

Evolving Efficient Security Systems Under Budget Constraints Using Genetic Algorithms

Michael L. Gargano, William Edelson, Paul Benjamin,
Paul Meisinger, Maheswara Kasinadhuni, and Joseph DeCicco

ABSTRACT

The EASI model (estimate of adversary sequence interruption model) is a dynamic, analytic method widely used by security professionals to evaluate a physical protection security system (ppss). Our methods involve using genetic algorithms to evolve such systems when budget constraints or detector sequencing must be considered.

Keywords: physical protection security system, genetic algorithms, EASI

Introduction

Here we present genetic algorithm [1,9,11] models that can help security analysts evaluate the efficiency and effectiveness of a physical protection security system against adversarial threats attempting to reach a protected target. These models can be easily implemented and are useful tools in the initial and intermediate analysis of major physical protection security systems for high consequence loss of assets. They also aid in the designing of protection systems for low consequence loss of assets.

The EASI model (estimate of adversary sequence interruption) [4] is a simple, easy to use method of evaluating physical protection security system performance along a specific path and under specific conditions of threat and of system operation. This model assesses the interruption of an attack based on the complex interactions amongst detection, delay, response, and communication using the effectiveness measure of timely detection (sequence interruption). This model is often used as the basis for other more complicated models (e.g., ASSESS, FESEM, ISM, SAFE, SAVI, SNAP, etc. see [4]).

This research considers three models:

- effectively allocating budget monies to control the probability of detection at each level in a physical protection security system with a fixed sequence of detectors by using genetic algorithms with EASI as a measure of fitness
- effectively sequencing re-locatable detectors (e.g., guards) in a single path detection system using genetic algorithms with multiple genomes as a genotype with EASI as a measure of fitness
- effectively allocating budget monies in a multi-path model using the co-evolution paradigm with EASI as a measure of fitness.

MODEL I Allocating budget monies to control the probability of detection at each level in a physical protection security system with a fixed sequence of detectors using genetic algorithms with EASI as a measure of fitness.

After a new (or upgraded existing) physical protection security system has been developed it should be quantitatively analyzed. Even if adversary risk is low if asset loss risk is high a rigorous analyze should be done. A physical protection security system is a complicated interaction of detection, delay, communication, and response that determines overall effectiveness.

Analysis and evaluation is based on studying an adversary path to an asset (i.e., a target). An ordered sequence of actions to penetrate a facility is called an adversary path. Effective usage of elements that comprise potential detection along that path is the goal of a physical protection security system. Detection, together with communication, delay, and response, attempts to interrupt an adversary attempt on the target. Performance measures consist of probability of detection and communication as well as delay and response times. Since the goal of the adversary is to reach the target before detection and being stopped, the adversary will choose a path that increases the chances of success. Timely detection is therefore very important in arresting an attempt of intruder penetration.

Assume the adversary must follow the actions A_1, A_2, \dots, A_n in sequence (e.g., A_1 may be penetration of an outside perimeter fence, A_2 may be the penetration of an entrance door, etc.). Each A_i can be done in time t_i with pd_i being the probability of detection at that level. Detection involves sensing, assessing, and transmitting a breach. We can calculate the cumulative probability of detection at A_j to be the probability of interruption

$$PI = 1 - \left(\prod_{i=1}^j (1 - pd_i) \right) \text{ and } TR = \text{the time remaining to reach the target } A_n.$$

Each t_i is assumed to be normally distributed. If the response time is TG (also normally distributed), we require the expected values to satisfy $E(TR) > E(TG)$ if we are to have enough time to respond to the adversary attack before they reach their goal (i.e., the target). Each pd_i can be thought of as being under management control through appropriate budget allocations.

This section extends the standard analytical EASI model of physical protection security systems [4] by considering budget constraints that affect detection at level i and thus indirectly the probability of system interruption. We propose assigning a fitness based on the probability of intruder interruption. A GA (genetic algorithm) [1, 2, 5, 6, 7, 8, 9, 11] employed an EASI model for measuring its fitness. In this research we also used a sigmoidal function to more realistically relate budget spending to detection at each level in a non-linear fashion.

