Multi-Objective Optimization of a Port-of-Entry Inspection Policy

Christina M. Young, Mingyu Li, Yada Zhu, Minge Xie, Elsayed A. Elsayed, and Tsvetan Asamov

***Abstract*—At the port-of-entry containers are inspected through a specific sequence of sensor stations to detect the presence of nuclear materials, biological and chemical agents, and other illegal cargo. The inspection policy, which includes the sequence in which sensors are applied and the threshold levels used at the inspection stations, affects the probability of misclassifying a container as well as the cost and time spent in inspection. In this paper we consider a system operating with a Boolean decision function combining station results and present a multi-objective optimization approach to determine the optimal sensor arrangement and threshold levels while considering cost and time. The total cost includes cost incurred by misclassification errors and the total expected cost of inspection, while the time represents the total expected time a container spends in the inspection system. Examples which apply the approach in various systems are presented.**

Note to practitioners—Inspection of containers arriving at the port-of-entry is becoming a challenging problem as both the number of containers and inspection attributes of the containers increase. The sequence of inspections and the level of inspection have a major impact on the total cost of inspection and delay of containers at a port. This paper presents methods that search for the optimum threshold levels at inspection stations as well as the optimum inspection sequence to minimize the total cost and delays. Several methods are presented and their performances are compared. These methods provide the border protection and customs agencies with approaches based on theoretical foundations yet the results are readily applicable.

***Index Terms*—Boolean function, probability of false accept, probability of false reject, sensor threshold levels, multi-objective**

I. Introduction

THE significant increase in trade agreements and the growth in the world economy have propelled unprecedented increase in maritime traffic. The value of export goods produced and transported globally in 2000 is about $6.186 trillion [1]. Disruption of such a system has catastrophic consequences on the world economy and our daily needs. In order to minimize sources of disruptions, the United Nations passed several resolutions with the objective of improving security in maritime trade. Likewise, the United States initiated the Container Security Initiative (CSI) to ensure container security through different approaches starting from the origin port of the container and ending at the delivery port in the United States. One of these approaches involves the assignment of a “risk” factor associated with containers bound for the United States. Based on this factor, containers might be subjected to screening at the port of origin and might be “pre-cleared” for importation. Moreover, when containers arrive at United States ports they are subjected to further security inspection systems. Containers can be randomly selected and subjected to inspection at the port-of-entry. The type of inspections, number of containers to be inspected, and the inspection policy have a profound effect on the cost of the system, risk of accepting undesired containers and potential delays and congestion at the ports.

In this paper we consider a port-of-entry (POE) container inspection system where a fraction of the arriving containers at a port is subjected to a sequence of inspections at different stations. A typical inspection system begins with radiation detection. Containers are driven through a Radiation Portal Monitor (RPM) at approximately five miles per hour, where radiation emissions are detected. The equipment is passive in that it absorbs radiation from containers as they pass through the RPM. A graphic profile of the radiation reading is produced and if the profile suggests the presence of radioactive material, an alarm is activated. Once an alarm is activated, the container is then subjected to further inspection to determine the source of radiation. This is usually accomplished by using a lightweight hand-held Radiation Isotope Identification Device (RIID) or an Advanced Spectroscopic Portal (ASP) to identify the radiation isotope. The RIID or ASP can differentiate from naturally occurring, harmless radiation emitted by materials such as: ceramic tile, granite, kitty litter, fertilizer, or food products containing potassium, including bananas or avocados. The RIID is more sensitive than the traditional Geiger counter, and takes an isotope reading to determine the type of radiation being emitted. The RIID is capable of distinguishing between naturally occurring or weapons grade radioactive and nuclear materials, including highly enriched uranium or plutonium used in nuclear devices [2].

There are two common approaches that use radiation for container inspection: the first is based on x-ray systems which generally take a few minutes to scan a standard 40-foot

This research is conducted with partial support from ONR grant numbers N00014-05-1-0237, N00014-07-1-029 and NSF grant number NSFSES 05-18543 to Rutgers University.

E. A. Elsayed, C. M. Young, and Y. Zhu are with the Department of Industrial and Systems Engineering, Rutgers University, Piscataway, NJ 08854 USA (e-mail: elsayed@rci.rutgers.edu)
M. Xie and M. Li are with the Department of Statistics, Rutgers University, Piscataway NJ 08854 USA (e-mail: mxie@stat.rutgers.edu)
T. Asamov is with Kenyon College

container. More advanced x-ray systems can take only a few seconds [3]. However, total inspection cycle times may range from 7-15 minutes or longer due to image analysis. The second approach includes gamma-ray inspection systems. These systems directly use gamma-rays or use pulsed fast neutrons to generate gamma-rays to produce images of the container's contents, 3-D mappings of content location, as well as other important information [4]. Unlike x-ray systems, the gamma-ray systems can scan standard 40-foot containers in a few seconds and generate a total inspection time of less than a minute [5]. The average inspection throughput of gamma-ray systems is more than 10 times greater than the fastest x-ray system [5].

Gamma-ray systems can cost from 3-20 times less than x-ray systems in terms of initial capital investment, 4-5 times less in terms of installation and when considering other benefits, gamma-ray systems can yield a cost per inspection that is 50 times less than that of conventional x-ray systems [6], [7].

In addition to inspecting containers for radioactive material and image analysis, the inspection stations may identify biological warfare agents by using biosensors, currently under research and ready for deployment on an experimental basis, that can detect trace amounts of viral or bacterial pathogens in situ and provide immediate analysis. The detection system uses chip-based technology, which requires low voltages and can easily be incorporated into portable, wireless devices [8]. Other equipment for detecting biological agents uses fluorescent particle counters for detecting airborne bacteria. In this case the threshold level of the decision is related to the count of these particles. Likewise, methods for detection of chemical agents, currently sarin, cyanide, and pesticides, may be applied at other inspection stations using different sensors [9].

Researchers have investigated the problem of container inspection with different objectives. Lewis *et al.* [10] develop a best-first heuristic search procedure to model the problem of moving containers from inbound ships to staging areas where security inspections can occur and moving containers from staging areas and areas where security inspections have been completed to outbound ships. The inspection procedures and sequences have not been considered. Stroud and Saeger [11] consider the problem where a stream of containers arriving at a port and sequential inspections (diagnosis) are conducted to decide whether to pass the container or subject it to further inspection. The container can leave the sequence of inspection stations when some conditions are met or it continues through other stations until completion of the entire inspection system. Containers that leave the inspection system during inspection can either be accepted or subjected to "manual" inspection. This problem is considered as binary decision tree problem. Madigan *et al*. [12] extend the work of Stroud and Saeger by incorporating the threshold levels of the inspection "sensors" and develop a novel binary decision tree search algorithm that operates on a space of potentially acceptable binary decision tree. They describe computationally more efficient approaches for this binary decision tree problem and obtain optimum sensor threshold levels that minimize the total cost of the inspection system.

