Analysis and Mathematical Justification of a Fitness Function used in an Intrusion Detection System

Pedro A. Diaz-Gomez*
Ingenieria de Sistemas
Universidad El Bosque
Bogota, Colombia
pdiazg@ou.edu

Dean F. Hougen
Robotics, Evolution, Adaptation, and Learning Laboratory (REAL Lab)
School of Computer Science
University of Oklahoma
Norman, OK, USA
hougen@ou.edu

ABSTRACT

Convergence to correct solutions in Genetic Algorithms depends largely on the fitness function. A fitness function that captures all goals and constraints can be difficult to find. This paper gives a mathematical justification for a fitness function that has previously been demonstrated experimentally to be effective.

Categories and Subject Descriptors: I.2.6 [Artificial Intelligence]: Learning, D.4.6 [Operating Systems]: Security and Protection.
General Terms: Algorithms, Security.
Keywords: Genetic Algorithms, Fitness Function, Intrusion Detection Systems.

1. INTRODUCTION

This paper focuses on an off-line intrusion detection system known as *GASSATA* [5] that uses a Genetic Algorithm (GA) to search for matches in the audit trail. Unfortunately the parameters for the fitness function cannot be tuned to effectively detect all possible attacks found in an audit trail while still avoiding false positives (warnings of attacks that do not exist) and false negatives (failing to detect real intrusions). Here we mathematically justify an improved fitness function independent of parameters.

2. *GASSATA* & INTRUSION DETECTION

GASSATA [5] is a tool for security audit trail analysis that performs misuse detection by comparing the user's behavior (OV vector) to a matrix of known attacks (AE). *GASSATA* explains the audit trail data by hypothesizing one or more attacks (I vector), and uses a heuristic method—GAs—because explaining the data is an NP-Complete problem. The fitness function used in *GASSATA* is:

$$F(I) = \alpha + \sum_{i=1}^{N_a} W_i * I_i - \beta * T^2$$

where α maintains $F(I) > 0$, N_a is the number of known attack types, W is a one-dimensional array of length N_a called the *weighted vector* that gives the risk of each attack, I is a one dimensional vector of length N_a called the *hypothesis vector* ($I_i = 1$ if attack i is present, $I_i = 0$ otherwise), and β slopes the penalty function T^2.

To explain the audit trail data by the occurrence of one or more attacks, *GASSATA* attempts to find an I that maximizes $W \cdot I$. To evaluate the constraint, the algorithm counts the number of events of each type generated by all the attacks hypothesized in I. If these numbers are less than or equal to the number of events OV, then the hypothesis is realistic but if some of those numbers are greater than the number of events that occurred, then the hypothesis is penalized, i.e., if $(AE \cdot I)_i > OV_i$, a failure is added [5].

Good results with *GASSATA* have been reported [5] but we had poor results when we attempted to duplicate it [1]. We set various values to parameters α, W and β, however we always obtain many false positives and some false negatives. As an example, we use the fitness function $F(I) = 4.0 + \sum_{i=1}^{N_a} I_i - (1/20) * T^2$ and obtain 217% false positives and 20% false negatives—100% corresponds to 4 actual intrusions. Further experimental results led us to propose a new fitness function [2].

3. NEW FITNESS FUNCTION PROPOSED

The term $\sum_{i=1}^{N_a} I_i$ was incorrectly guiding the GA and the term T^2 was excessively penalizing false positives [2]. So the solution proposed removes $\sum_{i=1}^{N_a} I_i$ and it uses as a penalty function T^1, taking into account that if two intrusions relate to the same event with an overestimate of the number of events hypothesized then they should be counted them twice, and so forth. Call this T'.

With this in mind and the experience gained with testing, the fitness function proposed only has the penalty function. As the number of events is N_e, the new fitness function suggested is $F(I) = N_e - T'$. This is paired with a combination method based on a bitwise logical 'or' for alleles at each locus [2]. With this system *there are no false positives* and the number of *false negatives decreases dramatically.* This time 70 runs were performed with different data (some data was downloaded from the Lincoln Laboratory [3]) and only one time a false negative was present.

While the fitness function suggested here certainly appears to function well, given the empirical data we have

*Conducting research at the Robotics, Evolution, Adaptation, and Learning Laboratory (REAL Lab), School of Computer Science, University of Oklahoma.

GECCO'05, June 25–29, 2005, Washington, DC, USA
ACM 1-59593-010-8/05/0006.

obtained, the question remains as to whether this fitness function can be justified mathematically. The following section shows that it can be.

4. MATHEMATICAL JUSTIFICATION

The problem is to find the vector I that maximizes

$$F(I) = W \cdot I \tag{1}$$

subject to

$$(AE \cdot I)_i \leq OV_i \tag{2}$$

and $$I_i \in \{0,1\} \text{ for } 0 \leq i \leq N_a$$

We can abbreviate the constraints given in Equation 2 using polynomials c_j in I and $a_{jk} \in AE$, following the notation suggested by Ham [4] as

$$c_j(I) = a_{j1}I_1 + a_{j2}I_2 + \cdots + a_{jn}I_n - OV_j \text{ for } 0 \leq j \leq N_e \tag{3}$$

and we can join the objective function in Equation 1 with constraint in Equation 2 to obtain the *energy function* [4]

$$E(I,K) = W \cdot I + K \sum_{j=1}^{N_e} \Phi[c_j(I)] \tag{4}$$

where the positive parameter K controls how well the unconstrained optimization problem in 4 to 6 approximates the original *Linear Problem* in 1 to 2

$$\Phi[c_j(I)] = \begin{cases} = 0 & \text{if } c_j(I) \leq 0 \\ > 0 & \text{if } c_j(I) > 0 \end{cases} \tag{5}$$

$$I_i \in \{0,1\} \text{ for } 0 \leq i \leq N_a \tag{6}$$

and as we want to maximize E, $\Phi(t)$ must be differentiable with the property in Equation 5.