Specifically, consider a ppss with n access levels. Let $T = F + B$ be the total budget in dollars to be allocated amongst the levels of the network in order to increase

probability of system interruption. F is the fixed portion of the total budget required to insure that each level is minimally protected, thus $F = m_1 + m_2 + \dots + m_n$. This allows us to allocate to each of the n levels the amount m_i required to provide a minimal detection probability $pd_i = f(m_i)$. The detection probability is obtained from a generalized sigmoidal function with two parameters and relates budget expenditure to node reliability. Let $B = T - F$ be the remaining amount of the total budget which can be additionally allocated to increase detection probability at some or all of the levels.

The vector $(\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_n)$ is used to denote the proportion of B to be spent at each node in order to increase the survivability of node i to $pd_i = f(\rho_i \bullet B + m_i) = f(b_i)$ and is called the additional budget allocation vector ABAV.

The genetic algorithm paradigm is an adaptive method based on Darwinian natural selection. It applies operations of reproduction (i.e., crossover), and mutation, and also uses a paradigm based on survival of the fittest, to a given population of potential solutions to generate a new, more fit, population of potential solutions. The process repeats itself until it converges to a stable optimal (or near optimal) solution. In this paper, we propose efficient genetic algorithms to solve real world security problems.

In Model I we use an encoding that provides feasible search space membership (after normalization) [7, 8] in each generation of the genetic algorithm, that is, not only during population initialization (in generation 0) but also in all other generations following matings and mutations.

ENCODING MEMBERS OF THE POPULATION

Each population member $(\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_n)$ is encoded as a proportion vector of length $n =$ number of levels to be penetrated with $\rho_i \geq 0$ for $1 \leq i \leq n$ and $\sum \rho_i = 1$. It is easy to generate an initial population member using a normalization method. First simply generate a vector $(x_1, x_2, \dots, x_i, \dots, x_n)$ of n IID (independent identically distributed) $x_i \sim \text{uniform}[0, 1]$ for $1 \leq i \leq n$. By considering the normalized vector $(x_1 / \sum x_i, x_2 / \sum x_i, \dots, x_i / \sum x_i, \dots, x_n / \sum x_i)$ we create a proportion vector (of course we must insure that $0 < \sum x_i$). The ABAV is simply $(\rho_1, \rho_2, \dots, \rho_j, \dots, \rho_n)$ where $\rho_j = x_j / \sum x_j$.

Detection probabilities at each level can now be easily calculated using $f(b_j)$ where $b_j = B \bullet \rho_j + m_j$. The detection probability at level j is obtained from a generalized sigmoidal function with two parameters and relates budget expenditure to detection probability.

$$\text{Specifically, } f(x) = 0.5 + \frac{\text{signum}(x - m)}{2} \bullet \frac{2^{s \bullet (\text{abs}(x-m))} - 1}{2^{s \bullet (\text{abs}(x-m))}}$$

where m is a centralizing parameter and s is a stretching parameter.

FITNESS

The fitness value of each population member is based on the results of the EASI model (i.e. the probability of intruder interruption). Through the use of a sigmoidal function that more realistically relates budget spending to the probability of detection on each level in a non-linear fashion, our goal is to allocate the budget optimally.

MATING and MUTATING

Selection of parents for mating involves choosing one high fitness member of the population and the other member randomly. The reproduction process is a simple crossover operation whereby two selected parent members swap elements in randomly chosen positions to create two new offspring members. The crossover operation produces an encoding for the offspring members having element values which always satisfies the membership requirements after normalization.

