检测器的生成时间与自体和检测器集合规模成线性关系,因此,这两种算法的检测器生成效率并不高。基于 EDGA 算法随机产生二进制串和 LTDGA 算法产生检测器模板的优点,本文引入自体变异机制,结合空位模板技术,提出了一个自体变异的检测器生成算法(Self-Mutation Detector Generating Algorithm,SMDGA)。SMDGA 算法通过保留一定数量空位变异自体来生成合格的检测器,能够尽可能地覆盖非自体空间,从而可以有效提高检测器的生成效率。

3.1 算法模板定义

定义 1(变异自体串 $\hat{s}$) s 表示一个自体串,l 为自体串的长度,r 为匹配阈值;$\hat{s}$ 表示 s 的变异自体串,s 的任意 c 位变异(即取反)。

定义 2(所有位变异串 $\bar{s}$) 表示 s 所有位都变异的串,当 $l=c$ 时,$\hat{s}=\bar{s}$。例如,$l=4,c=2,s=0010$,$\hat{s}=1110$,$\bar{s}=1101$;

定义 3(模板 T) 一个秩为 i 的模板 T 是指一个长度为 l,包含 $l-i$ 空位(用"*"表示)和 i 个完全确定的位的字符串。例如:"11*1*"是一个秩为 3,有两个空位的模板。设 $l=4,r=3$,自体集为{0010,1001},模板"111*"就是一个合格的检测器。

定义 4(自体串的候选检测器模板 T_S) 给定一个自体

串 $s=x_1x_2x_3\wedge x_l$，随机选择 s 的 $c(c=l-r+1)$ 位取反，剩余 $(r-1)$ 个空位，组成秩为 c 的候选检测器模板 T_S。

定义 5(自体串和模板 T_S 的候选检测模板 $T_{t,s'}$) 给定任一自体串 $s'(s'\neq s)$和一个秩为 c 的模板 T_s，随机选择模板 T_s 的 $l-c$ 个空位中的 k 位，用 $\bar{s'}$对应的位代替，剩余 $l-c-k$ 个空位，组成秩为 $c+k$ 的模板 $T_{T_s,s'}$，其中 $k\leqslant l-c$(如果 $k>l-c$，则 $c+k>l$ 与字符串长度为 l 相悖)。例如：$l=4,r=3$，$s\in\{0010,1001\}$，$T_{T_s,s'}=111*$。如果 $t=\overline{x_1x_2}\wedge\overline{x_c}*\wedge *$，则 T_S 可以从下列模板中进行选取：

$\overline{x_1x_2}\wedge\overline{x_cy_{c+1}}\wedge\overline{y_{c+k}}*\wedge *$，

$\overline{x_1x_2}\wedge\overline{x_cy_{c+1}}\wedge\overline{y_{c+k-1}}*\overline{y_{c+k+1}}\wedge *$，

$\overline{x_1x_2}\wedge\overline{x_cy_{c+1}}\wedge\overline{y_{c+k-1}}**\overline{y_{c+k+2}}\wedge *$，

……，

$\overline{x_1x_2}\wedge\overline{x_c}\wedge\overline{y_{l-k}}*\overline{y_{l-k+2}}\wedge\overline{y_{l-1}y_l}$。

在 SMDGA 算法中，检测器是由{0，1，*}组成的二进制串，$*\in\{0,1\}$，即 * 可能是“0”或“1”。因此，如果一个模板不匹配任何自体，那么这个模板就是一个候选检测器。

3.2 SMDGA 算法实现步骤

SMDGA 算法通过保留一定数量空位的变异自体生成候选检测器，候选检测器经过自体耐受、否定选择等过程生成合格检测器。候选检测器的生成采用了 EDGA 算法随机产生候选检测器的方法；合格检测器的生成采用了 LTDGA 算法检测器模板的生成原理。SMDGA 算法分为两个阶段：第一阶段由保留一定数量空位的变异自体生成候选检测器模板；第二阶段经过递归过程产生合格的检测器模板，即由候选检测器模板和自体串为基础，产生新的模板，再经过自体耐受、否定选择等过程生成合格的检测器。

