ARTICLE IN PRESS

Computers & Operations Research ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Sequencing counts: A combined approach for sequencing and selecting costly unreliable off-line inspections

Tal Avinadav [a], David Sarne [b,*]

[a] Department of Management, Bar-Ilan University, Ramat Gan 52900, Israel
[b] Department of Computer Science, Bar-Ilan University, Ramat Gan 52900, Israel

ARTICLE INFO

Keywords:
Off-line inspections
Branch and bound
Sequencing
Monte-Carlo simulation

ABSTRACT

We study the case of "inspect-all" policy, using off-line quality inspections to prevent non-conforming items from reaching the final consumer, in domains where an item is rejected upon first "failure" classification. Given a set of inspections with known inspection costs and error probabilities of two types (classifying conforming items as non-conforming and vice versa), the goal is to find a sequenced subset of inspections that maximizes the expected overall profit, taking into account the revenue from delivering conforming items, the penalty of delivering non-conforming ones, and the overall cost of the inspections used. Our model allows an additional degree of freedom, in comparison to prior work in this domain, enabling the selection of inspections sequence along the selection of which inspections to use. We present an efficient branch and bound algorithm for finding the optimal solution, and two types of heuristics: greedy-based and preliminary sort-based, differing in their accuracy and calculation time. The optimal and heuristic methods are extensively evaluated, using a factorial experimental design that includes 65 610 problem instances. For each instance we compared the methods performance in terms of reaching optimality, deviation from the optimal solution and calculation-time. The results reflect a substantial influence of the sequence over the expected profit. An interesting finding is that the suggested preliminary sort-based heuristics achieve a relatively accurate solution in a reasonable calculation-time and outperform the commonly used greedy-based heuristics. The usefulness of the different methods is illustrated using sample problems from the biometric inspection security domain.

1. Introduction

Off-line inspections are a fundamental means for fault detection and quality assurance in various domains from health care through software development to assembly lines. In many cases, off-line inspections are the only means available as process control or on-line quality assurance is infeasible (as in the case of antivirus inspections or medical tests). In this paper we focus on costly inspections with a dichotomous output: the item is either classified as "conforming" or "non-conforming" and every item must pass a series of off-line inspections [24–26,2,29]. While in some domains the inspection policy used is sampling-based (in order to reduce inspection costs) [3,10,30–33], the "inspect-all" policy is highly common whenever the consequences of failing to identify a problem are substantial. For example, in the car industry, some critical safety-systems are inspected one by one. Similarly, in health-care screening exams and in multi-biometric security system (MBSS), each person needs to go through the full set of exams. Finally, when downloading a set of files to a computer, the antivirus software executes its full set of testing algorithms over all files.

As accurate as inspections can be, and despite technology advances, they are vulnerable to errors. For example, in many inspections the classification is based on norm ranges, resulting with misclassification (e.g., medical tests). Other inspections are very reliable, however, vulnerable to errors associated with calibration of the inspection equipment.[1] Similar to most works in this domain [28,35,27,15,14,34,3,31], we classify the errors according to two types: Type I—classifying a conforming item as non-conforming, and Type II—classifying a non-conforming item as conforming. As in the majority of applications, both Type I and Type II errors incur costs. For example, Type I errors may lead to loss of income, as well as loss of work and materials used to produce the item. On the other hand, Type II errors may lead to penalties, reputation loss and replacement costs resulting from supplying a non-conforming item.

* Corresponding author.
E-mail address: sarned@cs.biu.ac.il (D. Sarne).

[1] E.g., according to the National Institute of Standards and Technology report on the impact of calibration error in clinical decision making and health care costs, calibration error in measurements of serum calcium levels led to analytic bias in 15% of tests results with estimated errors costs, on a national scale, ranging from \$60 million to \$199 million per year [19].

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.12.018

Commonly, several, and possibly numerous, independent classification inspections are available, varying in their reliability and inspection cost. The inspection plan may include any subset of the potential inspections in any order. For example, a medical diagnosis usually relies on several tests, an antivirus software uses several different checks and algorithms for diagnosing a file, and biometric-based identification systems usually rely on different metrics for classifying an item. There are two main approaches for determining when to terminate an inspection process. The first does not immediately reject an item if classified as non-conforming by an inspection, and may let the item go through further inspections in order to reduce the loss associated with Type I error [25,26,7]. The second approach terminates upon receiving a non-conforming classification from an inspection, i.e., rejects upon first "failure" classification [22,9,29,2]. This latter approach is commonly used in many legacy domains. For example, antivirus software quarantines files if one of the inspections labels them as infected or a threat; blood and sperm donations are rejected if at least one inspection indicates that they are infected or below some quality threshold; candidates in a hiring process do not proceed to the next phase upon failing in an interview. In particular, the latter approach is necessarily used when no evidence that the item has actually failed an inspection is allowed.[2] The model analyzed in this paper relies on the latter approach.

The objective function, commonly used in such environments, is to maximize the expected profit per item, given the revenues and losses associated with delivering a conforming or a non-conforming item correspondingly. The problem is to decide which inspection subset should be used and in what execution order.[3] Adding more inspections reduces the Type II error, as the number of non-conforming items decreases between subsequent inspections. However, at the same time, it increases the Type I error (as some of the conforming items are rejected in the transition between inspections) as well as the total cost of inspections along the process.

Models combining selection and sequencing of offline inspections have been proposed primarily for domains in which items are not immediately rejected when an inspection classifies them as non-conforming. The most relevant to our case is the work of Raz and Bricker [26], which presents three types of inspection selection methods: complete, fixed and variable. In the complete sequence method, all units go through all the selected inspections; thus, there is no importance to the internal order of inspections. The number of possible policies in this case is 2^n for a set of n potential inspections, since each inspection can be either used or not. A branch and bound algorithm for finding the optimal subset of inspections for this method is proposed, relying on calculating lower and upper bounds of the potential expected revenue that can be obtained from each subset. In the fixed sequence method, the sequence of inspections selected is also the same for all units; however, the decision whether to continue to the next inspection in the sequence depends on the results obtained in previous inspections (thus, the decision to continue to the k-th inspection is based on up to 2^{k-1} possible combinations). The number of possible sequences for a set of n potential inspections is approximately $O(n!)$, and for possible sequence an exponential number of policies needs to be devised. A branch and bound algorithm is proposed for finding the optimal sequence in this case. Finally, in the variable sequence method, each unit may continue through a different sequence based on the results of prior inspections. The number of possible sequences is approximately $O(n^n)$, and again for each possible sequence an exponential number of policies needs to be devised. For this latter method a heuristic solution is provided based on limiting the search to the subset of inspections with the highest improvement potential. For the variable sequence method, two additional heuristics, rank-based and difference-based, are given in [25]. Both heuristics use a greedy approach, considering, at each iteration, the contribution of a single inspection to the cost of the current sequence. Despite the economic usefulness of this latter approach, it is inapplicable in our case, as our model does not allow an item to be delivered if any inspection has found it to be non-conforming.

