## Annual Report

## DIMACS Project on Algorithms for Port of Entry Inspection


| | |
|---|---|
| PI: | Fred S. Roberts |
| PI Institution: | Rutgers University |
| PI Phone: | 732-445-4303 |
| PI Fax: | 732-445-5932 |
| PI email: | froberts@dimacs.rutgers.edu |
| PI URL Home Page: | http://www.dimacs.rutgers.edu/People/Staff/froberts/index.html |


| | |
|---|---|
| Grant/Contract Number: | N00014-05-1-0237 |
| Period of Grant: | February 1, 2005 – January 31, 2007 |
| Period of Report: | February 1, 2005 – June 30, 2005 |


## Participants

***Rutgers Faculty***
Tayfur Altiok, Rutgers University, Industrial Engineering
Endre Boros, Rutgers University, RUTCOR
Elsayed A. Elsayed, Rutgers University, Industrial and Systems Engineering
Paul Kantor, Rutgers University, SCILS
Alexander Kogan, Rutgers University, Accounting & Information Systems
Brenda Latka, Rutgers University, DIMACS
Paul Lioy, joint Rutgers University, EOHSI and University of Medicine and Dentistry of NJ
David Madigan, Rutgers University, Statistics
Rick Mammone, Rutgers University, CAIP
Benjamin Melamed, Rutgers University, RBS
S. Muthukrishnan, Rutgers University, Computer Science
Fred S. Roberts, Rutgers University, DIMACS

***Los Alamos National Laboratory (LANL)***
Sallie Keller-McNulty, Los Alamos National Laboratory
Feng Pan, Los Alamos National Laboratory
Richard R. Picard, Los Alamos National Laboratory
Kevin Saeger, Los Alamos National Laboratory
Phillip Stroud, Los Alamos National Laboratory

***Rutgers Students and Student Visitors***
Ozlim Alpinar, Rutgers University, Industrial and Systems Engineering
Lilya Fedzhora, Rutgers University, RUTCOR
Abdullah Karaman, Rutgers University, Industrial and Systems Engineering
Devdatt Ladd, Rutgers University, CAIP
Francesco Longo, University of Calabria, Italy
Milanic Martin, Rutgers University, RUTCOR
Paul Raff, Rutgers University, Mathematics
Hao Zhang, Rutgers University, Industrial and Systems Engineering

## Objectives and Approach

Finding ways to intercept illicit nuclear materials and weapons destined for the U.S. via the maritime transportation system is an exceedingly difficult task. Until recently, only about 2% of ships entering U.S. ports have had their cargoes inspected. The percentage at some ports has now risen to 6%, but this is still a very small percentage. The purpose of this project is to develop decision support algorithms that will help us to optimally intercept illicit materials and weapons. The algorithms we seek will find inspection schemes that minimize total cost, including the "cost" of false positives and false negatives.

We envision a stream of entities arriving at a port and a decision maker having to decide how to inspect them, which to subject to further inspection and which to allow to pass through with only minimal levels of inspection. This is a complex sequential decision making problem. Sequential decision making is an old subject, but one that has become increasingly important with the need for new models and algorithms as the traditional methods for making decisions sequentially do not scale.

Existing algorithms for optimally intercepting illicit cargo assume that sensor performance, operating characteristics of ports, and overall threat level are all fixed. The approach in this project involves decision logics and is built around problem formulations that lead to the need for combinatorial optimization algorithms as well as methods from the theory of Boolean functions, queuing theory, and machine learning. Practical complications of any such approach involve economic impacts of surveillance activities, errors and inconsistencies in available data on shipping and import terminal facilities, and the tradeoffs between combinations of sensors. A full-blown approach to the port-of-entry inspection problem includes the decision problem of when to initiate different levels of inspection if there are seasonal variations in cargo flows and cargo types, sensor reliability effects, and changing threat levels. In general terms, it is necessary to explore new sensor deployment methods and sensor configurations, the problem of false alarms from naturally occurring radiation sources (which vary spatially) and from innocent cargos (such as medical waste), and models of "information sensors." Moreover, existing algorithms for designing port-of-entry inspection are rapidly coming up against the combinatorial explosion caused by so many possible alternative inspection strategies. In this project, we are attempting to develop an approach that brings into the analysis many of these complications. Details of the approach are summarized in the paper "Decision support algorithms for port-of-entry inspection." They were presented by the PI at the DHS Meeting on Research and Development Partnerships in Homeland Security in Boston in April 2005 and are to appear in the conference proceedings.