There are similarities and dissimilarities between the problem of container inspection at POE and the baggage screening in airports. The similarities are related to potential error in inspection. This problem is investigated by Canadalino *et al.* [13] where they introduce a comprehensive cost function that includes direct costs associated with the purchase and operation of baggage screening security devices and the indirect costs associated with device errors. They present a methodology to determine the best selection of baggage screening security devices that minimizes the expected annual total cost of a baggage screening strategy [13]. The dissimilarities arise from the fact that container inspection requires several inspection stations for detecting the presence of different nuclear materials, drugs, explosives and others while baggage screening is performed at one station.

This raises several issues of concern stemming from the inspection sequence (or flow of container in the system) and the acceptance threshold levels of sensors at inspection stations. Moreover, the container can leave the inspection system when a partial list of attributes deem it safe to accept or unsafe to manually unpack and inspect. Inspection error is a function of the threshold values set at inspection stations. Threshold levels have a major impact on inspection errors, inspection time, and throughput of the system. The problem becomes a multi-objective problem with two objectives: minimization of the total inspection cost which includes cost of inspection and cost of misclassifying containers (acceptable when they are not or unacceptable when they are) and the minimization of delay time.

The problem of determining the optimum inspection sequence has been investigated by many researchers. Since this problem is NP-hard, researchers have focused their investigation on known configured inspection systems such as series, series-parallel, parallel-series and *k*-out-of-*n* systems in order to obtain optimum solutions for a small number of attributes. Lee [14], Raouf *et al*. [15], and Dufuaa and Raouf [16] consider a similar problem for inspection of units in a typical production system where a product is tested on a number of its characteristics, failure of which results in the rejection of the product. There are associated costs for each of the tests and the probability that each test will fail the item is known. In some models, tests are not perfect, and there are associated costs of shipping a faulty product to the customer. The problem is to sequence the tests to minimize the total expected cost.

Branch and bound and dynamic programming formulations have been proposed to solve the general problem of sequential diagnosis, both of which run in exponential time in general. A dynamic programming approach has been proposed in [17] for threshold functions. Bioch and Ibaraki [18] and Chang *et al.* [19] propose polynomial time algorithms that produce optimal solutions for *k*-out-of-*n* system. These algorithms are generalized further in Boros and Unluyurt [20] by providing a generalized algorithm that is optimal for double regular systems (having identical components). Other approaches such as genetic algorithms and artificial intelligence have been utilized.

Elsayed *et al.* [21] present a unique approach to the formulation of the port-of-entry inspection problem as an analytical model. Unlike previous work which determines threshold levels and sequence separately, they consider an integrated system and determine them simultaneously. They decompose the POE problem into two sub-problems. One 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. This problem is formulated and investigated using approaches parallel to those used in the optimal sequential inspection procedure for reliability systems as described in [17], [22]-[28]. The other problem deals with the determination of the optimum thresholds of the inspection stations so as to minimize the cost associated with false reject and false accept. As indicated earlier, the delay in inspection system is also a major concern as it has significant economic consequences.

In this paper we develop a new formulation for the POE problem by considering both the minimization of the total cost and the delay time of the containers simultaneously as a multi-objective optimization problem. We seek the optimum inspection sequence and the optimum threshold levels of sensors at inspection stations in order to minimize the total cost and total delay time.

This paper is organized as follows. Section II describes the port-of-entry container inspection problem. Section III describes the multiple objectives of the optimization problem: the cost of misclassifications, the cost of inspection, and the time spent in inspection. Section IV details three approaches for solving the multi-objective problem. Section V presents numerical examples of the methods discussed and finally the last section offers a discussion of the work presented.

II. Problem Description

A. Port-of-Entry Container Inspection System

In modeling the port-of-entry container inspection system it is assumed that containers arriving for inspection are inherently acceptable or contain unacceptable materials, and that they have several attributes which may reflect the status (presence or no presence) of such material. The inspection system is viewed as a collection of stations, over which the inspection of a given container is performed sequentially. Each station inspects one specific attribute and returns a pass-or-fail decision (0 or 1 respectively). At each individual station the decision is dependent on a preset threshold level. Varying this threshold level affects the probability of misclassifying an acceptable container as suspicious or vice versa. The sequence in which stations are to be visited, along with threshold levels to be applied, establishes the inspection policy which is applied to every container arriving for inspection. The final decision to accept a container or reject it, thereby subjecting the container to further manual inspection, often including a manual "unpacking" method, is determined based on the evaluation of a predefined Boolean decision function of the individual station decisions.

The Boolean decision function F assigns to each binary string of attributes $\left(a_1, a_2, ..., a_n\right)$ a category. In other words $F(a_1, a_2, \ldots, a_n) = 0$ indicates negative class and that there is no suspicion with the container and $F(a_1, a_2, \ldots, a_n) = 1$ indicates positive class and that additional inspection is required, usually manual inspection.

By definition, for instance, a series Boolean function is a decision function F that assigns the container a value of "1" if any of the attributes is present, i.e. $a_i = 1$ for any $i \in \{1, 2, \ldots, n\}$, and a parallel Boolean function is a decision function F that assigns the container a value of "1" if all of the attributes are present, i.e. $a_i = 1$ for all $i \in \{1, 2, \ldots, n\}$. The Boolean function to be used depends on the nature of the inspection system; the container attributes being inspected, and other factors. The work presented here is designed so that it can be applied with any Boolean function. A few common Boolean functions are used in the numerical examples.

B. Modeling of Sensor Measurements

Let x represent true status of a container, and code $x = 1$ if it should be rejected and $x = 0$ if it should be accepted. We assume this container is a sample from a population of interest under which the probability of $x = 1$ is $P(x = 1) = \pi$ and the probability of $x = 0$ is $P(x = 0) = 1 - \pi$.

Let r be the measurement taken by a sensor. This measurement r can in general be a numerical (continuous or discrete) reading or a graphical image, as described in the Introduction section. To simplify the presentation of our development and following [11] and [21], we assume

$$r \sim N\left(\mu_0, \sigma_0^{\ 2}\right) \text{ when } x = 0 \text{ and}$$
$$r \sim N\left(\mu_1, \sigma_1^2\right) \text{ when } x = 1, \qquad (1)$$

where $\mu_0 \neq \mu_1$. We choose to use the normal distribution because normally distributed data are the most commonly seen data in practice and it is has been used in port-of-entry inspection applications [11], [21]. Also, continuous measurements can often be transformed into a normal distributed data by the well known inverse transformation method [29]. Likewise, discrete data sometimes can be well approximated by a normal distribution either by central limit theorem or by some special techniques such as variance stabilization transformation. Our development, in principle, can be extended to some non-normal cases.