自体变异检测器生成算法的具体实现过程如图 1 所示。SMDGA 算法首先定义自体集 S，最小匹配长度 r。其次随机选择一个自体串 s_n，由 s_n 随机生成一个候选检测器模板 d(有 $r-1$ 个空位)。然后将 d 与自体集中除 s_n 外的任一元素 s_i 匹配，如果 d 确定的位与 s_i 匹配数 k 大于 r，则删除检测器模板 d，否则由 s_i 取反代替 d 对应的位。最后重复以上操作，直到产生合格的检测器加载到检测器集中。自体变异检测器生成算法过程如下所示。

```
1 Initialize                                    //1 初始化
  1.1 Denote l,r                                //   设定字符串长度l,最小匹配长度r;检测器个数N_R
  1.2 DenoteN_R(or P_f)                         //   检测器个数N_R,或者检测失败概率P_f
  1.3 Denote S={s_1,s_2,∧,S_Ns}                 //   建立自体集S
  1.4 Denote R=ϕ                                //   检测器集R初始化为空
2 Generating a candidate detector               //2 生成一个候选检测器
  2.1 Randomly select a self string s_n(1≤n≤N_S)//   随机选择一个自体串s_n
  2.2 Randomly generating a detector template d //   随机生成s_n的一个秩为c的候选检测器模板d
        with order c(c=l-r+1) of s_n            //
  2.3 Let m=r-1                                 //   设置参数m为候选检测器模板d的空位数
3. Initialize i=0                               //3 由候选检测器产生成检测器前的初始化
4. Set i=i+1                                    //4 循环生成合格检测器集
  4.1 If i=n goto 4                             //   d由s_n产生不用匹配
  4.2 If i≤N_S, Calculate k=Match'(d,s_i)       //   Match'(d,s_i)为d除空位后与s_i的匹配值
    4.2.1 If k≥r Delete d goto 2                //     删除候选检测器d
    4.2.2 If k=r-1 replace(d,flip(s_i))         //     flip(s_i)为把s_i的每一位取反后的串
          And Set m=0 goto 4                    //     replace(d,flip(s_i))用flip(s_i)代替d所有空位
    4.2.3 If (k<r-1)∧(k+m≤r-1)                  //     d保持不变
          goto 4                                //     随机生成d和s_i的一个秩为l-(r-1-k)
    4.2.4 If (k<r-1)∧(k+m>r-1)                  //       的候选检测器模板T_d,A
      Randomly generate a setector Tempate T_d,A//       同时设置d=T_d,m=r-1-k
      with order l-(r-1-k) Set d=T_d,           //
          m=r-1-k goto 4                        //   把候选检测器加入检测器集
  4.3 If i>N_S, R←R∪{d}                         //   如果检测器集数小于预设的值继续生成
  4.4 If Num(R)<N_R goto 2                      //5 检测器集生成,算法结束
5 End return
```

3.3 SMDGA 算法分析

EDGA 算法和 SMDGA 算法都是随机生成候选检测器，EDGA 算法随机直接生成候选检测器。SMDGA 算法随机选择自体变异加一定数据的空位生成检测器，在不与自体集匹配的前提下，能尽可能多地匹配非自体集中的元素。SMDGA 算法的检测器生成中的数据用二进制表示，即自体、非自体和检测器都用二进制串表示，算法的环境参数定义如表 1 所列。

3.3.1 EDGA 算法基本公式

由穷举检测器生成算法的原理和汉明距离规则可知，P_m,P_f,f 和 N_R 满足式(2)—式(5)。

$$P_m=\sum_{i=r}^{l}\binom{l}{i}\cdot\left(\frac{1}{2}\right)^i\cdot\left(\frac{1}{2}\right)^{l-i}=\frac{1}{2^l}\sum_{i=r}^{l}\binom{l}{i}\tag{2}$$

$$P_f=(1-P_m)^{N_R}\tag{3}$$

$$f=(1-P_m)^{N_S}\tag{4}$$

$$N_R=N_{R_0}\cdot f\tag{5}$$

其中，$\binom{l}{i}$表示从二进制串 l 位数中随机取出 i 位的组合数。

由式(3)可得 $\ln P_f=N_R\cdot\ln(1-P_m)$，有 $P_f=e^{N_R\cdot\ln(1-P_m)}$，当 P_m 足够小时，根据泰勒公式展开 $\ln(1-P_m)\approx-P_m$，因此可得式(6)和式(7)，将式(4)和式(7)代入式(5)可得候选测试集规模式(8)。

$$P_f\approx e^{-P_mN_R}\tag{6}$$

$$N_R=-\frac{\ln P_f}{P_m}\tag{7}$$

$$N_{R_0}=\frac{N_R}{(1-P_m)^{N_S}}=-\frac{\ln P_f}{P_m\cdot(1-P_m)^{N_S}}\tag{8}$$