The project is being carried out in collaboration between a university team of faculty and students and a team from the Los Alamos National Laboratory. The university team is based at DIMACS, the Center for Discrete Mathematics and Theoretical Computer Science. DIMACS is a partnership of Rutgers and Princeton Universities with industrial partners at AT&T Labs, Bell Labs, NEC Laboratories America, and Telcordia Technologies, and affiliated partners at Avaya Labs, HP Labs, IBM Research, Microsoft Research, Stevens Institute of Technology, and

Georgia Institute of Technology. We have in place a team of DIMACS members from Rutgers University that reflects the multi-disciplinary nature of the port-of-entry inspection problem.

## Technical Progress

After exchanges of ideas by telephone and email, the project team met at DIMACS on May 25-26, 2005 to review in depth the initial approach to the port-of-entry inspection problem taken by the Los Alamos team. Our Los Alamos partners studied four tests for deciding if a cargo was positive, that is, contained illicit material. These tests (we will call them all sensors) were evaluation of ships manifests, passive radiation signature, radiographic image, and induced fission. All of these have costs associated with them, including the cost of a reading indicating illicit material when there is none, a false positive (FP), the cost of a reading indicating there is no illicit material when there is, a false negative (FN), time costs of using the sensor, delay costs of waiting for the sensor, and fixed cost of equipment, labor, etc. For each sensor the readings for cargo containing illicit material (positives) and readings for cargo not containing illicit material (negatives) are random variables. It is assumed that each is distributed normally and that the mean and standard deviation for each of these distributions is known. Setting a threshold level for when a reading is considered positive controls the performance characteristics of each sensor, that is the probability of FP and the probability of FN. For example, a false positive occurs when a reading from a sensor for a cargo that does not contain illicit material falls in the range where that sensor gives a positive reading. The model our Los Alamos partners created assigned an output of 0 (absence of illicit material) or 1 (presence of illicit material) for each sensor. In general, n sensors will yield a string (vector) of 0's and 1's of length n. A decision function is a Boolean function F on an n-dimensional vector with output 0 (negative) indicating the cargo is not suspected of containing illicit material and an output of 1 (positive) indicating the cargo is suspected and must be "unstuffed." The cost of a false positive is the cost of unstuffing, $600. The cost of a false negative was based on the estimated cost of the destruction of the World Trade Center, $50 billion, times the estimated fraction of imports with weapons of mass destruction (WMD), 1 per 5 years. To which sensor a cargo is sent depends on the output of the previous sensor. This can be modeled with a binary decision tree (BDT). The best LANL could accomplish was to find the binary decision tree in the case of 4 sensors that would minimize total cost. Restricting to complete (every variable is required) and monotone (if F(0,1,1,0) is 1 then F(1,1,1,0) must also be 1) Boolean functions, there are 114 possible functions and 11,808 possible binary decision trees. Using two months data from the LA Long Beach port, by exhaustive search it was determined that there was 1 best, the best 100 fell into 10 patterns, and there were about 300 that were close enough to optimal.

In reality, we will want to use many more sensors and the large number of possible trees makes an exhaustive search infeasible. One goal of this project is to understand the characteristics and behaviors of the solution space with the objective of developing heuristics that will allow rapid computation in finding optimal and near optimal trees. This heuristic will need to be able to scale up to 12, 20 or even higher numbers of sensors.

Several problems were recognized with this model. There are many different types of costs involved. There are fixed costs and salary costs for the inspection stations. There are delay costs that are primarily borne by the shippers. Also, a sufficiently long delay could cause the entire system to collapse causing proliferating economic costs throughout the country and the world.

