Appl Intell (2015) 42:289–302
DOI 10.1007/s10489-014-0599-9

An immune optimization based real-valued negative selection algorithm

Xin Xiao · Tao Li · Ruirui Zhang

Published online: 8 October 2014

Abstract Negative selection algorithms are important for artificial immune systems to produce detectors. But there are problems such as high time complexity, large number of detectors, a lot of redundant coverage between detectors in traditional negative selection algorithms, resulting in low efficiency for detectors' generation and limitations in the application of immune algorithms. Based on the distribution of self set in morphological space, the algorithm proposed in this paper introduces the immune optimization mechanism, and produces candidate detectors hierarchically from far to near, with selves as the center. First, the self set is regarded as the evolution population. After immune optimization operations, detectors of the first level are generated which locate far away from the self space and cover larger non-self space, achieving that fewer detectors cover as much non-self space as possible. Then, repeat the process to obtain the second level detectors which locate close to detectors of the first level and near the self space and cover smaller non-self space, reducing detection loopholes. By analogy, qualified detector set will be obtained finally. In detectors' generation process, the random production range of detectors is limited, and the self-reaction rate between candidate detectors is smaller, which effectively reduces the number of mature detectors and redundant coverage. Theoretical analysis demonstrates that the time complexity is linear with the size of self set, which greatly reduces the influence of growth of self scales over the time complexity. Experimental results show that IO-RNSA has better time efficiency and generation quality than classical negative selection algorithms, and improves detection rate and decreases false alarm rate.

Keywords Anomaly detection · Artificial immune system · Negative selection algorithm · Detector generation · Immune optimization

1 Introduction

The negative selection algorithm (NSA) is one of the main algorithms in artificial immune systems. It simulates the immune tolerance in T-cell maturation process of biological immune system, and achieves effective recognition of non-self antigens by clearing self-reactive candidate detectors without any prior knowledge. It is widely applied in the fields of fault diagnosis, intrusion detection, pattern recognition and etc. [1–8, 24, 29]

Forrest et al. [9] originally established the framework of negative selection algorithms in 1994, and adopted binary string representation of antigens (samples) and antibodies (detectors), and r-contiguous-bit matching method to compute the matching degree between antibodies and antigens, which is successfully applied in anomaly detection system. Balthrop et al. [10] pointed out the vulnerabilities which exist in the r-contiguous-bit matching algorithm and presented an improved r-chunk matching mechanism.

For the lack of binary representation in dealing with numerical data and multi-dimensional data, Gonzalez and Dasgupta [11] proposed a real-valued negative selection algorithm (RNSA), which encodes antigens and antibodies in n-dimensional [0,1] space and adopts Minkowski

X. Xiao (✉) · T. Li
School of Computer Science, Sichuan University,
Chengdu 610000, China
e-mail: xiaoxin618@gmail.com

R. Zhang
School of Business, Sichuan Agricultural University,
Chengdu 610000, China

distance to compute the matching degree between antigens and antibodies. Because radiuses of detectors are the same size and are difficult to accurately determine, there are holes in the algorithm and the detection rate is not high. Zhou et al. [12, 13] presented a real-valued negative selection algorithm with variable-sized detector (V-Detector), which dynamically determines the detector's radius by calculating the distance between the center of the candidate detector and the closest self antigen. The work also proposed a method for computing detectors' coverage rate in non-self space based on the probability, and obtained better detection results. Joseph et al. [14] introduced hyper-ellipsoidal detectors in negative selection algorithm, Ostaszewski et al. [15] introduced super-rectangular detectors, and Zhang et al. [16] introduced a matrix-based detector model, which all achieve the original coverage with fewer detectors. Gao et al. [17] put forward a negative selection algorithm based on genetic algorithms, which searches optimal detectors by genetic mechanism. Yang et al. [18] proposed a multi-population genetic based negative selection algorithm in which the self set was divided according to features and sub populations evolved independently, reducing redundant coverage between detectors. Stibor [19] put forward a classification method of self detectors, which dynamically adjusted the radius of self by ROC analysis to balance the detection rate and the false alarm rate. Chen et al. [20] presented a negative selection algorithm based on hierarchical clustering of self set, which preprocessed the self set to increase the efficiency of detectors' generation.

How to generate efficient detector set is the key of negative selection algorithms. The work in [1, 9, 20, 21, 23] pointed out that current problems of negative selection algorithms are as follows. First, the time complexity is $O(-\ln P_{tp} \cdot N_s/(P' \cdot (1-P')^{N_s})$, where P_{tp} is the detection rate, N_s is the size of self set, and P' is the self-reaction rate between detectors. The cost of detectors' generation is exponential to the size of self set, the generation efficiency of mature detectors is low, and the execution time severely limits practical applications. Second, for a given detection rate P_{tp}, the size of required detector set $N_d \approx \ln(1-P_{tp})/\ln(1-P')$. The number of detectors is large, and there exists a lot of redundant coverage between detectors, making the detection time much longer. Third, pathogens are always evolving in the direction of vulnerabilities. Holes exist more or less in various negative selection algorithms, resulting in low detection rate.

This paper presents a real-valued negative selection algorithm based on immune optimization (IO-RNSA). The main contributions are as follows. First, introduces the immune optimization mechanism. Detectors are not randomly produced throughout the shape space, but evolved from successive generations with the self set as the initial population, which maintains the diversity of detectors and reduces the redundancy. Second, generates detectors hierarchically, and limits the random generation range of detectors. The algorithm gives priority to producing detectors of large size which are distributed in low coverage areas, decreasing the number of mature detectors. And then the algorithm generates detectors with small size which are distributed in the area close to the self space, reducing the number of vulnerabilities. Third, makes performance analysis of detectors' generation, and shows effectiveness of the algorithm through simulations.

The remainder of this paper is organized as follows. Related work is introduced in Section 2, including two classical negative selection algorithms RNSA and V-Detector, and basic concepts of the immune optimization. The idea, implementation strategies, and theoretical analysis of IO-RNSA are described in Section 3. The effectiveness of IO-RNSA is verified in Section 4 through experiments. Finally, the conclusion is given and further work is proposed in the last section.

2 Related work

2.1 Basic definitions of RNSA

Immune events occur in the shape space S, and the process of antibodies recognizing antigens is the process of antibodies matching and binding antigens [2, 9, 22]. The algorithm discussed in this paper is based on real value. So, S is the n-dimensional [0,1] space, and antibodies and antigens are hyper-spheres in the space.

Definition 1 The artificial immune system is expressed as $\Sigma_{AIS} = (X_{AIS}, \gamma_{AIS}, G_{AIS})$. X_{AIS} is the input to be detected, which may be network packets, or file signatures. The input domain can be divided into two mutually exclusive sets, which are a normal set and an abnormal one. γ_{AIS} is the output of Σ_{AIS}. G_{AIS} represents the nonlinear function of the relationship between the input and output.

$$\gamma_{AIS} = G_{AIS}(X_{AIS}) = \begin{cases} 0 & X_{AIS}\ belongs\ to\ selves \\ 1 & X_{AIS}\ belongs\ to\ non-selves \end{cases} \tag{1}$$

Definition 2 Antigens are represented as the tuple $ag =< x, r_s >$. x is the location of the sample ag, and is expressed as $x =< x_1, x_2, \ldots, x_n >$. n is the data dimension, $x_i \in [0, 1](1 <= i <= n)$ is the normalized value of the i^{th} attribute of ag, and r_s is the variation range of ag. The antigen set $AG = \{ag|ag =< x, r_s >, r_s \in [0, 1]\}$ is the collection of all the samples in the space.

Definition 3 The self set $Self \subset AG$ represents all the normal samples, and the non-self set $Nonself \subset$

AG represents all the abnormal samples. Then, $Self \cap Nonself = \emptyset$, $Self \cup Nonself = AG$.

Definition 4 The training set $Train \subset Self$ is a subset of *Self*, and is the priori knowledge of detections.

Definition 5 The structure of detectors is similar to antigens, $d =< y, r_d >$. y is the location of the detector d, and is expressed as $y =< y_1, y_2, \ldots, y_n >$. $y_i \in [0, 1](1 <= i <= n)$ is the i^{th} position vector of detector d, and r_d is the radius of d. The detector set is expressed as $D = \{d | d =< y, r_d >, r_d \in [0, 1]\}$.