We assume two normal distributions in (1) because we expect to have different sensor readings for a container with true status $x = 1$ and $x = 0$. We also assume that the parameters of the two normal distributions in (1) are known or can be estimated from past inspection history. Note that the task of distinguishing acceptable and unacceptable containers is location and scale invariant to the readings. Without loss of generality, we can assume that $\mu_0 = 0$ and $\mu_1 = 1$. See also [21] for further discussions on this assumption.

C. Threshold Approach

To make a decision based on the sensor measurement r, the r value is compared against a given threshold value T. We reject the container ($d=1$) if the reading r is higher than T and accept it ($d=0$) if the reading is lower than T. The decision d at this level of decision making does not always agree with the true status x. There are two types of potential errors:

Type I Error: decision $d=1$ when the true status of the container is $x=0$,

and

Type II Error: decision $d=0$ when the true status of the container is $x=1$.

The probability of these two types of errors can be computed by

$$P(d=1\,|\,x=0)=P(r>T\,|\,x=0)=1-\Phi\left(\frac{T}{\sigma_0}\right)$$

and

$$P(d=0\,|\,x=1)=P(r\le T\,|\,x=1)=\Phi\left(\frac{T-1}{\sigma_1}\right).$$

D. System Inspection Policy

The minimization of costs associated with performing inspection and misclassification of containers has been formulated in Elsayed *et al*. [21]. Here we expand the optimization objective to include the time required for inspection, which takes into account the effect of delays on the overall system. The time incurred in inspection is added to the objectives because it may be very important in some situations. The performance of the inspection system is determined by both the sequence in which inspection stations are visited and the threshold levels applied at those stations, which we denote collectively as the inspection policy.

Since the optimal parameter values for the cost minimization problem may not minimize time, some compromise may be required. A particular balance of the importance of cost and time may be represented by weights. We consider the case where the relative importance of cost and time is unspecified and therefore we use different importance weights to generate possible solutions that produce a Pareto frontier as described in section IV.

III. Performance measures of inspection policy

A. Cost of Misclassification and Inspection

The cost involved in this inspection problem is the sum of any cost incurred as a result of misclassifying a container's status and the actual cost of performing the inspection. As Elsayed *et al.* [21] note, there are two types of misclassification errors at the systems level: falsely rejecting a container that should be cleared and falsely accepting a container that should be rejected. These errors are associated with the probability of false reject (PFR) and the probability of false accept (PFA), respectively. The complementary probabilities of these two errors are true reject (PTR) and true accept (PTA). If D denotes the decision of the entire inspection system of sensors, where $D=1$ means to reject and $D=0$ to accept, the four probabilities can be written as follows:

$PFR=P(D=1\,|\,x=0)$, $PTA=P(D=0\,|\,x=0)=1-PFR$,

$PFA=P(D=0\,|\,x=1)$, and $PTR=P(D=1\,|\,x=1)=1-PFA.$

The inspection decision D depends on the individual inspection results and the system Boolean function. The probability equations just mentioned can be rewritten in terms of the threshold value T_i and σ values related to the inspection station for any given Boolean function. Several examples are given in Elsayed *et al.* [21].

The cost of misclassification arises when a cost is associated with PFR and PFA. Let c_{FA} be the cost of the system accepting a "bad" container and c_{FR} be the cost of the system rejecting a "good" container. Then the total cost of system misclassification error is $C_F=\pi\ PFA\ c_{FA}+(1-\pi)PFR\ c_{FR}$ as described in [21].

The expectation of the cost of inspection is a function of the unit cost to operate each sensor (station) and the probabilities of passing each station. Given a particular set of threshold values, an optimal sequence in which to visit the sequence can be found following the conditions in Elsayed *et al*. [21]. This is explained in the two Theorems in Section IV. The total cost arising from misclassification errors and inspection is denoted by $c_{total}=C_F+E[C_{inspection}]$.

B. Time for Inspection

The time required for a container to pass through an inspection station is an important measure of the inspection system performance. It is possible that this time would be related to some characteristic of the inspection station, that is to say the inspection may be sped up or slowed down depending on some operational setting of the sensor. For example the inspection time may be related to a variable that represents the resolution or other settings of the sensor. Following Jupp *et al*. [30], we assume the time spent at each station could be related to the threshold level T_i at that station. This relationship is expressed as $t_i=a\exp(b\cdot T_i)$ for illustration purposes. Here the time of inspection decreases as the applied threshold level increases.

To find the total expectation of time spent in the system for a given container we first denote p_i, the probability of passing station *i*, by:

$$p_i=P(d_i=0)=\sum_{j=0}^{1}\left[P(d_i=0\,|\,x=j)P(x=j)\right]$$
$$=(1-\pi)\Phi\left(\frac{T_i}{\sigma_{0i}}\right)+\pi\,\Phi\left(\frac{T_i-1}{\sigma_{1i}}\right),$$

and $q_i=1-p_i$ where p_i and q_i are functions of threshold values T_i. Then, the total expected inspection time t_{total} can be expressed as $t_{total} = t_1 + \sum_{i=2}^{n} \left[\prod_{j=1}^{i-1} p_j \right] t_i$, where t_i is the inspection time at station *i*, for a system with *n* stations using a series Boolean decision function. For a parallel Boolean decision function, the total expected inspection time is $t_{total} = t_1 + \sum_{i=2}^{n} \left[\prod_{j=1}^{i-1} q_j \right] t_i$.