Given two parents randomly chosen from the population P_1 and P_2

$$P_1 = (\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_n) \text{ and } P_2 = (\pi_1, \pi_2, \dots, \pi_i, \dots, \pi_n)$$

we can mate them to produce two children (offspring) C_1 and C_2 for example

$$C_1 = (\rho_1 / s_1, \pi_2 / s_1, \rho_3 / s_1, \dots, \rho_i / s_1, \pi_{i+1} / s_1, \dots, \rho_n / s_1)$$

$$\text{where } s_1 = \rho_1 + \pi_2 + \rho_3 + \dots + \rho_i + \pi_{i+1} + \dots + \rho_n$$

$$\text{and } C_2 = (\pi_1 / s_2, \rho_2 / s_2, \pi_3 / s_2, \dots, \pi_i / s_2, \rho_{i+1} / s_2, \dots, \pi_n / s_2)$$

$$\text{where } s_2 = \pi_1 + \rho_2 + \pi_3 + \dots + \pi_i + \rho_{i+1} + \dots + \pi_n .$$

Similarly, a random population member can be mutated. Mutation is carried out by randomly choosing a member $P = (\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_n)$ of the population then randomly changing the value of its encoding at randomly chosen positions subject to the range constraints for that position. For example,

$$M = (\rho_1 / s, \pi_2 / s, \rho_3 / s, \dots, \rho_i / s, \pi_{i+1} / s, \dots, \rho_n / s)$$

$$\text{where } s = \rho_1 + \pi_2 + \rho_3 + \dots + \rho_i + \pi_{i+1} + \dots + \rho_n$$

Thus, after normalization population membership requirements are met.

GENETIC ALGORITHM METHODOLOGY

A genetic algorithm (GA) is a biologically inspired, highly robust heuristic search procedure that can be used to find optimal (or near optimal) solutions to hard problems. The GA paradigm uses an adaptive methodology based on the ideas of Darwinian natural selection and genetic inheritance on a population of potential solutions. It employs the techniques of crossover (or mating), mutation, and survival of the fittest to generate new, typically fitter members of a population over a number of generations [1, 9, 11].

We propose GAs for the three pps problems listed earlier. Our GAs create and evolve an encoded population of potential solutions so as to facilitate the creation of new feasible members by standard mating and mutation operations. (A feasible search space contains only members that satisfy the problem constraints. When feasibility is not

guaranteed, numerous methods for maintaining a feasible search space have been addressed in [11], but most are elaborate and complex. They include the use of problem-dependent genetic operators and specialized data structures, repairing or penalizing infeasible solutions, and the use of heuristics.) By making use of problem-specific encodings, each class of problem insures a feasible search space during the classical operations of crossover and mutation and, in addition, eliminates the need to screen during the generation of the initial population.

We adapted many of the standard GA techniques found in [1, 7, 9, 11] to these specific problems. A brief description of these techniques follows. Selection of parents for mating involves randomly choosing one very fit member of the population and the other member randomly. The reproduction process is a simple crossover operation whereby two selected parent members swap elements in randomly chosen positions to create two new offspring members. In our applications the crossover operation produces an encoding for the offspring that have element values that always satisfy the membership requirements after normalization. Mutation is performed by randomly choosing a member of the population, cloning it, and then changing values in its encoding at randomly chosen positions. The mutation operation too produces an encoding for the offspring that have element values that always satisfy the membership requirements after normalization. A grim reaper mechanism replaces low scoring members in the population with newly created more fit offspring and mutants. The GA is terminated when, either no improvement in the best fitness value is observed for a number of generations, a certain number of generations have been examined, and/or a satisficing solution is attained (i.e., the probability of interruption is not necessarily optimum, but is satisfactory).

The Generic Genetic Algorithm

We can now state the generic genetic algorithm we used for each application:

- 1) Randomly initialize a population of encoded potential solutions.
- 2) Map each encoded population member genotype to its equivalent phenotype.
- 3) Calculate the fitness of any population member not yet evaluated.
- 4) Sort the members of the population in order of fitness.
- 5) Randomly select parents for mating and generate offspring using crossover.
- 6) Randomly select then clone members of the population and generate mutants.
- 7) Sort all the members of the expanded population in order of fitness.
- 8) Use the grim reaper to eliminate the members of the population with poor fitness.
- 9) If (a termination criteria is met) then return solution(s) which is(are) best
else go to step 5.

MODEL II Sequencing re-locatable detectors (e.g., guards) in a single path detection system using genetic algorithms with multiple genomes as a genotype with EASI as a measure of fitness.

In this problem we wish to sequence re-locatable detectors (e.g., guards) in a single path detection system.

