Optimization Problems for Port-of-Entry Detection Systems

Endre Boros, Elsayed Elsayed, Paul Kantor, Fred Roberts, Minge Xie[1]

Rutgers University, Piscataway, NJ 08854

Abstract

The problem of container inspection at ports-of-entry is formulated in several different ways as an optimization problem. Data generated from different analytical methods, x-ray detectors, gamma-ray detectors and other sensors used for the detection of chemical, biological, radiological, nuclear, explosive, and other illicit agents are often relied upon to make critical decisions with regard to the nature of containers presented for possible inspection and the appropriate response mechanism. Several important questions related to the utilization and coordination of multiple sensors for container inspection are discussed. New and efficient algorithms for finding the best inspection strategy, including the optimal sequencing of sensors and optimal assignment of thresholds for interpreting sensor readings, are described. Models and algorithms that can be used by decision makers, allowing them to minimize expected cost of inspection, minimize inspection errors (both false positives and false negatives), and/or maximize the throughput of containers, are outlined.

1. Introduction

Finding ways to intercept illicit materials, in particular weapons, destined for the U.S. via the maritime transportation system is an exceedingly difficult task. Practical complications of inspection approaches involve the negative economic impacts of surveillance activities, errors and inconsistencies in available data on shipping and import terminal facilities, and the tradeoffs between costs and potential risks, among others.

---

[1]All five authors were supported by ONR grant number N00014-05-1-0237 and NSF grant number NSFSES 05-18543 to Rutgers University. The authors also thank Sushil Mittal and Richard Hoshino for their helpful comments.

Until recently, even with increased budget and emphasis, and rapid development of modern technology, only a very small percentage of ships entering U.S. ports have had their cargo inspected. Thus there is a great need to improve the efficiency of the current inspection processes. There has been a series of attempts to develop algorithms that will help us to inspect for and intercept chemical, biological, radiological, nuclear, and explosive agents, as well as other illicit materials. Such algorithms need to be developed with the constraint that seaports are critical gateways for the movement of international commerce. More than 95 percent of our non–North American foreign trade arrives by ship. With "just-in-time" deliveries of goods, the expeditious flow of commerce through these ports is essential. Slowing the flow long enough to inspect either all or a statistically significant random selection of imports would be economically intolerable (Loy and Ross 2002).

Potential goals for container inspection include:

- Minimizing inspection errors, both false positive and false negative, subject to constraints on cost of inspections
- Maximizing inspection system "throughput"
- Minimizing total expected cost of inspection.

Here, false positive means a container that has no suspicious contents is rejected or goes through extensive manual examination and false negative means that a container that has suspicious contents is accepted. These criteria are inter-related and it is unlikely that they would be optimized by a single set of parameter values. Thus, we are dealing with a typical *multi-objective programming problem*, and one seeks ways to formulate this precisely and use methods of multi-objective programming to find solutions under different formulations.

As to the data itself, at present there are only two techniques for non-invasively "seeing" into a container, both involving projection and the resultant transmission or reflection of waves: 1) Electromagnetic (EM) waves (radio waves, light, x-rays, gamma rays, etc.) and 2) Material vibration waves (ultrasound). A variety of means exist for converting these waves into images suitable for human inspectors to interpret (James et al. 2002). In practice, there are only a small number of types of sensors used in port-of-entry inspection protocols. In part this is due to the reality that there are not many different types of inspections that can be carried out short of actually opening a container, and in part it is due to the fact that algorithmic approaches to container inspection run afoul of combinatorial explosions when more than a small number of sensors are used. We will describe new approaches to the container inspection problem that we can hope could lead to almost doubling the number of sensors whose outputs could be considered. Such methods should prepare us well for the time when additional practical detection methods are made available for port-of-entry inspections.

Although the techniques of "seeing" into containers are limited, various types of sensors, testing for the same types of things, are being used in the current practice of port-of-entry inspection. They range from hand-held or portable devices, to heavy equipment, and from counts to numerical measurements to shape (image) classifications. These sensors vary widely in cost, resolution and sensitivity, and they are evolving quickly with innovations leading to both significant decreases in costs and to many new features as well. We rely upon their measurements to make critical decisions with regard to the nature of the containers being inspected and to detect unwanted agents. Interpretation of the output from a sensor is not always straightforward and requires calibration. The interpretation may lead to accepting a container that contains undesirable material (false negative) or may lead to unnecessary inspection of a container that is acceptable (false positive), which results in delays and added cost. Clearly, interpretation of sensor readings has a direct effect on these errors and on the entire inspection process.

Typical techniques transform sensor readings into a real output, and generally thresholds are used to separate *suspicious containers* from *innocent looking* ones. There are many critical issues involved in determining what thresholds to use, such as noise (in the detector measurements), the sequencing of different sensors, the determination of the threshold levels of the sensors that minimize the probabilities of accepting undesirable containers and rejecting acceptable containers, and the minimization of system delays, among others. These issues must be considered simultaneously in configuring inspection and detection systems and research needs to address these threshold-related issues. We will describe a polyhedral approach to this problem, building on earlier work that formulated a large-scale linear programming model yielding optimal strategies for container inspection under certain assumptions, and we will also describe a dynamic programming approach and a new algorithm that combines the well-known optimization known as the Gradient Descent Method and Newton's Method.