3.3.2 SMDGA 算法公式

采用运用概率方法对 SMDGA 生成算法的匹配概率、检测器的规模、检测概率等性能指标进行理论分析。在自体变异的检测器生成算法中，$P_{m,a}$表示匹配概率，即随机二进制串与秩为 $l-a$ 的检测器匹配的概率。依据模板 T_s 的定义，模板 T_s 有 $r-1$ 个空位和 c 个确定的位，显然 T_s 的个数 $Num(T_s)$满足式(9)。模板 $T_{T_s,s'}$有 $l-r-k$ 个空位和 $c+k$ 个确定的位，显然 $T_{t,s'}$的个数 $Num(T_{t,s'})$满足式(10)。

$$Num(T_s)=\binom{l}{r-1}\tag{9}$$

$$Num(T_{t,s'})=\binom{l-c}{l-c-k}=\binom{l-c}{k}\tag{10}$$

秩为 $l-a$ 的检测器有 a 个空位，那么任一随机产生的字符串与这个检测器的匹配概率 $P_{m,a}$满足式(11)。

$$P_{m,a}=\begin{cases}1 & a>r\\ \sum_{i=r-a}^{l-a}\binom{l-a}{i}\cdot(\frac{1}{2})^i\cdot(\frac{1}{2})^{l-i}=\frac{1}{2^{l-a}}\sum_{i=r-a}^{l-a}\binom{l-a}{i} & a\leqslant r\end{cases}\tag{11}$$

假设检测器集 $R=\{d_1,d_2,\cdots,d_{N_R}\}$，每一个检测器的空位为$\{a_1,a_2,\cdots,a_{N_R}\}$，则检测器集 R 检测失败概率 P_f 满足式(12)。

$$P_f=\prod_{i=1}^{N_R}(1-P_{m,a_i})\tag{12}$$

任一秩为 $c(c=l-r+1)$的候选检测器(有 $r-1$ 空位)，它相当于穷举检测器生成算法中的 2^{r-1}个候选检测器，因此，它与自体集中的 N_S 个字符串都不匹配的概率 f 满足式

(13)。将式(13)代入式(5)可得候选测试集规模式(15)。

$$f=2^{r-1}(1-P_m)^{N_S} \tag{13}$$

$$N_R=N_{R_0}\cdot f=2^{r-1}(1-P_m)^{N_S}\cdot N_{R_0} \tag{14}$$

$$N_{R_0}=\frac{N_R}{2^{r-1}(1-P_m)^{N_S}} \tag{15}$$

3.4 SMDGA 算法分析

算法的性能和复杂性是衡量算法优劣的重要指标。在衡量检测器生成算法的性能时，普遍将合格检测器的产生分两个基本过程考虑：① 产生一个固定长度的候选字符串；② 将该字符串与自体集 S 中的所有元素比较，分析它们是否满足匹配条件。假设这两个操作每次所花费的时间分别都是固定的。在第一个过程中，产生候选检测器集合 R_0，其时间复杂度与 N_{R_0} 成正比；第二个过程中，将每个产生的随机字符串与集合 S 的所有元素进行比较，其所花费的时间复杂性和 N_S 成正比。因此，检测器生成算法总的时间复杂性既与 N_{R_0} 成正比，又与 N_S 成正比，即 $O(N_{R_0}\cdot N_S)$。本文提出的 SMDGA 算法的空间复杂性由自体集 S 决定，由算法公式推导可知 $Num(T_s)=\binom{l}{r-1}$，由此可计算 SMDGA 的空间复杂性。穷举检测器生成算法(EDGA)、线性检测器生成算法(LTDGA)、贪婪检测器生成算法(GDGA)、阴性变异检测器生成算法(MDGA)和自体变异检测器生成算法(SMDGA)的时间复杂性和空间复杂性如表 2 所列。