ENCODING MEMBERS OF THE POPULATION

Here, a standard permutation code [2, 5, 6] is used to identify the sequencing (i.e., ordering) of a solution. An n permutation [10, 12] of the integers $\{ 1, 2, \dots, n \}$ can be encoded by an array of size n where the value of the k^{th} position can range over the values $1, 2, \dots, n - k + 1$. Thus, an encoding of a permutation of the levels of a pps can be represented as an array C where $1 \leq C[k] \leq n - k + 1$ for $1 \leq k \leq n$. In order to decode a permutation code C to obtain the permutation that it represents, begin with an empty array P of size n , then for $1 \leq i \leq n$ fill in the $C[i]^{\text{th}}$ empty position (from left to right starting at position 1) of P with the value i . Consider a simple example, with $n = 5$ and $C[1] = 2, C[2] = 4, C[3] = 3, C[4] = 1, \text{ and } C[5] = 1$ (or 24311) which represents the permutation $P[1] = 4, P[2] = 1, P[3] = 5, P[4] = 3, \text{ and } P[5] = 2$ (or 41532).

Given a permutation array P , the reverse process begins with an empty array C of size n , then for $1 \leq i \leq n$ starting with $i = 1$ and ending with $i = n$ fill in the i^{th} position of C (from left to right starting at position 1) with the value $k - (\# \text{ of values } \leq i \text{ that occur before position } k \text{ in } P)$ where $P[k]$ contains the value i . (Note that $C[n]$ will always be 1, so that, we can shorten C to an $n - 1$ element array if we wish.) For example, an ordering of detectors $D = \{ d_1, d_2, \dots, d_5 \}$ in a pps of 5 levels based on the code (24311) would then be $(d_4, d_1, d_5, d_3, d_2)$.

Sometimes a genotype (encoding) is somewhat biased toward erratic moves through the mating and/or mutation process. This is called the locality problem. Sometimes this is just what we need (e.g., when the solution space is highly discontinuous or multi-modal) [6]. However, for this model we may encounter difficulties. Thus we propose the use of a multi-genome genotype where each genome represents the same phenotype. For example, $(d_4, d_1, d_5, d_3, d_2)$ could also be represented by the reverse code 31221. An n permutation [10, 12] of the integers $\{ 1, 2, \dots, n \}$ can be encoded by an array of size n where the value of the k^{th} position can range over the values

$1, 2, \dots, n - k + 1$. Thus, an encoding of a permutation of the levels of a pps can be represented as an array C' where $1 \leq C'[k] \leq n - k + 1$ for $1 \leq k \leq n$. In order to decode a permutation code C' to obtain the permutation that it represents, begin with an empty array P of size n , then for $1 \leq i \leq n$ fill in the $C'[i]^{\text{th}}$ empty position (from left to right starting at position 1) of P with the value $n - i + 1$. Consider a simple example, with $n = 5$ and $C'[1] = 3, C'[2] = 1, C'[3] = 2, C'[4] = 2, \text{ and } C'[5] = 1$ (or 31221) which also represents the permutation $P[1] = 4, P[2] = 1, P[3] = 5, P[4] = 3, \text{ and } P[5] = 2$ (or 41532). Given a permutation array P , the reverse process begins with an empty array C' of size n , then for $1 \leq i \leq n$ starting with $i = 1$ and ending with $i = n$ fill in the i^{th} position of C' (from left to right starting at position 1) with the value $k - (\# \text{ of values } \geq n - i + 1 \text{ that occur before position } k \text{ in } P)$ where $P[k]$ contains the value $n - i + 1$. (Note that $C'[n]$ will always be 1, so that, we can shorten C' to an $n - 1$ element array if we wish.). An

ordering of detectors $D = \{ d_1, d_2, \dots, d_5 \}$ in a ppss of 5 levels based on the code (31221) would then again be $(d_4, d_1, d_5, d_3, d_2)$.

The multi-genotype for the phenotype $(d_4, d_1, d_5, d_3, d_2)$ would be (2431131221) where the first half represents C and the second half represents C' . After mating or mutating the two genomes comprising the genotype may be different. In this case each half is evaluated separately and the better phenotype is encoded into a new genotype where both halves again represent the same phenotype in a multi-genome genotype.