2. Model Formulation

In the port-of-entry inspection process, we consider containers (entities) being off- loaded from ships. Containers are inspected and classified according to observations we make regarding their attributes. In the simplest case, each attribute is either "present" (1) or "absent" (0). There are several categories into which we seek to classify entities. In the simplest case, these are positive and negative, 1 or 0, with "0" designating entities that are considered "acceptable" and "1" designating entities that raise suspicion and require special treatment. After each observation, we either classify the entity as 0 or 1 or subject it to another inspection process. Thus, a container can be thought of as corresponding to a string of 0's and 1's, a binary string. Note that we might make a decision about a container before knowing all the terms in this string. The classification can be thought of as a *decision function* F that assigns to each binary string of attributes a category. In this paper, we focus on the case where there are two categories. Thus, F is a *Boolean function*. For instance, consider the Boolean function defined by $F(111) = 1$ and $F(x_1x_2x_3) = 0$ otherwise, where x_1, x_2, and x_3 are 0 or 1. This F is the function that classifies an entity as positive if and only if it has all of the attributes. Boolean functions provide the selection logic for an inspection scheme. If F is known, we seek to determine its value by testing the attributes one by one. In a typical case, the attributes are assumed to be independent. We shall describe models where the distribution of the attribute states is known and also where it is not. In particular, in the binary case, we consider the probability p_i that the i^{th} attribute takes on the value 0, and the probability $q_i = 1-p_i$ that it takes on the value 1. The inspection policy determines the order of testing the container's attributes. At any point, the inspection scheme might tell us to stop inspecting and decide the value of F based on the outcomes of inspections so far. We must make enough observations to allow us to classify the entity and decide whether to designate it as an entity that needs special handling or accept it. The notation we use and assumptions we make about a port-of-entry inspection system can be formalized 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^{th} attribute.

2. We think of observations (measurements) as taking place sequentially. After each observation, we either classify the entity being inspected or subject it to another inspection process. As noted above, the classification of each container is thought of as a decision function F that assigns to each string of attribute values a class C. We focus on the case where there are only two classes, $C = 0$ and $C = 1$, i.e. $F(x_1x_2...x_n) = 0$ means the negative class and that there is no suspicion with the container, and $F(x_1x_2...x_n) = 1$ means the positive class and that additional special treatment such as manual inspection of the container is required.

3. We assume that we gather information about the probability that the attribute i is present or absent for a group of containers, which are imported by the same company or from the same origin at the same time. This information can be obtained, for example, from inspection history or other sources. As noted, we use p_i to be the probability that the attribute i is absent and q_i to be the probability that the attribute i is present, i.e., $p_i = P(x_i = 0)$, $q_i = P(x_i = 1) = 1 - p_i$, for $i = 1, 2, \ldots, n$.

4. The inspection stations are assumed at first to be perfect, which means that the true attributes can always be identified at the corresponding inspection stations without any error. However, later we will assume that the sensor readings are either continuous or categorical measures and consider errors in identification. In particular once "thresholds" are included in our model, we will consider the problem with the possibility that resetting thresholds will lower the probability of misclassifying a "good" container as needing further inspection or a "bad" container as being acceptable.

5. The port-of-entry inspection problem involves three kinds of costs (Stroud and Saeger (2003)): costs of making observations, costs of false positives, and costs of false negatives. There are many possible ways to calculate the cost of obtaining a sensor

reading. For instance, we can break down the cost into two components: unit cost and fixed cost. The *unit cost* is just how much it costs to use the sensor to inspect one item, and the *fixed cost* is the cost of the purchase and deployment of the sensor itself. In many cases, the primary cost is the unit cost since many inspections are very labor intensive. The fixed cost is usually a constant and often does not contribute in optimization equations, so for simplicity we will disregard it. The *inspection cost* c_i is basically the expected cost of making observations for an average container. Stroud and Saeger (personal communication) also assign costs to false positives and false negatives, though especially the latter is very difficult to do. The cost of the former is essentially the cost of labor to manually inspect the container, though more sophisticated analysis would take into consideration the economic cost of delays in shipping. Stroud and Saeger then seek to minimize the sum of inspection costs and false positive and negative costs. Some other researchers also consider such a sum of costs, whereas others concentrate on inspection costs only.

The whole inspection process can be represented as a binary decision tree (BDT), the nodes of which correspond to sensors or decisions (0 or 1), and branches of which correspond to the decision we make after learning the sensors' readings. For concreteness, we think of a BDT as having a left branch from a sensor node meaning that the sensor gives an outcome 0, and a right branch meaning that it gives an outcome 1. Decision nodes are all 0 or 1 and are leaves of the tree, i.e., have no outgoing branches. The container inspection problem was considered by Stroud and Saeger (2003), who provided a complete enumeration of all possible binary decision trees built from no more than 4 sensors and corresponding to Boolean functions satisfying two assumptions: completeness (all variables are needed) and monotonicity (finding a more suspicious reading on any one sensor must not decrease the probability that the container itself should be inspected). A Boolean function F is *monotone* if given two strings $x_1x_2\ldots x_n$, $y_1y_2\ldots y_n$ with $x_i \geq y_i$ for all i, then $F(x_1x_2\ldots x_n) \geq F(y_1y_2\ldots y_n)$. F is *incomplete* if it can be calculated by finding at most n-1 attributes and knowing the value of the input string on those attributes. Stroud and Saeger then modeled sensor outcomes as following independent Gaussian distributions, one for "good" containers and one for "bad" ones, and studied thresholds for container readings above which an inspection outcome of 1 would be reported. They computed approximately optimal thresholds for each sensor (one threshold per sensor) by a non-linear grid-search approach. Their method cannot be extended to more than 4 sensors, due to combinatorial explosion of the number of binary decision trees. For example, for *n* = 4 different sensors as nodes, there are 114 complete, monotone Boolean functions and 11,808 distinct corresponding BDTs. Compare this with 1,079,779,602 BDTs for all Boolean functions, which explains the need for special assumptions such as completeness and monotonicity. For *n* = 5, there are 6,894 complete, monotone Boolean functions and 263,515,920 corresponding BDTs. Even worse: compare 5×10^{18} BDTs corresponding to all Boolean functions. (Counts are from Stroud-Saeger.)