Prior work in the "reject upon first failure" domain has considered only limited versions of the problem. Avinadav and Raz [2] present a model similar to ours; however, in their model the inspections must follow a pre-defined order based on, e.g., technological or logistical considerations. The number of potential inspection sequences for a set of n potential inspections in their case is 2^n, since each inspection in the pre-defined sequence can either be used or not. A branch and bound algorithm is provided for finding the optimal solution, as well as two types of greedy-based heuristics: activation and deactivation of inspections, one at a time.

Several papers have addressed the sequencing problem, assuming that all inspections must be carried out [22,9,8]. The optimal solution is obtained using ordering rules, which are based on the ratio of the inspection cost to its probability of rejecting a non-conforming unit. Shaoxiang and Lambrecht [29] generalize these models, allowing free order and inspection selection; however, their model assumes that the probability of rejecting a unit is independent of its location in the sequence. Furthermore, their model assumes that inspections are inconsistent, i.e., a repeated execution of the same inspection may result in different outcomes, and that each result is independent of previous ones. The number of potential inspection policies in their model is $O(n^n)$, and a branch and bound algorithm, based on the ordering rule of [9], is given.

Overall, the model used in this paper generalizes the "reject upon first failure" model in [2], allowing selection of the inspections to be executed as well as their order of execution. It shares some of the characteristics of "non-reject upon first failure" models [25,26], e.g., the probability of an inspection to reject a unit is assumed to be dependent on the results of former inspections, and the inspections are consistent in their observations if repeated. As we show in the evaluation section, when these two degrees of freedom (selection and sequence) are enabled, the profit obtained substantially improves. Yet, the transition to two degrees of freedom adds substantial computational complexity that needs to be handled.

The contributions of this paper are twofold: First, an efficient branch and bound algorithm for finding the optimal inspection sequenced subset for the augmented model is introduced. While the algorithm is proved efficient experimentally for most cases, its theoretical complexity is factorial of the number of inspections. Therefore, the second contribution is in developing heuristics that are shown experimentally to produce highly accurate solutions, in comparison to traditional greedy-based heuristics, in a substantially shorter calculation-time. The developed heuristics provide a tradeoff between calculation-time and accuracy. The paper

[2] We distinguish between the case of a non-conforming item that actually failed one of the inspections and yet was delivered (since further inspections resulted with a reduced probability of being non-conforming) and an item that passed all the inspections successfully and turned out to be non-conforming. While in both cases the probability of the item being non-conforming may be equal, from the user's perspective there is a great difference. Consumers are sensitive to cases where there are "documented" doubts and are often more lenient towards cases where an item is found eventually to be non-conforming despite passing successfully the sequence of inspections. A good example for this can be found in the recent investigation by the House of Representatives Oversight Committee into Toyota Motor Corp's response to complaints of uncontrolled engine acceleration that led to a recall of more than 6 million vehicles in the United States. As part of the process, some claims were made that Toyota concealed evidence for documented possible defects (NY Times, August 31, 2009).

[3] Obviously, when there is no constraint over the overall inspection time and items are rejected upon receiving a non-conforming classification, a sequential execution of inspections dominates a parallel one.

provides a thorough analysis of the methods, using advanced statistical methods, enabling identification of the cases where each method is recommended.

In the next section we present the model and detail its assumptions. The branch and bound algorithm and its analysis are described in Section 3. Section 4 presents five heuristics, classified either as greedy-based or preliminary sort-based. Heuristics of the first class add inspections according to classical greedy approaches, while those of the latter decompose the problem into a sequencing and selection problems. A thorough evaluation of the branch and bound algorithm, as well as the suggested heuristics, is given in Section 5. Based on the results, a detailed drill down analysis is presented for two of the heuristics that provide a better tradeoff between calculation-time and accuracy than the other heuristics. An additional evaluation of these two heuristics and the branch and bound algorithm based on a comprehensive set of biometric inspections is provided in Section 6. Finally, we conclude and suggest directions for future research in Section 7.

2. The model

For exposition purposes, we adopt the terminology of the "product inspection" domain, which is widely used in research of inspection planning problems (see [23] for an extensive survey). We consider a setting where an item needs to be classified dichotomously either as "conforming" or "non-conforming". We denote the a priori probability of an item being non-conforming by q (see Table 1 for a summary of notations used). We use $S=\{s_1,..,s_n\}$ to denote the set of n potential inspections available for classification purposes. Each inspection $s_i \in S$ is characterized by three parameters: (1) the cost of executing the inspection, denoted c_{s_i}; (2) the probability of Type I error when using inspection s_i, denoted α_{s_i}; and (3) the probability of Type II error when using inspection s_i, denoted β_{s_i}. The result of any given inspection s_i is assumed to be independent of any other inspections' results. Furthermore, we assume that inspections are consistent in a sense that repeating an inspection results in the same output (and thus it is useless to repeat an inspection already carried out). The revenue from supplying a conforming item is denoted by *REV*. The penalty, in addition to the loss of revenue, for supplying a non-conforming item is denoted by *PEN*. Inspections are assumed to be carried out sequentially, where if the item is classified by any inspection in the sequence as non-conforming, then no more inspections are carried out, and the item is rejected without any salvage value. Otherwise, the item is submitted to the next inspection in the sequence or, if it is the last one, it is classified as conforming and delivered. This process is illustrated in Fig. 1.

Table 1
Summary of notations.

Notation	Meaning
q	The a priori probability that the item is "non-conforming"
$S=\{s_1,..,s_n\}$	The set of potential inspections
S'	A sequenced subset of inspections from S ($S' \subseteq S$)
S''	The remaining inspections from S, given the selection S' ($S=S'+S''$)
s_i	The i-th inspection in S
c_{s_i}	The cost of executing inspection s_i
α_{s_i}	The probability of Type I error using inspection s_i
β_{s_i}	The probability of Type II error using inspection s_i
REV	The revenue from delivering a conforming item
PEN	The penalty for delivering a non-conforming item

The goal is to construct a sequenced subset of inspections $S' \subseteq S$ (denoted as sequence from this point onwards) that maximizes the expected profit. For exposition purposes we denote $S'=\{s'_1,\ldots,s'_k\}$ where $k \le n$. The expected profit calculation includes the revenue of delivering a conforming item, the penalty of delivering a non-conforming one, and the cost of the inspections used. Formally, the expected profit from using the sequence S' is given by:

$$Profit_{S'} = (1-q)\prod_{i=1}^{k}(1-\alpha_{s'_i})REV - q\prod_{i=1}^{k}\beta_{s'_i}PEN - \sum_{i=1}^{k}\left((1-q)\prod_{j=1}^{i-1}(1-\alpha_{s'_j}) + q\prod_{j=1}^{i-1}\beta_{s'_j}\right)c_{s'_i} \quad (1)$$

An exhaustive-search-based solution would require evaluating the expected profit of all possible sequences of all possible subsets of S (i.e., a total of $\sum_{k=0}^{n} n!/(n-k)!$ sequenced subsets). For large values of n the number of evaluations converges to $en!$. For comparative purposes, if the order of the inspections is fixed, limiting the decision to the selection of inspections to be carried out, the number of required evaluations in a similar exhaustive search is 2^n [2].