The delays can be a random variable; for example there is variability in the time to read the radiograph.

The Rutgers team has subdivided their approach into three interdependent parts. One group is studying the sensitivity of the determination of optimal and near optimal trees to the input parameters. As input parameters such as the costs of false positives and false negatives, the costs of delays, etc., are estimated with more or less accuracy, one wants solutions whose sensitivity to changes in these parameters is known and tolerable. This group is also applying datamining techniques to study the dataset of 11,808 possible binary decision trees provided by LANL for the case of 4 sensors. A second group is considering the optimization problem in the context of a shipping port and building a simulation model of inspection stations as one part of an operating port. Such a model will allow the estimation of some of the cost parameters by, for example, providing estimates of delays. A third group is developing new modeling approaches that are computationally cheap, highly scalable, and able to incorporate various cost factors with enough flexibility to include future technologies.

*Sensitivity Analysis and Datamining*

In general, analyzing the effect of varying one or more parameters on an optimal solution of a generalized decision tree cost-minimizing problem may be classified as sensitivity analysis if it concerns a single parameter or as parametric programming if more than one parameter needs to be varied.

The cost function can be formulated as follows. Each decision (at each node) has a certain cost associated with it. This cost would include the costs of false positives and false negatives, which can take place at the sensor/node, the cost associated with the utilization of the sensor itself as well as the costs that would accumulate as a result of the decision of the sensor. We assume that for each sensor the readings for cargo containing illicit material (positives) and readings for cargo not containing illicit material (negatives) are each distributed normally and that the mean and standard deviation for each of these distributions is known. If a reading is positive (negative), we assume the decision tree is traversed to the left (right). We formulate a probabilistic cost model for a sensor/node in the decision tree and also incorporate costs of each branch in a recursive fashion.

$$C_{Node} = (P_{FalsePositive}*C_{FalsePositive}) + (P_{FalseNegative}*C_{FalseNegative}) + C_{Observation}$$
$$+ (P_{Positive}*C_{LeftTree} + P_{Negative}*C_{RightTree})$$

where $C_{Node}$ = cost of tree this node downwards

$P_{FalsePositive}$ = probability of a false positive decision in this sensor/node

$C_{FalsePositive}$ = cost incurred as a result of a false positive at this node on the system

$P_{FalseNegative}$ = probability of a false negative decision in this sensor/node

$C_{FalseNegative}$ = cost incurred as a result of a false negative at this node on the system

$C_{Observation}$ = the cost incurred as a result of operating the sensor at this node. This cost includes the unit, fixed and delay costs.

$P_{Positive}$ = probability of a positive decision in this sensor/node.

$C_{LeftTree}$ = cost of the entire left tree with the current node as the root, computed recursively until terminal nodes are reached.

$P_{Negative}$ = probability of a negative decision in this sensor/node.

$C_{RightTree}$ = cost of the entire right tree with the current node as the root, computed recursively until terminal nodes are reached.

($P_{Positive}$*$C_{LeftTree}$ + $P_{Negative}$*$C_{RightTree}$) is the recursive term, which can be run over the entire tree.

Here we have labeled the Boolean decision of a sensor as positive or negative. The $C_{LeftTree}$ is the recursively accumulated cost of the left subtree of the current sensor/node and the $C_{RightTree}$ is a similar cost for the right subtree. Thus each node in the tree has a cost connected to the measurement by the sensor, a cost of false classifications as well as the costs of additional sensors that are brought into use depending on our current sensor's decision. These additional sensors are the nodes on either side of the subtree of which the current sensor can be considered as root node.

