Internal vs. External Parameters in Fitness Functions

Pedro A. Diaz-Gomez

Computing & Technology Department
Cameron University
Lawton, Oklahoma 73505, USA
pdiaz-go@cameron.edu

Dean F. Hougen

School of Computer Science
University of Oklahoma
Norman, Oklahoma 73019, USA
hougen@ou.edu

Abstract

A fitness function is needed for a Genetic Algorithm (GA) to work, and it appears natural that the combination of objectives and constraints into a single scalar function using arithmetic operations is appropriate. One problem with this approach, however, is that accurate scalar information must be provided on the range of objectives and constraints, to avoid one of them from dominating the other. One possible solution, then, is to try to join the objectives with the constraints with internal parameters, i.e., information that belongs to the problem itself, thereby avoiding external tuning. The building of the fitness function is so complex that, using internal or external parameters, any optimal point obtained will be a function of the coefficients used to combine objectives and constraints. However, it is possible that using internal parameters will increase performance compare to external ones.

1. Introduction

A fitness function is needed to construct a GA—the combination of objectives and constraints in a single function using arithmetic operators seems to be an appropriate way to define it. There are, however, problems with this approach. The first is that accurate scalar information must be provided on the range of objectives, to avoid some of them dominating the others. The second is the difficulty in determining appropriate weights when there is not enough information about them. In this case, any optimal point obtained will be a function of the coefficients used to combine the objectives and constraints [1].

As an approximation to avoid the tuning of external parameters for some linear fitness functions, this paper uses internal parameters, i.e., parameters that belong to the problem itself. There is no external tuning required, just the fitness function that must be logically constructed. The general hypothesis of this paper is that a GA using a fitness function F_i with internal parameters is going to have better performance than the same GA using a fitness function F_e with external parameters. What is called external parameters in this paper are for example the terms α, β and p as in Equation 2, that need to be tuned before the algorithm runs. This is different than when the algorithm runs with Equation 1 which has information from the problem itself and no external parameters to be tuned.

In order to perform a specific test of the general hypothesis, the misuse detection problem is tested with F_i Equation 1 and F_e Equation 2. The misuse detection is an off-line detection system, where the profile activity of the user (OV vector as in equation 1), and the misuse to be detected (AE matrix as in Equation 1) are known in advance. The goal of the GA is to find the best chromosome, I that holds the corresponding misuse, where a 1 in position k of I means misuse k present, with $1 \leq k \leq N_a$ and N_a the number of attacks [8].

F_i the fitness function that uses information from the problem itself is:

$$F_i(I) = \frac{\sum_{j=1}^{N_e}(AE * I)_j - \sum_{j=1}^{N_e} max[0, (AE * I)_j - OV_j]}{\sum_{j=1}^{N_e}(AE * I)_j}. \tag{1}$$

with N_e the number of events, AE the misuse matrix, I the chromosome, OV the observed activity, the term $\sum_{j=1}^{N_e} max[0, (AE * I)_j - OV_j]$ corresponds to the aggregation of faults, and $\sum_{j=1}^{N_e}(AE * I)_j \neq 0$ for $1 \leq j \leq N_e$ (see more details in [3]).

F_e the fitness function that has external parameters is [7, 8]:

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

with $\alpha > 0$, $\beta > 0$ and $p \geq 1$ external parameters to be tuned, N_a is the number of attacks, T the counting of failures as described in [8], and $W_i = 1$ for all $1 \leq i \leq N_a$ [8].

Parameters		*Number*	
α	β	*False +*	*False -*
50.0	1.0	6	0
50.0	7.0	3	7
392.0	1.0	5	0
4.0	0.05	296	1
50.0	0.5	32	0

Table 1: Difficulty in tuning of external parameters as in Equation 2. 30 runs per parameter setting.

Fitness functions as in Equations 1 and 2 have been tested in various projects (see [3, 5, 8]) for finding intrusions in audit trail files .

2. External parameters

This section is going to outline the difficulties of using a fitness function with external parameters as in Equation 2 for joining the objective (finding of misuse I) and the constraint (no more than the activity recorded OV must be guessed, i.e., $(AE * I)_j \leq OV_j$ for all $1 \leq j \leq N_e$ [7, 8]). α, β and p must be chosen. For simplicity, in this paper we choose $p = 2$ constant as suggested by [8], so we need to search for values for α and β.

