Available online at www.sciencedirect.com
ScienceDirect
Reliability Engineering and System Safety 93 (2008) 1698–1709
RELIABILITY ENGINEERING & SYSTEM SAFETY
www.elsevier.com/locate/ress

Port-of-entry safety via the reliability optimization of container inspection strategy through an evolutionary approach

Jose Emmanuel Ramirez-Marquez*

School of Systems & Enterprises, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ 07030, USA

Received 4 September 2007; received in revised form 29 November 2007; accepted 6 January 2008
Available online 19 January 2008

Abstract

Up to now, of all the containers received in USA ports, roughly between 2% and 5% are scrutinized to determine if they could cause some type of danger or contain suspicious goods. Recently, concerns have been raised regarding the type of attack that could happen via container cargo leading to devastating economic, psychological and sociological effects. Overall, this paper is concerned with developing an inspection strategy that minimizes the total cost of inspection while maintaining a user-specified detection rate for "suspicious" containers. In this respect, a general model for describing an inspection strategy is proposed. The strategy is regarded as an $(n+1)$-echelon decision tree where at each of these echelons, a decision has to be taken, regarding which sensor to be used, if at all. Second, based on the general decision-tree model, this paper presents a minimum cost container inspection strategy that conforms to a pre-specified user detection rate under the assumption that different sensors with different reliability and cost characteristics can be used. To generate an optimal inspection strategy, an evolutionary optimization approach known as probabilistic solution discovery algorithm has been used.

Keywords: Sensor; Reliability; Decision tree; Evolutionary optimization

1. Introduction

During the last three decades cargo carriage practices have considerably changed around the world making the global economy highly dependent on container cargo trade. Currently, the United States of America accounts for roughly 20% of the global cargo trade and according to the World Shipping Council, the total value of goods arriving and departing USA ports was approximately $2.23 trillion, of which roughly 55% was cargo carried on container vessels. There is no clear estimate of the number of containers in circulation worldwide. However, there is a rough estimate of around 4 million containers in USA ports at any given day. The origin of these containers span the entire globe—from China to Chile, Costa Rica to India, France to Australia, etc.—and arrive at locations that cover the entire United States. Once any of these containers clears the port-of-entry (POE), by way of an inspection, it can move freely throughout the country via the roadway and railway network.

As discussed by Wein et al. [1], "all containers shipped from an overseas port of embarkation to a US debarkation are deemed to be either trusted or untrusted [sic] by US Customs' Automated Targeting System (ATS)". The ATS is part of the US Customs 24-h manifest rule, which is implemented to any container, bound to the US. However, estimates of the percentage of containers that are further scrutinized to determine if they could cause some type of danger or contain suspicious goods range between 2% [1] and 5% [2]. Historically, the purpose of container inspection at the POE has been to stop suspicious or illegal goods (drugs, hazardous material, contaminated food, etc.) from entering the country. However, since the events of September 11, this historical focus has changed dramatically to deterring possible attacks on one of the key components of the national infrastructure [3].

In general, concerns have been raised regarding the type of attack that could happen via container cargo [2]: (1) damaging the port infrastructure itself and/or (2) delivering

*Tel.: +1 201 216 8003.
E-mail address: jmarquez@stevens.edu

0951-8320/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2008.01.003

Nomenclature

i	index representing the ith location in the inspection strategy decision tree, $i=(1,\ldots,2^{n+1}-1)$
j	index for the sensor types available, $j=1,\ldots,n$
h	index representing the hth echelon of inspection strategy decision tree, $h=0,\ldots,n$
r_j	reading obtained from the jth sensor type
X	a random variable defining the status of a container
$N(\mu_{jz},\sigma_{jz})$	normal probability density function with mean, μ_{j0} and standard deviation, σ_{j0}
z, y	binary (0, 1) variables
D_j	container classification based on the reading of the jth sensor
$\mathbf{t}$	vector of sensor thresholds, $\mathbf{t}=(t_1,t_2,\ldots,t_n)$
$\Phi_j^z(t_j)$	cumulative distribution function for $r_j\|X=1\sim N(\mu_{jz},\sigma_{jz})$
Q_i^z	percentage of containers type z at L_i, $Q_1^z=100$
$\Omega(i)$	the location immediately preceding location i
I_i^0	indicator variable defining the arc preceding location i in DT, is not suspicious
$L(i)$	type of sensor at the ith location in the inspection strategy decision tree
$\mathbf{L}$	decision-tree structure vector $\mathbf{L}=(L(1),L(2),\ldots,L(2^{n+1}-1))$
$R(\mathbf{L})$	detection rate under $\mathbf{L}$
$S^{-}(\mathbf{L})$	false-suspicious rate under $\mathbf{L}$
$U_j(\mathbf{L})$	utilization of sensor type j under $\mathbf{L}$
$C(\mathbf{L})$	inspection strategy cost under $\mathbf{L}$
S_{ij}	decision variable defining the type of sensor j used in location i
γ_{iu}	appearance probability vector. Defines the probability that a sensor type j will be present in the optimal inspection strategy
$\wedge,\vee$	logical and, or
U	number of cycles in PSDA
V	solutions generated per cycle U
S	sub-sample size
b_i	uniform random variable i

Acronyms

POE	port-of-entry
PSDA	probabilistic solution discovery algorithm
MC	Monte Carlo

Assumptions

1. Containers can only be classified as "suspicious" or "not suspicious"
2. Sensor reliability and cost characteristics are known for every sensor type
3. Thresholds for container classification for each sensor are given
4. At the $(n+1)$-echelon of the decision tree, each container has to be either manually checked (CHK) or released (OK)
5. There are no space, time or manpower limitations regarding container screening by any of the sensors

a weapon that could be shipped to any point in the country for future use. Both these scenarios would lead to devastating economic, psychological and sociological effects. However, the rationale for deterrence of these types of attack defer significantly. While the first one is related to obtaining a priori information that can be used to stop the container or vessel before reaching the POE, the second one relates to obtaining the necessary information on-site, to decide if the container should enter the country or not. The focus of this paper is related to the second type of attack and in general, to the historical roll of US Customs at the POE. Overall, this paper is concerned with developing an inspection strategy (via the strategic allocation of sensors) that minimizes the total cost of inspection while maintaining a user-specified detection rate for "suspicious" containers.

As discussed by Martonosi et al. [17]: "adopting a policy of 100 per cent inspection with current technology is not viable because of restrictions on land and personnel". The practical interpretation of such statement is that, currently many containers arriving at POE are released without ever being thoroughly inspected. From an economical perspective, to open each container and, to remove and inspect its cargo is unfeasible with the kind of resources available. Also, such an action would be both time-consuming and unrealistic. Thus, the release of "suspicious" containers is a threat the country has to live with. Yet, reducing such a threat to acceptable low levels is of utmost importance [4].

Container inspection technology has evolved significantly during the last decades. Currently, different sensors can be utilized to analyze the contents within a container. A frequently used non-invasive technique [5] involves the use of γ-rays that look for suspicious images and radiation detection, which traces radioactive signatures. Similarly, Hitachi has developed large-size X-ray digital radiography equipment to inspect containers loaded aboard a trailer. Material vibration waves, or ultrasound, have also been used as a means to look into these containers. Most of these non-invasive inspection techniques are sensor reading dependent. That is, once a reading is obtained, the container is identified as suspicious or not suspicious depending on a pre-defined threshold level. Whenever a container is defined as suspicious by the readings of a given sensor, it can be further inspected via another sensor or manually opened and inspected to check for suspicious material. Conversely, when the readings indicate the container is not suspicious, it can be further inspected via another sensor or definitely released.