Among important extensions of this work are: doing sensitivity analysis on their results (Anand, et al (2006)); making more restrictive assumptions about the nature of the binary decision trees (Zhang, Schroepfer, and Elsayed (2006)); introducing a new and promising polyhedral approach (Boros, et al. (2006)); and broadening the class of binary decision trees considered in order to introduce more computationally-efficient search procedures for optimal inspection strategies (Madigan, et al. (2007)).

3. The Polyhedral Formulation

Boros, et al. (2006) extended the work of Stroud and Saeger (2003) and formulated a large-scale linear programming model yielding optimal strategies for container inspection. This model is based on a polyhedral description of *all decision trees* in the space of possible *container inspection histories*. The dimension of this space, while quite large, is an order of magnitude smaller than the number of decision trees. This formulation allowed them to incorporate both the problem of finding optimal decision trees and optimal threshold selection for each sensor into a single linear programming problem. The model can also accommodate budget limits, capacities, etc., and one can solve it to maximize the achievable detection rate. Boros, et al. have been able to solve this model for 4 sensors, and branching that allows up to 7 possibly different routing decisions at each sensor (in contrast to the binary routing solved by Stroud and Saeger (2003), and implicit in Boolean models) in a few minutes of CPU time, on a standard desktop PC. They are also able to run the model for as many as 7 sensors, when they allow only binary decisions, as in Stroud and Saeger (2003). A challenge is to extend and improve this polyhedral formulation.

To describe the polyhedral approach in more detail, note that to containers passing through the inspection station, one can associate a "history," which is the sequence of pairs naming a sensor and its reading. Readings at a given sensor fall within a range of values, and we can partition this range into several classes. For instance, if we have sensors *a,b,c* and we present a grid of partitions for these sensors, consisting of, say, 2 possible readings for sensor *a*, 3 for sensor b, and 5 for sensor *c*, then sequences like (*c*,4;*a*,1;*b*,3), (*b*,1;*a*,2), etc. are all possible histories. (In the former, we mean that sensor *c* has a reading in its fourth region, *a* in its first region, and *b* in its third region). Let us note that the number of possible histories is smaller by a full exponential order (that is, it is only singly exponential instead of being doubly exponential) than the number of decision trees (utilizing the same branching parameters and same set of sensors). Denoting our terminal decisions by 0 and 1, we can define decision variables $y(\mathrm{H},0)$ and $y(\mathrm{H},1)$ representing, respectively, the (unknown) fraction of containers having history H and final decisions 0 (ok) or 1 (check further with special handling). Clearly, the equality $\Sigma_{\mathrm{H,D}} y(\mathrm{H,D}) = 1$ must hold, where the summation is taken over all histories H and final decisions D. Let us further call an initial segment of a history a pre-history, and denote by P<H the statement that P is an initial segment in H. With this notation we can describe a set of consistency equalities corresponding to the knowledge that at each sensor *s* and for each reading range *r* at this sensor the fraction of containers receiving reading *r* is *g(s,r)* out of all containers that arrive at *s* with the same pre-history (where *g(s,r)* are considered as input parameters, derivable from a physical sensor model, or from past data, etc.). In other words, we must have:

$$g(s,r) * \Sigma_{\{\mathrm{H,D} \mid (\mathrm{K},s) < \mathrm{H}\}}\, y(\mathrm{H,D}) = \Sigma_{\{\mathrm{H,D} \mid (\mathrm{K},s,r) < \mathrm{H}\}}\, y(\mathrm{H,D})$$

for all sensors s, for all pre-histories K not involving sensor *s*, and for all reading ranges *r* possible at sensor *s*.

Boros, et al. (2006) prove that the above set of equalities, together with the nonnegativity of the decision variables, describes a polytope, the vertices of which correspond to decision trees. Since many other characteristics of an inspection policy, such as unit inspection cost, detection rate, sensor loads, etc., can all be expressed as linear functions of the decision variables, they are able to formulate various problems related to container inspection as linear programs (see more in Boros, et al, 2006)).

The preliminary results of Boros, et al. demonstrate that (1) allowing multi-fold discretization of sensor readings provides substantially better results (higher detection rate at a lower cost) with the same set of sensors; (2) optimal strategies in fact involve a mixture of several decision trees, rather than just a *single best* decision tree. There are various ways to extend this work.

3.1 Extension to more sensors and variables

The current formulation involves a quite large system of linear inequalities, which leads to numerical instability, and with this there are challenges in expanding to a larger number of sensors and larger branching factors at each of these sensors. One could exploit the algebraic structure of this system, and reformulate it to yield a numerically more stable formulation. Developing a column generation technique for this type of problem might also help. Extending the method to systems involving up to 7 to 8 sensors, and as many as 5 to 10 different possible decisions at each of these sensors, would allow us to evaluate the effects of new technologies, budget changes, etc., in realistic sizes.

3.2 Selection of threshold grids or partitions

As noted in Section 2, another important area for further research is the selection of threshold grids. There are many important issues involved in determining what thresholds to use, including sequencing of sensors, the desire to minimize false positive or false negative results, the need to minimize system delays for commerce, and problems caused by noise and errors in measurement. Stroud and Saeger (2003) developed a grid-optimization technique for selecting the thresholds. This is incorporated into the large-scale linear programming formulation of Boros, et al. (2006) by choosing a grid of possible threshold values, and letting the optimization model select which ones of them are actually utilized in an optimal inspection strategy. The large-scale linear programming formulation can be solved efficiently by standard LP packages for up to 5 sensors and 5 to 7 thresholds per sensor, even if the decision trees involved are not binary. As noted above, the results of Boros, et al. (2006) show that multiple thresholds provide substantial improvement at no added cost. They also show that the optimal solution is not a single decision tree, but a convex mixture of several decision trees. Of course, this approach is sensitive to the selection of the initial grid. In particular, computational complexity forces us to start with a relatively coarse grid initially. A challenge is to develop a better method for selecting initial grids. One could use the fact that decisions based on sensor readings do not depend on the actual reading, but rather on the odds ratio of dangerous to innocent container contents corresponding to the actual reading (Kushner & Pacut, 1982). One could also develop an iterative approach for refining the partitioning of sensor readings, so that a k-fold partition at a given sensor is focused on the most critical sensor readings, as revealed by the model with k-1 partitions.