Definition 6 The affinity between antibodies and antigens is the binding strength between them. For real coding, it is usually related to the distance between antibodies and antigens. Euclidean distance is adopted in this paper.

$$affinity\,(ag, d) = dist\,(ag.x, d.y) = \sqrt{\sum\nolimits_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

In the process of detectors' generation, if $dist(ag.x, d.y) <= r_s + r_d$, the detector d triggers the immune self-reaction, and cannot become mature. In the process of detection, if $dist(ag.x, d.y) < r_d$, the detector d recognizes the antigen ag as a non-self.

Definition 7 The detection rate *DR* is the ratio of non-self samples which are correctly recognized by detectors to the entire non-selves, and is expressed as (3), where *tp* represents the number of non-selves correctly recognized by detectors, and *fn* represents the number of non-selves wrongly recognized.

$$DR{=}R_{DR}\left(\sum\nolimits_{AIS}\right){=}P\,(\gamma_{AIS}{=}\ 1/X_{AIS}{\in}Nonself){=}\frac{tp}{tp{+}fn} \tag{3}$$

Definition 8 The false alarm rate *FAR* is the ratio of self samples which are wrongly recognized as non-selves by detectors to the entire selves, and is expressed as (4), where *fp* represents the number of selves wrongly recognized by detectors, and *tn* represents the number of selves correctly recognized.

$$FAR{=}R_{FAR}\left(\sum\nolimits_{AIS}\right){=}P\,(\gamma_{AIS}{=}\ 1/X_{AIS}{\in}Self){=}\frac{fp}{fp{+}tn} \tag{4}$$

In real-valued negative selection algorithms, the detection mechanism is shown in Table 1.

Table 1 Mechanism of detection

```
Input: the detector set $D$, the antigen set to be judged *Ag'*
Output: *tp, fn, fp, tn* are the number of antigens which mean true positive, false negative, false positive, and true negative
**Step 1.** $tp = 0, fn = 0, fp = 0, tn = 0$;
**Step 2.** Select an antigen *ag* from *Ag'* in sequence;
**Step 3.** Calculate the Euclidean distances between *ag* and all the detectors in $D$. If $dist(d.y, ag.x) < r_d$ for at least one detector $d$ and this antigen *ag* is non-self, then $tp++$; If $dist(d.y, ag.x) < r_d$ for at least one detector $d$ and this antigen *ag* is self, then $fp++$; If $dist(d.y, ag.x) < r_d$ for non-detectors and this antigen *ag* is self, then $tn++$; If $dis(d,\ ag_t) < r_d$ for non-detectors and this antigen *ag* is non-self, then *fn*++.
**Step 4.** If antigens in *Ag'* aren't all tested, jump to Step 2; if not, return*tp, fn, fp, tn* and the process ends.
```

2.2 RNSA

RNSA adopts detectors with fixed size and the preset number of detectors as the termination condition [9]. The algorithm randomly generates a candidate detector $d_{new} =< y, r_d >$, and then calculates the distance between d_{new} and any self $ag =< x, r_s >$ in the training set. If the candidate does not react with any self, put d_{new} into the detector set D. This algorithm is expressed in Table 2.

2.3 V-Detector

V-Detector adopts detectors with variable size and the expected coverage rate as the termination condition [12, 13]. The algorithm randomly generates the center of candidate detector y in the shape space, and then obtains the minimum Euclidean distance between y and any self $ag{=}{<}x{,}r_s{>}$in the training set. If $dist(y, ag.x) > r_s$, generate a detector $d_{new} =< y, r_d >$ with y as the center and $r_d = dist(y, ag.x) - r_s$ as the radius.

Figure 1 shows the contrasts of RNSA and V-Detector. Where, the blue area represents selves, the light gray area

Table 2 Procedure of RNSA

```
Input: the self training set *Train*,
the number of required detectors *maxNum*
Output: the detector set $D$
**Step 1.** Initialize the self training set *Train*;
**Step 2.** Randomly generate a candidate detector $d_{new}$. Compute Euclidean distances between $d_{new}$ and all the selves in *Train*. If $dis(d_{new}.y, ag.x) <= r_d + r_s$ for at least one self *ag,* execute Step 2; if not, execute Step 3.
**Step 3.** Add $d_{new}$ into the detector set $D$;
**Step 4.** If the size of $D$ satisfies $N_d > maxNum$, return $D$, and the process ends; if not, jump to Step 2.
```

represents vulnerabilities, and the unfilled area represents detectors. In RNSA, there are many vulnerabilities and the detection rate is low because the radius of detectors is fixed and hard to be determined. In V-Detector, due to variable-sized detectors, detectors with large radius cover most of the non-self space and detectors with small radius cover holes, which not only reduces the number of detectors but the number of holes. As can be seen from Fig. 1, there are general problems which are also raised by the introduction section in these two algorithms: low generation efficiency of mature detectors; large number of detectors and a lot of redundant coverage resulting in longer detection time; existence of holes causing low detection rate.

2.4 Immune optimization

Immune optimization mechanism simulates the activation process of immune cells in the immune response [26, 27], which is used to solve problems of function optimization, combinatorial optimization and etc. When immune cells are stimulated by antigens, clonal proliferation occurs, a large number of clones are created, and then these cells differentiate into effector ones and memory ones through hyper-mutation. During proliferation, effector cells will produce a large number of antibodies, and then antibodies will replicate and hyper-mutate to increase their affinities and reach affinity maturation ultimately in order to eliminate antigens quickly. Table 3 shows the general flow of the immune optimization algorithm.

Table 3 General flow of the immune optimization algorithm

Step 1. Initialize, and generate the initial population randomly in the domain;
Step 2. Calculate the affinity and stimulation level of each antibody in the population. If the termination condition is satisfied, output the memory set and the process ends; if not, execute Step 3;
Step 3. Select better individuals from the population according to the stimulation level to be cloned and activated;
Step 4. Make clones to mutate. The lower the affinity of a clone is, the higher the mutation rate is;
Step 5. Choose best individuals to join the memory set from the clone collection;
Step 6. Generate new antibodies randomly, update the population, and execute Step 2.

3 IO-RNSA

3.1 Basic idea of the algorithm

In practical problems, it is impossible for normal data to be distributed randomly in the shape space. They are highly concentrated, and are located in a very small part of the space [8, 9, 28]. The main idea of the algorithm is as follows. According to the distribution of self set in shape space, the algorithm introduces the immune optimization mechanism, and generates candidate detectors hierarchically from far to near with selves as the center, which decreases the redundancy between detectors and reduces the detection holes. The algorithm adopts detectors with variable size, and expected coverage of non-self space as the termination condition for the process of detectors' generation. Self set is regarded as the evolutionary population on which immune optimization operations will be performed. Then, the algorithm does immune selection operation, clonal proliferation operation and hyper-mutation operation on this population, and performs negative selection to obtain detectors of the first level locating far away from selves in the non-self space where coverage rate is low. Repeat the process to get detectors of the second level which locate close to the first level detectors and away from selves but closer than the first level detectors in the non-self space. The antibody mutation rate will decrease with the evolution generation increasing. When detectors are close to selves, detectors have small

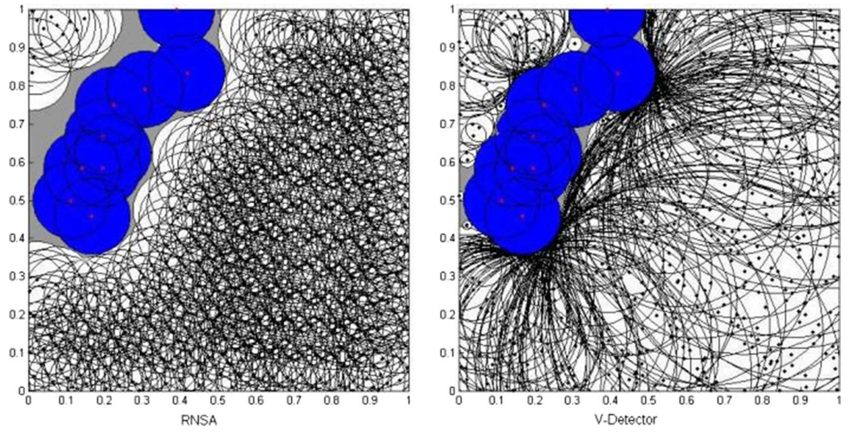


Fig. 1 Contrasts of RNSA and V-Detector

coverage, reducing detection holes, and when detectors are far from selves, detectors have large coverage, achieving that fewer detectors cover as much non-self space as possible. That is to say, the algorithm produces detectors with large size in low coverage areas, to be followed with detectors of small size in areas close to self space. By analogy, the detector set which satisfies conditions will be obtained finally. Steps of the algorithm are shown in Table 4.