Under the assumption that all these different sensor technologies are readily available, the POE faces a decision-making problem regarding how containers should be inspected. That is, build an inspection strategy that conforms to specific constraints. Thus, the purpose of this paper is two-fold.

First, a general model for describing an inspection strategy is proposed. In this respect, the inspection strategy is regarded as an $(n+1)$-echelon decision tree where at each of these echelons, a decision has to be taken, regarding which sensor to be used, if at all. In the last stage, the containers are either released, if their status is not suspicious, or checked manually, if their status is suspicious. The paper proposes a general $(n+1)$-echelon decision tree for any inspection strategy and via this representation, presents closed form mathematical expressions for inspection reliability, cost and sensor utilization.

Second, based on the general decision-tree model, this paper presents a minimum cost container inspection strategy that conforms to a pre-specified user detection rate under the assumption that different sensors with different reliability and cost characteristics can be used. To generate an optimal inspection strategy, an evolutionary optimization approach known as probabilistic solution discovery algorithm (PSDA) [6,7] has been used.

For this problem, a general overview of the PSDA is as follows. Initially, inspection strategies (i.e. $(n+1)$-echelon decision trees) are generated based on a randomly assigned probability distribution of sensor appearance at each echelon. For each of these strategies, reliability (understood in this paper as the detection rate) is obtained via a closed-form expression and, its associated cost is penalized depending on how they compare to a pre-defined user detection rate. Once this step is performed, strategies are ranked from lowest to highest with respect to the penalized cost and the probability distribution of sensor appearance is updated. The cycle is restarted until this distribution converges to a single solution or a stopping rule is enforced. Once the algorithm stops, a set of solutions is analyzed to choose the optimal strategy.

The remainder of the paper is organized as follows. Section 2 presents an overview of optimal inspection strategy development and work related to such a topic. In Section 3, inspection strategy modeling is discussed while, Section 4 illustrates the PSDA algorithm when applied to cost minimization subject to a user-defined detection rate. Finally, Section 5 presents examples and results while Section 6 concludes the paper.

2. Literature review

Boros et al. [8] originally analyzed and solved this problem through a large-scale linear programming model. Their solution provides not a single optimal strategy but a mixture of strategies each applied to a certain fraction of the containers. Also, it allows various container classifications outside the binary case considered in this paper. However, the development of the actual optimization model is quite involved for analysts without extensive background on linear programming and experience in mathematical modeling. Furthermore, there is no general description regarding the development of a decision tree for inspection strategies.

Elsayed et al. [9] analyzed the problem of developing strategies for containers that are inspected through a specific sequence to detect the presence of nuclear materials, biological and chemical agents, and other illegal shipments. These authors presented several optimization approaches on how to select sensor threshold levels under considerations of misclassification errors, total cost of inspection and budget. Two different perspectives were analyzed: first, the determination of the optimum configuration of the inspection strategy to achieve the minimum expected inspection cost and second, the determination of the optimum threshold of the inspection stations so as to minimize the cost associated with false reject and false accept. However, the strategies considered are focused on sensor networks configured as series, parallel and combination of these two configurations.

Although Boros et al. [8] and Elsayed et al. [9] are the only authors that develop an optimal inspection strategy via the allocation of sensors with different characteristics, the general problem incorporating sensor information to improve the reliability of decision making is not new. Chair and Varshney [10] employed readings from several sensing techniques (sonic, microwave, infrared and X-ray) to generate a multiple sensor system that improves individual sensor performance. They derived optimum fusion rules under the assumption that individual sensor characteristics were known. These rules are a function of the probability of false alarm and the reliability of individual sensors.

In similar studies, reliability has been defined as the probability that when an aggregated system of sensors or decision-making units is given a proposition, it makes the right decision. As discussed in Refs. [11–13], these systems have become critical because of their potential applications in security, target identification, safety and monitoring. Nordmann and Pham [11] provide an example in target identification and pattern recognition where an object needs to be identified based on some of its attributes. Similarly, Levitin [12] and Ramirez-Marquez [13] note that for safety, security and monitoring, different sensors may be used to observe some proposition and make a decision about its criticality. These last two authors have analyzed this type of sensor aggregation via multi-state reliability modeling [14].

In summary, although recent studies have contributed to the development of the-state-of-the-art in sensor reading analysis, data fusion and decision making for improving system inspection there is a lack of techniques to provide optimal solutions to the inspection strategy optimization problem.

3. Reliability of container inspection strategies

This section discusses a decision-tree-based approach for modeling the reliability of an inspection strategy. In the first part, the reliability of each of the sensors is discussed. This is followed by a discussion regarding the construction of a general decision tree for reliability analysis of container inspection and finally, a step-by-step example is introduced.

3.1. Sensor reliability

Classification of containers may include several categories usually depending on the observations made regarding specific attributes [13]. The case considered in this paper involves containers that can be classified by a sensor either as "suspicious" or as "not-suspicious".

Let r_j be the reading obtained from the jth sensor type regarding the status of a container and, the binary random variable X define the status of a container as follows:

$$X = \begin{cases} 0 & \text{if the container is not suspicious,} \\ 1 & \text{if the container is suspicious.} \end{cases}$$

Based on historical data [9] and without loss of generality, assume that $r_j|X=0 \sim N(\mu_{j0}, \sigma_{j0})$ and $r_j|X=1 \sim N(\mu_{j1}, \sigma_{j1})$. For each container, the vector $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ defines the threshold value for assigning one of the container classifications based on the reading of the jth sensor. To illustrate these concepts, Fig. 1 illustrates a graph of historical data associated with container readings for a POE sensor, as presented in Ref. [8].

Finally, let D_j define the container classification based on the reading of the jth sensor. Under the assumption that

$$D_j = \begin{cases} 0 & \text{if } r_j \leqslant t_j, \\ 1 & \text{otherwise,} \end{cases}$$

the following probabilities can be obtained for each sensor reading:

$$P^j_{yz} = P(D_j = y | X = z) = \begin{cases} \Phi^z_j(t_j) & \text{if } y = 0, \\ 1 - \Phi^z_j(t_j) & \text{if } y = 1, \end{cases} \tag{1}$$

where $\Phi^z_j(t_j)$ is the cumulative distribution function under $r_j|X=1 \sim N(\mu_{jz}, \sigma_{jz})$ for $z = 0$ or 1.

It is important to note that Eq. (1) provides insight into the reliability of the jth sensor by accounting for the probability of

(1) labeling a not suspicious container as suspicious, P^j_{10},
(2) labeling a suspicious container as not suspicious, P^j_{01}.