4. Different Allowable Topologies for the Decision Trees

The port-of-entry inspection problem can be decomposed into two sub-problems. The first problem deals with the determination of the optimum sequence of inspection or the structure of the inspection decision tree in order to achieve the minimum expected inspection cost; the second problem with the determination of the optimum thresholds of the sensors at inspection stations so as to minimize the cost associated with false positive (false alarm, which results in additional manual inspection) and false negative (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 sequences of inspection and the structure of the inspection decision tree are determined, one determines the optimum thresholds of the sensors at inspection stations.

The Stroud-Saeger (2003) approach to port-of-entry inspection concentrated on limiting the number of decision trees we must consider by making special assumptions about the underlying Boolean function, i.e., that it is complete and monotone. An alternative approach is to make special assumptions about the topology of the decision tree.

4.1. Series, parallel, and other topologies

A topology-limiting approach can take advantage of the literature of systems reliability. Here, one considers a multi-component system, where the state of all the components is described by a vector $(x_1,x_2,\ldots,x_n)$, and the operational state of the system is a function $F(x_1,x_2,\ldots,x_n)$ of the states of its components. (In our earlier discussion, we spoke of strings $x_1x_2\ldots x_n$ rather than vectors, but the distinction is not important.) In the so-called testing or diagnosis problem, we want to learn the state of the system (the value of F) by testing some of the components (learning the value of some of the variables x_i). It is assumed that we know the cost c_i of testing component x_i, and we also know the distribution of the values of x_i, for all components i=1,..,n. The problem is to determine that sequence of tests that minimizes the total expected cost. A simplified version considers only expected cost of inspections, disregarding costs of false positives and negatives. The word "sequence" used here is in fact misleading, since, as we have noted, the real testing strategy is in fact a decision tree, in which the next component to be tested depends on the values we learned for the components tested earlier. Such problems arise in fault testing in systems engineering, medical diagnosis, search problems in data bases, and even in quiz shows (see e.g., Butterworth (1972), Chang and Slagle (1971), Duffuaa and Raouf (1990), Greiner (1990), Joyce (1971), Kadane (1969), Kowalski (1969, 1972), Nilsson (1971), Pohl (1971)). Numerous papers in the extensive literature have considered this problem in one or another special case.

For instance, consider series or parallel systems, i.e., those in which components have only working or failing states, and the function F is a simple conjunction or disjunction of its components. More precisely, a *series Boolean function* is a decision function F that assigns the container class "1" if any of the attributes is present, i.e., if $x_i = 1$ for any $i \in \{1, 2, \ldots, n\}$, and a *parallel Boolean function* is a decision function F that assigns the container the class "1" if all of the attributes are present, i.e., if $x_i = 1$ for all $i \in \{1, 2, \ldots, n\}$.

Consider a container with n independent attributes and described by a series Boolean function. Let the inspection procedure be such that attribute $i+1$ is inspected only if attribute i is found absent, for all attributes $i = 1, 2, \ldots, n-1$. If attribute i is present, the inspection of further attributes is halted and the container is subject to special handling/manual inspection. In this situation, the following result holds: For a series Boolean decision function, inspecting attributes $i = 1, 2, \ldots, n$ in sequential order is optimum, i.e., minimizes total expected inspection cost, if and only if:

$$c_1 / q_1 \leq c_2 / q_2 \leq \ldots c_n / q_n .$$

In this case, the expected inspection cost is given by

$$C = c_1 + \sum_{i=2}^{n} [\prod_{j=1}^{i-1} p_j] \, c_i .$$

In other words, the optimal strategy is to test the components in the order of the ratio c_i/Prob(x_i=1). This strategy is dependent on the sensor configuration. (See e.g., Mitten (1960), Butterworth (1972), Natarajan (1986), Alidaee (1994) and many others for proofs.)

It is natural to ask: How can this be generalized for different Boolean decision functions other than series? For more complex *k*-out-of-*n* systems, the optimal strategy was determined by Chang, Shi and Fuchs (1990). Series-parallel systems (i.e., systems in which components can hierarchically be grouped into series or parallel subsystems) are also frequently considered in the literature. One natural idea is to test subsystems one-by one, without interruption, in the order of their cost/probability ratios. This was proposed by Joyce (1971), and was claimed (mistakenly) to be optimal for series-parallel systems. Ben-Dov (1981) showed that this idea is indeed optimal for simple, depth-2 series-parallel systems, while Natarajan (1986) proved that it is optimal within those strategies that test subsystems without any interruption. Boros and Ünlüyurt (2000) showed that this algorithm can unfortunately fall short of the optimal value, and by an arbitrary factor, for series-parallel systems of depth-3. A generalization was also proposed, and was shown to be optimal for the very special case of systems composed of identical components (see

Ünlüyurt (2005) for a survey on this topic). Zhang, Schroepfer, and Elsayed (2006) have developed general total cost of inspection equations for *n* sensors in series and parallel configurations. It is natural to seek to develop similar equations for other configurations such as series-parallel, parallel-series, *k*-out-of-*n* systems and consecutive *k*-out-of-*n* systems. The approach for other complex configurations such as networks of sensors can be based on either a path-tracing approach or cut sets (minimum cut sets) depending on the characteristics of the sensor network. It is likely going to be difficult to obtain closed form expressions but the derived equations can be numerically evaluated through efficient algorithms.