3. Branch and bound algorithm

We present an efficient branch and bound algorithm, denoted as *BB*, for finding the optimal sequenced subset for the "reject upon first failure" model. The algorithm uses an upper bound to the expected profit obtained from extending further a given sequence. The upper bound is calculated by using the permissive assumption that all the currently unassigned inspections can be

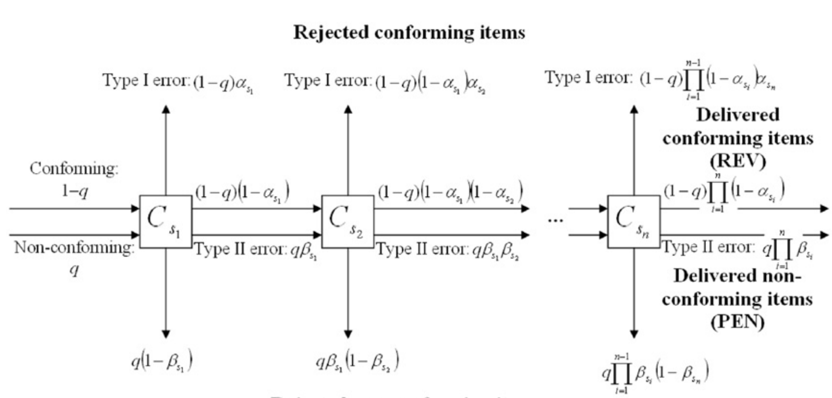

Fig. 1. Items' transition probabilities in an inspections sequence $S=\{s_1,..,s_n\}$.

Please cite this article as: Avinadav T, Sarne D. Sequencing counts: A combined approach for sequencing and selecting costly unreliable off-line inspections. Computers and Operations Research (2012), doi:10.1016/j.cor.2011.12.018

used to reduce Type II error with a one time cost, which equals the cost of the cheapest inspection among them, and that none of these inspections incur a Type I error.

Lemma 1. *Given a sequence of inspections* $S'=\{s'_1,\ldots,s'_k\}$, *an upper bound to the expected profit for any extended sequence* $S'+S''$ ($S''=\{s_i|s_i\in S, s_i\notin S', S''\neq\emptyset\}$), *denoted* $UB_{S'+S''}$, *is given by*:

$$UB_{S'+S''}=(1-q)\prod_{i=1}^{k}(1-\alpha_{s'_i})REV \\ -q\prod_{i=1}^{n}\beta_{s_i}PEN-\sum_{i=1}^{k}\left((1-q)\prod_{j=1}^{i-1}(1-\alpha_{s'_j})+q\prod_{j=1}^{i-1}\beta_{s'_j}\right)c_{s'_i} \\ -\left((1-q)\prod_{j=1}^{k}(1-\alpha_{s'_j})+q\prod_{j=1}^{k}\beta_{s'_j}\right)\min\{c_{s_i}|s_i\in S''\} \quad (2)$$

Proof. The expected profit of carrying $S'+S''$ inspections sequence includes three components as described in Eq. (1): (a) the expected revenue from delivering conforming units; (b) the expected penalty from delivering non-conforming units to the customer; and (c) the expected cost of inspections. The revenue component in Eq. (2) is the expected revenue of using only S' according to Eq. (1). This is an upper bound to the revenue of using $S'+S''$, since any additional inspection decreases the revenue because of its Type I error (reflected by the multiplication by a coefficient smaller than 1). The cost component in Eq. (2) that is associated with the penalty from delivering non-conforming units to the customer is a lower bound to the $S'+S''$ case, since it adds all remaining inspections, and according to Eq. (1) this component decreases with each inspection that is added to the sequence (reflected by the multiplication by a coefficient smaller than 1). Finally, the last two terms sum up the inspection costs of S' and the cost of the cheapest inspection in S'', weighted according to its occurrence probability given the sequence S'. Thus, the sum is a lower bound to the inspection costs of $S'+S''$. Since Eq. (2) uses an upper bound to the revenue and lower bounds to both the penalty and inspection costs resulting from adding at least one inspection to the sequence S', it actually yields an upper bound for the profit in Eq. (1). □

Algorithm 1. A branch and bound algorithm for finding the optimal sequenced subset.

Input: $S=\{s_1,\ldots,s_n\}$;$(c_{s_i},\alpha_{s_i},\beta_{s_i})\forall s_i\in S$; REV; PEN; q;
Output: $S_{optimal}$, $Best$—the optimal sequenced subset of inspections and its expected profit, respectively.
1: Set $S'=\emptyset$; $L=\emptyset$
2: Calculate $Profit_{S'}$ according to Eq. (1)
3: Set $Best=Profit_{S'}$
4: Set $S_{optimal}=S'$
5: **repeat**
6:   **for** each $s_i\in S-S'$ **do**
7:     Set $S''=S'\cup s_i$
8:     **if** $UB_{S''}>Best$ **then**
9:       Set $L=L\cup S''$
10:     **end if**
11:     **if** $Profit_{S''}>Best$ **then**
12:       Set $Best=Profit_{S''}$
13:       Set $S_{optimal}=S''$
14:     **end if**
15:   **end for**
16:   Set $L=L-\{I|UB_I<Best, I\in L\}$
17:   Set $S'=argmax_{S'\in L}(UB_{S'})$
18: **until** $L=\emptyset$
19: **return** $S_{optimal}$,$Best$

The branch and bound algorithm procedure is described in Algorithm 1. The set L is used for storing the sequences that need to be branched further. The algorithm starts with an empty set of inspections and gradually builds new sequences by adding inspections, one at a time. On each step, the algorithm branches further the sequence associated with the highest upper bound (which has a potential for a high improvement in the expected profit) and prunes branches associated with an upper bound smaller than the highest expected profit calculated so far. The sequence S' is used to denote the current sequence that needs to be branched further. This set is initialized as an empty set (Step 1) and its expected profit is calculated accordingly (Step 2). The variables $S_{optimal}$ and $Best$ are used to store the best sequence found so far and its expected profit (Steps 3–4). The branching takes place in Steps 6–15. The values $Profit_{S''}$ and $UB_{S''}$ can be calculated according to Eqs. (1) and (2), respectively. Only a new sequence that has an upper bound greater than the maximum profit found so far joins the set of sequences, L, that needs to be branched further (Steps 8–10). The best profit found so far is updated accordingly (Steps 11–14). Following the latter update, all sequences in L that are associated with an upper bound lower than $Best$ are pruned (Step 16). The process resumes with the sequence from L associated with the highest upper bound (Step 17). The algorithm terminates when all potential further sequences are associated with an upper bound lower than the highest expected profit found so far (i.e., when L is an empty set, as given in Step 18).