3.2. Inspection strategy as a decision tree

Decision trees allow for the clear understanding of a complex decision problem through a graphical representation. They also contribute to structuring a problem so that potential alternative outcomes can be analyzed. The first step towards developing a decision tree for describing a container inspection strategy is to graphically illustrate its structure. Fig. 2 presents a general $(n+1)$-echelon decision tree where, at each decision node of each echelon, one of

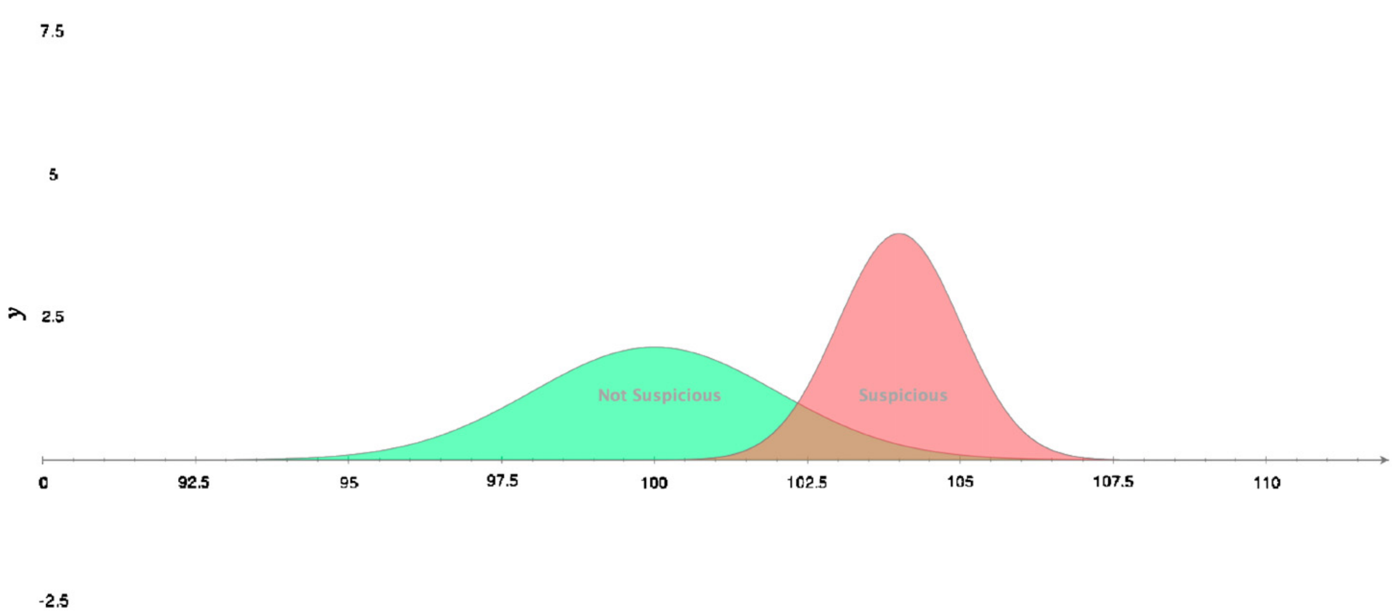

Fig. 1. Density function from historical sensor readings for containers.

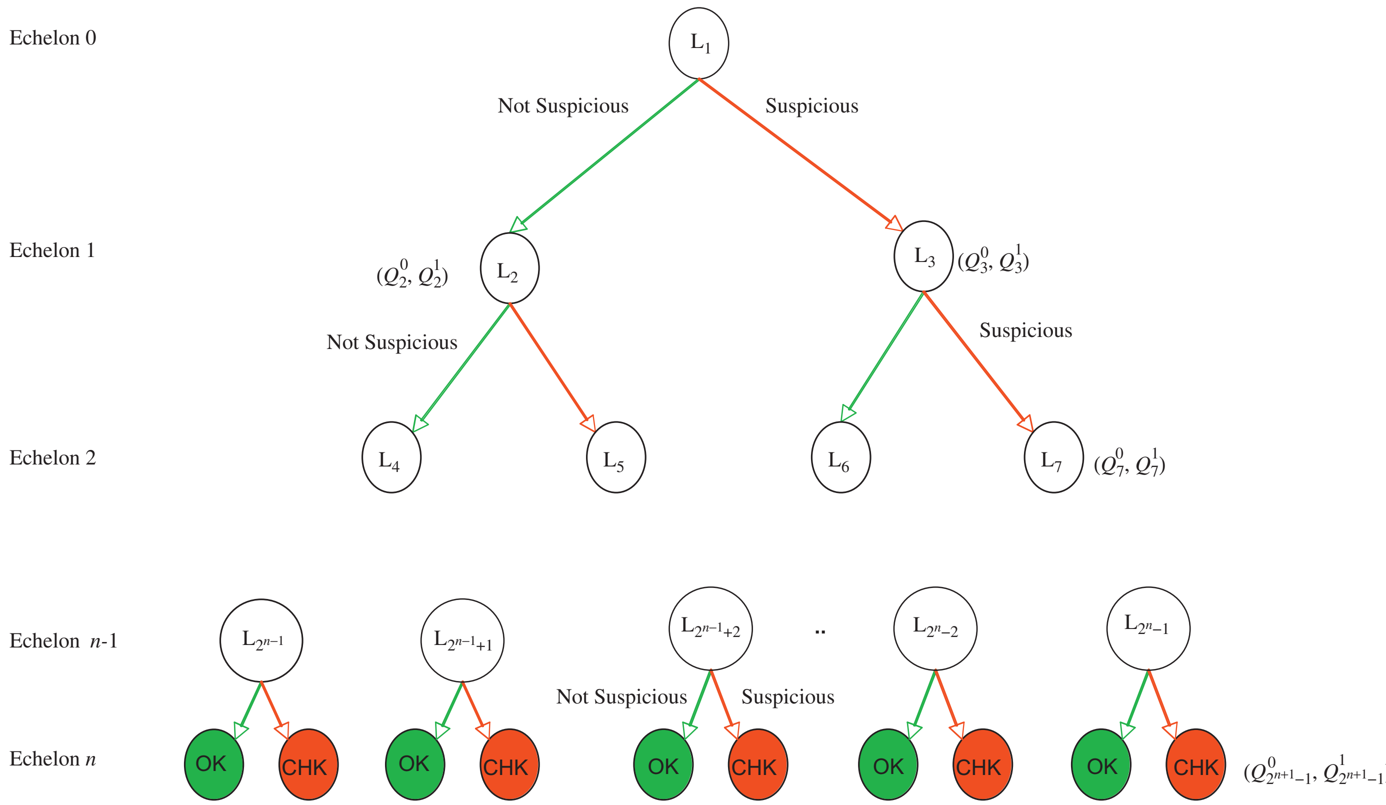

Fig. 2. Decision tree structure for container inspection strategy.

the n sensor types available can be located. The directed arcs that originate at each node, define the status of a container following such an arc. At the last echelon of the tree, and depending on its final location, a container can either be released if its final location is in a node labeled "OK" or, manually inspected if its final location is a node labeled "CHK".

In Fig. 2, the possible location for each of the n sensors has been labeled i, $i = 1,\ldots,2^{(n+1)}-1$. It is important to note that location i will receive an amount of "not suspicious" and "suspicious" containers, as dictated by

$$Q_i^z = Q_{\Omega(i)}^z\left[I_i^0 \Phi_{L(\Omega(i))}^z\left(t_{L(\Omega(i))}\right) + \left(1 - I_i^0\right) \times\left(1 - \Phi_{L(\Omega(i))}^z\left(t_{L(\Omega(i))}\right)\right)\right]. \quad (2)$$

In Eq. (2), Q_i^z mathematically represents the percentage of containers type z ($z = 0$ (not suspicious) or, $z = 1$ (suspicious)) at location i, as a function of (1) the percentage of containers in the location immediately preceding location i, $Q_{\Omega(i)}^z$, (2) the type of labeling of the container, I_i^0 and (3) the reliability of the sensor contained the location immediately preceding location i, $\Phi_{L(\Omega(i))}^z(t_{L(\Omega(i))})$. In this respect, $\Omega(i)$, represents the location immediately preceding location i. For example, for location $i = 2$, $\Omega(2) = \{1\}$. Similarly, $L(\Omega(i))$ contains the type of sensor used in the location immediately preceding location i. Also, the indicator variable I_i^0 defines the type of arc that precedes location i in the decision tree as follows:

$$I_i^0 = \begin{cases} 1 & \text{if the arc between } i \text{ and } \Omega(i) \text{ is of type } 0, \\ 0 & \text{otherwise.} \end{cases}$$

Before describing the quantifiers of interest, it is necessary to describe the rules that generally govern inspection strategies and also, how these rules impact Q_i^z.