3.2 Comparisons with RNSA and V-Detector

Iris data set is one of the classic machine learning data sets published by the University of California Irvine (UCI) [25], which is widely used in pattern recognition, data mining and other fields by researchers. We select data records of the category "setosa" in the data set Iris as self antigens, select "sepalL" and "sepalW" as the first dimension attribute and the second dimension attribute of antigens, and select top 20 records as the self training set. Figures 2 and 3 illustrate the idea of IO-RNSA, and differences between IO-RNSA and classic negative selection algorithms RNSA and V-Detector. Filled circles represent self individuals in the space, and unfilled circles represent detectors. RNSA produces detectors with fixed size, and V-Detector dynamically produces variable-sized detectors according to the distance between the center of detectors and the nearest self antigen. For these two algorithms, there is redundant coverage in non-self space between mature detectors with the increase of coverage rate. IO-RNSA hierarchically produces detectors from far to near, and newly-generated candidate detectors are distributed around those of the last level, avoiding from repeated coverage with mature detectors and achieving fewer detectors covering as much non-self space as possible.

Table 4 Steps of IO-RNSA

Input: the self training set *Train*, expected coverage c_{exp}
Output: the detector set D
t: the evolution generation
n_0: sampling times in non-self space, $n_0 > max(5/c_{exp}, 5/(1-c_{exp}))$
i: the number of non-self samples
m: the number of non-self samples covered by detectors
CD: the candidate detector set $CD = \{d|d =< y, r_d >, r_d \in [0, 1]\}$
Step 1. Initialize the self training set *Train* as the evolutionary population, $t = 1, i = 0, m = 0, CD = \emptyset$;
Step 2. If the mutation rate is less than or equal to r_s, return D and the program ends; if not, perform the immune selection, clonal proliferation and hyper-mutation to get the mutation population;
Step 3. Fetch an individual d_{new} in sequence from the mutation population, if the population is empty, jump to Step 2;
Step 4. Compute distances between $d_{new}$and all selves in the training set *Train*. If $d_{new}$is recognized by at least one self, discard d_{new} and jump to Step 3; if not, increase$i$;
Step 5. Calculate distances between d_{new} and all detectors in the set D. if $d_{new}$is not identified by any detector, add d_{new}into the candidate detector set *CD*; if not, increase m, and determine whether the expected coverage c_{exp} is reached. If so, reduce the mutation rate, increase t, and jump to Step 2;
Step 6. Determine whether the number of non-self samples i r eaches the sampling times n_0. If $i = n_0$, integrate the candidate detector set *CD* into D, reset i, m, *CD*. Jump to Step 3.

3.3 Immune optimization mechanism

The algorithm introduces the immune optimization mechanism, including immune selection, clonal proliferation and hyper-mutation. And it adopts the self set as the initial population, and searches in the non-self space to get optimum detectors.

The immune selection operator chooses part of antibodies to enter the next operation - clonal proliferation according to the stimulation levels of antibodies. The aim of negative selection algorithms is to cover all non-self space with optimum detectors, and it is necessary to cover the area around each self. Therefore, the algorithm selects all self individuals as the evolutionary population.

The clone operator simulates the clonal expansion mechanism in the immune response. When antibodies identify foreign antigens, clonal proliferation will occur. The number of clones for each detector is limited in this paper. Set the maximum of clones for each detector is $c_{max}$and the minimum of clones is c_{min}, then the number of clones $c(d)$for detector$d$ is calculated as follows.

$$c(d) = ceil(c_{max} - (1/t)(c_{max} - c_{min})) \tag{5}$$

Where, t is the evolution generation. And *ceil()* is a function which returns a minimum integer greater than or equal to the specified expression. The formula shows the relationship between evolution generation and clone scale. In the beginning of evolution, detectors are far away from selves, and the radius of detectors is large. So, the duplication level of detectors is low, trying to cover larger non-self space with fewer detectors. In the last stage of evolution, detectors are close to selves, and the radius of detectors is small. So, the duplication level is high, giving detectors more opportunities to fit the self space.

The mutation operator simulates the hyper-mutation mechanism in the immune response. Antibodies can change their genes randomly through mutation, and antibodies with higher affinities will be produced. In this paper, different candidate detectors will be generated in a wider scope through mutation, and variation ranges of candidate

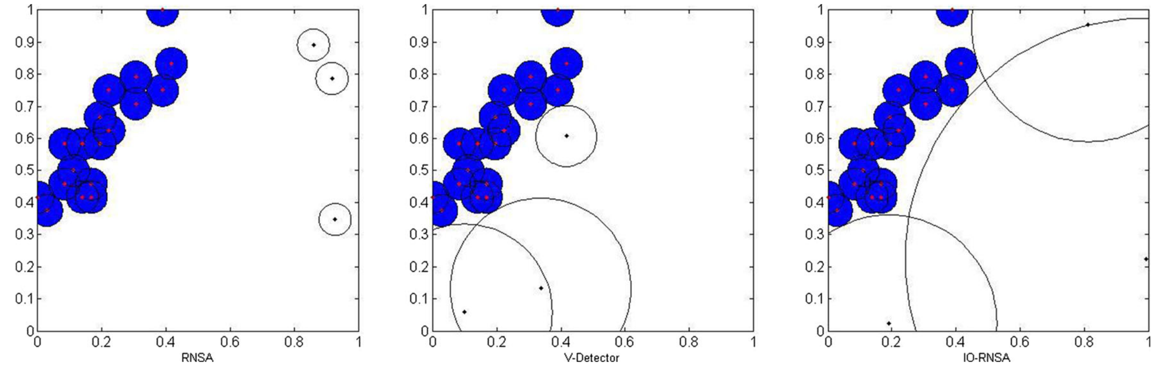


Fig. 2 Comparisons of detectors' generation for RNSA, V-Detector and IO-RNSA (simultaneously generate three detectors)

detectors in each generation are limited in different locations. The formulas are as follows.

$$d'.y_i = d.y_i + \beta \cdot N(0,1) 1 <= i <= n \quad (6)$$

$$\beta = \sqrt{n}/exp(t-1) \quad (7)$$

$$\sqrt{\mathrm{n}}/exp(t) < dist(d'.y, d.y) <= \sqrt{\mathrm{n}}/exp(t-1) \quad (8)$$

Where, d' is the newly-generated candidate detector. *N(0,1)* is the random variable. β is the mutation rate, adjusting the variation range. *n* is the data dimension. It is difficult to compute for the above formulas when randomly generating new candidate detectors. So, polar coordinates are adopted. Set the polar diameter is ρ in n-dimensional space, and the polar angles are $\theta_1\theta_2, \ldots, \theta_{n-1}$. The above formulas are expressed as follows.

$$\begin{aligned} d'y_1 &= dy_1 + \rho \cdot cos(\theta_1) \\ d'y_2 &= dy_2 + \rho \cdot sin(\theta_1)cos(\theta_2) \\ d'y_{n-1} &= dy_{n-1} + \rho \cdot \sin(\theta_1)\sin(\theta_2)\ldots cos(\theta_{n-1}) \\ d'y_n &= dy_n + \rho \cdot sin(\theta_1)\sin(\theta_2)\ldots sin(\theta_{n-1}) \end{aligned} \quad (9)$$

Where, ρ is a random variable in [$\sqrt{n}$*/exp(t),* $\sqrt{n}$*/exp(t-1)*], and $\theta_1\theta_2, \ldots, \theta_{n-1}$ are random variables in [0,360]. Therefore, variation range of candidate detectors in each generation is limited to a super ring.