While Algorithm 1 guarantees reaching the optimal solution, its complexity remains $O(n!)$. Nonetheless, as shown in the experimental analysis section, its performance, in terms of calculation-time, for the majority of the cases is satisfactory (even for the extreme case of $n=40$ inspections). As a complementary means for cases where the branch and bound algorithm exceeds its allotted running time, we introduce several heuristics characterized by different levels of calculation-time and accuracy tradeoffs.

4. Heuristics

Two types of heuristics are proposed to solve the sequencing and selection problem: greedy-based and preliminary sort-based. Heuristics of the first type are trivial and widely used in this domain [27,14,2]. The heuristics of the second type make use of the decomposition concept, by which the problem is divided into a sorting problem and a selection problem [29], and solved in this order. This paper presents several heuristics of each type for extracting a near-optimal inspections sequence. A thorough comparative experimental analysis of these heuristics' performance is given in Section 5.

4.1. Greedy-based

The general greedy-based heuristic used in our experiment is of the form G_k. It starts with an empty sequence and adds on each step a sorted combination of i inspections ($i=1,2,\ldots,k$). The selected addition is the one associated with the maximum marginal improvement to the expected profit, according to Eq. (1), out of the set of all size-i ($i\le k$) ordered combinations. This procedure continues until no improvement is achieved by adding an additional inspection/s. The complexity of this heuristic is $O(n^{k+1})$. Three greedy-based variants of the general heuristic are used in our experiments: G_1, G_2 and G_3.

The advantage of evaluating a set of inspections instead of a single inspection is that for a single inspection the greedy approach is most likely to select the inspection which offers the lowest Type II error even if its cost is relatively high. This results

Please cite this article as: Avinadav T, Sarne D. Sequencing counts: A combined approach for sequencing and selecting costly unreliable off-line inspections. Computers and Operations Research (2012), doi:10.1016/j.cor.2011.12.018

from the reduction in the penalty associated with delivering a non-conforming item, which is greater in magnitudes in comparison to the inspection cost. Choosing a set of two or more inspections enables grouping cheap inspections with relatively high Type II errors to an equivalent inspection that is characterized by a very low Type II error.

The main benefit of the greedy-based heuristics is their polynomial computational complexity. Among the above greedy-based heuristics, G_1 is the most common in sequence-based problems [27,14,2]. Heuristics G_2 and G_3 are less common, however, guarantee finding the optimal sequence if its size (i.e., the number of inspections) is smaller or equal to the size of the set that is being evaluated on each step. Theoretically, one could use even G_4 and up to G_n heuristics of this kind, however, from the practical aspect (and as demonstrated in the results section), the computational complexity increases exponentially in the set size.

4.2. Preliminary sort-based

The preliminary sort-based heuristics divide the optimization problem into two sub-problems: finding an efficient order of inspections and consequently finding which inspections should be used (the selection problem), taking the order found as a constraint. While this approach is common in many domains [6], it does not necessarily guarantee finding the optimal sequence for the original composite problem. The inspections order used in this work is based on the ordering rule suggested by Duffua and Raouf [9]. According to this rule, the inspections are sorted in an ascending order according to the ratio between the cost of the inspection, c_{s_i}, and the inspection's a priori probability to reject an item. The latter probability in our model, denoted by $P_{reject}(s_i)$, is given by $(1-q)\alpha_{s_i}+q(1-\beta_{s_i})$. It is notable that this initial sorting according to Duffua and Raouf [9] is known to be optimal if each inspection tests a different characteristic of the item (i.e., when $P_{reject}(s_i)$ is independent in the location of the inspection in the sequence). Nonetheless, in our model the result of each inspection updates the a priori probability of conformance for the following inspections. Therefore, the value of $P_{reject}(s_i)$ changes according to the location of the inspection in the sequence, and consequently, ordering the inspections according to this rule does not guarantee the optimal sequence.

Once the order is set, the selection of inspections to be used is made. This selection process has a computational complexity of $O(2^n)$ [2], which is a substantial improvement in comparison to the complexity of the original problem ($O(n!)$). Given any sequence of potential inspections, we can either use techniques that guarantee the optimal selection for the specific sequence (e.g., branch and bound, dynamic programming), or use polynomial-time selection heuristics. In this context we propose both types of solution, based on [2]:

- Sort-based Branch and Bound Heuristic (SBBH)—uses an improved version of the branch and bound algorithm suggested by Avinadav and Raz [2] for finding the optimal subset of inspections for the given pre-defined order. The improvement is achieved by using a tighter upper bound (considering the cost of the cheapest inspection which is not included in the current sequence, as in Eq. (2), rather than disregarding it as originally proposed). The computational complexity of this heuristic remains $O(2^n)$.
- Sort-based Greedy, Ascending and Descending (SGAD)—takes the best of the sequenced subsets obtained by: (1) Sort-based Greedy, Ascending—starts with an empty sequence and adds a single inspection on each step, constrained by its place in the pre-defined ordering, according to its marginal improvement if added; and (2) Sort-based Greedy, Descending—starts with the full ordered set of inspections, and removes a single inspection at a time, according to its marginal improvement if removed. Note that the descending greedy heuristic cannot be used without a pre-defined order of the potential inspections, unlike the ascending greedy heuristic. The complexity of the *SGAD* heuristic is $O(n^2)$, as the complexity of its two components.

5. Evaluation

The *BB* algorithm and the proposed heuristics were evaluated using simulation, based on 65 610 different problem instances. This section describes the experimental design and detail the results of each method in terms of reaching optimality, deviation from the optimal profit and calculation-time (measured in milliseconds).