There is an infinite number of possibilities for $\alpha > 0$ and $\beta > 0$. We can begin with the values suggested by [7] with $\alpha = 50.0$ and $\beta = 1.0$). Table 1 shows in the first row the result. There were 6 false positives and no false negatives. So as the algorithm gives false positives we can increase the value of β, the factor that is influencing the penalty term. Lets say that we are going to use $\alpha = 50.0$, as before, and $\beta = 7.0$ (increased). Then, the number of false positives decreases (from 6 to 3), but the number of false negatives increases (from 0 to 7, see Table 1 second entry). We need to decrease the β factor and to increase the α factor in order to compare the effect. Lets choose $\alpha = 392.0$ and $\beta = 1.0$. The algorithm performs better this time (see Table 1 third entry), but we still have the problem of false positives. We can continue trying, and report the best set of parameters that maybe were tuned just for a specific input (OV vector of activity). How can we reconcile this external tuning? Section 3 will address the use of internal parameters in order to correct this.

3. Internal Parameters

Different than Equation 2, Equation 1 uses only information that belongs to the problem itself. It has no external parameters and it can be rewritten in a more general form as

$$F_i(I) = 1 - \left(\frac{1}{\sum_{j=1}^{N_e} (AE * I)_j} \right) \sum_{j=1}^{N_e} max[0, (AE*I)_j - OV_j] \tag{3}$$

and though of $\alpha = 1$ and $\beta = 1/\sum_{j=1}^{N_e} (AE * I)_j$. β is an *adaptive* parameter that depends on I, and $Penalty = \sum_{j=1}^{N_e} max[0, (AE * I)_j - OV_j]$ which takes into account the violation of the constraint in a *dynamic* way, i.e., it does not count the number of violations, but instead it finds, in a finer way, the positive difference of each $(AE * I)_j$ from OV_j and adds them up to get the net penalty for that particular I. For example if the target is the analysis of an observed vector that contains the count of a user's activity performed in a computer, then a chromosome of the population might incorrectly hypothesize with respect to one category of the observed vector. There is a fault in this particular case. However, how far (or near) was that fault? If the fault corresponds, for example, only to an entry in the observed vector OV which value is 0 and the chromosome I is hypothesizing for that entry 299, then the distance is 299 because $\sum_{j=1}^{N_e} max[0, (AE * I)_j - OV_j] = 299$. On the contrary, if the hypothesis were a count of 1 (for the same specific entry), then the distance is 1 because $\sum_{j=1}^{N_e} max[0, (AE * I)_j - OV_j] = 1$. This is the type of internal tuning that the algorithm performs. In the first case the chromosome is penalized with 299 and in the second case it is penalized with 1.

4. Parameter Setting and Method

The GA parameters used were: population size $N = 50$, chromosome length $l = 24$, 2-tournament selection ($75\% - 25\%$ which means that two chromosomes are selected randomly and with a probability of 75% the best is chosen, and with a probability of 25% the worst is chosen), probability of 1-point crossover $p_c = 0.6$, probability of mutation $p_m = 0.0083$ per bit, stop criterion is $1,000$ generations. Besides that, [7] utilizes a threshold D in order to differentiate an intrusion or a non-intrusion. For example, in the final population, each locus of the entire population is analyzed: if the number of $1s$ in column k is greater than D, then there is a possible intrusion. This is different from what is used in this paper in the sense that the algorithm is storing the best solution so far and, if a new intrusion k is found, it is checked to see if it is already in the current solution [3]. If not, then it is checked to see if it violates the constraint in union with the previous ones in order to be added to the current solution. If when in union with the previous ones it violates the constraint, then it is marked as an exclusive intrusion (see more details in [3]).

Parameter Justification: This set of parameters was used by [7], except the 2-tournament selection pressure ($75\% - 25\%$), which is not defined in [7], but that we used in previous tests with good results [3, 5].

In order to compare the effectiveness of internal vs. external parameters, 5 different sets of external parameters (α, β) for fitness function 2 where used:[1] $(392.0, 1.0)$, $(50.0, 7.0)$, $(50.0, 1.0)$, $(50.0, 0.5)$ and $(4.0, 0.05)$.[2] The GA was run 30 times for each set of parameters finding the average number of false positives and negatives in the 30 runs, in order to obtain the best set of parameters in terms of the Hansen and Kuipers' score.