4.2. Optimal threshold setting

In Section 3.2, we discussed a polyhedral approach to optimal threshold setting. The problems of optimal threshold setting become much more complex and difficult to solve for a larger number of sensors. An alternative approach to determining threshold levels involves a simplifying assumption about the tree topology. Assuming a "series" topology (looking at one sensor at a time in a fixed order), one can first determine an optimal sequence of sensors. Once an optimum sequencing of sensors is obtained, the threshold level problem is then formulated. Zhang, Schroepfer and Elsayed (2006) have used a complete enumeration approach to determine the optimum sequence of inspection stations and the corresponding sensors' threshold levels to solve problems with up to three sensors in series and parallel systems. The computational time increases exponentially as the number of sensors increases. They are developing efficient algorithms for solving different sensor configurations and hope to be able to utilize the topology to obtain tractable solutions for systems having 6 to 7 sensors without resorting to complete enumeration, through a combinatorial optimization algorithm based on a dynamic programming approach.

4.3. Unknown Boolean function F

A similar, though very different problem, arises in the study of systems reliability when we do not explicitly know the system function *F*, but we know a way to determine for any proposed function $G(x_1,\ldots,x_n)$ the probability that it describes the true state of our system (we call this probability the detection rate). This variant arises in several applications, including medical diagnosis (Greiner (1990)), organizing call centers (Cox, Qiu, and Kuehner (1989)), and container inspections at ports (Stroud and Saeger (2003)).

There are several possible ways to formulate problems in this setting (still assuming that we know the cost of testing and probability distributions of the variables). One problem is to find that function *G* for which the detection rate is the highest, and then to determine the optimal decision tree for "testing its components" (that is, determining the actual value of each of the components) to minimize expected expenses. A more general problem is to consider all of the functions *G* for which the detection rate is above a given threshold, and minimize the expected testing cost among all these. This particular formulation was considered by Stroud and Saeger (2003).

Stroud and Saeger (2003) solved the above problems at once, by enumerating all the possible decision trees representing all possible complete, monotone Boolean functions. The difficulty with this approach is that the number of decision trees grows doubly-exponentially with the number of sensors. In particular, they demonstrated that these problems can be handled for up to 4 sensors, but that this approach is not feasible for 5 or more sensors, with current computing technology, due to the intensity of the combinatorial explosion.

Both of the above problem formulations are further complicated in practice by the fact that testing the components typically yields a result on a continuous scale, not a discrete or binary scale. The solution must discretize this (that is, assign some meaning to the readings), changing thus implicitly the functions (or sets of functions) considered by the model. For instance, if we decide to binarize all readings, then one possibility is to use thresholds that divide the range of readings into two, and this leads to the problem of finding the thresholds for which the optimum in the previously considered cases (whichever model we use) is the best. One approach to this problem is to bisect the range of the threshold into two equal regions and then select two threshold levels, the first in the middle of the first region and the second in the middle of the second region. One can then estimate the minimum cost for each and select the level corresponding to the lower cost. A promising approach is to repeat the process by bisecting the range between the selected level and the original level and continue the process until the global minimum cost is obtained. The computational time required for this approach will of course be dependent on the initial range and the "fine" sectioning of the region.

The effectiveness of a partition having a given number of subsets *K* will be increased if the partition is not defined by thresholds placed on the natural sensor reading scale, but are defined to mark off connected portions of the ROC curve. This curve plots the conditional probability of sending a dangerous container for inspection as a function of the probability that a harmless container will be unnecessarily inspected. This curve rises smoothly as the required posterior odds that a container is "bad" are lowered. With complex sensors, the natural sensor readings corresponding to a particular portion of the ROC curve may be quite widely separated. On the other hand, statistical theory tells us that such a portion should be treated in the same or very similar ways. Therefore one can anticipate that defining partitions in terms of the ROC curve, rather than in terms of the natural sensor reading, will increase the power of an inspection scheme for any given number of partitions *K*

4.4. Complete, Monotone Binary Decision Trees

Stroud and Saeger (2003) formulate the port-of-entry inspection problem as a sequential decision making problem that involves finding an optimal (least cost) binary decision function. They reformulate this by considering the possible binary decision trees that correspond to that decision function, and seek through the space of possible binary decision trees to find least cost trees realizing each function. The problem becomes rapidly intractable unless special assumptions are made about the binary decision function. As noted earlier, Stroud and Saeger limit their analysis to complete, monotone binary decision functions. They enumerate all complete, monotone Boolean functions and then calculate the least expensive corresponding BDTs under assumptions about various costs associated with the trees. Their method is practical for n up to 4, but not $n = 5$. The problem is exacerbated by the number of BDTs (see counts in Section 2).

Madigan, et al. (2007) generalized the notion of complete, monotonic Boolean functions and defined notions of complete and monotonic binary decision trees. They developed a search algorithm that identifies (locally) optimum (least cost) complete, monotonic binary decision trees that is more efficient than the method developed by Stroud and Saeger, and makes it possible to analyze trees of at least 5 sensors (types of tests). The search method is based on a notion of neighborhood in the space of complete, monotonic binary decision trees, built on work of Chipman, George, and McCullough (1998a,b) and Miglio and Soffritti (2004).

5. **Multi-Objective Programming Approaches**

As mentioned earlier, in algorithmic approaches to inspection at ports of entry, we hope to find the optimal design of (sensor) system configuration and the best sets of threshold levels that can achieve a variety of objectives, such as maximizing inspection system throughput, minimizing the expected cost of inspection per container, and minimizing inspection errors, including both false positive and false negative. These objectives or criteria are interrelated. It is unlikely that they would be optimized by the same alternative parameter values, and there exists some trade-off between the criteria. It is a typical *multi-objective optimization* problem (see, for instance, Eschenauer, et al. (1990), Statnikov and Matuso (1995), Fonseca and Fleming (1998a, b), and Leung and Wang (2000), among others). Depending on the application, one can formulate different multi-objective algorithms for the problem. In general, there may be a large number or infinite number of optimal solutions (e.g., optimal sets of threshold values, etc.), in the sense of *Pareto-optimality*. It is desirable to find as many (optimal) solutions as possible in order to provide more choices to decision makers.