At the early stage of detectors' generation, the mutation rate is greater, and detectors locate mainly in the area of non-self space far from selves. After some generations, the mutation rate decreases, and detectors locate mainly in the area of non-self space close to selves. New mature detectors of each generation have different variation ranges, and cover different regions in non-self space, which reduces redundancy between detectors. When the mutate rate β is less than or equal to r_s, that is to say, when the evolutionary generation *t>=floor(1+log(*$\sqrt{n}/r_s$*))*, the population stops evolving and the coverage of non-self space for mature detectors has reached the requirement. *floor()* is a function which returns a maximum integer less than or equal to the specified expression.

Adopting Iris data set as well, Fig. 4 shows the process of detectors' generation. The navy blue filled circles represent self individuals in the space. Figure 4a shows variation ranges of two generations for a self, where the gray area is the variation range of first generation, and the light blue area is the variation range of second generation. Mature detectors of each generation are limited to a certain area. In Fig. 4b,

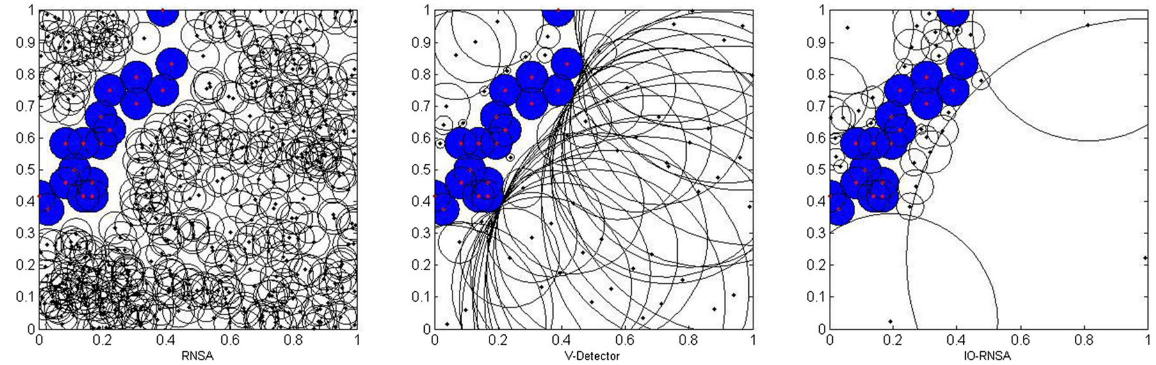
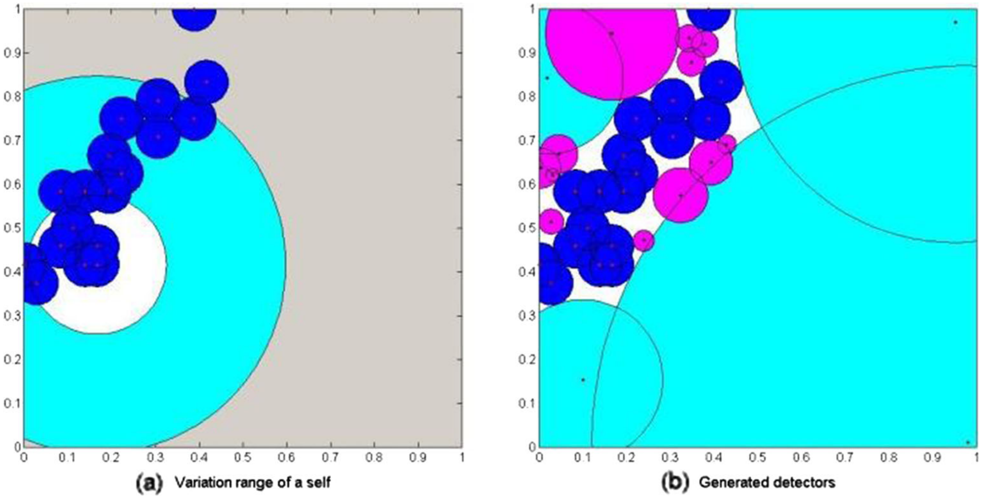


Fig. 3 Comparisons of detectors' generation for RNSA, V-Detector and IO-RNSA (to reach the expected coverage 90 %, three algorithms need 408, 56 and 32 detectors respectively)

Fig. 4 The process of detectors' generation



Variation range of a self

Generated detectors

the light blue area is coverage of first generation detectors, and the purple area is coverage of second generation. Detectors of the first level are distributed in the non-self space far away from selves, and have larger coverage; while detectors of the second level are distributed in the non-self space close to detectors of the first level and nearer to selves, and have smaller coverage.

3.4 Calculation method of coverage in non-self space

The non-self space coverage of detectors is equal to the ratio of the non-self space volume covered by detectors to the total volume of non-self space [12]. It is demonstrated in (10).

$$P=\frac{V_{covered}}{V_{nonself}}=\frac{\int_{covered}dx}{\int_{nonself}dx} \tag{10}$$

Because of duplicated coverage between detectors, the equation above cannot be directly calculated. Method of probability estimation is introduced to calculate the non-self space coverage of detectors P in this paper. In the non-self space, the probability of sampling one time which is covered by detectors obeys the binomial probability *b(1,P*), and the probability of sampling $n$times obeys the binomial probability $b(n, P)$. If the number of times continual sampling in the non-self space$i$ is not larger than n_0, and $\frac{m}{\sqrt{n_0P(1-P)}} - \sqrt{\frac{n_0P}{1-P}} > Z_\alpha$, the coverage of non-self space of detectors reaches P [12, 13]. Z_α is a quantile of the standard normal distribution, and the value of a determines the precision. m is the number of times sampling in the non-self space covered by detectors in one round. n_0 is the minimal positive integer greater than both $5/P$and 5/(1-P), and is a determined value.

Therefore, the algorithm needs to record i and m in the procedure of detectors' generation. At first, i =0 and m =0. When i is less than n_0, it belongs to one round. After sampling randomly in the non-self space, the algorithm judges if the sample is covered by any detector in D. If it is not covered, produce a candidate detector using this sample's position vector and put it in CD. If it is covered, calculate if $\frac{m}{\sqrt{n_0P(1-P)}} - \sqrt{\frac{n_0P}{1-P}} > Z_\alpha$. If so, the non-self space coverage of detectors reaches P and stop sampling. If not, increase i. When i is equal to n_0, unite CD into D in order to alter the non-self space coverage, reset i =0m =0, and start a new round. After several rounds of sampling, the detector set D gradually increases, and the non-self space coverage of detectors becomes greater.

3.5 Performance analysis

Set that the total number of samples in the problem space is N_{Ag}, the size of self set is N_{Self}, the size of training set is N_s, and the size of detector set is N_d. The matching probability between an arbitrary given detector and an antigen is P', which is related with specific matching rules [9, 10]. $P(A)$is defined as the probability of event A happening.

Theorem 1 *The probability of matching an un-described self for an arbitrary detector which passes self-tolerance is* P_d=$(1 - P')^{N_s} \cdot (1 - (1 - P')^{N_{Self}-N_s})$. *The probability of correct identification for an arbitrary non-self is* P_{tp}= 1−$(1 - P')^{N_d \cdot (1-P_d)}$, *and the probability of being misidentified for an arbitrary non-self is* P_{fn}=$(1 - P')^{N_d \cdot (1-P_d)}$. *The probability of correct identification for an arbitrary self is* P_{tn}=$(1 - P')^{N_d \cdot P_d}$, *and the probability of being misidentified for an arbitrary self is* P_{fp}= 1−$(1 - P')^{N_d \cdot P_d}$.

Proof Known from the proposition, the given detector passes self-tolerance, which means that it does not match any self in the training set. Let event A be "the given detector does not match any self in the self set", and event

B be "the given detector at least match one self which is undescribed", then $P_d = P(A)P(B)$. In event A, the number of times that detectors match selves satisfies the binomial distribution $X \sim b(N_s P')$. So, $P\,(A) = P\,(X = 0) = (1 - P')^{N_s}$. In event B, the number of times which detectors match un-described selves satisfies the binomial distribution $Y \sim b(N_{Self} - N_s P)$. So,$P\,(B) = 1 - P\,(Y = 0) = 1 - (1 - P')^{N_{Self} - N_s}$. $P_d = P\,(A)\,P\,(B) = (1 - P')^{N_s} \cdot (1 - (1 - P')^{N_{Self} - N_s})$.