FITNESS

The fitness value of each population member is based on the results of the EASI model (i.e. the probability of intruder interruption). Through the use of a sigmoidal function that more realistically relates budget spending to the probability of detection on each level in a non-linear fashion, our goal is to allocate the budget optimally.

MATING AND MUTATING

The reproduction process is a simple crossover operation whereby two selected parent members swap elements in randomly chosen positions to create two new offspring members. The crossover operation produces an encoding for the offspring members having element values that always satisfy the position bounds. Mutation is carried out by randomly choosing a member of the population then randomly changing the value of its encoding at randomly chosen positions. Mating and mutating are feasible operations that always obey range constraints for each position. Thus, population membership requirements are always met.

MODEL III Allocating budget monies in a multi-path model using the co-evolution paradigm with EASI as a measure of fitness.

This model assumes that there are multiple ways to access entry at each level i , $A_{i1}, A_{i2}, \dots, A_{i m_i}$. Since the EASI model assumes only one path, this multi-path model can only assess EASI results for a specific path (i.e., one at a time).

ENCODING MEMBERS OF THE POPULATION

A co-evolution paradigm that employs two competing genetic algorithms will be used. First, we propose using the following feasible encodings for the adversary path genetic algorithm. Let (m_k) be the number of ways to access entry on the k^{th} level of the ppss. For each of the $(m_1) \cdot (m_2) \cdot \dots \cdot (m_s)$ ordered path sequences possible, we can associate a unique code consisting of an s -tuple of length s (v_1, v_2, \dots, v_s) where

$(m_1 + m_2 + \dots + m_{i-1} + 1) \leq v_i \leq (m_1 + m_2 + \dots + m_i)$ for each $1 \leq i \leq s$ with $m_0 = 0$ for convenience. These s values represent a path sequence chosen by the adversary.

A modified ppss ABAV additional budget allocation vector $(\rho_{11}, \rho_{12}, \dots, \rho_{1m_1}, \rho_{21}, \dots, \rho_{ij}, \dots, \rho_{s1}, \dots, \rho_{sm_s})$ will be employed in the other genetic algorithm. Here ρ_{ij} represents the amount of budget spent for detection at access j on level i .

For example, let $s = 4$ and $m_1 = 3, m_2 = 5, m_3 = 2$ and $m_4 = 3$.

The modified ppss ABAV additional budget allocation vector would be

$(\rho_{11}, \rho_{12}, \rho_{13}, \rho_{21}, \rho_{22}, \rho_{23}, \rho_{24}, \rho_{25}, \rho_{31}, \rho_{32}, \rho_{41}, \rho_{42}, \rho_{43})$ or
 $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \rho_7, \rho_8, \rho_9, \rho_{10}, \rho_{11}, \rho_{12}, \rho_{13})$ with size $m_1+m_2+m_3+m_4 = 13$.

Similarly the accesses would be

$(A_{11}, A_{12}, A_{13}, A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{31}, A_{32}, A_{41}, A_{42}, A_{43})$ or
 $(A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}, A_{13})$ with size 13.

There would be $(m_1) \cdot (m_2) \cdot (m_3) \cdot (m_4) = 90$ possible adversary paths. The code for path $A_{12}, A_{23}, A_{31}, A_{42}$ (or A_2, A_6, A_9, A_{12}) would be $(v_1 = 2, v_2 = 6, v_1 = 9, v_2 = 12)$.

FITNESS

Since each population member of the first genetic algorithm represents one path, the fitness value of each population member is based on the results of the EASI model based on how the second genetic algorithm allocated the budget through the use of a sigmoidal function to more realistically relate budget spending to the probability of detection at each level in a non-linear fashion. Conversely, each member of the second genetic algorithm has its fitness evaluated by the results of the EASI model based on how the second genetic algorithm sequenced its paths.

MATING and MUTATING

Note that in both genetic algorithms mating and mutating are straightforward operations on members of their respective populations.