Except for the first location, where only sensors can be allocated, in the remaining locations either one of the sensor types can be situated or, the container may be released or be manually inspected depending on the type of arc preceding such a location. This leads to the following rules governing the decision tree.

If at echelon h, $L(\Omega(i)) = \{\text{"OK"}\}$ (i.e. release) then at L_{2^h+i} and at L_{2^h+i+1}, $L(\Omega(2^h+i)) = \{\text{"OK"}\}$ and $L(\Omega(2^h+i+1)) = \{\text{"CHK"}\}$, respectively. This rationale is equivalent for the case when $L(\Omega(i)) = \{\text{"CHK"}\}$. To clarify, once a decision node in the tree is of the form "OK" (or "CHK"), its immediate successors should also be of the form "OK" or "CHK", respectively. The rationale for this rule is that there is no point in analyzing a container via a sensor once it has been manually checked. Equivalently, it is impossible to further analyze a container that has already been released. As an example, assume that in Fig. 2 at echelon 2, $L(\Omega(4)) = \{\text{"OK"}\}$ then, at echelon 3, $L(\Omega(8)) = \{\text{"OK"}\}$ and $L(\Omega(9)) = \{\text{"CHK"}\}$.

If a sensor has been used to inspect container type z, subsequent readings, of the same sensor and for the same container type z, at higher echelons, provide the exact same reading.

Based on these rules, Eq. (2) has to be transformed to

$$Q_i^z = \begin{cases} Q^z_{\Omega(i)} & \text{if } \left(L(\Omega(i)) = \{\text{“OK”}\} \wedge I_i^0 = 1\right) \\ & \vee\left(L(\Omega(i)) = \{\text{“CHK”}\} \wedge I_i^0 = 0\right) \\ & \vee\Big(\left(L(\Omega(i)) = L\left(\Omega\left(\alpha_i^1\right)\right) \vee L(\Omega(i)) = L\left(\Omega\left(\alpha_i^2\right)\right) \vee \cdots \vee L(\Omega(i)) = L\left(\Omega\left(\alpha_i^{h-1}\right)\right)\right) \wedge I_i^0 = I^0_{\alpha_i^1}\Big), \\ 0 & \text{if } \left(L(\Omega(i)) = \{\text{“OK”}\} \wedge I_i^0 = 0\right) \\ & \vee\left(L(\Omega(i)) = \{\text{“CHK”}\} \wedge I_i^0 = 1\right) \\ & \vee\Big(\left(L(\Omega(i)) = L\left(\Omega\left(\alpha_i^1\right)\right) \vee L(\Omega(i)) = L\left(\Omega\left(\alpha_i^2\right)\right) \vee \cdots \vee L(\Omega(i)) = L\left(\Omega\left(\alpha_i^{h-1}\right)\right)\right) \wedge I_i^0 \neq I^0_{\alpha_i^1}\Big), \\ \text{Eq. (2)} & \text{otherwise,} \end{cases}$$

where

$$\alpha_i^h = \left\lfloor \frac{i}{2^{h-1}} \right\rfloor.$$

Now, under the assumption that

$$S_{ij} = \begin{cases} j & \text{if location } i \text{ of the decision tree contains sensor type } j, \\ 0 & \text{otherwise.} \end{cases}$$

The vector $\mathbf{L}$ completely describes the decision tree at each echelon:

$$\mathbf{L} = \left(\underbrace{\sum_j S_{1j}}_{L_1}, \underbrace{\sum_j S_{2j}}_{L_2}, \ldots, \underbrace{\sum_j S_{2^n-1j}}_{L_{2^n-1}} \right).$$

Given that Q_i^z can be obtained for each and every location of the decision tree, each Q_i^z associated with the last echelon ($i = 2^n, \ldots, 2^{n+1}-1$) allows for the complete characterization of the following:

- The detection rate, $R(\mathbf{L})$, defined as the expected fraction of suspicious containers manually checked at the end of the inspection strategy. Mathematically

$$R(\mathbf{L}) = \sum_{l=0}^{2^n-2} Q^1_{2^n+(2l+1)}.$$

- The false-suspicious rate, $S^{-}(\mathbf{L})$, defined as the expected fraction of not suspicious containers manually checked at the end of the inspection strategy. Mathematically

$$S^{-}(\mathbf{L}) = \sum_{l=0}^{2^n-2} Q^0_{2^n+(2l+1)}.$$

- The utilization of each sensor type in the decision tree, $U_j(\mathbf{L})$, defined as the number of “not suspicious” containers inspected with sensor type j. Mathematically

$$U_j(\mathbf{L}) = \frac{1}{j} \sum_i S_{ij} Q_i^0.$$

- The inspection strategy cost, $C(\mathbf{L})$, defined as the cost of utilizing each different sensor plus the cost of manually inspecting false-suspicious containers. Mathematically

$$C(\mathbf{L}) = \sum_{j=1}^{n} c_j U_j(\mathbf{L}) + c_{\text{CHK}} S^{-}(\mathbf{L}).$$

Finally, it is important to note that based on the proposed model, time and manpower considerations can be addressed as a function of the utilization of each sensor.

3.3. Illustrative example

This example considers three types of sensors positioned at different locations of the decision tree. The reliability and cost characteristics are illustrated in Table 1 while Fig. 3 illustrates the inspection strategy. Finally, Table 2 illustrates quantifiers associated with each of the locations in the tree.

For this inspection strategy, rule 1 has been implemented in locations 8, 9, 12, 13, 14 and 15. Based on the quantities received at the locations in the last echelon, the quantifiers of interest have obtained as presented in Table 3.

Table 1
Reliability and cost characteristics of sensor types

Sensor	Container				Cost (in \$)
	Not suspicious		Suspicious		
	$X=0$	$X=1$	$X=0$	$X=1$	
1	0.995	0.005	0.517	0.483	10
2	0.858	0.142	0.323	0.677	14
3	0.877	0.123	0.041	0.959	55
CHK	1	0	0	1	285