IV. Multi-Objective Optimization

A. Total Expected Cost and Time

As noted in the problem description, we need to determine the optimal design or configuration of sensors in the system and the optimum sets of threshold levels that can achieve the objectives of maximizing inspection system throughput and minimizing the expected total cost. These objectives are interrelated. It is unlikely that they would be optimized by the same set of parameter values, and there exists some trade-off between them. This is a typical multi-objective optimization problem. See, for instance, Eschenauer *et al*. [31], Statnikov and Matuso [32], Fonseca and Fleming [33], [34], and Leung and Wang [35], among others. We formulate the POE problems as a multi-objective optimization problem:

$$\min_{Sequence,Threshold} \{c_{total}, t_{total}\}.$$

In general, there may be a large number or infinite number of optimal solutions 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. A commonly used method to deal with the multi-objective optimization problem is to use the weighted sum approach, where we optimize fitness functions (i.e., weighted sums of the objective functions) for various choices of fixed weights w_1 and w_2, $w_1 + w_2 = 1$.

$$f_{w_1,w_2}(S,T) = w_1 c_{total} + w_2 t_{total}$$

Here, *S* and *T* stand for sequence and threshold levels. Thus, the multi-objective optimization problem becomes a sequence of single objective optimization problems, in which we minimize the fitness function for a set of fixed weights w_1 and w_2.

$$\min_{S,T} f_{w_1,w_2}(S,T) \tag{2}$$

In this paper, we employ a modified weighted sum approach, in which we utilize some theoretical results to deal with the arrangement of system sequences. Note that the function $f_{w_1,w_2}(S,T)$ is highly discrete with regards to the system sequence and arrangements. The number of arrangements also grows exponentially as the number of inspection stations increase. It is computationally expensive to directly solve the minimization problem in (2). For the system Boolean functions considered in this paper, the optimal sequence can be obtained for a given set of weights and thresholds, as stated in the theorem presented below So, for a given set of weights and threshold we are able to compute the function

$$f_{w_1,w_2}(T) = \min_{S} f_{w_1,w_2}(S,T). \tag{3}$$

In the modified weighted sum approach, we in fact solve the minimization problem $\min_{T} f_{w_1,w_2}(T)$.

This modified approach can provide an efficient method to deal with the multi-objective optimization problem under the current context. Note that avoiding the consideration of all possible sensor arrangements improves the efficiency of the modified algorithm.

Theorem 1:

(a) For a series Boolean decision function, inspecting attributes $i = 1, 2, \ldots, n$ in sequential order is optimum, in the sense of minimizing the fitness function for the given set of weights (w_1, w_2) and a given set of threshold, if and only if

$$(w_1c_1 + w_2t_1)/q_1 \le (w_1c_2 + w_2t_2)/q_2 \le \ldots \le (w_1c_n + w_2t_n)/q_n$$

(condition 1a).

In this case, the minimal value of the fitness function is given by

$$f_{w_1,w_2}(T) = (w_1c_1 + w_2t_1) + \sum_{i=2}^{n} \left[\prod_{j=1}^{i-1} p_j \right] (w_1c_i + w_2t_i) + w_1C_F.$$

(b) For a parallel Boolean decision function, inspecting attributes $i = 1, 2, \ldots, n$ in sequential order is optimum, in the sense of minimizing the fitness function for the given set of weights (w_1, w_2) and a given set of threshold, if and only if

$$(w_1c_1 + w_2t_1)/p_1 \le (w_1c_2 + w_2t_2)/p_2 \le \ldots \le (w_1c_n + w_2t_n)/p_n$$

(condition 1b).

In this case, the minimal value of the fitness function is given by

$$f_{w_1,w_2}(T) = (w_1c_1 + w_2t_1) + \sum_{i=2}^{n} \left[\prod_{j=1}^{i-1} q_j \right] (w_1c_i + w_2t_i) + w_1C_F.$$

The results in Theorem 1 for series system and parallel system can be extended to systems with arrangements of parallel-series and series-parallel sensors, given in Theorem 2.

Theorem 2:

(a) Consider a parallel-series decision function with n paths of m sensors each (see Fig. 1). If an inspection system with attributes $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$ arranged in parallel-series is optimal, it satisfies the following conditions: the inspection sequence of the series of sensors within each path should be arranged in the order of

$$(w_1c_{i1} + w_2t_{i1})/q_{i1} \le (w_1c_{i2} + w_2t_{i2})/q_{i2} \le \ldots \le (w_1c_{im} + w_2t_{im})/q_{im},$$

and the inspection sequence of parallel paths should be arranged in the order of $F_1/P_1 \le F_2/P_2 \le \ldots \le F_n/P_n$ (condition 2a). Here, F_i and P_i are the (minimal) combined *expanse (fitness) of cost and time and the probability of acceptance of the i*th *path:*

$$F_i = (w_1c_{i1} + w_2t_{i1}) + \sum_{j=2}^{m}\left[\prod_{k=1}^{j-1} p_{ik}\right](w_1c_{ij} + w_2t_{ij})$$

$$= (w_1c_{i1} + w_2t_{i1}) +$$

$$\sum_{j=2}^{m}(w_1c_{ij} + w_2t_{ij})\prod_{k=1}^{j-1}\left[(1-\pi)\Phi\left(\frac{T_{ik}}{\sigma_{0ik}}\right) + \pi\Phi\left(\frac{T_{ik}-1}{\sigma_{1ik}}\right)\right]$$

and $P_i = P(D_i = 0) = \prod_{j=1}^{m} p_{ij}$ *. In this case, the minimal value of the fitness function is:*

$$f_{w_1,w_2}(T) = F_1 + \sum_{i=2}^{n}\left[\prod_{j=1}^{i-1}(1-P_j)\right]F_i + w_1C_F$$

$$= F_1 + \sum_{i=2}^{n} F_i \prod_{j=1}^{i-1}\left(1-\prod_{k=1}^{m} p_{jk}\right) + w_1C_F .$$

(b) Consider a series-parallel decision function that has n subsystems in series with m units in parallel in each subsystem (see Fig. 2). If an inspection system with attributes $i = 1, 2, \ldots, n$ *and* $j = 1, 2, \ldots, m$ *arranged in series-parallel is optimal, it satisfies the following conditions: the inspection sequence of the series of each subsystem should be arranged in the order of*

$(w_1c_{i1} + w_2t_{i1}) / p_{i1} \le (w_1c_{i2} + w_2t_{i2}) / p_{i2} \le \ldots \le (w_1c_{im} + w_2t_{im}) / p_{im}$ *and the inspection sequence of parallel paths should be arranged in the order of* $F_1 / Q_1 \le F_2 / Q_2 \le \ldots \le F_n / Q_n$ *(condition 2b). Here,* C_i *and* Q_i *are the (minimal) combined expanse (fitness) of cost and time and the probability of rejection of the i*th *subsystem:*

$$F_i = (w_1c_{i1} + w_2t_{i1}) + \sum_{j=2}^{m}\left[\prod_{k=1}^{j-1} q_{ik}\right](w_1c_{ij} + w_2t_{ij})$$

$$= (w_1c_{i1} + w_2t_{i1}) +$$

$$+\sum_{j=2}^{m}(w_1c_{ij} + w_2t_{ij})\prod_{k=1}^{j-1}\left[(1-\pi)\left\{1-\Phi\left(\frac{T_{ik}}{\sigma_{0ik}}\right)\right\} + \pi\left\{1-\Phi\left(\frac{T_{ik}-1}{\sigma_{1ik}}\right)\right\}\right]$$

and $Q_i = P(D_i = 1) = \prod_{j=1}^{m}(1-p_{ij})$ *. In this case, the minimal value of the fitness function is*

$$f_{w_1,w_2}(T) = F_1 + \sum_{i=2}^{n}\left(\prod_{j=1}^{i-1} P_j\right)F_i + w_1C_F$$

$$= F_1 + \sum_{i=2}^{n} F_i \prod_{j=1}^{i-1}\left\{1-\prod_{k=1}^{m}(1-p_{jk})\right\} + w_1C_F$$