Equation 2 was compared with Equation 1 using the best set of external parameters (α, β) found in previous steps for Equation 2 according to the Hansen and Kuipers' score. We Ran the GA 30 times with the fitness function as in Equation 2, and 30 times with Equation 1, using 25 different scenarios (from no intrusion to a full vector of intrusions). Each scenario is tested 30 times, so the average number of false positives and false negatives is recorded for each scenario, in order to find the Hanssen and Kuipers' score.

The Hanssen and Kuipers' score was used because it is considered a good predictor. It has the peculiarity that it takes into account not only the false positives (B) and false negatives (C), but the correct positives (A) and correct negatives (D). The formula used for the Hanssen and Kuipers' score is $Skill = A/(A+C) - B/(B+D)$ which tell us how well the algorithm, with the corresponding fitness function, separates the "intrusion" from the "non-intrusion". The range of the Hanssen and Kuipers' score is $[-1, 1]$, being 1 perfect score, and 0 an indication of no skill.[3] Once the average of the Hanssen and Kuipers' scores for each Equation is found they are compared statistically using the Z score, in order to check for the 95% confidence level.

Data Analysis

First the Hanssen and Kuipers' scores for Equations 2 and 1 will be calculated for each scenario. After that, the corresponding average of the Hanssen and Kuipers' scores will be calculated for each Equation, and they will be compared looking at the Z score, in order to check if both are statistically significantly different at the 95% confidence level. In order to corroborate the accuracy of the Hanssen and Kuipers' averages, the bootstrapping technique was applied taking $1,000$ samples (samples from 25 Hanssen and Kuipers' means respectively).

In order to see if there is a statistically significant difference between the Hanssen and Kuipers' scores obtained with each Equation, and the union of both, a combined bootstrapping was applied using both sets (i.e., they form a set of 50 combined Hanssen and Kuipers' scores) and the corresponding comparison was made, in order to check if they are statistically different at the 95% confidence level.

[1] Other integer values for $1 < \beta < 7$ with $\alpha = 50.0$ were tested until the average of false negatives were 0.0 No one of those gives as result 0.0 for averages of false positive and false negatives. Less than 30 runs were done until the 0.0 average of false negatives appeared.

[2] [7] used $\alpha = 50$ and $\beta = 1.0$, so we begin from there to test if a better set of parameters could be obtained,as it happened with parameters $\alpha = 392.0$ and $\beta = 1.0$.

[3] Thanks to Dr. Mike Richman for suggesting this score.

Parameters		*Average*		*H.&K.*
α	β	*False +*	*False -*	*Score*
392.0	1.0	0.16	0.00	0.988
50.0	7.0	0.10	0.23	0.971
50.0	1.0	0.20	0.00	0.985
50.0	0.5	1.06	0.00	0.918
4.0	0.05	9.86	0.03	0.239

Table 2: Average of false positives, false negatives, and the corresponding Hanssen & Kuipers' Score given by Equation 2, using different setting for external parameters α and β. 30 runs per parameter setting.

5. Experimental Results

As addressed in Section 4, first using 5 different sets of (α, β), 30 times the GA with the fitness function as in Equation 2 was executed, with the corresponding set of GA parameters proposed, recording the average number of false positives and false negatives and the Hanssen and Kuipers' score as is shown in Table 2.[4]

A new test set was performed for 30 runs with the best external parameters available so far ($\alpha_1 = 392$, $\beta_1 = 1.0$ that corresponds to the best Hanssen and Kuipers' score of 0.988) for Equation 2, and various scenarios as shown in Tables 3 and 4. Table 3 shows the cases where Equation 2 gives false negatives besides false positives, and Table 4 shows the remaining scenarios in order to covert the whole spectrum, from no intrusion to a full vector of intrusions. Table 5 shows the corresponding Hanssen and Kuipers' scores obtained for Equation 2.

In order to compare Equation 2 with Equation 1, the GA was run 30 times with the fitness function free of external parameters as in Equation 1, with the same set of GA parameters, recording the number of false positives and false negatives. No false positives, nor false negatives were present, so the Hanssen and Kuipers' score was perfect, at 1. Equation 1 with no external tuning, then outperforms Equation 2 in all scenarios, for the set of parameters as defined in Section 4.

Taking together all 25 scenarios, i.e., from no intrusion to 24 intrusions (no exclusive ones), the estimated average of the Hanssen and Kuipers' scores, for Equation 2, gives 0.840 with a standard deviation of 0.279 and the estimated standard error of the average of 0.056. As the av-

[4] Previous tests used an artificial vector of 11 intrusions.