In the adversary population of paths, the reproduction process is a simple crossover operation whereby two selected parent members swap elements in randomly chosen positions to create two new offspring members. The crossover operation produces an encoding for the offspring members having element values that always satisfy the position bounds. Mutation is carried out by randomly choosing a member of the population then randomly changing the value of its encoding at randomly chosen positions. Thus, population membership requirements are always met.

In the ppss population of ABAVs, mating and mutating are similar to Model I. The reproduction process is a simple crossover operation whereby two selected parent members swap elements in randomly chosen positions to create two new offspring members. The crossover operation produces an encoding for the offspring members having element values that always satisfy the restrictions on proportion vectors after normalization. Mutation is carried out by randomly choosing a member of the population

then randomly changing the value of its encoding at randomly chosen positions subject to the restrictions on proportion vectors after normalization. Thus, after normalization population membership requirements are met.

THE CO-EVOLUTION PARADIGM

Our co-evolution paradigm begins by randomly assigning a population to each genetic algorithm (i.e., the first being the adversary population of paths and the second being the pps population of budget allocations). Using EASI results (i.e., probability of interruption), we can evaluate each member of the second population (pps budget allocations) based on how they interrupt members from the first population (adversary paths). Then the first genetic algorithms (adversary paths) can evaluate its members based on the EASI results on how they interrupt members from the second population (pps budget allocations). This process is iterated repeatedly as one population evolves based on the evolution of the other in the previous generation.

Conclusions and Future Research Plans

These algorithms will be implemented using customized software programs developed by the authors and will generate experimental and empirical results. Compared with other models this model is robust, easy to implement, easy to use, gives optimal / near optimal results, converges quickly, uses feasible encodings up to normalization, allows the flexibility of seeding the initial population with expert knowledge, and can handle constraints and/or incorporate heuristics.

The methods proposed here on applications of genetic algorithms to evaluate physical protection security system performance present promising areas of research.

Future work includes implementing and analyzing the genetic algorithms proposed here and further investigation of other security models.

Acknowledgements

We wish to thank Pace University's School of Computer Science and Information Systems (SCSIS) for partial support of this research (specifically a summer 2002 research grant).

References

- [1] Davis, Lawrence, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
- [2] Edelson, W. and Gargano, M.L., Minimal Edge-Ordered Spanning Trees Solved By A Genetic Algorithm With Feasible Search Space, Congressus Numerantium 135, 1998, pp. 37-45.
- [3] Edelson, W. and Gargano, M. L., A Modified Prufer Code, Graph Theory Notes of New York (New York Academy of Sciences).
- [4] Garcia, Mary Lynn, The Design and Evaluation of Physical Protection Systems, Butterworth-Heinemann, 2001.
- [5] Gargano, M.L., Edelson, W., A Genetic Algorithm Approach to Solving the Archaeological Seriation Problem, Congressus Numerantium 119, 1996, pp. 193-203.
- [6] Gargano, M.L., Edelson, W., “Optimal Sequenced Matroid Bases Solved By Genetic Algorithms With Feasibility Including Applications”, Congressus Numerantium 150 (2001), pp 5-14.
- [7] Gargano, M.L., Edelson, W., He, Y., “Evolving Optimal Reliability Networks Subject To Budget Constraints Using Genetic Algorithms II, GECCO, 2002 Workshop Proceedings-7/9, pp 178-179.
- [8] Gargano, M.L., Rajpal, Neetu, Using Genetic Algorithm Optimization to Evolve Popular Modern Abstract Art, Proceedings of the Long Island Conference on Artificial Intelligence and Computer Graphics, Old Westbury, NY, 1994, pp.38 - 52.
- [9] Goldberg, David E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, 1989.
- [10] Grimaldi, R.P., Discrete and Combinatorial Mathematics, 4th ed., Addison-Wesley, 1999.
- [11] Michalewicz, Z., Heuristic Methods For Evolutionary Computation Techniques, Journal of Heuristics, vol. 1, no. 2 pp 177-206, Spring 1996.
- [12] Rosen, Kenneth H., Discrete Mathematics and Its Applications, 3rd Edition, Random House, New York, 1994, pp. 135, pp. 596-597.