5.1. Experimental design

Simulation settings were generated by varying seven parameters: (1) n; (2) q; (3) the distribution from which costs of the different inspections are drawn; (4) the distribution from which Type I errors are drawn; (5) the distribution from which Type II errors are drawn; (6) *REV*; and (7) *PEN*. For each parameter we used three levels of values: low, medium and high. The levels are given in Table 2. Full factorial experimental design was used, covering all $3^7=2187$ cross-level combinations.

Due to the complexity of the model and the number of uncertain parameters, a Monte-Carlo simulation was used, generating different input values for each parameter. Overall, 30 repetitions were carried out for each of the 2187 settings, resulting in a total of 65 610 problem instances. Simulations were carried out using a computer equipped with Core 2 Duo 3.16 GHz processor. For each problem instance we logged the sequence produced, the profit and calculation-time achieved by executing each of the methods. All the data collected is available from the authors.

5.2. Results and analysis

The performance of the methods tested, according to calculation-time and optimality, is summarized in Table 3. The first row of the table presents the percentage of instances in which the optimal solution was not obtained. The second set of rows relates to the calculation-time achieved in different percentiles in addition to the mean and the maximum values. Finally, the deviation of the expected profit (obtained by the heuristics) from the optimal one (obtained by the *BB* algorithm), in case such deviation exists, is given for different percentiles. In order to eliminate the influence of the magnitude of the net profit when calculating

Table 2
Parameters' values in the experiment.

Parameter	Level		
	1	2	3
n	10	20	40
q	0.02	0.2	0.4
C_{s_i}	$\sim$U(1, 19)	$\sim$U(5, 15)	$\sim$U(9, 11)
α_{s_i}	$\sim$U(0.0005, 0.0015)	$\sim$U(0.005, 0.015)	$\sim$U(0.05, 0.15)
β_{s_i}	$\sim$U(0.005, 0.015)	$\sim$U(0.025, 0.075)	$\sim$U(0.125, 0.375)
REV	100	1000	10 000
PEN	1000	10 000	100 000

Table 3
Results summary.

Measures	Optimal *BB*	Greedy-based heuristics			Sort-based heuristics	
		G_1	G_2	G_3	*SBBH*	*SGAD*
% of non-optimal sequences	N/A	44.3%	20.4%	9.3%	5.7%	9.5%
Calculation-time (ms)						
50%	2	0.1	1.3	32.1	0.5	0.4
Mean	3749	0.1	3.8	178.3	17.8	1.0
95%	1521	0.3	13.1	514.7	36.3	2.0
99%	66 096	0.4	16.7	677.7	398.4	3.6
max	2 930 216	2.3	20.8	827.3	3 807.9	31.7
Deviation from the optimal profit (as % of *REV*)						
50%		0.1%	0.1%	0.1%	0.0%	0.0%
Mean		0.9%	0.8%	0.8%	0.1%	0.2%
95%		5.6%	4.7%	5.1%	0.3%	1.2%
99%		12.1%	9.3%	9.6%	1.0%	3.2%
Max		21.5%	20.3%	18.2%	3.2%	14.2%

the relative deviation, the absolute deviation in the expected profit is divided by the value of the revenue parameter.[4] The choice of the revenue seems to be the most reasonable benchmark, as companies are usually measured according to their profitability on top of their revenue. The profit-deviation analysis is based only on those problem-instances in which the optimal sequence is not reached.

As can be seen from Table 3, G_1 is the best heuristic (in order of magnitude) as far as calculation-time is concerned. Nevertheless, it achieves poor performance as far as optimality is concerned (in 44.33% of the cases it failed to obtain the optimal sequence, with a maximum deviation of 21.53%). Heuristic G_2 provides better performance than G_1 by factor of 2 in terms of not reaching the optimal sequence, however, it is dominated by *SGAD* for all measures except for the maximum calculation-time (a difference of 11 ms, which is negligible in light of the fact that up to the 99 percentile the *SGAD* is in fact better).[5] Heuristic G_3 is also dominated by *SGAD* for all measures except for the percentage of non-optimal sequences obtained. The difference between the two heuristics is statistically insignificant at the 0.05 level according to a two-proportion z-test with unequal variances. It is notable that G_3 guarantees finding the optimal sequence for those cases, in which the optimal solution includes up to three inspections. Nonetheless, comparing G_3 with *SGAD*, for settings where the optimal sequence includes more than three inspections, reveals that G_3 failed to reach the optimal sequence in 87% of these cases while *SGAD* failed only in 43.6%. Furthermore, the mean calculation-time of *SGAD* is smaller in about two orders of magnitude than in G_3. Thus, we consider *SGAD* to outperform G_3. At this point, an interesting observation can be made according to the experimental evaluation results: an order-constrained greedy heuristic (*SGAD*) reaches the optimal sequence more frequently than all the non-constrained greedy-based heuristics, which are highly used in literature.

Other than *SGAD*, the *SBBH* heuristic is the only one among those tested that provides a reasonable tradeoff between calculation-time and optimality. While *SGAD* excels in terms of calculation-time, *SBBH* is by far the most accurate heuristic (both in terms of reaching the optimal sequence and deviating from the optimal expected profit). In the remaining of this section, we introduce a drill down analysis of the methods: *BB*, *SBBH* and *SGAD*, divided into optimality and calculation-time performance. The reasons for excluding G_1 from the in-depth analysis is that accuracy is an important measure and G_1, which is the commonly used heuristic in the inspections literature [14,25–27], is simply inaccurate in a large portion of the settings. The reason for excluding G_2 and G_3 is that they are dominated by *SGAD* as discussed above.

Table 4
Percentage of classes where the optimal sequence was received in at least T confidence level.

Method	T=99%	T=95%
SBBH	44%	78%
SGAD	22%	64%

5.2.1. Optimality

By definition, the *BB* algorithm always results with the optimal sequenced subset, thus our optimality analysis applies only to *SBBH* and *SGAD*.[6] The percentage of cases where the optimal sequence is not obtained, as given in Table 3, is 5.7% for *SBBH* and 9.5% for *SGAD*.

In order to predict when each of the two heuristics is likely to result with an optimal sequence, a C4.5 classification-tree analysis [21] was carried out using a binary classifier, indicating whether the optimal sequence was reached in each problem instance. To this end, we define a threshold T, distinguishing classes where the heuristic is likely to result with the optimal sequence. A class was classified as likely to result with the optimal sequence if its prediction accuracy, calculated as the proportion of instances where the optimal sequence is reached within the class, is greater than T.[7] The threshold T thus reflects the level of confidence in the prediction based on the class. The percentages of problem instances covered by "optimality recommendation" in *SBBH* and *SGAD*, for two values of T, are given in Table 4. Based on these results, we conclude that for a very large number of settings we can predict, with a considerable level of

[4] The profit when using the optimal sequence may be zero, in which case any relative deviation results with an infinite value in percentages.