There are variables associated with each node/sensor of the tree that are provided as *a priori* knowledge. The purpose of the sensitivity analysis would be to vary these parameters, individually as well as in n-tuples, to see how the change in parameters affects the cost structure of each tree and hence, which tree minimizes cost. This analysis is an optimization problem but with constraints which are not clearly defined, apart from overall cost-minimization. At present, algorithms to enumerate the feasible binary decision trees and the recursive cost formulation model are being implemented. The optimal solutions can be found for the values provided for the parameters. In the second stage, basic sensitivity tests will be carried out on the optimal and nearly optimal solutions and the effects of varying parameter values on the costs and optimality of trees will be analyzed. Based on these results, further tests can then be carried out extensively on the effectiveness and sensitivity of each parameter in determining the optimality of a tree structure. In particular, we will attempt to see whether BDTs which stay above a chosen level of optimality can be characterized by their structure and whether certain parameters affect the sensitivity of all optimal BDTs in a uniform way or whether some parameters affect certain tree structures more while causing less variations in other tree structures. The other test to be carried out is to find out by how much parameters with high underlying costs overshadow other relatively cheaper parameters. In addition, methods for pruning the decision trees will be developed. The parameters on which the cost structure is formulated include the mean and standard deviation of the probability distribution of a *positive* decision and the parameters which relate it to the distribution of the *negative* decision, the costs of *false positive* and *false negative* decisions, as well as the intrinsic cost associated with each sensor.

Phillip Stroud of LANL did extensive calculations to evaluate 11,808 possible binary decision trees for the case of 4 sensors, using specific assumptions about costs. Datamining techniques can be applied to this data set to study the behavior of optimal and nearly optimal trees. If in a binary decision tree, a sensor appears at the same level, no matter on which branch of the tree it appears, then the sensors have a well-defined order. For example, the tree in Figure 1 with 3 sensors indicates that the sensors should be used in the order $a_2$, $a_0$, and $a_1$. It can be argued that there is a straightforward cost heuristic, perhaps based on a measure of power/cost for each sensor, that can be used to specify the order in which the sensors should be used. We are formulating a method to use the LANL dataset to test the following conjectures:

1. For nearly all top scoring trees, the sensors are used in a well-defined order.
2. Nearly all top scoring trees have the same well-defined order.
3. Nearly all top scoring trees have the well-defined order that is predicted using the cost heuristic.
4. All the trees which have the well-defined order predicted using the cost heuristic will be fairly close to having highest score.
5.

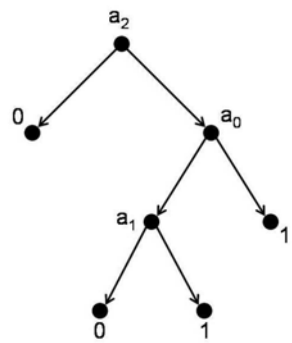


Figure 1. Binary Decision Tree

*Simulation Modeling and Analysis of Cargo Flow*

The binary decision tree is a model of the security operations of using various sensors to obtain readings of various types on cargo containers. These security operations take place in the context of a port in which there are many, many other activities that affect and are affected by the security operations. This part of the project is focusing on modeling and analysis of security operations for cargo flow through marine ports, with special emphasis on containerized cargo subject to the Container Security Initiative (CSI) program in the context of the entire port operation. We are building a simulation model of cargo security operations in a container port including handling, storage and inspection. An initial simulation model has been developed using the ANYLOGIC simulation tool that is web-based. The objective is to be able to understand the efficiency of the security operations and their impact on various delays in a typical marine port. Vessel arrival processes, vessel unloading operations and the various inspection processes are critical components of the modeling. The impact of the arrival process and the inspection ratio on the overall marine port operation is being studied. The simulation model will provide input into parameter ranges on costs and delays that will be used in the sensitivity analysis and will provide insight into other parameters that may be required in the optimization model. It can be used to test the effectiveness of a decision algorithm for cargo inspection in the context of the overall port operation.

*New Optimization Models*

One of the key elements in this project is the development of an optimization model for finding optimal and near optimal binary decision trees. Several approaches are being explored at this stage of the project.