Let event E be "the given non-self at least matches one detector in the detector set". In event E, the number of times which non-selves match detectors satisfies the binomial distribution $Z \sim b(N_d \cdot (1 - P_d)\,P')$. Then,$P_{tp} = P\,(E) = 1 - P\,(Z = 0) = 1 - (1 - P')^{N_d \cdot (1 - P_d)}$ and $P_{fn} = 1 - P_{tp} = (1 - P')^{N_d \cdot (1 - P_d)}$.

Let event F be "the given self does not match any detector in the detector set". In event F, the number of times that selves match detectors satisfies the binomial distribution $W \sim b(N_d \cdot P_d P')$. Then, $P_{tn} = P\,(F) = P\,(W = 0) = (1 - P')^{N_d \cdot P_d}$,$P_{fp} = 1 - P_{tn} = 1 - (1 - P')^{N_d \cdot P_d}$. Proved.

In Theorem 1, P' is the matching probability between any given detector and any antigen, which is the self-reaction rate of a candidate detector. For RNSA and V-Detector, P' is the probability that a candidate detector falls in the self space. For IO-RNSA, P' is the probability that a candidate detector falls in the intersection between its random range and the self space. To simplify the discussion, assume that there are no overlaps between self antigens.

Algorithms of RNSA and V-Detector randomly generate candidate detectors in the n-dimensional [0,1] space, and the self-reaction rate of a candidate detector is the ratio of the self space to the entire shape space.

$$P'_1 = \frac{V_{Self}}{V_S} = \frac{N_s r_s^n \pi^{\frac{n}{2}}}{\tau(\frac{n}{2} + 1)} \tag{11}$$

In IO-RNSA, when the evolution generation is t, candidate detectors which are mutated from an individual d are limited in the region between two hyper-spheres with the individual as the center and $\sqrt{n}$*/exp(t-1)* and $\sqrt{n}$*/exp(t)* as the radiuses. Self antigens may intersect with this region, or not. The self-reaction rate of a candidate detector is the ratio of the intersection space to the space between the two hyper-spheres. Suppose that the number of self antigens which satisfy $\sqrt{n}/exp(t) - r_s < dist(d.y, ag.x) < \sqrt{n}/exp(t-1) + r_s$ is N_z, $N_z < N_s$.

$$\begin{aligned} P'_2 &= \frac{V_{Cross}}{V_t - V_{t+1}} \le \frac{\frac{N_z r_s^n \pi^{\frac{n}{2}}}{\tau(\frac{n}{2}+1)}}{\frac{\left(\frac{\sqrt{n}}{exp(t-1)}\right)^n \pi^{\frac{n}{2}}}{\tau(\frac{n}{2}+1)} - \frac{\left(\frac{\sqrt{n}}{exp(t)}\right)^n \pi^{\frac{n}{2}}}{\tau(\frac{n}{2}+1)}} \\ &= \frac{N_z r_s^n}{\left(\frac{\sqrt{n}}{\exp(t-1)}\right)^n - \left(\frac{\sqrt{n}}{\exp(t)}\right)^n} \end{aligned} \tag{12}$$

To compare the self-reaction rate of a detector for the three algorithms, set $\zeta = P'_2/P'_1$.

$$\zeta = \frac{P'_2}{P'_1} \le \frac{\frac{N_z}{\left(\frac{\sqrt{n}}{exp(t-1)}\right)^n - \left(\frac{\sqrt{n}}{exp(t)}\right)^n}}{\frac{N_s \pi^{\frac{n}{2}}}{\tau(\frac{n}{2}+1)}} = \frac{N_z}{N_s}(\frac{1}{V_t - V_{t+1}}) \tag{13}$$

When the data dimension and the evolution generation are determined, v_t and v_{t+1} can be calculated. If distributions of self set in the shape space are concentrated, N_z is far less than N_s. When the evolution generation is small, $\zeta < 1$, it indicates that the self-reaction rate of IO-RNSA is less than that of RNSA and V-Detector. Known from [4], the number of candidate detectors required for generating N_d mature detectors is $N_c = N_d/(1 - P')^{N_s}$. Therefore, the smaller the self-reaction rate is, the smaller N_c is, that is, the smaller the cost of detectors' generation is.

Set the ratio of the training set size N_s to the self set size N_{Self} is f, $P_d = \left(1 - P'\right)^{N_s} \cdot (1 - (1 - P')^{N_{Self} - N_s})$. Figure 5 is the matlab simulations of Theorem 1. As can be seen, when N_s is large enough, the effect of f and P' on P_d is small. For example, when $N_s = 500$, $P_d < 1\,\%$ and reaches satisfactory values. The false alarm rate $FAR = P_{fp}$ and the detection rate $DR = P_{tp}$ are related to the self-reaction rate of detectors P', the number of mature detectors N_d, the size of training set N_s and the size of self set N_{Self}. As can be seen, the effect of P' on P_{fp} and P_{tp} is small. With the increase of N_s and N_d, P_{fp} also increases. When N_s is small, P_{tp} increases with the increase of N_d. When N_s is large enough, P_{tp} is small and approaching 0.