表 2 免疫检测器生成算法时间与空间复杂性对比表

算法名称	时间复杂性	空间复杂性
EDGA	$O\left(\frac{-\ln P_f}{P_m\cdot(1-P_m)^{N_S}}\cdot N_S\right)$	$O(l\cdot N_S)$
LTDGA	$O((l-r)\cdot N_s)+O((l-r)\cdot 2^r)+O(l\cdot N_R)$	$O((l-r)^2\cdot 2^r)$
GDGA	$O((l-r)\cdot 2^r\cdot N_R)$	$O((l-r)^2\cdot 2^r)$
MDGA	$O(2^l\cdot N_s)+O(N_R\cdot 2^r)+O(N_R)$	$O(l\cdot(N_S+N_R))$
SMDGA	$O\left(\frac{N_R}{2^{r-1}(1-P_m)^{N_S}}\cdot N_S\right)$	$O\left(\binom{l}{r-1}\cdot N_S\right)$

在穷举线性检测器生成算法中，大多数候选检测器都被抛弃，因此 EDGA 算法的时间复杂性最高，但 EDGA 适合任意一种匹配规则。对于 r-连续位匹配规则，采用线性检测器生成算法比穷举法的效率明显更高，LTDGA 能在参数(字符串长度 l 和连续位长度 r)一定的前提下，使运行时间与系统输入成线性关系。建立在 r-连续位匹配规则的基础上的贪婪检测器生成算法 GDGA，通过消除冗余的检测器改进了线性检测器生成算法，不仅提高了算法的效率，保证生成的检测器能够尽可能多地覆盖非自体空间，而且其空间复杂性与 LTDGA 基本相同。变异的检测器生成算法 MDGA 把每个候选检测器与自体数据作比较，如果能成功匹配，则对候选检测器进行有导向性的变异，以使它远离自体。其变异能够根据候选检测器与自体的亲合力大小进行调节，亲合力越大，变异的几率就越大，因此能较大程度地提高算法的效率。本文提出的 SMDGA 算法在变异检测器生成算法的基础上，引入空位模板技术，通过保留一定数量的空位变异自体来生成合格的检测器，从而进一步提高了免疫检测器生成算法的效率。

4 仿真实验

通过实验来检证算法分析的结果。仿真实验的自体集 S 和测试集(非自体集)T 由不同的二进制串组成，随机从 0 到 2^l-1 的整数中取 N_S 个，转化为二进制串，建立自体集。测试集的建立和自体集类似，首先，随机从 0 到 2^l-1 的整数中取 1 个，转化为二进制串；然后，把这个二进制串与所有的自体集和测试集中的元素匹配(匹配规则为完全匹配规则，即汉明距离为 1)，如果不匹配，则把该二进制串加入测试集 T 中。在实验中，检测器是由{0,1,*}组成的二进制串，其中每一位用两比特表示，例如："01"表示"0"，"10"表示"1"，"11"表示"*"。仿真实验的硬件环境：CPU 为 Intel Pentium 1.8GHz，内存为 1G，硬盘容量为 80G；软件环境：操作系统为 Windows XP Professional，实验在随机二进制字符串的基础上进行。

4.1 SMDGA 算法参数 a 匹配概率实验

SMDGA 算法检测器匹配概率实验过程为：①确定 l,r 的值。②对不同的 l,r 和 a 参数值，用式(11)计算 $P_{m,a(HDGA)}$，其中 a 表示秩为 $l-a$ 的检测器有 a 个空位。SMDGA 算法的检测器匹配概率实验结果如表 3 所列。

表 3 用不同 l,r,a 参数的 SMDGA 算法检测器匹配概率 $P_{m,a}$

l	r	a	$P_{m,a}$	l	r	a	$P_{m,a}$
16	14	0	0.0021	32	28	0	9.6506e-6
16	14	2	0.0065	32	28	2	2.9738e-5
16	14	4	0.0193	32	28	4	8.9996e-5
16	14	6	0.0547	32	28	6	0.0003
16	14	8	0.1445	32	28	8	0.0008
16	14	10	0.3438	32	28	10	0.0022
16	14	12	0.6875	32	28	12	0.0059