The multi-objective problem is almost always solved by combining the multiple objectives into one scalar objective whose solution is a Pareto optimal point for the original problem. Most algorithms for the problem have been developed in combination with techniques, such as minimizing/maximizing weighted sum of the objective functions, the goal programming method, a normal-boundary intersection method, and multilevel programming, among others. A particularly promising idea is to employ a combination of a goal programming method and a modified method of using weighted sums of the objective functions.

The *goal programming* method (Scniederjans 1995, Jones and Tamiz 2002) is a branch of multiple objective programming. In the goal programming approach, we optimize one objective while constraining the remaining objectives to be less than given target values. One can, for example, set constraints on the inspection cost, false positive rate and false negative rate, and maximize the objective function of throughput. An alternative approach is to use a modified weighted sum approach (a hybrid version of goal programming and the weighted sum approach), where we optimize, for example, a *fitness function* (i.e., weighted sum) of the objective functions under a constraint that both the false negative rate and false positive rate are controlled within their respective tolerance levels. In both the goal programming method and the modified weighted sum approach, by using the constraints, one can avoid the problem of how to subjectively choose appropriate weights for the false positives and false negatives, as discussed in Stroud and Saeger (2003). In a small system with only 2 to 4 sensors, this multi-objective programming problem can be solved by a grid search method similar to that discussed in Zhang, Schroepfer, and Elsayed (2006) and Stroud and Saeger (2003); see, also Sections 4.2 and 3.2. However, for a large system with more sensors, it is computationally challenging to use this simple enumeration method, since the computational time increases exponentially as the number of sensors increases. Hence, one needs to investigate alternative methods to simple enumeration.

To effectively solve the problem for a system with more sensors and a more complex configuration, one needs to develop more efficient algorithms as well as more objective ways of choosing the weights. This could be tackled by using a *genetic algorithm* in combination with studying statistical designs (such as uniform designs) on the weights and by systematically searching through the domain of parameters (i.e., threshold values and configurations) for optimal values. A similar algorithm was developed by Leung and Wang (1995), who used a uniform design to study the weights and search for optimal parameter values.

6. **Generalizations and Complications**

6.1 Allowing for stochastic dependence of the sensor readings

The research described above has all been done for the case of stochastically independent sensors. The model can be extended to consider stochastically dependent sensors. In the current models, sensor readings are taken to be independent of one another, although, of course, the branching actions at a later sensor can be quite different depending on the readings provided by an earlier sensor. The polyhedral method described in Sec. 3 can in principle be extended to deal with the case of stochastically *dependent* sensors, although the constraints required to represent the inspection effort needed to identify a particular region of the "sensor reading space" become rather more complicated.

6.2 Measurement Error and Optimal Threshold Setting

One important issue in modeling and computing sensor reading data and the objective functions is *measurement error.* The measurement error is also known as 'error-in-variables' (Carroll et al., 1984, Carroll *et al*., 1995, Kalbfleisch and Prentice 2002). Some work on port-of-entry inspection optimization algorithms has simplified the analysis by assuming precise and accurate measurements of sensors at inspection stations. This is a reasonable first approach. However, most measurements are subject to errors due to external sources, which include environmental conditions such as temperature fluctuations, humidity, dust particles, vibration, natural radiation; and internal sources that are dependent on the sensing component material and its ability to provide accurate measurements over a wide range of measurements as well as its manufacturing and assembly issues.

As noted earlier, Stroud and Saeger (2003) assume that both "good" containers and "bad" containers follow Gaussian distributions of readings. When readings exceed a threshold value, the outcome is viewed as suspicious. Based on whether or not an outcome is suspicious, the next test to be applied is determined or the container is finally categorized as "ok" or not. Of course, depending on the distribution of readings and the threshold values, errors can occur with certain probabilities. Anand, et al (2006), also using

Gaussian distributions, experiment with models for setting the thresholds so as to minimize total cost of the corresponding binary decision tree. Their approach involves incrementing individual sensor thresholds in fixed-size steps in an exhaustive search for threshold values that will minimize the expected cost of a binary decision tree. More efficient algorithms, based on combinations of the Gradient Descent Method and Newton's Method in optimization, are reported on in Madigan, et al. (2007).

Other approaches are also of interest. For example, building on work of Yi and Lawless (2006), one can aim to "extract" the "true" values of the measurements in order to determine accurate threshold levels and minimize the probability of misclassification of inspected items (containers) investigated using sensor observations. One can also apply analysis of the "meaningfulness" of conclusions from combinatorial optimization, developed in the measurement theory literature. (See Mahadev, Pekec, and Roberts (1997, 1998), Roberts (1990,1994,1999).) This approach analyzes the sensitivity of conclusions about optimality if parameters are measured on different kinds of scales and the scales change in permissible ways. Conditions are given for conclusions of optimality to be invariant under permissible changes of scale. A major application of this line of work has been to scheduling problems and one can seek to modify the methods of Mahadev, Pekec, and Roberts (1997, 1998) to apply to the port-of-entry inspection problem, which involves complications in the scheduling problems considered previously in the literature.

7. Case Study: Container Risk Scoring

The first step in the container inspection process actually starts outside the United States. To determine which containers are to be inspected, the United States Customs and Border Protection (CBP) uses a layered security strategy. One key element of this strategy is the Automated Targeting System (ATS). CBP uses ATS to review documentation, including electronic manifest information submitted by the ocean carriers on all arriving shipments, to help identify containers for additional inspection. CBP requires the carriers to submit manifest information 24 hours prior to a United States-bound sea container being loaded onto a vessel in a foreign port.