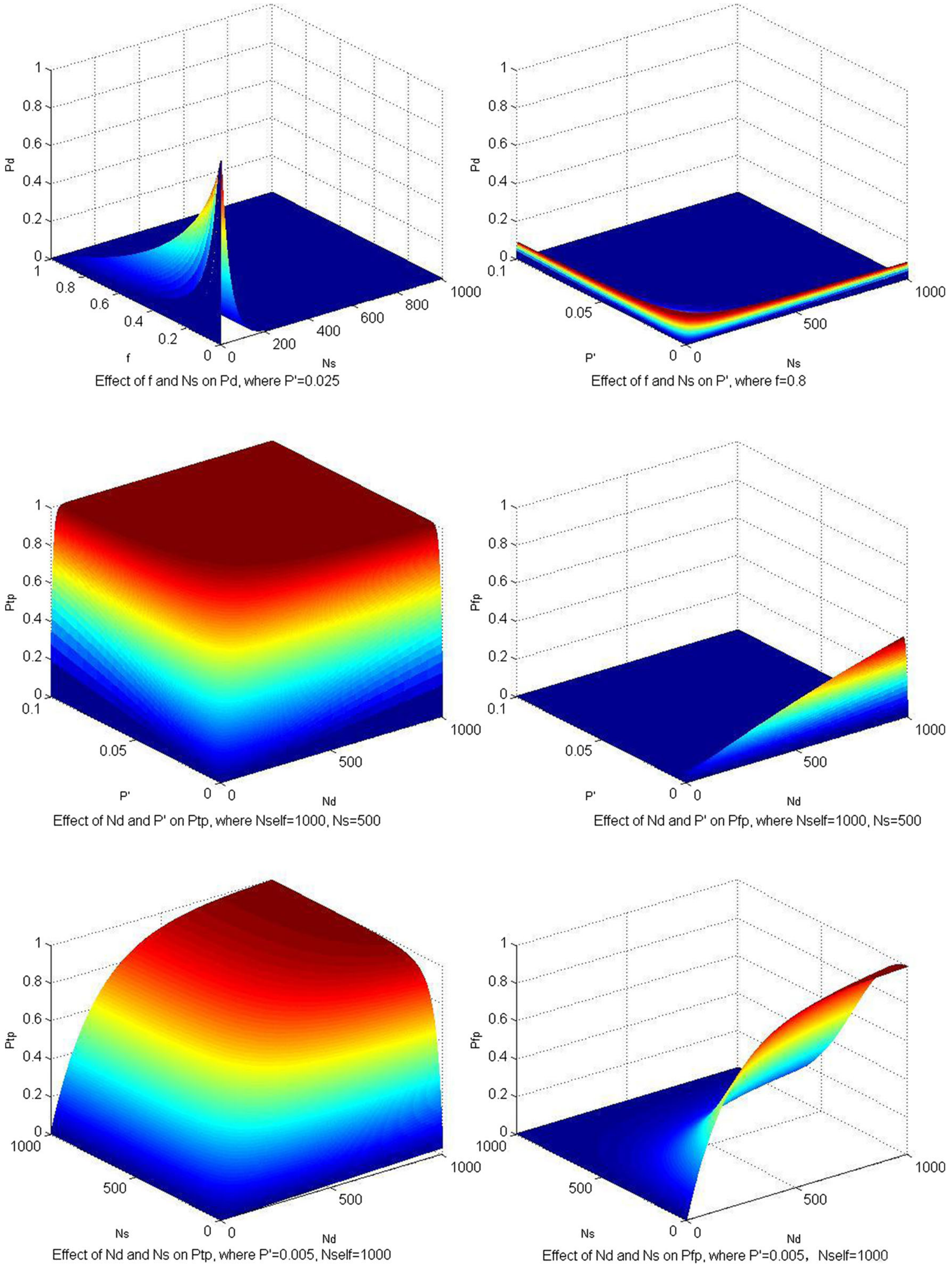
□

3.6 Time complexity analysis

Theorem 2 *The time complexity of IO-RNSA is* $O(\frac{N_d(N_s + c_{max} \cdot n + (1 - P') \cdot N_d)}{1 - P'})$. *Where,* N_d *is the number of mature detectors,* N_s *is the size of self set,* c_{max} *is the maximum of clones, n is the data dimension, and* P' *is the average self-reaction rate of detectors.*

Proof IO-RNSA produces mature detectors through the immune optimization and self-tolerance of generations of evolutional population. The main time costs are focused in step 2, step 4 and step 5.

In step 2, the algorithm performs operations of the immune selection, clonal proliferation and mutation. For a single individual, the number of calculation times for clone operation does not exceed c_{max}, where c_{max} is a determined value and can be negligible, the number of calculation times for mutation operation does not exceed $c_{max} \cdot n$, and the time complexity is O($c_{max} \cdot n$).



Effect of f and Ns on Pd, where P'=0.025

Effect of f and Ns on P', where f=0.8

Effect of Nd and P' on Ptp, where Nself=1000, Ns=500

Effect of Nd and P' on Pfp, where Nself=1000, Ns=500

Effect of Nd and Ns on Ptp, where P'=0.005, Nself=1000

Effect of Nd and Ns on Pfp, where P'=0.005， Nself=1000

Fig. 5 Simulations of Theorem 1

In step 4, the algorithm determines whether a candidate detector falls in the self space, and the time complexity is O(N_s). If it is not covered by selves, perform step 5.

In step 5, the algorithm judges whether a candidate detector is covered by mature detectors, the number of calculation times does not exceed N_d, and the time complexity is O(N_d).

Table 5 Comparisons of time complexities

Algorithms	Time complexities
RNSA	$O\left(\frac{N_d \cdot N_s}{(1-P')^{N_s}}\right)$ [11]
V-Detector	$O\left(\frac{N_d \cdot N_s}{(1-P')^{N_s}}\right)$ [12, 13]
IO-RNSA	$O\left(\frac{N_d(N_s+c_{max} \cdot n+(1-P') \cdot N_d)}{1-P'}\right)$

Suppose that in generation t, the number of mature detectors is N_{dt}, the number of candidate detectors is N_t, the self-reaction rate of candidate detectors is P_t, and then $N_t = N_{dt}/(1\text{-}P_t)$. So, the time complexity of generating mature detectors in generation t is $O(N_t \cdot (N_s + c_{max} \cdot n) + (1\text{-}P_t) \cdot N_t \cdot N_d) = O(N_t \cdot (N_s + c_{max} \cdot n) + N_{dt} \cdot N_d)$. Known from Section 3.3, the number of generations for evolution population is $k = floor(1+log(\sqrt{n}/rs))$. So, the total time complexity of the algorithm is $O(\sum_{t=1}^{k} (N_t \cdot (N_s + c_{max} \cdot n) + N_{dt} \cdot N_d))$.

Let the average self-reaction of candidate detectors is P', the required number of candidate detectors is $N = \sum_{t=1}^{k} N_t = N_d/(1 - P')$, and the number of mature detectors is $N_d = \sum_{t=1}^{k} N_{dt}$. So, the total time complexity of the algorithm is simplified to $O(N \cdot (N_s + c_{max} \cdot n) + N_d^2)) = O(\frac{N_d(N_s+c_{max} \cdot n+(1-P') \cdot N_d)}{1-P'})$. Proved.

RNSA and V-Detector are classical immune algorithms for detectors' generation, which are widely applied to pattern recognition, anomaly detection, immune optimization and other fields. Table 5 lists contrasts of time complexities of the two algorithms and IO-RNSA. From Table 5, the time complexity is exponential to N_s in traditional negative selection algorithms. When the number of selves rises, the time cost will increase fast even to the point of unbearable. The time complexity is linear to the size of self set in IO-RNSA, which greatly lowers the impact on the time cost with the growth of self set. Therefore, IO-RNSA reduces the time complexity and enhances detectors' generation efficiency. □

4 Experiments

This section verifies the effectiveness of IO-RNSA by experiments. The algorithm chooses two types of data sets which are widely used in the study of real-valued negative selection algorithms, including 2-D synthetic data sets [13] and UCI data sets [25]. 2-D synthetic data sets are authoritative for performance testing of real-valued negative selection algorithms [12, 13, 20], which is provided by the research team of professor Dasgupta in the University of Memphis. UCI data sets are classical machine learning datasets, widely used in performance testing and generation efficiency analysis [12–20]. And the algorithm is compared with two classical real-valued negative selection algorithms RNSA and V-Detector.

The experiments used the number of mature detectors *DN*, detection rate *DR*, false alarm rate *FAR* and the time of detectors' generation *DT* to measure the effectiveness of algorithms. Because RNSA adopts the default number of detectors as the termination condition, this paper modified RNSA to use the expected coverage of non-self space, in order to make valid comparisons of the three algorithms in the same experimental conditions.

4.1 2-D synthetic data sets

The data sets contain a plurality of different sub datasets. In general, Ring, Stripe and Pentagram sub datasets are selected to test the performance of detectors' generation for IO-RNSA. Figure 6 shows distributions of the experimental data in 2-dimensional space.

The size of self sets for three datasets is 1000. The training data are points randomly picked up from the self set, and the testing data are points randomly selected from the 2-dimensional space. Experiments were repeated 20 times and average values were taken. Experimental results are shown in Tables 6 and 7, where values in parentheses are variances. Table 6 lists the contrasts of detection rates and false alarm rates in three datasets for the algorithm, with

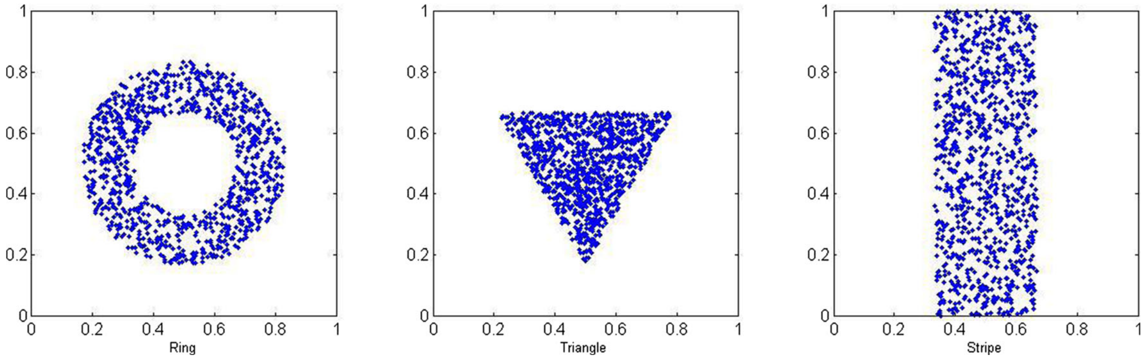


Fig. 6 Distributions of ring, triangle and stripe

Table 6 Effects of different self radius on the algorithm

Datasets	Radius of self r_s =0.02		Radius of self r_s =0.1		Radius of selfr_s =0.2	
	DR%	FAR%	DR%	FAR%	DR%	FAR%
Ring	91.56 (1.13)	31.70 (1.87)	81.74 (1.82)	10.22 (2.28)	45.44 (2.75)	0.00 (0.00)
Triangle	92.33 (1.02)	28.42 (1.41)	85.45 (1.77)	10.65 (2.10)	48.72 (2.12)	0.00 (0.00)
Stripe	93.02 (1.06)	30.15 (1.32)	88.61 (1.67)	9.43 (1.92)	49.71 (2.15)	0.00 (0.00)