	Equation 2 *with* $\alpha = 392.0$, $\beta = 1.0$						Equation 1 *with Union Operator*					
	Scenario						*Scenario*					
	No Intrusion		1 *Intrusion*		24 *Intrusion*		*No Intrusion*		1 *Intrusion*		24 *Intrusion*	
	False+	False-	False+	False-	False+	False-	False+	False-	False+	False-	False+	False-
Ave.	9.67	0.00	9.00	0.03	0.00	3.03	0.00	0.00	0.00	0.00	0.00	0.00
Std.	1.35	0.00	0.00	0.18	0.00	1.43	0.00	0.00	0.00	0.00	0.00	0.00

Table 3: Comparison between best fitness functions as shown in Table 2 and fitness function as in Equation 1. 30 runs per scenario per Equation.

Number of Intrusions												
0	1	2	3	4	5	6	7	8	9	10	11	12
-0.403	0.579	0.675	0.908	0.900	0.892	0.987	0.879	0.875	0.867	0.862	0.992	0.978
Number of Intrusions												
13	14	15	16	17	18	19	20	21	22	23	24	-
0.999	0.977	0.967	0.967	0.953	0.967	0.940	0.942	0.900	0.800	1.000	1.000	-

Table 5: Hanssen & Kuipers' scores given by Equation 2 for averages as in Tables 3 and 4.

	2	3	4	5	6	7	8	9	10	12	13
Ave.	7.16	1.93	2.00	2.06	0.23	2.06	2.00	2.00	1.93	0.26	0.01
Std.	0.37	0.36	0.00	0.36	0.77	0.18	0.00	0.00	0.36	0.45	0.30
	14	15	16	17	18	19	20	21	22	23	-
Ave.	0.23	0.30	0.26	0.33	0.20	0.30	0.23	0.30	0.40	0.00	-
Std.	0.43	0.46	0.45	0.47	0.40	0.46	0.43	0.46	0.49	0.00	-

Table 4: Average of false positives given by Equation 2. 30 runs per scenario.

erage of the Hanssen and Kuipers' score of Equation 1 is 1.00 with standard deviation of 0.00 and estimated standard error of 0.00 then the difference between the two averages is $1.00 - 0.840 = 0.160$ with an estimated standard error of 0.056. The observed difference 0.160 is $0.160/0.056 = 2.86$ estimated standard errors greater than zero, which means that the Hanssen and Kuipers' scores for Equation 2 is statistically significantly different than the average of the Hanssen and Kuipers' scores of Equation 1 at the 95% confidence level according to the Z score [2].

In order to corroborate the accuracy of the estimated average of 0.84, the bootstrapping technique was applied [6]: $1,000$ samples were taken at random from the 25 scenarios. The average of the $1,000$ sample means was 0.8482 with an estimated standard error of 0.054. These values are quite similar to the ones previously calculated for the Hanssen and Kuipers' scores. Figure 1 shows the histogram of the average of the $1,000$ samples means and the one that corresponds to 0.84.

How different is the result of each Equation from the possible combination of both? If we combined the two data sets of Hanssen and Kuipers' scores obtained independently for each Equation, and perform the bootstrapping technique again over $1,000$ samples, an estimated

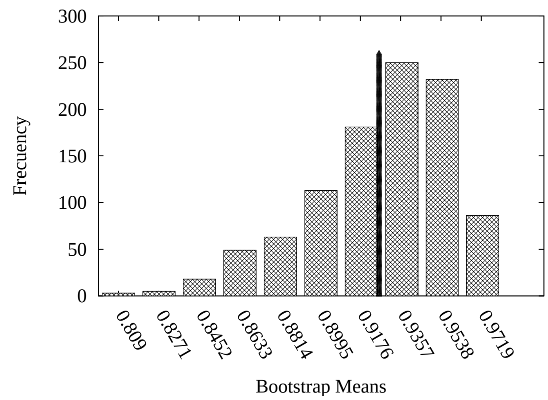

Figure 1: Distribution of the means of a randomized $1,000$ Hanssen and Kuipers' samples taken from 25 intrusion scenarios. Line corresponds to the mean 0.84 of the observed data for Equation 2.

averaged combined mean of 0.9258 with estimated standard error of 0.0284 is obtained. Comparing the combined estimated mean (0.9258) with the estimated mean of Equation 2 using the Z score, the result obtained is $(0.8482 - 0.9258)/(0.0559 - 0.0284) = -2.82$ which implies that both means correspond to statistically significantly different distributions at the 95% confidence level. If the two estimated means that are compared are combined with the one that corresponds to the average of Equation 1, the Z score is $(1.00 - 0.9258)/0.0284 = 2.61$ which indicates that the distributions of each estimated mean are statistically significantly different at the 95% confidence level. The corresponding histogram for the combined case is in Figure 2.