<u>Optimization Models Based on Network Flows</u>

The Los Alamos method of finding the optimal decision tree consisted of an exhaustive search. This method does not scale, in fact, it is computationally infeasible for even five sensors. One approach is to model the problem as a minimum cost network flow problem with some additional linear constraints. For this type of linear programming problem many computational tools are available, and these are typically solvable within fractions of a second of CPU time for up to hundreds of nodes. This efficiency will allow us to try several parameter settings, and make a detailed computational analysis of the effects of various cost factors, capacities, etc.

Description of model without sensor reading history

In our first version of the optimization model, we assume that containers are directed to the next sensor based only on the current sensor reading, regardless of what readings a container may have had at prior sensors. This is mathematically the simplest case to consider. Since the resulting mathematical model is very robust and highly scalable and the execution of this model is computationally very cheap, it may provide us with valuable information, including (i) a lower bound on expected inspection costs, (ii) an analysis of queuing/throughput, (iii) information on the most cost effective sequencing of sensors, (iv) and a measure of the added value coming from decisions based on inspection history. The model is still rich enough to be able to incorporate various cost factors and capacities, perhaps even beyond the scope of current practices.

We consider every sensor i to be a subgraph, consisting of nodes $u_i$, $v_i$ and $w_{ij}$ for $j=1,..,k_i$, and directed arcs $(u_i, v_i)$ and $(v_i, w_{ij})$ for $j=1,...,k_i$. The arc $(u_i, v_i)$ corresponds to the container flow arriving at sensor i, while the outgoing arcs $(v_i, w_{ij})$ for $j=1,..., k_i$ correspond to the outgoing container flow of containers with different ranges of sensor readings (here $k_i$ is the number of different levels of sensor readings which we want to distinguish at sensor i.) For each sensor we can introduce a high number of different ranges, due to the computationally robust nature of this model. We also introduce an additional node s, representing all the containers arriving at the port, and two terminal nodes $t_{OK}$ and $t_{CHK}$, the first representing containers which pass the inspection with the decision that they are not suspected of containing illicit material and are moved out of the inspection area, and the second representing those that will be opened and inspected manually (unstuffed). We add arcs $(s, u_i)$, $(w_{ij}, u_{i'})$ for all $i\neq i'$ and $j=1,...,k_i$, $(w_{ij}, t_{OK})$ and $(w_{ij}, t_{CHK})$ for all $j=1,...,k_i$, and $(t_{OK},s)$ and $(t_{CHK},s)$.

To every arc in this network we associate a variable representing the container flow crossing this arc, a cost function and a capacity. For each arc of the network we require that the flow not exceed the capacity. Furthermore, for each node we require flow conservation, i.e., that the incoming flow is the same as the outgoing flow. For each sensor we set (normalized) threshold values $0=t_0 < t_1 < t_2 < \dots < t_{k_i-1} < t_{k_i}=1$, and pre-compute (using historical data, or a physical model for the sensor) parameters $a_{ij}$ and $b_{ij}$ representing the lower and upper bounds on the fractions of containers which will have a reading between the (normalized) thresholds $t_{j}$ and $t_{j-1}$, for $j=1,...,k_i$. Then, for each sensor i we require that the number of containers with readings between $t_{j}$ and $t_{j-1}$ must be between $a_{ij}$ of the number of containers at sensor i and $b_{ij}$ of the number of containers at sensor i.

We also have the constraint that the number of containers marked okay and the number of containers marked to be checked equals the total number of containers arriving at the inspection station. Subject to the constraints, we minimize the total cost of inspection

This model allows us to incorporate and individually adjust the queuing cost at sensor i; the inspection cost at sensor i for which the reading is in the jth range, $j=1,...,k_i$ (these may or may not be the same); the cost of false negatives; the cost of manual inspection (including the cost of false positives); the increment in the cost of false negatives resulting from a reading at sensor i that falls into the jth range.

Properties of optimal solutions for initial model

There are some very useful properties of a (basic) optimal solution for such a problem, which makes it possible to interpret the results as an inspection policy, that is, we can derive an optimal sequence for the sensors. For every sensor and every sensor reading range, there is an “optimal,” a “next best,” etc., decision, where to send the containers next, together with some upper bounds on how many containers we can send the “optimal,” a “next best,” etc., ways.