ATS is a complex mathematical model that uses weighted rules that assign a risk score to each arriving shipment in a container based on manifest information. The CBP officers then use these scores to help them make decisions on the extent of documentary review or physical inspection to be conducted (United States Government Accountability Office (2006)). This can be thought of as the first inspection test and the “sensor” is the risk scoring algorithm. Thus, in some sense, all trees start with the first sensor and this sensor is then not used again. It is not unreasonable to think of more sophisticated risk scoring algorithms that also involve sequential decision making, going to more detailed analysis of risk on the basis of initial risk scoring results. The Canadian government uses similar methods. The Canadian Border Services Agency (CBSA) uses an automatic electronic targeting system to risk-score each marine container arriving in Canada. As with ATS, this Canadian system has several dozen risk indicators, and a score/weight for each indicator.

As an application of how these two risk-assessment systems can be enhanced using the methods discussed in this paper, one could think of the following approach. Each risk indicator could be categorized into a specific "risk category," based on general themes such as an unreliable trade chain partner or suspicious commodity information. Based on various algorithms, one could create a real-valued function or "super rule" to assess the potential risk in each of these risk categories, returning a value between 0 and 1. Each of these functions could be thought of as a "sensor". The methods described in this paper, in particular the polyhedral methods described in Section 3, could provide both the American and Canadian container targeting systems with tools to determine an optimal multi-level decision tree to determine whether a container should be targeted or authorized to clear based on the results of these super rules. These and other ideas are under consideration.

8. **Closing Comments**

It is quite impressive how much work has already been done to develop formal methods for improving container inspection procedures using optimization techniques. Future work would benefit greatly from agreement upon a method for representing the cost-effectiveness of various sensor strategies. The problem always involves representation of risks, with assumptions about probabilities and costs (or utilities). It would be very useful to find ways to make precise the range of possible parameter values for things like inspection costs, loss of trade costs, etc. The problem is particularly complex since we are dealing with low probability, high consequence events. Measurement of both the probability and the cost of such events is very difficult and of course risk assessment for such events is a central challenge in homeland security.

The many parameters involved and many criteria for a "good" inspection strategy suggest that there will be more than one way to formulate the inspection problem. The uncertainties involved about parameter values suggest that even for each formulation, a significant amount of sensitivity analysis should be carried out. Because the problem of finding solutions to the formalizations of the inspection problem becomes dramatically more difficult as the number of tests available increases by even a small number, there is great need to develop methods for finding such solutions in increasingly efficient ways. The chances of extending present methodology to the number of potential tests we might have in the future will be small without some dramatically new approaches.

Because our ports handle billions of dollars of goods each year, even small improvements in efficiency of handling these goods, or of inspecting them while not disrupting trade, can be very important. Because the consequences of mistakes in inspection can be catastrophic, even small improvements in the likelihood of successfully preventing weapons of mass destruction from passing through our borders can also be important.

References

Alidaee, B. (1994) "Optimal ordering policy of a sequential model" *Journal of Optimization Theory and Applications,* vol. 83, pp.199-205.

Anand, S., Madigan, D., Mammone, R., Pathak, S., and Roberts, F.S. (2006), “Experimental analysis of sequential decision making algorithms for port of entry inspection procedures,” *Intelligence and Security Informatics, Proceedings of ISI-2006*, Lecture Notes in Computer Science #3975, Springer-Verlag, New York, 2006 (Mehrotra, D. Zeng, H. Chen, B. Thuraisingham, and F-X Wang (eds.)).

Ben-Dov, Y. (1981) "Optimal testing procedures for special structures of coherent systems" *Management Science*, vol. 27, no.12, pp. 1410-1420.

Boros, E., Fedzhora, L., Kantor, P.B., Saeger, K., and Stroud, P. (2006) “Large scale LP model for finding optimal container inspection strategies” submitted to *Naval Research Logistics Quarterly*. (Preprint at http://rutcor.rutgers.edu/pub/rrr/reports2006/26_2006.pdf.)

Boros, E., and Ünlüyurt, T (1999) Diagnosing Double Regular Systems. *Annals of Mathematics and Artificial Intelligence* 26(1-4) (1999) pp. 171-191.

Boros, E., and Ünlüyurt, T (2000) “Sequential testing of series-parallel systems of small depth” In: *OR Computing Tools for the New Millennium*, pp. 39-74. (Manuel Laguna and José Luis Gonzáles Velarde, eds., INFORMS Computing Society, Cancun, Mexico, January 5-7, 2000).

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

Carroll, R. J., Ruppert, D. and Stefanski, L.A., *Measurement error in nonlinear models*, Chapman & Hall, London, 1995.

Carroll, R. J., Spiegelman, C. H., Lan, K. K., Bailey, K. T. and Abbott, R. D. (1984), “On errors-in-variables for binary regression models,” *Biometrika*, 71, 19-25.

Chang, C.L. and Slagle, J.R. (1971) “An admissible and optimal algorithm for searching and-or graphs”, *Artif. Intell.* 2 117-128.

Chang, M., Shi, W. and Fuchs, W.K. (1990) “Optimal diagnosis procedures for k-out-of-n structures”, *IEEE Trans. Comput*. 39(4) 559-564.

Chipman, H.A., George, E.I., and McCulloch, R.E. (1998a), "Bayesian CART model search," *Journal of the American Statistical Association*, 93, 935-960.

Chipman, H.A., George, E.I., and McCulloch, R.E. (1998b), "Extracting representative tree models from a forest," working paper 98-07, Department of Statistics and Actuarial Science, University of Waterloo.

Cox Jr., L.A., Qiu, Y. and Kuehner, W. (1989) "Heuristic least-cost computation of discrete classification functions with uncertain argument values", *Ann. Oper*. Res. 21. 1-21.

Duffuaa, S.O. and Raouf, A. (1990) "An optimal sequence in multicharacteristics inspection", *J. Optim. Theory Appl*. 67(1) 79-87.

Formatted: German (Germany)

Eschenauer, H., Koski J., and Osyczka, A. (1990). "*Multicriteria Design Optimization."* Berlin, Springer-Verlag, 1990.