From the theorems, we describe our modified weighted sum optimization algorithm as follows:

Step 1: Generate a large number, say N, sets of weight pairs (w_1 *,* w_2 *);*

Step 2: For each pair of weights, we solve the minimization problem $T_{\min}^{(w_1,w_2)} = \arg\min_T f_{w_1,w_2}(T)$ *where the function* $f_{w_1,w_2}(T)$ *is computed in a subroutine stated next utilizing the results of the theorems;*

Step 3: Obtain the optimal sequence corresponding to $T_{\min}^{(w_1,w_2)}$ *(using the results of the theorems) and compute the corresponding optimal throughput time and total cost* $(t_{total}^{(w_1,w_2)}, c_{total}^{(w_1,w_2)})$ *;*

Step 4: Plot the N pairs of optimal throughput time and cost $(t_{total}^{(w_1,w_2)}, c_{total}^{(w_1,w_2)})$ *which form the Pareto optimal solutions for the multi-objective optimization problem.*

For the parallel and series inspection Boolean systems, we use the following subroutine to calculate the function $f_{w_1,w_2}(T) = \min_S f_{w_1,w_2}(S,T)$:

1) For each inspection sensor i, calculate $w_1c_i + w_2t_i$*;*

2) For each inspection sensor i, calculate the ordering criterion ($w_1c_i + w_2t_i$*)/*p_i *or (*$w_1c_i + w_2t_i$*)/*q_i *;*

3) Sort the ordering criteria, thus finding the optimal arrangement of sensors, according to the theorems;

4) Calculate the total cost c_{total} *and the expected time of inspection* t_{total} *and return* $f = w_1c_{total} + w_2t_{total}$.

A similar subroutine can be developed for the series-parallel and parallel and series inspection Boolean systems.

In Step 2 of the modified weighted-sum algorithm, the minimization involves the function $f_{w_1,w_2}(T)$. We use two different ways to minimize this function and they lead to two different approaches described in the next section.

B. Implementation: Three Approaches

Three methods are distinguished in the implementation: Grid Search (GS), *fmincon* (FM) and Genetic Algorithm (GA).

The grid search method is a complete enumeration method and does not use any of the developments (Theorem or Algorithm) in this paper. But it sets a standard against which the GA and FM approaches may be compared.

FM and GA are based on the optimization algorithm developed in the previous subsection. The difference between these methods is how they solve the optimization problem $\min_T f_{w_1,w_2}(T)$.

1. Grid Search (GS)

The grid search method is a complete enumeration of possible threshold values and all inspection sequences. A discrete set of threshold values is formed from the range of 0-1 using a gradient of 0.05. The total cost and total time are calculated for each possible combination of threshold values and sequence. The resulting cost and time values are plotted and the outermost points along the curve are filtered to represent the solution set that forms the Pareto frontier. Thus the GS method yields a small number of true optimal points compared to the other methods.

2. *Minimization Using MATLAB fmincon (FM)*

The MATLAB (The MathWorks, Inc.) *fmincon* function attempts to find a constrained minimum of scalar function of several variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming. For each pair of weights, we use *fmincon* to minimize $f_{w_1,w_2}(T)$ and try different sets of initial thresholds. Note that the function $f_{w_1,w_2}(T) = \underset{S}{min} f_{w_1,w_2}(S,T)$ is highly discrete in T which is inherited from the sequence optimization. Therefore, direct use of the *fmincon* function does not always work. As the third subplot of Figure 1 shows, the optimal solutions vary significantly with different initial values used in the *fmincon* function.

3. *Genetic Algorithm (GA)*

A genetic algorithm is an iterative random search algorithm, which takes advantage of information in the previous steps (ancestors) to produce new searching points (off-springs). It is called “genetic” algorithm because the principle and design of this search algorithm mimics those of genetic evolution found in nature [36]. A genetic algorithm can be applied to solve “a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear” [37]. In the optimization algorithm developed here, the MATLAB function *ga* was used to minimize $f_{w_1,w_2}(T)$, replacing the MATLAB function *fmincon*, for each pair of weights. One advantage of using the *ga* is that it is insensitive to the initial values and we are able to obtain optimal solutions in all of our analyses.

V. Numerical Examples

Here the results of the multi-objective optimization are presented in graphical form. The three graphs in Figure 1 illustrate the optimal points obtained from the three methods discussed in the previous section applied to an inspection system using a parallel Boolean decision function. The system parameters in this example are as follows: n=3, c=[1 1 1], π=0.0002, μ_0=[0 0 0], μ_1=[1 1 1], σ_0=[0.16 0.2 0.22], σ_1=[0.3 0.2 0.26], c_{FA}=100000, c_{FR}=500, a=[20 20 20], b=[-3 -3 -3], w_1=[0: 0.004:1], w_2=1- w_1.

The grid search method produces optimal points that fall into distinct vertical segments due to the discrete nature of the method, and the minimum search gradient with an acceptable computation time was used. The leftmost graph contains only the outermost points with respect to the Pareto frontier from this method. Note that a small number of the points shown actually fall on the theoretical Pareto frontier, therefore the output from this method is not as useful compared to the others.

The center graph illustrates the optimal points obtained from the GA method. For the FM method it was discovered that the initial values used had a significant effect on the optimality of the points obtained. Therefore various sets of initial values were tested and the results overlaid on one graph to illustrate the phenomenon. The initial value sets are represented as two series in the rightmost graph. The initial values for threshold initial value (TIV) set 1 are T_0=[0.2 0.2 0.2] and the initial values for TIV-set2 are T_0=[0.2 0.6 0.2]. Another set, T_0=[0.8 0.8 0.8] gave similar results to TIV-set1, and is not shown.