Table 7 Effects of different sizes of self training set on the algorithm

Datasets	Size of training set N_s =100		Size of training set N_s =500		Size of training set N_s =800	
	DR%	FAR%	DR%	FAR%	DR%	FAR%
Ring	21.31 (3.23)	72.12 (2.55)	92.11 (1.93)	9.13 (1.86)	97.43 (1.81)	0.00 (0.00)
Triangle	24.49 (2.88)	65.58 (2.43)	94.35 (1.41)	9.01 (1.79)	98.26 (1.25)	0.00 (0.00)
Stripe	27.32 (2.76)	68.24 (2.01)	94.78 (1.53)	8.42 (1.72)	98.36 (1.27)	0.00 (0.00)

Table 8 Experimental parameters

Datasets	Records	Attributes	Type	Self set	Non-self set	Training set	Testing set
Iris	150	4	Real	Setosa: 50	Versicolour: 50 Virginica: 50	Setosa: 25 Versicolour: 25 Virginica: 25	Setosa: 25
Breast Cancer Wisconsin Diagnostic	569	30	Real	Benign: 444	Malignant: 239	Benign: 250	Benign: 150 Malignant: 150
Abalone	4177	8	Real, Integer	M: 1528	F: 1307 I: 1342	M: 1000	M: 500 F: 500 I: 500

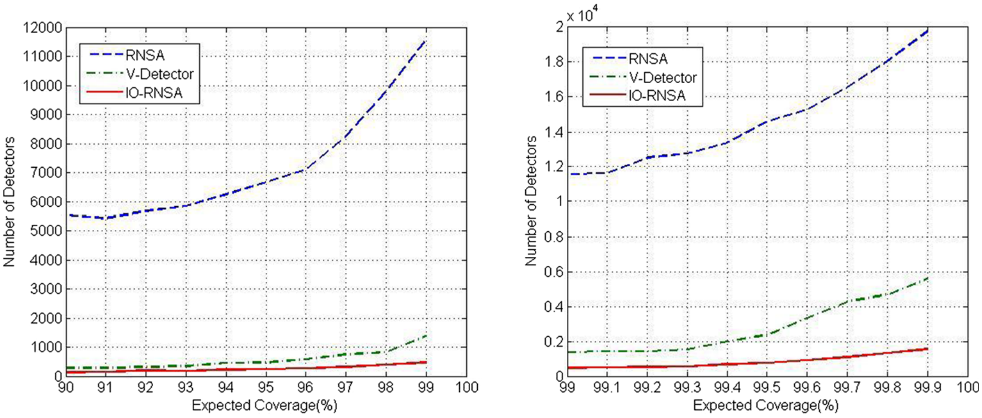


Fig. 7 Comparisons of the number of detectors for RNSA, V-Detector and IO-RNSA

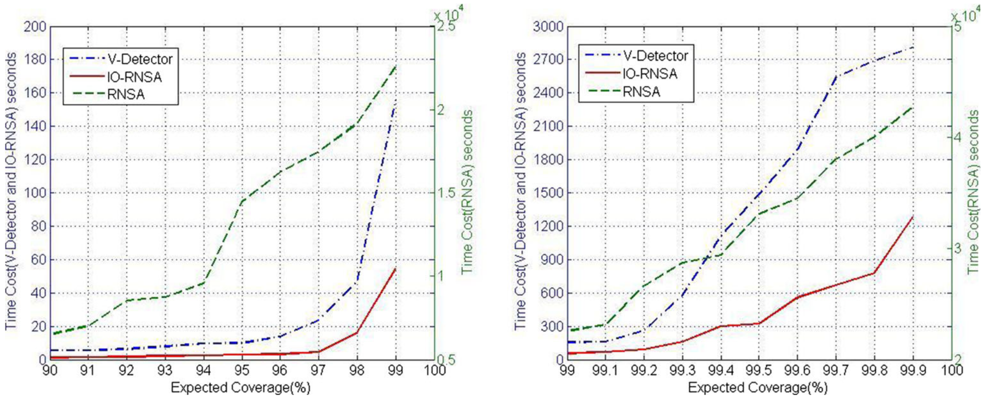


Fig. 8 Comparisons of costs of detectors' generation for RNSA, V-Detector and IO-RNSA

the same expected coverage of 90 %, the same training set size N_s =400 and different self radiuses. As can be seen, detection rate and false alarm rate are higher under small self size, while they are lower under large self size. Table 7 lists the contrasts of detection rates and false alarm rates in three datasets for the algorithm, with the same expected coverage of 90 %, the same self radius r_s =0.05 and different training set sizes. With the increase of the training set size, the detection rate gradually raises and the false alarm rate gradually decreases.

4.2 UCI data sets

Three UCI datasets are selected for the experiments, namely Iris, Abalone and Breast Cancer Wisconsin Diagnostic. Experimental parameters are shown in Table 8, where individuals of self set, non-self set, self training set and testing set are chosen randomly. The experiments were repeated 20 times and average results were taken. In this section the algorithm compared with RNSA and V-Detector from these aspects including the number of detectors, the time cost, the detection rate and the false alarm rate.

4.2.1 Comparisons of number of detectors

Figure 7 shows the comparisons of the number of mature detectors on Iris dataset for RNSA, V-Detector and IO-RNSA. Seen from Fig. 7, with the rise of the expected coverage, the number of required detectors generated by the three algorithms increases correspondingly, but the efficiency of IO-RNSA is better than RNSA and V-Detector. On the Iris dataset, in order to achieve the expected coverage 99 %, RNSA needs 11568 mature detectors, V-Detector needs 1371, and IO-RNSA needs 464 which is significantly reduced and has decrease of 96.0 % and 66.2 %. IO-RNSA generates detectors from far to near, improves the coverage of detectors, and achieves fewer detectors covering as much non-self space as possible. So, under the same expected coverage, IO-RNSA requires fewer detectors.

4.2.2 Comparisons of costs of detectors' generation

Figure 8 shows the comparisons of costs of detectors' generation on Breast Cancer Wisconsin Diagnostic dataset for RNSA, V-Detector and IO-RNSA. Seen from Fig. 8, with the increase of expected coverage, costs of detectors' generation for RNSA and V-Detector sharp rise, and cost for IO-RNSA grows slowly. When the expected coverage is 99 %, the time cost of detectors' generation for RNSA is 22560.4 seconds, the time cost for V-Detector is 155.8 seconds, and the time cost for IO-RNSA is 54.4 seconds which has decrease of 99.8 % and 65.1 %. By the analysis of Section 3.5, while the distribution of self set is concentrated, the reaction rate of detectors for IO-RNSA is smaller. So,

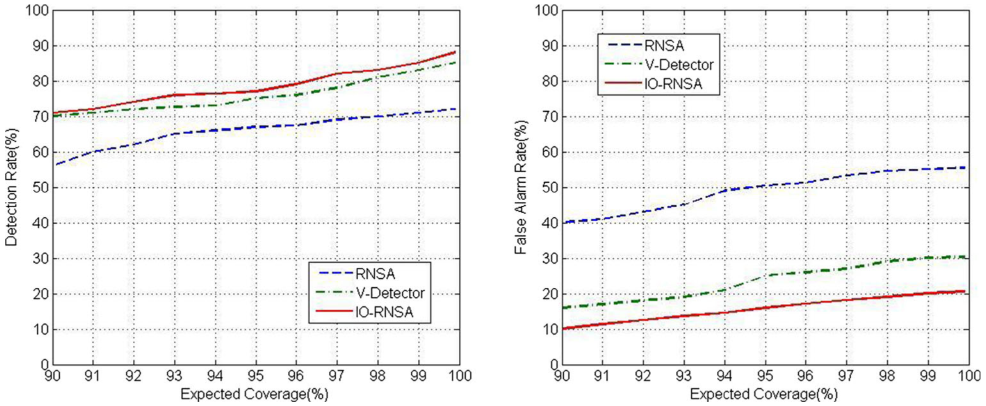


Fig. 9 Comparisons of detection rates and false alarm rates for RNSA, V-Detector and IO-RNSA

the required number of candidate detectors for IO-RNSA is less, which greatly reduces the time cost of self tolerance.