Note that if there are S sensors then the number of nodes in this model is linear in S and the sum of the number of threshold values at each sensor. Thus, this is a highly scalable model, in which we can assume a large number of different sensor ranges to fine tune the system.

Description of model with sensor reading history

In this model we assume that containers carry with them the sensor readings from the prior sensor stations, and that this “history” may influence the decisions made at the sensors that follow. We develop the very same model as before, with four important changes.

First, we fix an order of the sensors (e.g. the one derived from the historyless model) and assume that all containers will pass the sensors in the same order (with possible skips).

Second, all sensors will be included in several “copies” in this network. More precisely, instead of each sensor being labeled by a simple index i, we have each copy of the sensor labeled by a tuple carrying the history of prior readings. The tuples are of the form $(i, j_1,..., j_{i-1})$, representing the copy of sensor i which handles containers for which the first sensor reading belongs to the $j_1^{st}$ range, the second sensor reading belongs to the $j_2^{nd}$ range, etc. Third, we assume that all arcs (except $(t_{OK},s)$ and $(t_{CHK},s)$) are from left to right, s is the leftmost node, and $t_{OK}$ and $t_{CHK}$ are the rightmost nodes, and we include arcs $(w_{(i,j_1,...,j_{i-1}),j}, u_{(i',j_1',...,j_{i'-1}')})$ only for $i<i'$ and $j_1=j_1', ..., j_{i-1}=j_{i-1}'$.

Fourth, we need to change the capacity constraints, since a sensor's capacity applies to all containers, regardless of their history. We shall also need to add individual nonnegativity constraints for all arcs, that is, the flow in never negative on any arc.

### Properties of the model with history

In this model, we can have the cost of inspection as well as the distribution of sensor readings dependant on container history. On the negative side, the number of nodes depends on product of the number of threshold values rather than on the sum. This implies that computationally this model does not scale that well. It is therefore important to derive the right order of the sensors, as well as the right (small) number of threshold values for sensor readings. The historyless model can provide us with information regarding these settings. On the positive side, such models with linear costs can numerically be handled up to millions of nodes. Hence, for 5-6 sensors, we can still have 5-10 different threshold values, which is perhaps more than enough for practical purposes.

Perhaps the most important note is that this computationally much more involved model provides additional information, compared to the historyless model **only if** the distribution of sensor readings truly depends on the containers' history of readings, and/or if the expected incremental cost of false negatives truly depends on the inspection sequence. If this is not the case, or if we do not have enough information to derive different distributions for the different copies of sensors in this model, then there is no added benefit from using this model as opposed to the much simpler historyless case.

## Optimum Thresholds Model

In this approach we decompose the port-of-entry inspection problem into two sub-problems. The first problem deals with the determination of the optimum sequence of inspections or the structure of the inspection decision tree in order to achieve the minimum expected inspection cost. The second problem deals with the determination of the optimum thresholds of the sensors at inspection stations so as to minimize the cost associated with false positives (false alarm, which results in additional manual inspection) and false negatives (failure to identify illicit materials or weapons). The first problem can be formulated and investigated using approaches parallel to those used in the optimal sequential inspection procedure for reliability systems as described by Butterworth (1972), Halpern (1974, 1977), Ben-Dov (1981), Cox *et al.* (1989), Cox *et al.* (1996), and Azaiez *et al.* (2004). After the sequence of inspections and the structure of the decision tree are determined, we determine the optimum thresholds of the sensors at inspection stations. We first address the optimum thresholds problem.

### Description of the optimum thresholds problem

The inspection of a container at the port-of-entry is performed sequentially at inspection stations that form the inspection system, which is described as follows:

1. There are $n$ inspection stations in the system; each station is used to identify one attribute of the container being inspected. Let $x_i$ be the state of the $i^{\text{th}}$ attribute, $D_i$ be the decision of $i^{\text{th}}$ station after inspection. For simplicity we assume that there are only two states, 1 and 0, for presence or absence of the attribute respectively.