For simplicity, the function $\Phi(t)$ commonly selected is [4]

$$\Phi(t) = \begin{cases} = 0 & \text{for } t \leq 0 \\ = \frac{1}{2}\, t^2 & \text{for } t > 0 \end{cases} \tag{7}$$

and now finding the partial derivative of E in Equation 4 with respect to I we obtain

$$\frac{\partial E(I,K)}{\partial I} = W + K \sum_{j=1}^{N_e} \Psi[c_j(I)] \frac{\partial}{\partial I}[c_j(I)] \tag{8}$$

where $\Psi(v) = \frac{d\Phi(v)}{dv} = \Phi'(v) = v$.

Using Equation 3 to find $\frac{\partial}{\partial I}[c_j(I)]$, Equation 8 gives [4]

$$\frac{\partial E(I,K)}{\partial I} = W + K \sum_{j=1}^{N_e} \Psi[c_j(I)][a_{j1}\ a_{j2}\ \cdots a_{jn}]^T \tag{9}$$

and as $\Psi[c_j(I)] = c_j(I)$, substituting in Equation 9 and equating to 0—we are finding a maximum—we obtain

$$\frac{\partial E(I,K)}{\partial I} = W + K \sum_{j=1}^{N_e} c_j(I) * [a_{j1}\ a_{j2}\ \cdots a_{jn}]^T = 0. \tag{10}$$

Then using Equation 3 again we obtain

$$\begin{aligned} &\sum_{j=1}^{N_e} (a_{j1}I_1 + a_{j2}I_2 + \cdots a_{jn}I_n) * [a_{j1}\ a_{j2}\ \cdots a_{jn}]^T \\ &= -\tfrac{W}{K} + \sum_{j=1}^{N_e} OV_j * [a_{j1}\ a_{j2}\ \cdots a_{jn}]^T \end{aligned} \tag{11}$$

Taking vector components in Equation 11 we get for $1 \leq i \leq n = N_a$

$$\begin{aligned} &\sum_{j=1}^{N_e} (a_{j1}I_1 + a_{j2}I_2 + \cdots a_{jn}I_n) * a_{ji} \\ &= -\tfrac{W_i}{K} + \sum_{j=1}^{N_e} OV_j * a_{ji} \end{aligned} \tag{12}$$

that can be written, having in mind that a_{ji} corresponds to AE_{ji}, as

$$\sum_{j=1}^{N_e} (AE \cdot I)_j * a_{ji} = -\frac{W_i}{K} + \sum_{j=1}^{N_e} OV_j * a_{ji} \tag{13}$$

As W, the weighted vector, is such that $\forall i\ W_i \geq 0$ and K is a parameter that approaches positive infinity [4] then Equation 14 must be satisfied by a maximum I of Equation 4.

$$(AE \cdot I) \cdot \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{N_e i} \end{bmatrix} \leq OV \cdot \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{N_e i} \end{bmatrix} \tag{14}$$

And what we actually did, with the fitness function proposed, was to find I such that

$$(AE \cdot I)_j \leq OV_j, \quad \text{for } 1 \leq j \leq N_a \tag{15}$$

that clearly satisfies Equation 14, because

$a_{ji} \geq 0$, for $1 \leq j \leq N_e$, $1 \leq i \leq N_a$, with $a_{ji} \in AE$;
$I_i \in \{0,1\}$, for $1 \leq i \leq N_a$, with $I_i \in I$; and
$OV_j \geq 0$, for $1 \leq j \leq N_e$, with $OV_j \in OV$.

Thus, the fitness function we have proposed is justified mathematically.

5. CONCLUSIONS & FUTURE WORK

This paper shows some difficulties in providing accurate values to parameters in the fitness function suggested in *GASSATA* [5] and justifies a solution independent of variable parameters, making the fitness function to solve this problem quite general and independent of the audit trail data. This paper used, in part, methodology used in the *neural networks* field [4] for *linear programming with inequality constraints*.

6. REFERENCES

[1] P. Diaz-Gomez and D. Hougen. Improved off-line intrusion detection using a genetic algorithm. To appear in *Proceedings of the Seventh International Conference on Enterprise Information Systems*, 2005.

[2] P. A. Diaz-Gomez and D. F. Hougen. Analysis of an off-line intrusion detection system: A case study in multi-objective genetic algorithms. To appear in *the Florida Artificial Intelligence Research Society Conference.*, 2005.

[3] D. Fried and M. Zissman. Intrusion detection evaluation. Technical report, Lincoln Laboratory, MIT, 1998. http://www.ll.mit.edu/IST/ideval/, accessed March 2004.

[4] F. M. Ham and I. Kostanic. *Principles of Neurocomputing for Science & Engineering*. McGraw Hill, 2001.

[5] L. Mé. GASSATA, a genetic algorithm as an alternative tool for security audit trail analysis. In *First International Workshop on the Recent Advances in Intrusion Detection*, Belgium, 1998.