4.2.3 Comparisons of detection rate and false alarm rate

To further verify the effectiveness of IO-RNSA, detection rates and false alarm rates for RNSA, V-Detector and IO-RNSA are contrasted in this section. Abalone dataset is adopted, and experimental results are shown in Fig. 9. As can be seen, under the same expected coverage, IO-RNSA has significantly improved detection rate and lowered false alarm rate compared with RNSA and V-Detector.

5 Conclusions

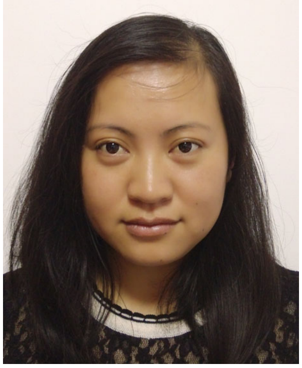
High time complexities, large number of detectors and redundant coverage are major problems in traditional negative selection algorithms, which limit applications of immune algorithms. An immune optimization based real-valued negative selection algorithm IO-RNSA is proposed. Based on the distribution of self set in morphological space, the algorithm introduces the immune optimization mechanism, and produces candidate detectors hierarchically. In the process of detectors' generation, the algorithm limits the random generation range of candidate detectors. It gives priority to producing detectors with large size which are distributed in low coverage areas, decreasing the number of mature detectors and redundancies Then it produces detectors with small size which are distributed in the area close to the self space, reducing the number of vulnerabilities. Theoretical analysis and experimental results show that IO-RNSA has better time efficiency and generation quality than classical negative selection algorithms, and is an effective artificial immune algorithm for generating detectors. The next step is to continue studying the immune mechanism of NSA, and propose more efficient negative selection algorithm for dynamic self set.

Acknowledgments This work has been supported by the National Natural Science Foundation of China under Grant No. 61173159, the National Natural Science Foundation of China under Grant No. 60873246, and the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China under Grant No. 708075.

References

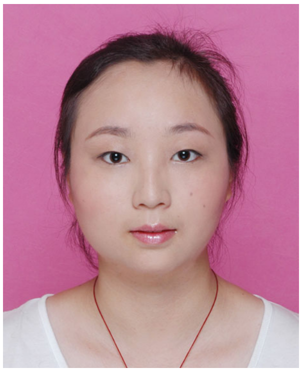
1. Jin ZZ, Liao MH, Xiao G (2013) Survey of negative selection algorithms. J Commun 34(1):159–170
2. Dasgupta D, Yu S, Nino F (2011) Recent advances in artificial immune systems–models and applications. Appl Soft Comput 11:1574–1587
3. Stibor T, Timmis J, Eckert C (2005) On the appropriateness of negative selection defined over hamming shape-space as a network intrusion detection system. In: Proceedings of IEEE evolutionary computation. IEEE Computer Society Press, Edinburgh, pp 995–1002
4. Timmis J, Hone A, Stibor T, Clark E (2008) Theoretical advances in artificial immune systems. Theor Comput Sci 403:11–32
5. Bretscher P, Cohn M (1970) A theory of self-nonself discrimination. Science 169:1042–1049
6. D'Haeseleer P, Forrest S, Helman P (1996). Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy, Washington, pp 110–120
7. Sobh TS, Mostafa WM (2011) A cooperative immunological approach for detecting network anomaly. Applied Soft Computing 11:1275–1283
8. Dasgupta D, Gonzalez F (2002) An immunity-based technique to characterize intrusions in computer networks. IEEE Trans Evol Comput 6(3):281–294
9. Forrest S, Perelson AS, Allen L, Cherukuri R (1994) Self–nonself discrimination in a computer. In: Proceeding of the IEEE Symposium on Research in Security and Privacy. Oakland: IEEE Computer Society Press, pp 202–212
10. Balthrop J, Esponda F, Forrest S et al (2002) Coverage and generalization in an artificial immune system. GECCO 2002. Morgan Kaufmann Publishers Inc, New York, pp 3–10
11. Gonzalez F, Dasgupta D (2003) Anomaly detection using real-valued negative selection. Genet Program Evolvable Mach 4:383–403
12. Zhou J (2006) Negative selection algorithms: from the thymus to V-detector. Ph. D dissertation, University of Memphis, Memphis, TN, USA
13. Zhou J, Dasgupta D (2009) V-detector: an efficient negative selection algorithm with "probably adequate" detector coverage. Inf Sci 19(9):1390–1406
14. Joseph M, Shapir O, Gary B (2005) An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection[A]. GECCO 2005[C]. Washington DC, USA, pp 337–344
15. Ostaszewski M, Seredynski F, Bouvry P (2006) Immune anomaly detection enhanced with evolutionary paradigms. In: 8th annual conference on genetic and evolutionary computation (GECCO 2006), Seattle, Washington, USA
16. Zhang XM, Yi ZX, Song JS et al (2010) Research on negative selection algorithm based on matrix representation. J Electron Inf Technol 32(11):2701–2706
17. Gao XZ, Ovaska SJ, Wang X (2006) Genetic algorithms-based detector generation in negative selection algorithm. In: 2006 IEEE mountain workshop on adaptive and learning systems
18. Yang DY, Chen JY (2009) Research on detector generation algorithm based on multiple populations GA. Acta Automatica Sinica 35(4):425–432
19. Stibor T (2008) An empirical study of self/non-self discrimination in binary data with a kernel estimator. In: 7th international conference on artificial immune systems, Phuket, Thailand
20. Chen W, Liu XJ, Li T et al (2011) A negative selection algorithm based on hierarchical clustering of self set and its application in anomaly detection. Int J Comput Intell Syst 4(4):410–419
21. Stibor T, Philipp M, Jonathan T (2005) Is negative selection appropriate for anomaly detection? In: Proceedings of IEEE Evolutionary Computation. IEEE Computer Society Press, Edinburgh, pp 569–576
22. Caldas B, Pita M, Buarque F (2007) How to obtain appropriate executive decisions using artificial immunologic systems. In: 6th international conference on artificial immune systems, Santos, Brazil

23. Ma W, Tran D, Sharma D (2008) Negative selection with antigen feedback in intrusion detection. In: 7th international conference on Artificial Immune Systems, Phuket, Thailand
24. Ou CM (2012) Host-based intrusion detection systems adapted from agent-based artificial immune systems. Neuro Comput 88:78–86
25. UCI Dataset. http://archive.ics.uci.edu/ml/datasets
26. de Castro LN, Timmis J (2002) An artificial immune network for multimodal function optimization. In: IEEE world congress on evolutionary computation, pp 699–704
27. de Castro LN, Fernando J (2002) Learning and Optimization Using the Clonal Selection Principle. IEEE transactions on evolutionary computation. Special Issue on Artificial Immune Systems 6(3):239–251
28. Cai T, Ju SG, Zhong W (2009) A cutting based detector generating and matching algorithm. Acta Electronica Sinica 7(B04):131–134
29. Lasisi A, Ghazali R, Herawan T (2014) Negative selection algorithm: a survey on the epistemology of generating detectors. Lect Notes Electr Eng 285:167–176



Xin Xiao is reading a doctorate of Sichuan University, China. She received her Bachelors degree in Computer Science from Sichuan University in 2004 and her Masters degree in Computer Science from Sichuan University in 2009. Her research interests are in network security, intrusion detection systems, artificial immune systems, and wireless sensor networks.

Tao Li received the Ph.D. degree in Computer Science from University of Electronic Science and Technology of China in 1994, and now is the professor of school of computer science in Sichuan University, China. His research interests are in network security, artificial immune systems, and cloud computing and cloud storage.



Ruirui Zhang received the Ph.D. degree in Computer Science from Sichuan University of China in 2012. Her research interests include network security, intrusion detection systems, and artificial immune systems.