2. The classification of each container is thought of as a decision function $F$ that assigns to each string of attributes a class. We focus on the case where there are only two classes, 0 and 1, i.e. $F(x_1, x_2, \ldots, x_n) = 0$ (negative) means there is no suspicion that the container has illicit material and $F(x_1, x_2, \ldots, x_n) = 1$ (positive) means that additional manual inspection is required (unstuffing). The decision as to the classification can be made before all attribute values are available as shown in Figure 1, which we repeat here.

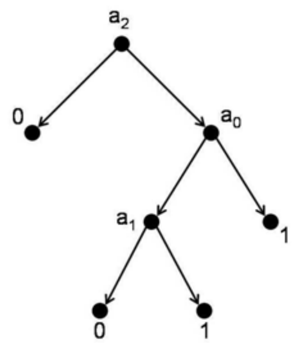


Figure 1. Binary Decision Tree

3. Let $c_i$ be the unit cost associated with the inspection at inspection station $i$, $i = 1, 2, \ldots, n$, and let $c_{FP}$ be the unit cost associated with false positive (the cost of additional inspection), and $c_{FN}$ be the unit cost associated with false negative.

4. From inspection history, we have information about the probability that the attribute $i$ is 0 or 1. Let $p_i$ be the probability that the attribute $i$ is 0 and $q_i$ be the probability that the attribute $i$ is 1, e.g.

$$p_i = P(x_i = 0),$$
$$q_i = P(x_i = 1).$$

5. For each inspection station $i$, the sensor at the station has possible discrete thresholds $T_{ij}, j = 1, 2, \ldots, k_i$ and $k_i \geq 1$. The decision of station $i$ is made based on the following criterion:

$$D_{ij} = \begin{cases} 1, & \text{if } r_i > T_{ij} \\ 0, & \text{if } r_i \leq T_{ij} \end{cases}$$

Where $r_i$ is the sensor measurement at station $i$.

For each threshold, we have following probabilities: $q_{ij} = P(D_{ij} = 1)$ and $p_{ij} = P(D_{ij} = 0)$. For every decision made based on the sensor measurement, there are potential decision errors. Let $\alpha_{ij}^1$ be the probability of a false positive from decision $D_{ij} = 1$, i.e. $\alpha_{ij}^1 = P(D_{ij} = 1 \mid x_i = 0)$. Similarly, we have the probability of a false negative $\alpha_{ij}^0 = P(D_{ij} = 0 \mid x_i = 1)$ at station $i$.

6. The classification of a container is based on all or partial decisions at inspection stations and the Boolean function $F(x_1, x_2, \ldots, x_n)$, i.e. $D = F(D_1, D_2, \ldots, D_n)$, which means that we may

not need all the values of $D_i$ 's to make the final decision as to classification. Here $D_i$ is the appropriate value of $D_{ij}$.

7. The probability of false positive of the final classification of the container status is $\alpha^1 = P(D=1 \mid F=0)$; and the probability of false negative of the final classification is $\alpha^0 = P(D=0 \mid F=1)$. Where F here indicates the true classification.
8. The total cost associated with false positive and false negative is $C_F = c_{FP}\alpha^1 + c_{FN}\alpha^0$.
9. The objective of the problem is to minimize $C_F$ and the decision variables are the thresholds of the sensors at inspection stations $T_i$, $i = 1, 2, \ldots, n$, where

$$T_i \subset \{T_{ij} : j = 1, 2, \ldots, k_i\}, \qquad \text{for all } i = 1, 2, \ldots, n.$$

The calculation of $\alpha^1$ and $\alpha^0$ depends on the Boolean function $F(x_1, x_2, \ldots, x_n)$. We assume that the inspections and decisions made based on the sensor measurements at different stations are independent.

## References


Azaiez, N., Bier, V. M., Optimal resource allocation for security in reliability systems, *CREAT Report*, 2004.