SMDGA 算法具有 a 个空位的检测器相当于 EDGA 算法中有 2^a 个检测器。SMDGA 算法不仅能有效降低检测器集的规模，而且生成检测器的匹配概率与 EDGA 算法相比会有显著提高。通过表 3 可以看出，SMDGA 算法随着空位 a 数目的增加，算法检测器的匹配概率也显著增加，具有良好的检测性能。

4.2 检测器集规模和漏检率实验

在免疫检测器生成算法中，LTDGA 和 GDTA 算法限制在只能使用 r-连续匹配规则，EDGA，MDGA 和 SMDGA 算法则可以采用除 r-连续匹配规则以外的其它匹配规则。本节针对相同的漏检率 P_f 和不同的自体集 N_S，对采用汉明距匹配规则的 EDGA，MDGA 和 SMDGA 算法的检测器集规模和实际漏检率进行对比实验。其过程为：①确定 l,r 的值。②对于相同的 P_f 和不同的 N_S，生成检测器集 R，并计算 N_R。③用检测器集 R 对测试集 T 实施检测，得出实际的漏报概率 $P_{f(actual)}$。实验设定 $l=16$，$r=14$，$a=8$，$P_f=0.1$，$N_T=10000$，结果如表 4 所列。

表 4 EDGA，MDGA 和 SMDGA 算法的 N_R 和 $P_{f(actual)}$

N_S	EDGA		SMDGA		SMDGA	
	N_R	P_f	N_R	P_f	N_R	P_f
100	1102	0.2352	360	0.1548	289	0.1562
200	1102	0.2581	402	0.1406	335	0.1421
300	1102	0.2433	458	0.1512	384	0.1387
400	1102	0.2486	573	0.1112	491	0.1269
500	1102	0.2399	643	0.1349	509	0.1284
600	1102	0.2612	685	0.1262	561	0.1105
700	1102	0.2478	762	0.1128	603	0.0968
800	1102	0.2395	893	0.1004	659	0.0873

从表 4 可以看出，当漏报概率 P_f 确定时，随着自体集规模 N_S 的增加，EDGA 算法的检测器规模保持不变，而 MDGA 和 SMDGA 算法的检测器规模均会增大，但明显小于 EDGA

算法的检测器规模，因为其采用了空位模板技术，在检测器规模上，SMDGA 算法性能要优于 MDAG 算法，如图 1 所示。在 3 个算法的漏检率对比中，采用 EDGA 算法的检测器集实际漏报概率高于 SMDGA 和 MDGA 算法。而 SMDGA 和 MDGA 算法的实际漏报率基本相同，如图 2 所示。

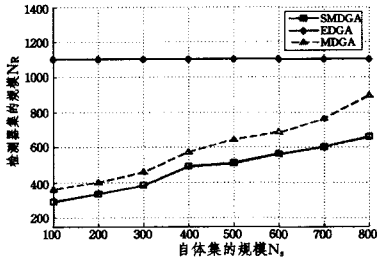

图 1　EDGA，SMDGA 等算法的生成检测器规模对比

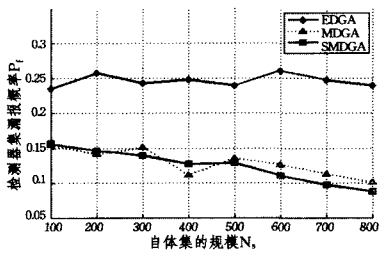

图 2　EDGA，SMDGA 等算法的生成检测器漏检率对比

4.3　检测器集生成时间对比实验

检测器集生成时间和检测概率的实验过程为：①确定 l，r 和 N_S 的值。②设定检测器集的规模 N_R，分别用 EDGA，LTDGA，GDGA，MDGA 和 SMDGA 算法生成检测器集 R，并记录生成检测器集的时间 G_R。③用检测器集 R 检测测试集 T，记录检测概率 D_R，有 $D_R=1-P_f$。检测器集生成时间实验序设定 $l=16$，$r=14$，$a=8$，$N_S=400$，$N_T=10000$，对每一个 N_R 值实验 10 次，不同算法检测器集的平均生成时间 $AveG_R$（单位为 ms）如表 5 所列。