[5] For the general case, the computational complexity of G_2 is greater than *SGAD* ($O(n^3)$ vs. $O(n^2)$).

[6] While *SBBH* is based on branch and bound, it does not result with an optimal solution to the sequencing problem, but rather with an optimal inspection selection given the sequence it receives as an input.

[7] While a similar prediction can be made based on clusters where the proportion of cases in which the optimal sequence was not found, none of the C4.5 clusters complied with T=95% accuracy level.

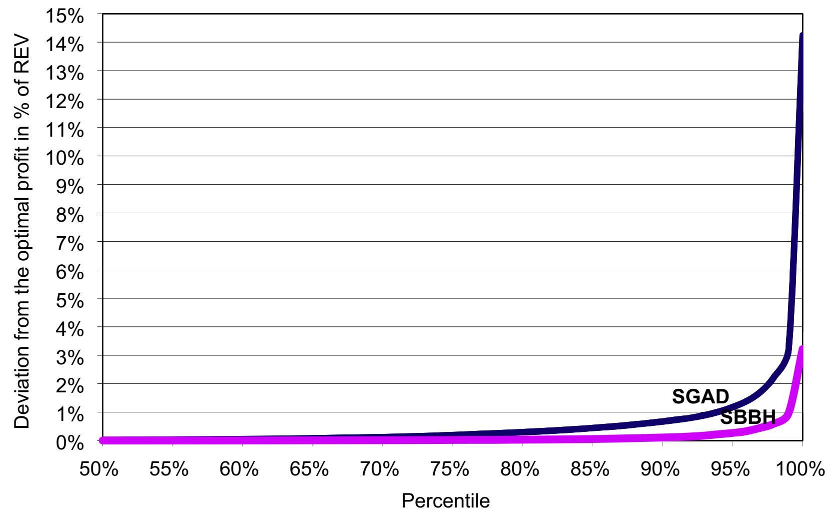

Fig. 2. Percentiles of deviation from the optimal expected profit in *SGAD* and *SBBH* heuristics, in case of failing to reach the optimal sequence.

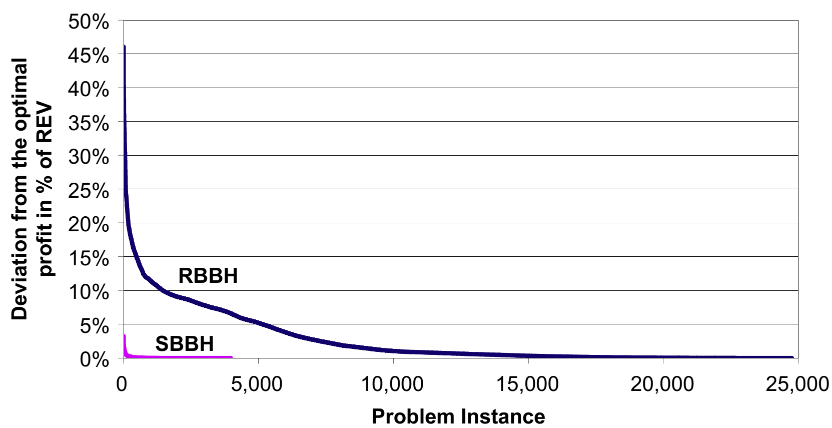

Fig. 3. *RBBH*'s accuracy in comparison to *SBBH*.

confidence, whether each of the two methods will find the optimal sequence.

Note that both heuristics provide a near optimal solution in cases where the optimal sequence is not reached. The deviation of the achieved expected profit from the one associated with the optimal sequence, in percentages of *REV*, is given for each heuristic in the last part of Table 3. In 99% of the cases where *SBBH* do not reach the optimal sequence (5.7% of all problem instances), the deviation is less than 1% and the maximum deviation overall is 3.2%. As expected, *SGAD* is dominated by *SBBH* in terms of deviation from the optimal expected profit value. In 95% of the cases where the optimal sequence is not reached (9.5% of all instances), the deviation is less than 1.2% and the maximum deviation overall is 14.2%. The deviation over the full spectrum of percentiles for the two heuristics is given in Fig. 2.

To conclude the optimality analysis of the sort-based heuristics, we present a complementary analysis supporting the efficiency of the preliminary sequencing stage used by these heuristics. For this purpose we use a revised variant of *SBBH*, denoted *RBBH*, that does not include the initial sorting stage but rather starts with a random order of inspections. Based on the same problem instances set, the percentage of cases where the optimal sequence is not obtained by *RBBH* is 37.8% (comparable to 5.7% in *SBBH*). The mean deviation of the achieved expected profit from the one associated with the optimal sequence is 2.6% (comparable to 0.1% in *SBBH*) and the maximal deviation is 46.1% (comparable to 3.2% in *SBBH*). Fig. 3 illustrates the deviation of the expected profit of the solutions produced by *RBBH* and *SBBH* from the optimal solution (in percentages of *REV*). The horizontal axis represents problem instances, sorted by the magnitude of the difference for each heuristic. As can be observed from the figure, the improvement in performance when adding the initial ordering phase is substantial and results both with fewer non-optimal solution and smaller deviations.

5.2.2. Calculation-time

We begin our calculation-time analysis with the branch and bound algorithm. Although the computational complexity of the *BB* algorithm is $O(n!)$, its actual performance (presented as calculation-time in this paper)[8] is quite phenomenal: the median value is 2 ms, which is one order of magnitude higher than in the other heuristics, and for 95% of the cases the calculation-time was less than 1521 ms, which is quite reasonable given the size of the problems analyzed and compared to its theoretical complexity. In particular, the actual calculation-time up to the 95 percentile is in the same order of magnitude as in G_3 heuristic, which complexity is $O(n^4)$. Nonetheless, the maximum calculation-time

[8] The number of calculations are available from the authors.