All three methods produce at least some portion of the same Pareto frontier of solutions with minimal time and cost. Each point represents the time and cost for one possible solution, and each solution is defined by a set of threshold values $\{T_i : i = 1,...n\}$ -each to be applied at one of the n inspection stations- and the sequence in which to visit those stations. Table 1 presents three examples of points chosen from the Pareto frontier of grid search.

Table 1. Examples of Pareto Optimal Solutions

T1	T2	T3	Sequence	Cost	Time
0.0	0.95	0.05	2-3-1	9.03	1.16
0.0	0.85	0.0	2-1-3	5.54	1.57
0.0	0.75	0.05	2-3-1	3.13	2.11

It is important to consider program running time in the comparison of methods. FM is the fastest of the three methods, requiring about 2 minutes for one initial set. However, it does not always return the correct Pareto frontier and thus different initial sets must be used, and without knowledge of the true Pareto frontier choosing a good initial set is difficult. The GS method with grid=0.05 runs in about 6 minutes, however only about 12 points of the output are considered to fall within the theoretical Pareto frontier. If the grid is decreased to 0.025, roughly 23 points on the theoretical Pareto frontier are produced but it takes 5 hours to run. Further reducing the grid to 0.01 requires more than 200 hours to finish. Therefore it becomes impractical to decrease the grid size in order to generate more optimal points on the theoretical frontier.

The GA method takes about 10.5 hours with the current choice of parameters (PopulationSize=80) and produces 251 points on the theoretical Pareto frontier. Note that the *ga* function of MATLAB is designed for general purpose use, and we anticipate that the running time can be significantly improved by using a specialized program. Moreover, the GA method produces optimal solutions in all trials that best represent the theoretical Pareto frontier, with all points falling on the frontier.

The GA method was applied to a system using a series Boolean decision function with the same parameters as the first example. The results are presented in Figure 2. Here it is evident that a change in Boolean function has an effect on the results.

In the third example the GA method was used to find the multi-objective optimal solution to an inspection problem that uses a series-parallel Boolean function with the system parameters: m=2, n=2, c=[1 1; 1 1], π=0.0002, μ_0=[0 0; 0 0], μ_1=[1 1; 1 1], σ_0=[0.16 0.2; 0.22 0.18], σ_1=[0.3 0.2; 0.26 0.18], c_{FA}=100000, c_{FR}=500, a=[20 20; 20 20], b=[-3 -3; -3 -3], w_1=[0: 0.004:1], w_2=1- w_1. Figure 3 gives the optimal points for this example.

VI. Discussion

This paper investigates and formulates the inspection systems at ports-of-entry. The formulation is general and applicable to different systems as the attributes of a typical container are expressed by a Boolean function. The inspection stations in the physical configuration of the system can be arranged in series (sequential inspection), parallel, series-parallel, parallel-series, k-out-of-n (where any k stations out of n indicate the presence of undesired attributes) or in any network configuration. Boolean functions corresponding to any of these configurations can be developed. The number of attributes and the inspection sequence have significant impact on the system performance. Likewise, the threshold levels of the sensors are critical in the decision process of accepting or classifying a container as suspicious. They influence the probability of making the "wrong" decision by accepting undesired containers or subjecting "good" containers to further unneeded inspections. The POE inspection system problem is formulated as a multi-objective optimization problem that attempts to minimize the total cost as well as the delay time of the containers. The paper presents three different approaches for determining the optimum inspection sequence and the threshold levels at each inspection station that result in the optimization of the system performance measures in terms of cost and time. They are: grid search, constrained nonlinear optimization function, and genetic algorithm. All result in the same values of the optimization function when the number of inspection stations and threshold levels are small. The first two approaches become impractical when more stations and threshold levels are introduced while the genetic algorithm provides optimum or near optimum solutions for such problems in much smaller computation times. As stated earlier, these approaches provide Pareto frontier optimal solutions where every solution includes the optimum sequence of the inspection stations and the corresponding optimum threshold levels. This will enable the decision maker to choose amongst solutions that meet other constraints such as budget, space or layout of the port.

Acknowledgment

This research is conducted with partial support from Office of Naval Research and the National Science Foundation. The authors wish to thank Fred Roberts, Paul Kantor, and other members of the DIMACS Port-of-Entry research group for their comments and input throughout the research.

References

[1] World Trade Organization, "International Trade Statistics 2001," WTO Publications, Geneva, Oct. 2001.

[2] U. S. Customs and Border Protection. (2007, Oct. 11). Secure Freight Inspection Technology. *Secure Freight Initiative Fact Sheets* [Online]. Available: http://www.cbp.gov/linkhandler/cgov/newsroom/fact_sheets/sfi/sfi_technology.ctt/sfi_technology.pdf

[3] G. F. Bowser and R. C. Huseman. (2002, Mar.). Advanced technology for security and customs enforcement. *Port Technology International* [Online]. Available: http://www.porttechnology.org/journals/ed12/pdfs/pt12-173.pdf, pp. 173-176

[4] B. R. Douglas and S. D. Botello. (2002, Mar.). Automatic cargo inspection. *Port Technology International* [Online]. Available: http://www.porttechnology.org/journals/ed12/pdfs/pt12-177.pdf

[5] S. P. Michael. (2002 Mar.). Gamma Ray Technology: The practical container inspection alternative. *Port Technology International* [Online]. Available: http://www.saic.com/products/security/pdf/saic.pdf

[6] M. J. Christopher, D. W. Bowlin, and V. J. Orphan. (2002 Mar.). Mobile cargo inspection provides improved throughput efficiency and flexibility. *Port Technology International* [Online]. Available: http://www.saic.com/products/security/pdf/PortTechIssue12.pdf

[7] B. M. Lewis, "Port security: container inspection technology, technical report," The Logistics Institute, Georgia Tech., and The Logistics Institute -- Asia Pacific, National University of Singapore, 2002.

[8] Maritime Transportation and Port Security Research Center. [Online]. Available: http://www.cs.sunysb.edu/PortSecurity/events/Newsday.html

[9] G. M. Murrar and G. E. Southard, "Sensors for chemical weapons detection," *IEEE Instrum. Meas. Mag.*, vol. 5, no. 4, pp. 12-21, Dec. 2002.

[10] B. M. Lewis, A. L. Erera, and C. C. White, "Optimization approaches for efficient container security operations at transshipment seaports," *Transportation Res. Rec.*, vol. 1822, pp. 1-8, 2003.

[11] P. D. Stroud and K. J. Saeger, "Enumeration of increasing Boolean expressions and alternative digraph implementations for diagnostic applications," *Proc. Volume IV, Computer, Communication and Technologies*, 2003, pp. 328-333.