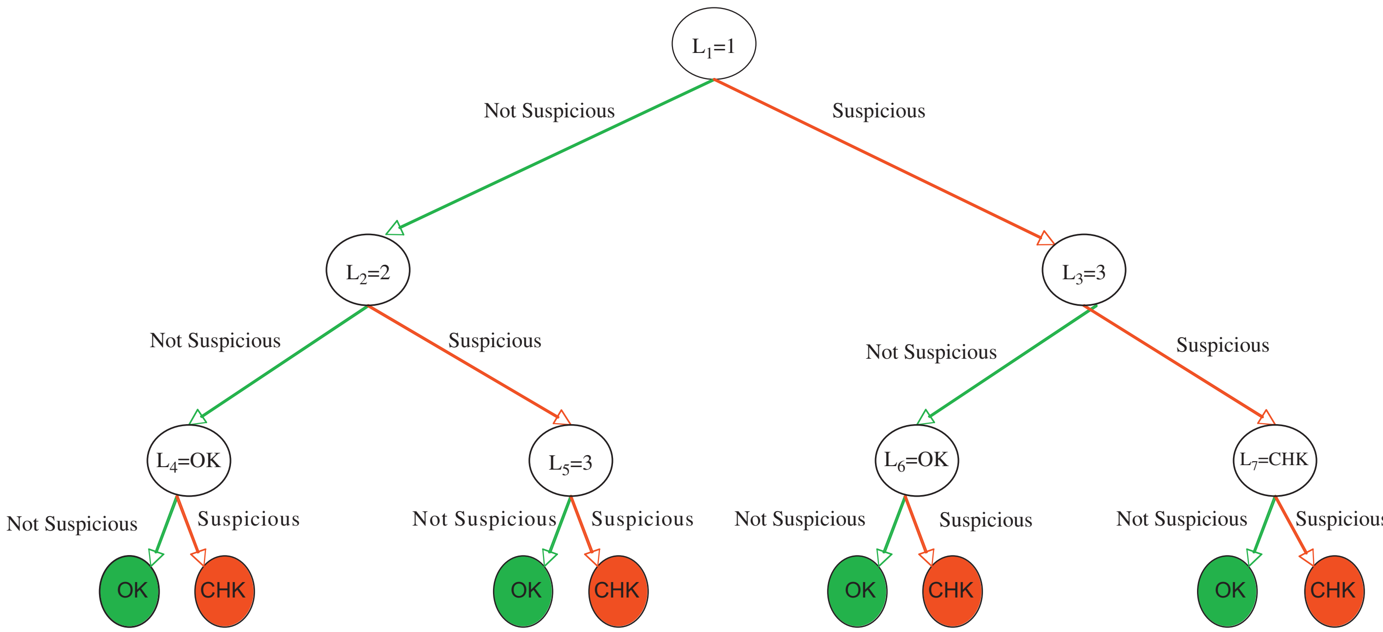

Fig. 3. Illustrative inspection strategy.

Table 2
Quantifiers associated with locations in the decision tree

i	Q_i^0	Q_i^1	$L(\Omega(i))$	$\Omega(i)$	I_i^0
1	100	100	$\emptyset$	$\emptyset$	0
2	99.53	51.69	1	1	1
3	0.47	48.31	1	1	0
4	85.37	16.68	2	2	1
5	14.16	35.01	2	2	0
6	0.41	1.98	3	3	1
7	0.06	46.34	3	3	0
8	85.37	16.68	OK	4	1
9	0	0	OK	4	0
10	12.42	1.43	3	5	1
11	1.74	33.58	3	5	0
12	0.41	1.98	OK	6	1
13	0	0	OK	6	0
14	0	0	CHK	7	1
15	0.06	46.34	CHK	7	0

Table 3
Results for illustrative inspection strategy

$U_1(\mathbf{L})$	100%
$U_2(\mathbf{L})$	99.53%
$U_3(\mathbf{L})$	14.63%
$R(\mathbf{L})$	79.91%
$S^-(\mathbf{L})$	1.80%
$C(\mathbf{L})$	37.11

4. Inspection strategy optimization

The mathematical model to be optimized in this paper is presented in Model 1. The objective of this model is to minimize the cost of an inspection strategy subject to a pre-specified reliability requirement. It is assumed that n sensors of equivalent functionality but with different reliability and cost can be used at any location of the decision tree.

Model 1:

$$\text{Min } C(\mathbf{L})$$

s.t.

$$R(\mathbf{L}) \geqslant R_0$$

$$S_{ij} \in (1, 2, \ldots, n, \text{“OK”}, \text{“CHK”})$$

4.1. Algorithm description

The algorithm proposed in this paper generates solutions to Model 1 through a global approach that contains three interrelated steps. These steps are based on the probabilistic generation of potential inspection strategies (via Monte Carlo (MC) simulation), the analysis of these solutions (via the technique described in Section 3) and the selection of new potentially optimal inspection strategies (via an evolutionary optimization technique).

In the Strategy Development step, the first of the three steps of the algorithm, and based on the probabilities defined by the vector $\boldsymbol{\gamma}_{iu}$, MC simulation is used to generate a specified number (defined by V) of potential inspection strategies, $\mathbf{L}_u^v$ ($v = 1, \ldots, V$). For each location, i, $\boldsymbol{\gamma}_{iu}$ defines the probability that a sensor type j will be present in the optimal inspection strategy. This step also contains the stopping rules of the algorithm. In essence, the first rule dictates the algorithm to be stopped once, for every i at the current cycle u, $\boldsymbol{\gamma}_{iu}$ can no longer be updated (i.e. all initial "appearance" probabilities are either zero or one). The second rule allows the user to set a specific number of cycles. In the context of this algorithm, a cycle is understood as every time the value u is updated.

The second step, Inspection Strategy Reliability Analysis, implements the approach discussed in Section 3 to obtain the reliability, the false suspicious rate, the utilization of each sensor and the cost associated with each of the potential inspection strategies, $\mathbf{L}_u^v$.

The final step in the algorithm penalizes the cost of potential inspection strategies whenever they violate the user-defined reliability requirement. The solutions are then ranked in decreasing order of magnitude with respect to the penalized cost. A specified number of these solutions (defined by TOP) are stored in set K and then, a subset of size S of the whole set of solutions, is used to update the probabilities defined by the vector $\boldsymbol{\gamma}_{iu}$. This new vector is sent to Step 1 to check for termination or for solution discovery. Finally, once either of the stopping rules has been met, the best solution in set K is chosen. The pseudo code of the SDA optimization is as follows.

Inspection Strategy Probabilistic Solution Discovery Algorithm:

$Initialize: \quad V, S, U, \alpha, \boldsymbol{\gamma}_{iu}, \quad u = 1,\ K = \emptyset$

Step 1: (*Strategy Development*)

For $v = 1, \ldots, V$, generate a potential inspection strategy $\mathbf{L}_u^v$, $\mathbf{L}_u^v = \left(L_{1u}^v, L_{2u}^v, \ldots, L_{(2^n-1)u}^v\right)$ with $L_{iu}^v = \sum_j S_{ij}$

{For $i = 1, \ldots, 2^n - 1$ and $j = 1, \ldots, n, n+1$ (where $n+1 =$ "OK" or "CHK") generate a random number b_i, $b_i \in (0,1)$ via MC simulation. As dictated by $\boldsymbol{\gamma}_{iu}$, with $\boldsymbol{\gamma}_{iu} = (\gamma_{iu}^1, \ldots, \gamma_{iu}^2, \ldots, \gamma_{iu}^n, \gamma_{iu}^{\mathrm{OK}}$ or $\gamma_{iu}^{\mathrm{CHK}})$, $\gamma_{iu}^j = P(S_{ij} = j)$ and $\sum_j \gamma_{iu}^j = 1$:

if $\gamma_{iu}^1 \geqslant b_i \rightarrow S_{i1} = 1$

else $\gamma_{iu}^{j-1} < b_i \leqslant \gamma_{iu}^j \rightarrow S_{ij} = j, \quad j = 2, \ldots, n+1$

$\sum_j \frac{S_{ij}}{j} = 1$

$v \rightarrow v + 1;$}

if $\left(\gamma_{im}^j = 1 \vee \gamma_{im}^j = 0 \forall i, j\right) \vee u = U$, then

$\mathbf{L}^* = \arg\max_{L \in K}\left\{\bar{C}(\mathbf{L})\right\}$. Stop.

Else Go to step 2.