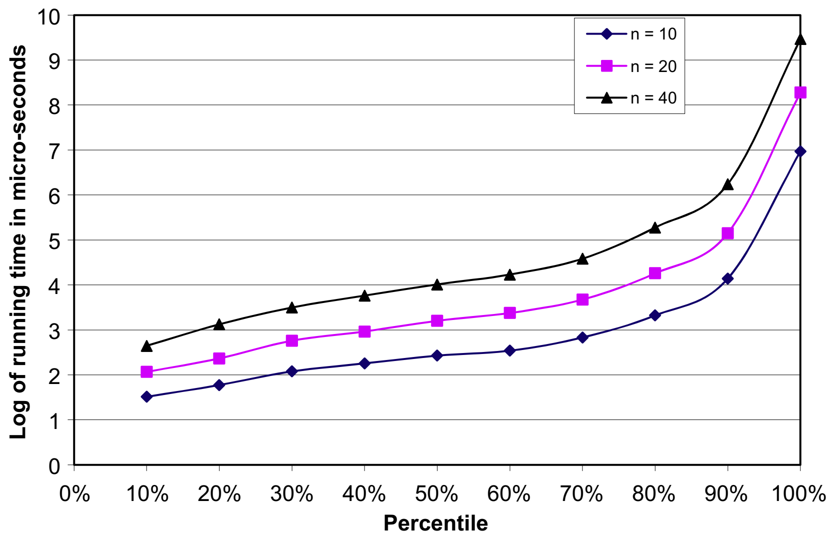

Fig. 4. Calculation-time performance of the *BB* algorithm.

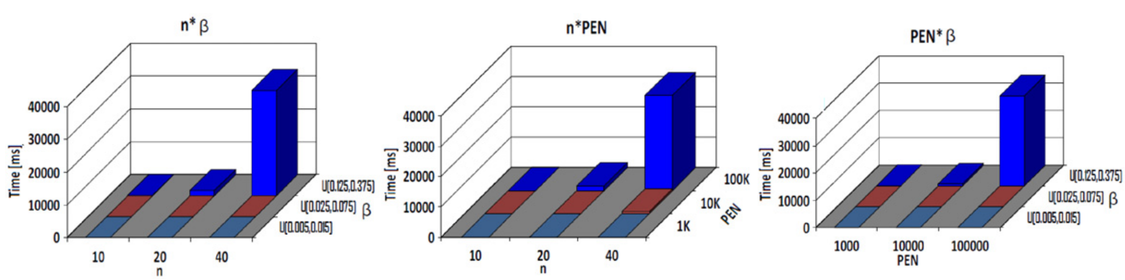

Fig. 5. Branch and bound calculation-time analysis correlations.

observed is substantial (almost an hour for several of the $n=40$ inspections problem instances), and hence the importance of the heuristics developed.

We analyze the calculation-time applying several complementary analysis methods. First we compare the calculation-time over different percentiles of problem instances, according to the three levels of the number of inspections, n. The results of the analysis are depicted in Fig. 4 using the log-scale for the calculation time (vertical axis). As can be seen from Fig. 4, the growth in calculation-time is quite mild up to the 80 percentile in all three levels of n (in comparison to the expected factorial theoretical growth).

Next, we performed ANOVA [18] of the calculation times achieved by the *BB* algorithm. The analysis considered the effect of the seven factors and all 21 first-order interactions over the calculation-time. The contribution to the total variance of each factor was found to be smaller than 2%.[9] A contribution to the total variance of more than 2% was found only in three first order interactions: $n{*}PEN$, $n{*}\beta$ and $PEN{*}\beta$. Fig. 5 depicts the mean calculation-time, according to different factor levels of the latter three interactions. The effect of n over the calculation-time is straightforward: any increase in the number of inspections increases the number of possible sequences that should be evaluated in a factorial order. As for *PEN* and β, these parameters directly affect the number of inspections that is likely to be used, as an increase in their values results with an increase in the expected penalty from delivering a non-conforming item, and consequently more inspections are required. Furthermore, all three parameters directly affect the tightness of the upper bound used for pruning in the *BB* algorithm, as the latter considers the product of all Type II errors of all inspections and *PEN* ($\prod_{i=1}^{n} \beta_{s_i} PEN$). Since each of the three parameters positively affects the calculation-time, their product has a much greater impact.

Table 5
Percentage of clusters with short/long *BB* calculation-time in at least *T* confidence level.

Classification	$T=99\%$	$T=95\%$
"Short" calculation-time	64%	75%
"Long" calculation-time	7%	11%
Undetermined calculation-time	29%	14%

Finally, we analyze our ability to predict when the *BB* algorithm is likely to result with a reasonable calculation-time and when its use should be avoided. This analysis is performed once again using C4.5 classification-tree with a binary classifier, indicating whether the calculation-time is reasonable or not. A reasonable calculation-time was defined arbitrarily as less than ten times (one order of magnitude) the time it takes *SBBH* (which is the most competent heuristic in terms of reaching optimality) to execute. The resulting classes were analyzed using the threshold-based classification, as detailed in the optimality analysis of *SGAD* and *SBBH* heuristics, based on calculation-time. This time, however, the C4.5 analysis also provided a complementary classification of clusters as "non-recommended" for use with the *BB* algorithm. The percentages of the problem instances that can be classified as either *BB*-suitable or non-suitable (based on the values of *T*) are given in Table 5.

A similar calculation-time analysis methodology was used for the *SBBH* heuristic. The importance of such an analysis is attributed to the fact that despite the heuristic's computational complexity of $O(2^n)$,

[9] The total explained variance of the model is 19.3%.

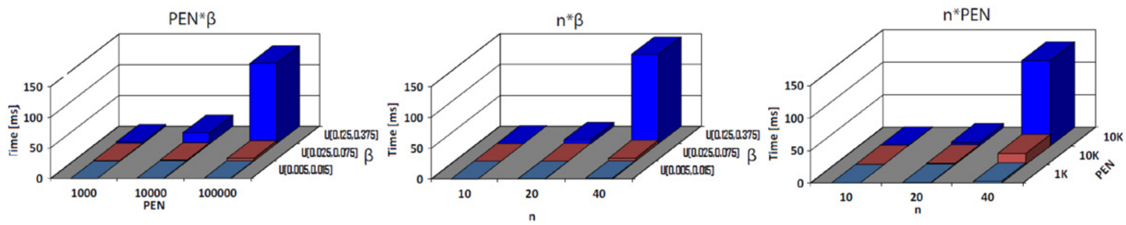

Fig. 6. *SBBH* calculation-time analysis correlations.

Table 6
Percentage of clusters with short/long *SBBH* calculation-time in at least *T* confidence level.

Classification	$T=99\%$	$T=95\%$
"Short" calculation-time	85%	85%
"Long" calculation-time	3%	7%
Undetermined calculation-time	12%	8%

its actual calculation-time, as found empirically in our experiments, is significantly lower, and thus can be considered as an appealing alternative in our case. The median value is 0.5 ms, which is of the same magnitude as of G_1 and G_2 heuristics, and for 95% of the cases the calculation-time is less than 36 ms. The maximum calculation-time observed is 3808 ms which might seem reasonable, however, it is two orders of magnitude higher than the maximum calculation-time of *SGAD*. Thus, there are cases in which the use of the *SBBH* heuristic is not favorable. According to ANOVA of the calculation time of *SBBH*, all factors have a relatively small contribution to the total variance except for n and β (contribution of 3.1% each). Overall, a relatively large portion of the total variance can be explained in *SBBH*[10] (33.2%), where the primary interactions contributing to the variance are the same as those found in the *BB* calculation-time analysis ($n*PEN$, $n*\beta$ and $PEN*\beta$, each contributing more than 4%). Fig. 6 depicts the mean calculation-time, according to different factor levels of the latter three interactions.