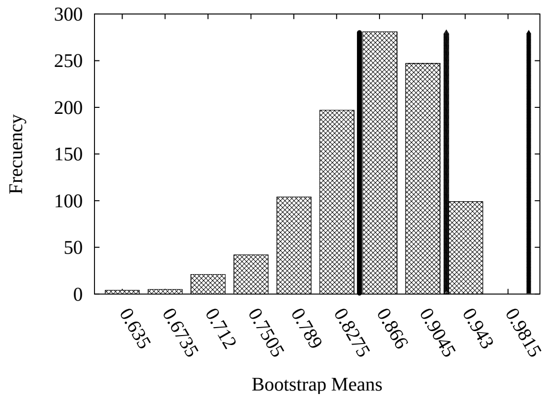

Figure 2: Distribution of the means of a randomized $1,000$ samples taken from 25 intrusion scenarios when original Hanssen and Kuipers' scores are combined for Equations 1 and 2. Left line corresponds to the mean 0.84 of the observed data for Equation 2. Middle line corresponds to the mean of the means over the $1,000$ samples of the combined data. Right line corresponds to the mean 1.00 of the observed data for Equation 1.

6. Conclusions and Future Work

This paper addressed the difficult problem of finding good parameters that join the objective(s) and the constraint(s). We used the intrusion detection problem (see Section 1) to perform the corresponding empirical study. Equation 1 penalizes each chromosome in a finer way with no use of external parameters. This characteristic makes Equation 1 good for avoiding false positives. Equation 1, free of external tuning, outperforms Equation 2, with external tuning, using the Hanssen and Kuipers' score as performance metric (see Section 4). Validating the specific hypothesis that states that the misuse detection problem has better performance when the fitness function used is Equation 1, free of external tuning, than when the fitness function used is Equation 2 with external parameters. As Equation 1 alone has the drawback of giving false negatives, the mechanism that stores the current solution and begins to add possible solutions—if the algorithm finds them—was used [5].

Certainly, Equation 1, which is free of external tuning of parameters outperforms Equation 2 (with external tuning) and contributes to the general knowledge. Work done elsewhere tried to generalize the concept of Equation 1 using it partially for finding longest connected paths in hypercubes that obey certain constraints [4].

This paper touches the core of GAs: fitness functions. Fitness functions guide GAs to possible solutions. They work in conjunction with the selection mechanism in choosing chromosomes that possibly are going to be mated and give rise to possibly better solutions. So, it is expected that if the fitness function is selected correctly, it will guide the algorithm to good solutions, however, if it is not set correctly, it can mislead the algorithm. The problem then is how to join objective(s) and constraint(s) in fitness functions to help in the setting of fitness functions; this hypothesis was tested, in this paper, in the context of the misuse detection problem.

Acknowledgments

We thank Ben Carlson, Doctoral student of Computer Science at the University of Oklahoma, for helping us in proofreading this paper.

References

[1] C. A. C. Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):269–308, 1998.

[2] P. R. Cohen. *Empirical Methods for Artificial Intelligence*. The MIT Press, 1995.

[3] P. A. Diaz-Gomez and D. F. Hougen. A genetic algorithm approach for doing misuse detection in audit trail files. In *Proceedings of the International Conference on Computing*, pages 329–335, 2006.

[4] P. A. Diaz-Gomez and D. F. Hougen. Genetic algorithms for hunting snakes in hypercubes: Fitness function analysis and open questions. In *Proceedings of the International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 389–394, 2006.

[5] P. A. Diaz-Gomez and D. F. Hougen. MISUSE DETECTION: An iterative process vs. a genetic algorithm approach. In *Proceedings of the International Conference on Enterprise Information Systems*, 2007.

[6] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

[7] L. Mé. Security audit trail analysis using genetic algorithms. In *Proceedings of the International Conference on Computer safety, reliability, and Security*, pages 329–340, 1993.

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