Step 2: (*Inspection Strategy Reliability Analysis*)

For $v = 1, \ldots, V$ obtain via decision-tree approach of Section 3:

$R\left(\mathbf{L}_u^v\right), S^-\left(\mathbf{L}_u^v\right), U_j\left(\mathbf{L}_u^v\right)$ and $C\left(\mathbf{L}_u^v\right)$

Go to Step 3.

Step 3: (*Solution Discovery*)

For $v = 1, \ldots, V$ {

$$\Delta\left(\mathbf{L}_u^{(v)}\right) = \begin{cases} \left[1 - \left(R_0 - R\left(\mathbf{L}_u^v\right)\right)\right]^\alpha & R\left(\mathbf{L}_u^v\right) \leqslant R_0 \\ 1 & \text{o.w.} \end{cases}; \bar{C}\left(\mathbf{L}_u^v\right) = \frac{C\left(\mathbf{L}_u^v\right)}{\Delta\left(\mathbf{L}_u^v\right)}; \}$$

List $\bar{C}\left(\mathbf{L}_u^v\right)$ by increasing order of magnitude:

$\bar{C}\left(\mathbf{L}_u^{(1)}\right) \leqslant \bar{C}\left(\mathbf{L}_u^{(2)}\right) \leqslant \cdots \leqslant \bar{C}\left(\mathbf{L}_u^{(v)}\right) \leqslant \cdots \leqslant \bar{C}\left(\mathbf{L}_u^{(V)}\right);$

$K \rightarrow K \cup \left\{\bar{C}\left(\mathbf{L}_u^{(1)}\right), \bar{C}\left(\mathbf{L}_u^{(2)}\right), \ldots, \bar{C}\left(\mathbf{L}_u^{(\mathrm{TOP})}\right)\right\};$

$u \rightarrow u + 1;$

For $i = 1, \ldots, 2^n - 1$ and $j = 1, \ldots, n$, "OK", "CHK" update vector $\boldsymbol{\gamma}_{ui}$ as follows:

$$\gamma_{iu}^j = \frac{\sum_{v=1}^{S} \left(\frac{L_{iu}^{(v)}}{j}\right)_{L_{iu}^{(v)} = j}}{S} \text{ where } S << V.$$

Go to Step 1.

4.2. Discussion parameter of tuning

4.2.1. Initial vector of probabilities (γ_{iu})

The rational behind assigning initial values to this vector is based upon the lack of knowledge regarding the sensors that should constitute the final inspection strategy so that Model 1 is optimally solved. Thus, this lack of knowledge, also known as "Laplace principle of insufficient reason", translates into initially providing the same probability of appearing in the final solution to each and every sensor type. Moreover, this same lack of information forces the choice of a sensor as part of the solution or not, to be equally likely.

4.2.2. Penalization factor (α)

The penalization factor employed in the algorithm is used to explore unfeasible neighborhoods of the solution space, with the expectation that, within the boundary of these areas, quasi-optimal solutions may be obtained. The rational behind setting this value has been first to define the maximum allowable increase in cost as a function of the distance between the reliability of unfeasible solutions and the requirement. Experimentation on these types of problem has shown that a value of $\alpha = 25$ provides good results.

4.2.3. Sub-sample size (S)

The sub-sample size S is used to update the probabilities defined by the vector $\boldsymbol{\gamma}_{iu}$. The rational for setting this value is based on work developed by Bäck and Schwefel [15], and Schwefel and Bäck [16] in the area of evolutionary strategies and its application to optimization algorithms that imitate certain principles of nature. In this respect, a population of individuals (a possible network configuration in this paper) collectively evolves towards better solutions by means of a parent's selection process, a reproduction strategy and a substitution strategy.

These studies [15,16] define a type of strategy, denoted by ES(μ,λ), where μ initial individuals, generate λ offspring and, selection takes place only among these λ offspring. A ratio of $\lambda/\mu \approx 7$ is suggested as a good setting for the relation between parent and offspring population size.

The algorithm described in Section 4.1 works in a similar fashion since from a sample of size V of potential inspection strategies, the algorithm selects S among the best solutions, according to the penalized cost. This means that, for example, from 100 potential strategy designs, the best 14 are selected. If $V=200$, then S should be approximately equal to 28. However, preliminary tests indicate that excellent results are obtained by using a fixed value of $S=20$.

4.2.4. Generation size (V)

The number of potential inspection strategies (V) has been set considering that the maximum allowable number of solutions searched ($U\times$ SAMPLE) would correspond to at most 10% of the solutions searched.

4.3. Algorithm complexity

Step 1 corresponds to the Strategy Development. First, the generation of a strategy $\mathbf{L}_u^v$, generation requires in the worst case $O(n\times[2^n-1])$ operations. Since V strategies are generated, Step 1 is $O(V\times n\times[2^n-1])$.

Step 2 corresponds to the Inspection Strategy Reliability Analysis of each potential design $\mathbf{L}_u^v$. In this respect, both the detection rate, $R(\mathbf{L}_u^v)$, and the false-suspicious rate, $S^-(\mathbf{L}_u^v)$ require each $O(2n\times[2^n-1])$. The utilization of each sensor type in the decision tree, $U_j(\mathbf{L}_u^v)$, requires $O(2^n-1)$ while the inspection strategy cost, $C(\mathbf{L}_u^v)$, requires $O(n+1)$.

Step 3, Solution Discovery, initially requires $O(V)$ to evaluate the penalized cost for each potential solution and $O(V^2)$ or $O(V\times\log(V))$ for sorting, depending on the sorting algorithm used. The update of vector γ_{iu} requires $O(S\times n)$ operations. So Step 3 is $O(V^2+[S\times n])$.

5. Illustrative example and results

This section describes the implementation of the strategy decision tree and of the optimization approaches previously described. As discussed by Boros et al. [8], currently the types of different sensors, V, available is relatively small (i.e. $V\leqslant 4$). Thus, a problem with four different sensors originally solved in Ref. [8] has been chosen for illustrative and comparison purposes.

The details of the problem are as follows. The readings of "not suspicious" containers follow a Normal distribution of the form $N(0,1)$ for all sensor types. The reading of "suspicious" containers for sensors 1–4, follow the subsequent distributions: (1) $0.5N(0,1)+0.5N(4.37,1)$, (2) $N(1.53,1)$, (3) $N(2.9,1)$ and (4) $N(4.6,1)$. Based on these distributions, Table 4 illustrates the probability of a container being classified as "not suspicious" or "suspicious" with respect to user-specific thresholds. Also, this table illustrates the cost incurred when employing each of the sensors as well as when manually checking a container. Finally, the strategy developed should provide a detection rate of at least 81.5% at minimum cost.

For these type of problems, the number of possible decision trees [8] is 2^{2^n}. For the case presented, 2^{16}. Based on the size of this search space, it was assumed the algorithm would search at most 5% of all the solutions in eight cycles. Thus, initialization parameters are as follows: $U=8$, $V=420$, $S=60$, $\alpha=25$.

Table 4
Reliability and cost characteristics of sensor types

Sensor	Container				t_j	Cost (in $)
	Not suspicious		Suspicious			
	$X=0$	$X=1$	$X=0$	$X=1$		
1	0.995	0.005	0.517	0.483	2.6	0.32
2	0.858	0.142	0.323	0.677	1.07	0.92
3	0.877	0.123	0.041	0.959	1.16	57
4	0.986	0.014	0.008	0.992	2.2	176
CHK						600

Table 5 illustrates the transformation of vector γ_{iu} from its initial setting up to cycle 7, immediately after which, the solution converges. Table 6 provides the best solution obtained at each of these cycles. Finally, Fig. 4 graphically illustrates the decision tree associated with these best solution obtained (Fig. 4A) together with the solution obtained in Ref. [8] (Fig. 4B). The algorithm obtained such a solution, however, after thorough examination, the solution is deemed infeasible since it violates the reliability constraint. The difference may be due to the cut off of significant figures. In order to avoid this concern, the algorithm has been implemented to obtain results with four significant figures.