表 5　不同算法检测器集平均生成时间

N_R	$AveG_{R(EDGA)}$	$AveG_{R(LTDGA)}$	$AveG_{R(GDGA)}$	$AveG_{R(MDGA)}$	$AveG_{R(SMDGA)}$
50	34339	23541	25041	24012	21946
100	69661	48538	50779	48735	43231
150	108788	75514	76775	72512	64578
200	142105	102374	103185	98813	86429
250	176583	132402	133736	119718	108240
300	217623	162115	165068	144612	130056
350	255471	195251	199532	170409	150841
400	285783	230524	239618	197323	173481
450	324924	271732	285692	220124	195645
500	353220	310341	334028	248468	216454

用 C_R 表示一个合格检测器的生成时间，则有 $C_R=AveG_R/N_R$。从表 5 可以看出，随着检测器集规模 N_R 的增加，所有检测器生成算法的检测器集算法的平均生成时间 $AveG_R$ 基本呈增长趋势，如图 3 所示。

由图 3 可以看出，LTDGA 算法检测器集平均生成时间与自体集的规模呈线性关系。GDGA 算法检测器集平均生成时间受自体集规模影响较大，随着自体集规模的增加，检测器集的生成时间呈指数级增长趋势。EDGA 算法检测器集平均生成时间虽与自体集规模无明显关系，但其生成时间明显高于其他免疫检测器生成算法。与以上 3 种算法相比，MDGA 算法检测器平均生成时间最优，并不随自体集规模的增加而增长。SMDGA 算法因采用了空位模板的自体变异技术，其检测器平均生成时间还要优于 MDGA 算法，更适应于要求检测实时性高的环境。

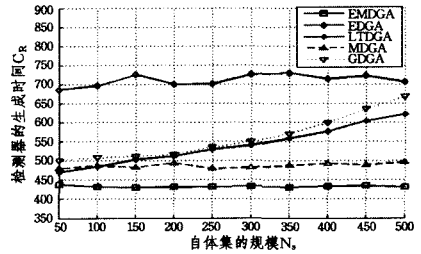

图 3　不同免疫检测器生成算法生成一个检测器的时间对比

结束语　本文依据否定选择原理，借鉴线性检测器生成算法产生检测器模板的方法，结合自体变异机制，在穷举检测器生成算法的基础上，提出一个新的自体变异的检测器生成算法。该算法解决了现有检测器生成效率低的问题，能有效降低生成检测器集的规模，减少检测器的平均生成时间，控制漏报概率，提高检测器生成效率。实验证明其性能明显优于 EDGA，LTDGA 和 GDGA 算法，与 MDGA 算法相比，也具有检测器集规模小和生成效率高的优势。SMDGA 算法具有良好的实时性和检测准确性，不仅可以应用在主机免疫系统模型中提高检测器的生成效率，而且可以推广到其它的人工免疫系统应用领域中，具有良好的工程实用价值。

参 考 文 献

[1] Dasgupta D. An overview of Artificial Immune Systems and their Application[M]. Berlin: Springer-Verlag, 1999

[2] 郑瑞娟，王慧强，等. 计算机免疫应用研究[J]. 计算机研究与发展，2006，43(8)：403-408

[3] Forrest S, Helman P. An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications[C]// Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy. 1996：192-211

[4] Forrest S, Perelson A S, Allen S, et al. Self-Nonself Discrimination in a Computer[C]// Proceedings of IEEE Symposium on Research in Security and Privacy. 1994：202-212

[5] Haeseleer P D, Forrest S, Helman P. An Immunological Approach to Change Detection: Algorithms, Analysis and Implications[C]// Proceeding the 1996 IEEE Symposium on Security and Privacy. 1996：110-119

[6] Ayara M, Timmis J, Deemos L, et al. Negative Selection: How to Generate Detectors[C]// Proceedings of 1st International Conference on Artificial Immune Systems (ICARIS-2002). 2002：89-98

[7] 张衡，吴礼发，等. 一种 r 可变否定选择算法及其仿真分析[J]. 计算机学报，2005，28(10)：1614-1619

[8] 罗文坚，曹先彬，等. 检测器自适应生成算法[J]. 自动化学报，2005，31(6)：907-916