[12] D. Madigan, S. Mittal, and F. Roberts, "Sequential decision making algorithms for port of entry inspection: Overcoming computational challenges," *IEEE Int. Conf. on Intelligence and Security Informatics (ISI-2007)*, New Brunswick, NJ, May 23-24, 2007.

[13] T. J. Canadalino, J. E. Kobza, and S. H. Jacobson, "Designing optimal aviation baggage screening strategies using simulated annealing," *Comput. and Oper. Res.*, vol. 31, no.10, pp. 1753-1767, Sep. 2004.

[14] H. L. Lee, "On the optimality of a simplified multicharacteristic component inspection model," *IIE Trans.* vol. 20, no. 4, pp. 392–398, 1988.

[15] A. Raouf, J. K. Jain, and P. T. Sathe, "A cost-minimization model for multicharacteristic component inspection," *IIE Trans.* vol. 15, no. 3, pp. 187–194, 1983.

[16] S. O. Duffuaa and A. Raouf, "An optimal sequence in multicharacteristic inspection," *J. Optim. Theory Appl.* vol. 67, no. 1, pp. 79–87, 1990.

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

[18] J. C. Bioch and T. Ibaraki, "Complexity of identification and dualization of positive Boolean functions," *Inform. and Comput.*. vol. 123, pp. 51–75, 1995.

[19] M. Chang, W. Shi, and W. K. Fuchs, "Optimal diagnosis procedures for k-out-of-n structures," *IEEE Trans. Comput.* vol. 39, no. 4, pp. 559–564, 1990.

[20] E. Boros and T. Unluyurt, "Diagnosing double regular systems," *Ann. Math. Artif. Intell.* vol. 26, pp. 171–191, 1999.

[21] E. A. Elsayed, C. M. Young, M. Xie, H. Zhang, and Y. Zhu, "Port-of-Entry inspection: Sensor deployment policy optimization," submitted to *IEEE Trans*.

[22] N. Azaiez and V. M. Bier, "Optimal resource allocation for security in reliability systems," *CREATE Report*, 2004.

[23] Y. Ben-Dov, "Optimal testing procedures for special structures of coherent systems," *Manage. Sci.*, vol. 27, no. 12, pp. 1410-1420, 1981.

[24] R. W. Butterworth, "Some reliability fault-testing models," *Oper. Res.*, vol. 20, pp. 335-343, 1972.

[25] L. Cox, S. Chiu, and X. Sun, "Least-cost failure diagnosis in uncertain reliability systems," *Rel. Eng. and Syst. Safety*, vol. 54, pp. 203-216, 1996.

[26] J. Halpern, "Fault-testing of a k-out-n system," *Oper. Res.*, vol. 22, pp. 1267-1271, 1974.

[27] J. Halpern, "A sequential testing procedure for a system's state identification," *IEEE Trans. Reliab.*, vol. R-23, no. 4, pp. 267-272, 1974.

[28] J. Halpern, “The sequential covering problem under uncertainty,” *INFOR*, vol. 15, pp. 76-93, 1977.

[29] L. Devroye, *Non-uniform Random Variate Generation*. New York: Springer-Verlag, 1986, ch. 2, sec. 2, pp. 28.

[30] I. D. Jupp, P. T. Durrant, D. Ramsden, T. Carte, G. Dermody, I. B. Pleasants, and D. Burrows, “The non-invasive inspection of baggage using coherent X-ray scattering,” *IEEE Trans. Nucl. Sci.*, vol. 47, no. 6, 2000.

[31] H. Eschenauer, J. Koski, and A. Osyczka, Editors, *Multicriteria Design Optimization*, Springer, Berlin, 1990.

[32] R. S. Statnikov and J. B. Matusov, *Multicriteria Optimization and Engineering*, New York, Chapman and Hall, 1995.

[33] M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part I: Unified formulation,” *IEEE Trans. Syst. Man Cybern. A*, *Syst. Humans*, vol. 28, pp. 26-37, 1998.

[34] M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms – Part II: Application example,” *IEEE Trans. Syst. Man Cybern. A*, *Syst. Humans*, vol. 28, pp. 38-47, 1998.

[35] Y. W. Leung and Y. Wang, "Multiobjective programming using uniform design and genetic algorithm," *IEEE Trans. Syst. Man Cybern. C*, *Appl. Rev.*, vol. 30, no. 3, pp. 293-304, 2000.

[36] M. Mitchell, *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press, 1996.

[37] *Genetic Algorithm and Direct Search Toolbox™ 2 User’s Guide*, The MathWorks Inc., Natick, MA, 2008.