To illustrate the algorithm, one can follow the convergence of the vector γ_{2u} (i.e. the sensor used in location 2). Initially, and based on the discussion in Section 4.2.1, every possible sensor type, along with the possibility of releasing a container (i.e. "OK"), is given the same probability of appearance. Thus, in the first step of the algorithm, vector γ_{iu} for $i=2$ and $U=1$ in Table 5 guarantees that the random variable b_i is equally likely to select any of these choices. Immediately, after the first cycle, γ_2 has been transformed from $\gamma_{21}=(0.2,\ 0.2,\ 0.2,\ 0.2,\ 0.2)$ to $\gamma_{22}=(0.325, 0.250, 0.250, 0.175, 0.000)$.

In Table 5, for purposes of illustration, the position of "OK"/"CHK" possibility in γ_{iu}, has been changed to the first position. The discussed transformation has the practical significance that, just after the first $V=420$ solutions were analyzed, the algorithm was able to identify the option of including sensor type 4 in the second position in the final solution as too costly an option. At cycle $U=3$, the algorithm decides that the viable options are sensor type 1 or sensor type 2 with $\gamma_{23}=(0.000,\ 0.525,\ 0.475,\ 0.000,\ 0.000)$. And, finally converging on the latter type at cycle $U=6$. A similar description can be given for the remaining vectors γ_{iu}.

Table 6 illustrates how the best solution per cycle changes. It should be noted that just after 840 solutions are analyzed, representing less than 1.5% of the solution space, good-quality solutions in terms of cost are obtained. The algorithm identifies the best solution during the fifth cycle

Table 5
Transformation of vector γ_{iu} per algorithm cycle

	γ_{1u}	γ_{2u}	γ_{3u}	γ_{4u}	γ_{5u}	γ_{6u}	γ_{7u}	γ_{8u}	γ_{9u}	γ_{10u}	γ_{11u}	γ_{12u}	γ_{13u}	γ_{14u}	γ_{15u}
$U=1$															
CHK/OK	0.0000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
1	0.2500	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
2	0.5000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000	0.6000
3	0.7500	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000	0.8000
4	0.1000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=2$															
CHK/OK	0.0000	0.3250	0.3000	0.5750	0.1111	0.1786	0.4250	0.8000	0.1176	0.3750	0.2593	0.3929	0.2174	0.0870	0.6250
1	0.2750	0.5750	0.4000	0.8000	0.4444	0.3571	0.5000	0.9250	0.5294	0.5000	0.4815	0.5357	0.2609	0.2609	0.6500
2	0.6000	0.8250	0.6250	0.9500	0.5185	0.5000	0.5750	1.0000	0.6471	0.5417	0.6296	0.6071	0.4783	0.3913	0.6750
3	1.0000	1.0000	0.7250	1.0000	0.8148	0.8214	0.6750	1.0000	0.7059	0.7917	0.8519	0.7500	0.6522	0.6957	0.8750
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=3$															
CHK/OK	0.0000	0.0000	0.2000	0.7500	0.0000	0.2188	0.4500	0.9750	0.2000	0.3750	0.2000	0.4688	0.3200	0.0909	0.8750
1	0.4750	0.5250	0.2500	0.9000	0.2250	0.3438	0.5750	1.0000	0.4000	0.5000	0.2250	0.6250	0.3200	0.1364	0.9000
2	1.0000	1.0000	0.4250	1.0000	0.2750	0.3750	0.6000	1.0000	0.5000	0.5000	0.4000	0.6875	0.4800	0.2273	0.9000
3	1.0000	1.0000	0.5250	1.0000	0.6250	0.7813	0.6500	1.0000	0.5000	0.7000	0.8500	0.7500	0.5600	0.7273	0.9500
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=4$															
CHK/OK	0.0000	0.0000	0.1000	0.6750	0.0000	0.3889	0.4750	0.9750	0.0000	0.5500	0.1500	0.6389	0.2727	0.0952	0.9500
1	0.7500	0.2500	0.1750	0.8750	0.2500	0.5000	0.6250	1.0000	0.3077	0.7750	0.2000	0.7500	0.2727	0.1429	0.9500
2	1.0000	1.0000	0.3250	1.0000	0.2750	0.5278	0.6750	1.0000	0.3846	0.7750	0.3500	0.7778	0.3636	0.2381	0.9500
3	1.0000	1.0000	0.5000	1.0000	0.7750	0.8333	0.6750	1.0000	0.3846	0.8750	0.9000	0.8611	0.4545	0.9048	0.9500
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=5$															
CHK/OK	0.0000	0.0000	0.0250	0.7250	0.0000	0.4615	0.5000	1.0000	0.0000	0.6500	0.2500	0.7949	0.2857	0.1000	1.0000
1	0.9750	0.0250	0.1000	0.9500	0.2500	0.5641	0.6250	1.0000	0.4545	0.9250	0.3000	0.8974	0.2857	0.1500	1.0000
2	1.0000	1.0000	0.2000	1.0000	0.2750	0.5897	0.6500	1.0000	0.5455	0.9250	0.5000	0.9231	0.2857	0.2000	1.0000
3	1.0000	1.0000	0.2250	1.0000	1.0000	0.8718	0.6500	1.0000	0.5455	0.9250	0.8750	0.9231	0.3333	0.8500	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=6$															
CHK/OK	0.0000	0.0000	0.0500	0.8750	0.0000	0.5526	0.4750	1.0000	0.0000	0.6000	0.2250	0.8684	0.3529	0.0952	1.0000
1	1.0000	0.0000	0.1000	0.9500	0.0000	0.6316	0.6250	1.0000	0.6000	1.0000	0.2250	0.8947	0.3529	0.1905	1.0000
2	1.0000	1.0000	0.1750	1.0000	0.0000	0.6316	0.6250	1.0000	0.8000	1.0000	0.2250	0.8947	0.3529	0.2381	1.0000
3	1.0000	1.0000	0.1750	1.0000	1.0000	0.8684	0.6250	1.0000	0.8000	1.0000	0.2250	0.8947	0.4118	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
$U=7$															
CHK/OK	0.0000	0.0000	0.0000	1.0000	0.0000	0.1000	0.6250	1.0000	1.0000	0.8000	0.0000	0.9750	0.3333	0.1333	1.0000
1	1.0000	0.0000	0.1500	1.0000	0.0000	0.1000	0.7500	1.0000	1.0000	1.0000	0.0000	1.0000	0.3333	0.1333	1.0000
2	1.0000	1.0000	0.1500	1.0000	0.0000	0.1000	0.7500	1.0000	1.0000	1.0000	0.0000	1.0000	0.3333	0.1333	1.0000
3	1.0000	1.0000	0.1500	1.0000	1.0000	0.9500	0.7500	1.0000	1.0000	1.0000	0.0000	1.0000	0.5278	1.0000	1.0000
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

converging to such a solution at cycle 8. In summary, at each cycle, the algorithm focuses in a new region of the search space that can, potentially or probabilistically, provide the best solution. As previously mentioned, the best solution obtained is illustrated in Fig. 4A together with Fig. 4B providing the decision tree described in Ref. [8].