Ben-Dov, Y., Optimal testing procedures for special structures of coherent systems, *Management Science*, **27** (1981), 1410-1420.

Butterworth, R. W., Some reliability fault-testing models, *Operations Research*, **20** (1972), 335-343.

Cox L., Qiu, Y., Kuehner, W., Heuristic least-cost computation of discrete classification functions with uncertain argument values, *Annals of Operations Research,* **21** (1989), 1-30.

Cox, L., Chiu, S., Sun, X., Least-cost failure diagnosis in uncertain reliability systems, *Reliability Engineering and System Safety*, **54** (1996), 203-216.

Halpern, J., Fault-testing of a k-out-n system, *Operations Research*, **22** (1974), 1267-1271.

Halpern, J., The sequential covering problem under uncertainty, *INFOR*, **15** (1977), 76-93.


## Software and Hardware Prototypes

Software prototypes are under development.

## Papers and Talks

Roberts, F.S., “Decision support algorithms for port-of-entry inspection,” in *Working Together: Research & Development Partnerships in Homeland Security, Proceedings of DHS/IEEE Conference*, Boston, 2005.
http://www.dimacs.rutgers.edu/People/Staff/froberts/PortofEntryPaper5-5-05rev2.doc

Roberts, F.S., “Decision support algorithms for port-of-entry onspection,” DHS Meeting on "Research and Development Partnerships in Homeland Security", Boston, April 2005.
http://www.dimacs.rutgers.edu/People/Staff/froberts/DHSBostonPorts4-19-05rev.ppt

We are planning a special session at the Institute for Operations Research and the Management Sciences (INFORMS) annual meeting, November 13 – 16, 2005. The titles, authors and abstracts of the papers that will be presented at the session are:

*Data Mining Complex Sensor Simulations for Optimal Security*
Paul B. Kantor, SCILS, Rutgers University
Alexander Kogan, Accounting & Information Systems, Rutgers University
Brenda Latka, DIMACS, Rutgers University
Rick Mammone, CAIP, Rutgers University
Phillip Stroud, Los Alamos National Laboratory

Rigorous calculation of screening schemes produces a rich database with over 10,000 instances of parameters settings and expected performance levels. These are mined to extract heuristic relations that will be useful when the number of sensors makes direct exploration computationally intractable.

*Decision Support Algorithms for Port-of-Entry Inspection*
Fred S. Roberts, DIMACS, Rutgers University
Phillip D. Stroud, Los Alamos National Laboratory

We describe approaches to efficiently discover smuggling attempts at U.S. ports of entry. We use a sequential decision making model: Containers are routed to different tests depending on outcomes of earlier tests. We describe ways to find inspection schemes minimizing cost, including "cost" of false positives and negatives.

*Modeling Cargo Flow Security Operations in Marine Ports*
Tayfur Altiok, Industrial and Systems Engineering, Rutgers University
Kevin Saeger, Los Alamos National Laboratory
Benjamin Melamed, Rutgers Business School, Rutgers University

We model cargo security operations in a container port including handling, storage and inspection. Here, vessel arrival processes, vessel unloading operations and the inspection process are critical modeling components. The impact of the arrival process and the inspection ratio on the overall marine port operation will be discussed.

*Network Models for Sequential Diagnosis*
Endre Boros, RUTCOR, Rutgers University

E. A. Elsayed, Industrial and Systems Engineering, Rutgers University
Liliya Fedzhora, RUTCOR, Rutgers University
Hao Zang, ISE, Rutgers University

We develop a network optimization type linear programming model for sequential container inspection. In this model we minimize the total "cost," including expenses arising from storage, inspection and delays, as well as the estimated cost incurred by false positives and negatives.

## Current Students and Recent Graduates Supported by ONR

Oelim Alpinar, Rutgers University,
Lilya Fedzhora, Rutgers University, RUTCOR
Abdullah Karaman, Rutgers University
Francesco Longo, University of Calabria, Italy
Hao Zhang, Rutgers University, Industrial and Systems Engineering