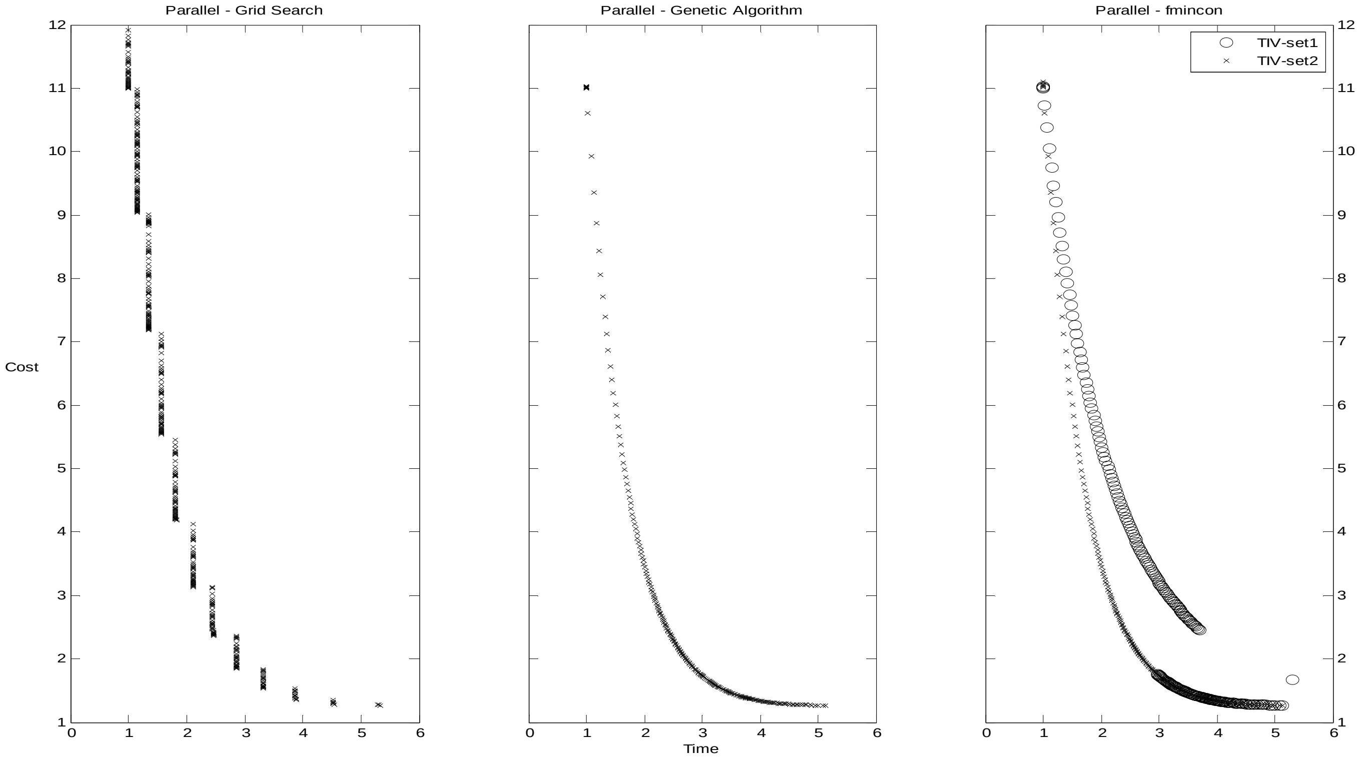

Figure 1. Comparison of Three Solution Methods to Multi-Objective Problem

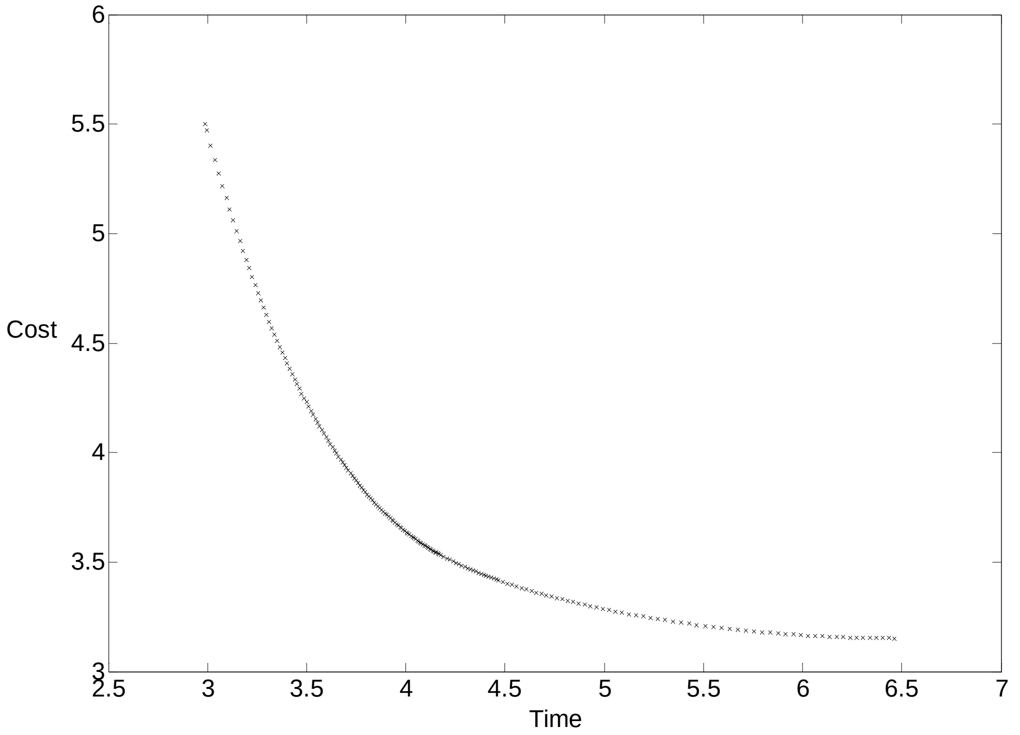

Figure 2. GA Method for Series Boolean

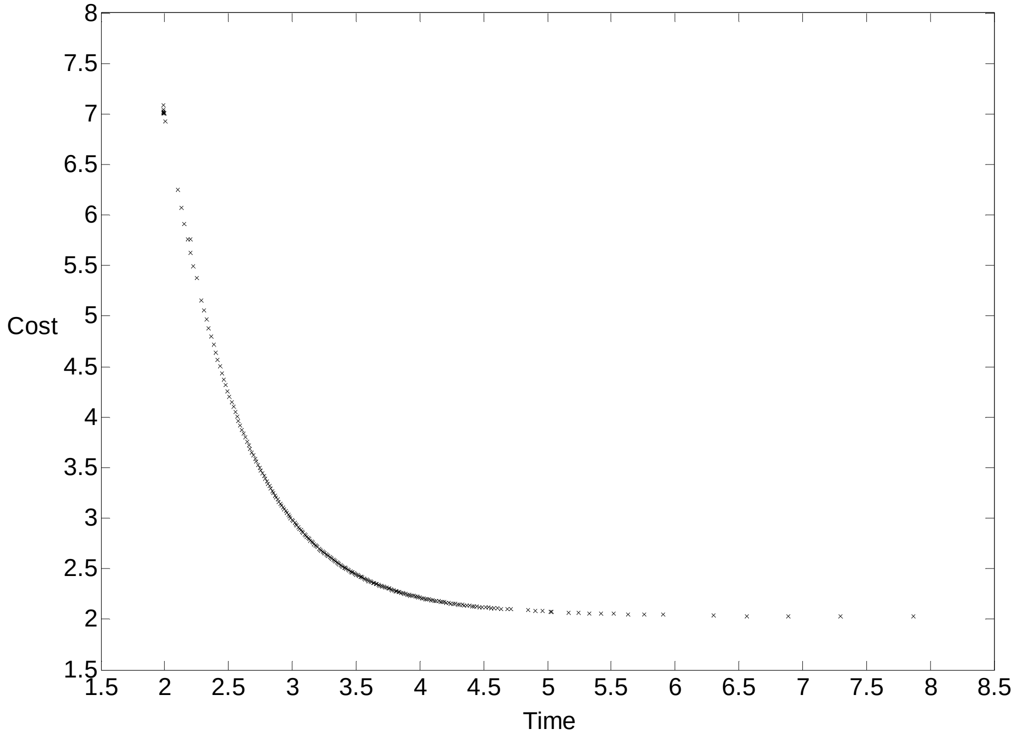

Figure 3. GA Method for Series-Parallel Boolean