6. Conclusions and future research

This paper presented an approach for developing an inspection strategy that minimizes the total cost of inspection while maintaining a user-specified detection rate for ‘‘suspicious’’ containers. The approach has been described as a two-part process. In the first part, a general model was presented for defining inspection strategy via an $(n+1)$-echelon decision tree where at each of these echelons, a sensor may be used. The second part, presented an evolutionary optimization approach, known as probabilistic solution discovery algorithm, to generate a minimum cost container inspection strategy that conforms to a pre-specified user detection rate under the assumption that different sensors with different reliability and cost characteristics can be used. Results show the approach can provide quasi-optimal solution by analyzing at most 5% of the solution space.

Table 6
Best solutions obtained per algorithm cycle

U	Location															$C(\mathbf{L})$	$R(\mathbf{L})$
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1	1	2	4	OK	1	3	4	OK	CHK	OK	3	2	2	1	CHK	21.34	0.8176
2	1	2	CHK	OK	3	OK	CHK	OK	CHK	OK	4	OK	CHK	OK	CHK	15.32	0.8161
3	1	2	4	OK	1	3	CHK	OK	CHK	4	3	OK	2	OK	CHK	20.98	0.8175
4	1	2	CHK	OK	3	OK	CHK	OK	CHK	OK	4	OK	CHK	OK	CHK	15.32	0.8161
5	1	2	4	OK	3	3	CHK	OK	CHK	OK	4	OK	CHK	OK	CHK	13.98	0.8159
6	1	2	4	OK	3	3	CHK	OK	CHK	OK	4	OK	CHK	OK	CHK	13.98	0.8159
7	1	2	4	OK	3	3	CHK	OK	CHK	OK	4	OK	CHK	OK	CHK	13.98	0.8159

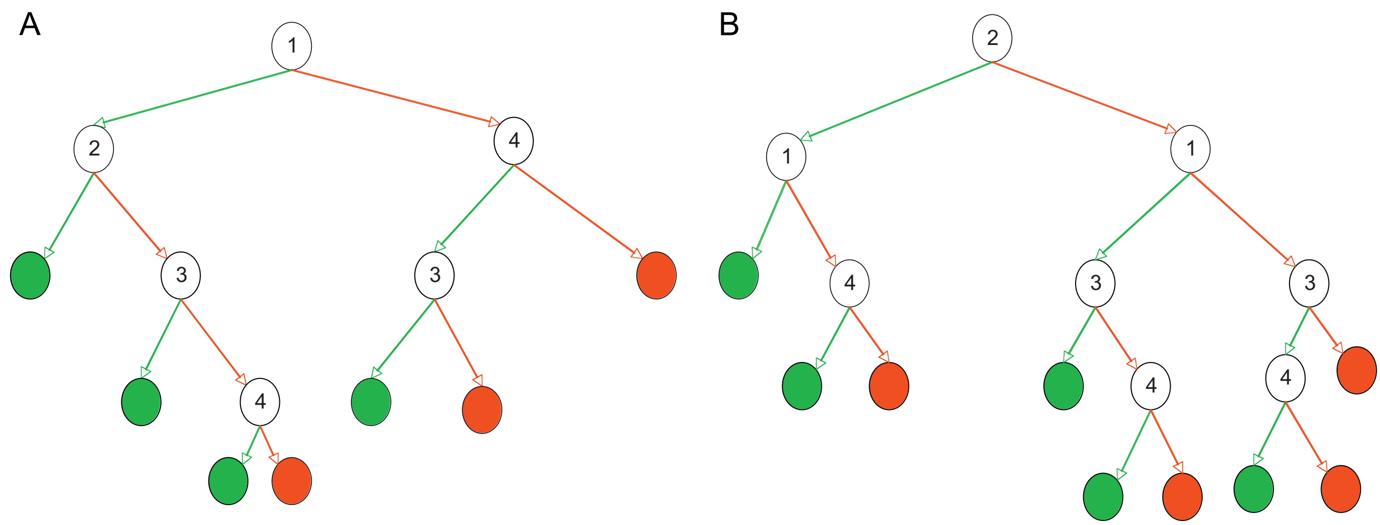

Fig. 4. Illustration of best inspection strategies. (A) PSDA and (B) [6].

Future research efforts will be devoted to improve the approaches presented to allow (1) multiple classification of containers and (2) search for the optimal value of the thresholds for each sensor.

References

[1] Wein L, Wilkins A, Baveja M, Flynn S. Preventing the importation of illicit nuclear materials in shipping containers. Risk Anal 2006; 26(5):1–35.
[2] Richardson H, Gordon P, Moore J, editors. The economic impacts of terrorist attacks. Edward Elgar; 2005. p. 218–41.
[3] Gully E. Assessment of our national infrastructure strategy. Strategy research project, 2006. United States Army War College.
[4] Intelligence assessment terrorist use of containerized shipping: a threat overview. State of New Jersey, Office of Homeland Security & Preparedness, October 12, 2007.
[5] James K, de Sulima-Przyborowski J, Haydn N. Non invasive means of investigating container contents for customs agents at port. University Southern California; 2002.
[6] Ramirez-Marquez JE, Rocco C. Probabilistic knowledge discovery algorithm in network reliability optimization. In: Proceedings of ESREL annual conference, Stavanger, Norway, June 2007.
[7] Ramirez-Marquez JE, Rocco CM. All-terminal network reliability optimization via probabilistic solution discovery. Reliab Eng Syst Saf 2008, doi:10.1016/j.ress.2008.01.001.
[8] Boros E, Fedzhora L, Kantor P, Saeger K, Stroude P. Large scale LP model for finding optimal container inspection strategies. Rutcor research report, no. RRR-26-2006, 2006.
[9] Elsayed A, Schroepfer C, Xie M, Zhang H, Zhu Y. Port-of-entry inspection: sensor deployment policy optimization. Rutgers IE working paper, 07-012, 2007.
[10] Chair Z, Varshney P. Optimal data fusion in multiple sensor detection systems. IEEE Trans Aerospace Electron Syst 1986; 22(1):98–101.
[11] Nordmann L, Pham H. Weighted voting systems. IEEE Trans Reliab 1999;48(1):42–9.
[12] Levitin G. Asymmetric weighted voting systems. Reliab Eng Syst Saf 2002;76:205–12.
[13] Ramirez-Marquez JE. A new holistic reliability analysis of weighted voting systems from a multi-state perspective. IIE Trans 2008; 40(2):122–32.
[14] Ramirez-Marquez JE, Coit D, Tortorella M. A generalized multi-state based path vector approach for multi-state two-terminal reliability. IIE Trans 2006;38(6):477–88.
[15] Bäck Th, Schwefel HP. Evolutionary computation: an overview. In: Proceedings of the 1996 IEEE international conference on evolutionary computation (IECC'96), Nagoya, Japan. NY: IEEE Press; 1996. p. 20–9.

[16] Schwefel HP, Back Th. Evolution strategies I: variants and their computational implementation. In: Periaux J, Winter G, editors. Genet algorithm in engineering and computer science. Wiley; 1995.

[17] Martonosi SE, Ortiz DS, Willis HH. Evaluating the viability of 100 percent container inspection at America's ports. In: Richardson H, Gordon P, Moore J, editors. The economic impacts of terrorist attacks. Cheltenham, UK: Edward Elgar Publishing; 2005. p. 218–41 [Chapter 12].