Fonseca, M. and Fleming, P.J. (1998a) "Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: Unified formulation," *IEEE Trans. Syst., Man. Cybern. A,* vol. 28, pp. 26-37.

Fonseca, M. and Fleming, P.J. (1998b) "Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part II: Application example," *IEEE Trans. Syst., Man. Cybern. A,* vol. 28, pp. 38-47.

Greiner, R. (1990) "Finding optimal derivation strategies in redundant knowledge bases", *Artif. Intell*. 50, 95-115.

Jones, D.F. and Tamiz, M. (2002) "*Goal programming in the period 1990-2000, in Multiple Criteria Optimization: State of the art annotated bibliographic surveys*," M. Ehrgott and X.Gandibleux (Eds.), 129-170. Kluwer.

Joyce, W.B. (1971) "Organizations of unsuccessful R&D projects" *IEEE Transactions on Engineering Management*, vol.18, no.2, pp 57-65.

Formatted: German (Germany)

Kadane, J.B. (1969) "Quiz show problems", *J. Math. Anal. Appl*. 27, 609--623.

Kalbfleisch, J. D. and Prentice, R. L., *The statistical analysis of failure time data* (second ed), Wiley, New York , 2002.

Kowalski, R. (1969) "Search strategies for theorem proving", in: Machine Intelligence, Vol. 5, eds. B. Meltzer and D. Mitchie, (Edinburgh University Press, Edinburgh, 1969) pp. 181-201.

Kowalski, R. (1972) “And-or graphs, theorem proving graphs and bi-directional search”, in: Machine Intelligence, Vol. 7, eds. B. Meltzer and D. Mitchie (Edinburgh University Press, Edinburgh, 1972) pp. 167-194.

Kushner, H and Pakut, A. (1982) “A Simulation Study of Decentralized Detection Problem," *IEEE Trans. On Automatic Control*, vol. AC-27, No. 5, pp.1116-1119

Leung, Y.W. and Wang, Y. (2000). "Multiobjective programming using uniform design and genetic algorithm," *IEEE Trans. Syst. Man Cyber*. C, vol. 30, no. 3, pp. 293-304.

Madigan, D., Mittal, S., and Roberts, F.S. (2007), “Sequential decision making algorithms for port of entry inspection: Overcoming computational challenges,” preprint, DIMACS Center, Rutgers University, January 2007 (submitted for publication).

Mahadev, N.V.R., Pekec, A., and Roberts, F.S. (1997) “Effect of change of scale on optimality in a scheduling model with priorities and earliness/tardiness penalties,” *Mathematical and Computer Modelling*, 25 (1997), 9-22.

Mahadev, N.V.R., Pekec, A., and Roberts, F.S. (1998) “On the meaningfulness of optimal solutions to scheduling problems: Can an optimal solution be non-optimal?” *Operations Research*, 46 supp. (1998), S120-S134.

Miglio, R., and Soffritti, G. (2004) “The comparison between classification trees through proximity measures,” *Computational Statistics and Data Analysis*, 45, 577-593.

Mitten, L.G. (1960) "An analytic solution to the least cost testing sequence problem" *The journal of Industrial Engineering*, Jan-Feb 1960, pp.17.

Natarajan, K.S. (1986) "Optimizing depth-first search of AND-OR trees" Technical report, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, 1986.

Nilsson, N.J. (1971) “Problem-Solving Methods in Artificial Intelligence*”* (McGraw-Hill, New York, 1971).

Pohl, I. (1971) “Bi-directional search, in: Machine Intelligence”, Vol. 6, eds. B. Meltzer and D. Mitchie (Edinburgh University Press, Edinburgh, 1971) pp. 127-140.

Roberts, F.S. (1990) “Meaningfulness of conclusions from combinatorial optimization,”*Discrete Applied Math.*, 29, 221-241.

Roberts, F.S., (1994) "Limitations on conclusions using scales of measurement," in A. Barnett, S.M. Pollock, and M.H. Rothkopf (eds.), *Operations Research and the Public Sector*, Elsevier, Amsterdam, 621-671.

Roberts, F.S. (1999) "Meaningless statements," in *Contemporary Trends in Discrete Mathematics*, DIMACS Series, Vol. 49, American Mathematical Society, Providence, RI, 257-274.

Scniederjans. M.J. (1995) "Goal programming methodology and applications", Kluwer publishers, Boston.

Statnikov, R.S. and Matusov, J.B. "Multicriteria Optimization and Engineering". New York, Chapman and Hall, 1995.

Stroud, P.D. and Saeger, K.J. (2003) "Enumeration of Increasing Boolean Expressions and Alternative Digraph Implementations for Diagnostic Applications", In: *Proceedings Volume IV, Computer, Communication and Control Technologies: I*, (eds. H. Chu, J. Ferrer, T. Nguyen, Y. Yu), pp. 328-333.

Programming: Theories and Applications, M. Tamiz, Ed., Berlin, Germany, Springer-Verlag, pp. 164-195.

Ünlüyurt, T., (2004) "Sequential testing of complex systems: A review", *Discrete Applied Mathematics,* 142*,* (1-3):189-205.

Ünlüyurt, T., (2005) "Testing systems of identical components", *Journal of Combinatorial Optimization,* 10, (3):261-282.

United States Government Accountability Office (2006), "Cargo container inspection," GAO-06-591T. March 30, 2006.

Yi, G. Y. and Lawless J. F. (2006), "A corrected likelihood method for the proportional hazards model with covariates subject to measurement error," *Journal of Statistical Planning and Inference,* In-press, 2006.

Zhang, H., Schroepfer, C. and Elsayed E. A. (2006) "Sensor Thresholds in Port-of-Entry Inspection Systems," *Proceedings of the 12th ISSAT International Conference on Reliability and Quality in Design*, Chicago, Illinois, USA, August 3-5, 2006, pages 172-176.