As for the prediction capability, a C4.5-based analysis was carried out using a binary classifier. A reasonable calculation-time was arbitrarily defined as less than ten times (one order of magnitude) the time it takes *SGAD* (which is the most polynomial time-competent heuristic) to execute. The percentages of the problem instances covered by the different types of clusters in *SBBH* for different values of *T* are given in Table 6.

The calculation-time of *SGAD* is $O(n^2)$ and thus there is no interest in a similar analysis for this heuristic.

Finally, we note that based on Table 3, while the median calculation-time in *BB* and *SBBH* is 1.672 ms and 0.543 ms, respectively, the mean is significantly higher (particularly in *BB*)—3749 ms in *BB* and 17.8 ms in *SBBH*. This indicates that the methods are generally efficient, except for cases where it is highly time consuming. This is not the case in the *SGAD* heuristic, where the ratio between the mean and the median is only 2.5.

6. Test case

In this section we evaluate the performance of the *BB* algorithm and of the two best-performing heuristics, *SBBH* and *SGAD*, using a comprehensive set of biometric inspections [16]. The biometric inspection domain encompasses many tradeoffs concerning reliability, false positives, false negatives, enrollment/authentication speed, privacy, invasiveness and cost. Systems in this domain are divided into verification systems (authentication, confirming or denying a person's claimed identity) and identification systems (recognition, establishing a person's identity) [12]. Identification is further divided into recognition from a set of already known identities (closed identification) and otherwise (open identification). The verification and identification problems are important in light of the substantial loss associated with failure to reveal impostors. Jain et al. [12] note that billions of dollars are lost annually in the United States alone as a result of fraud in welfare payments, unauthorized usage of ATM cards, and theft of cellular bandwidth made by importers using multiple identities. Furthermore, the identification problem is crucial in fighting terrorism and industrial espionage, as identification can be used to prevent unauthorized people from entering countries, secured facilities (e.g., intelligence agencies, nuclear power stations, etc.), factories and offices.

There are many biometric inspections that can be used to deal with the identification and authentication problems. In general, these inspections are divided into three major groups: physical, behavioral and chemical, where each group encompasses several measurements. A partial list of biometric inspections includes: inspections of fingerprints (optical/solid-state scanners), retinal vessels, iris, hand geometry (whole hand/several fingers), hand topography, vein features (fingers/palm), crease features (fingers/palm), facial features (landmark/infrared), voice (special word/text-independent), signature (pattern, acoustic emissions), keystroke dynamics, ear (shape/structure of cartilaginous tissue of the pinna), gait, DNA, blood pulse and odor [12]. For each inspection there are several vendors who provide scanners (or readers) and matching algorithms (including filters, sampling points and image processing), which differ from each other in their accuracy and cost.[11]

The accuracy provided by biometric identification devices is generally captured by two measures: FAR (false acceptance rate) and FRR (false rejection rate); however, there is a tradeoff between the two values for a given piece of inspection equipment, and most biometric readers offer an option to calibrate the acceptance threshold according to the customer's preferences [16]. Although manufacturers claim that their devices achieve low FAR and FRR values, these can never be obtained simultaneously at one security threshold [12]. Furthermore, different manufacturers do not use the same benchmark database to compute their devices' FAR and FRR values, and thus these values are not totally comparable [13]. For instance, Jain et al. [12] give an example of a case in which a manufacturer claimed that its biometric system achieved an FRR of 0.3% and an FAR of 0.1%, yet the system actually obtained an FRR of 25% on a different database. Furthermore, the actual values of FAR and FRR depend on the type of identification being used (open/closed), the environmental conditions (light, noise, temperature, dirt, etc.), and the size of the comparative database [12].

Until recently, most biometric systems for preventing impostors from accessing restricted places or data were designed as

[10] One interesting explanation for this is that the total variance in *SBBH* is substantially smaller than in *BB*.

[11] See [4] for a list of over 200 vendors and manufacturers of biometric-based personal identification/verification technology.

Please cite this article as: Avinadav T, Sarne D. Sequencing counts: A combined approach for sequencing and selecting costly unreliable off-line inspections. Computers and Operations Research (2012), doi:10.1016/j.cor.2011.12.018

Table 7
Biometric inspections data.

Inspection type	Unit inspection cost range	Type I error range	Type II error range
Fingerprints (optical scanners)	0.25–0.75	0.000005–0.000015	0.01–0.03
Fingerprints (solid-state scanners)	0.1–0.3	0.000005–0.000015	0.03235–0.09705
Retinal vessels	0.75–2.25	0.0005–0.0015	0.025–0.075
Iris	0.05–0.15	0.00005–0.00015	0.06–0.18
Hand geometry (whole hand)	1.1–3.3	0.0005–0.0015	0.01–0.03
Hand geometry (two fingers)	0.8–2.4	0.0005–0.0015	0.0005–0.0015
Hand topography (finger creases)+palm	0.25–0.75	0.001–0.003	0.0005–0.0015
Finger vein	0.05–0.15	0.00005–0.00015	0.0063–0.0189
Palm vein	0.05–0.15	0.00005–0.00015	0.02115–0.06345
Palm creases	2–6	1.25E-08–3.75E-08	0.005–0.015
Facial features (landmark)	0.25–0.75	0.005–0.015	0.05–0.15
Voice	0.1–0.3	0.01–0.03	0.075–0.225
Text independent voice	0.1–2.3	0.01–0.03	0.1–0.3
Signature	0.25	0.025–0.075	0.05–0.15
Keystroke dynamics	0.05	0.0005–0.0015	0.035–0.105

single biometric security systems (SBSS), relying on a single identification technology. Naturally, the devices used in such systems were calibrated to produce the minimal possible FAR, even at the price of higher FRR value. This limitation resulted in high costs from Type I errors and rendered the use of such systems inconvenient. Current systems, however, are designed as multi-biometric security systems (MBSS), relying on a combination of more than one biometric technology [20]. The main advantage of the MBSS over the SBSS is their ability to obtain very low values for both FAR and FRR. However, when a "reject upon first failure" policy is executed in MBSS, it is beneficial to calibrate the threshold of each inspection to have a very small Type I error, even at the price of a larger Type II error. Such calibration achieves excellent results when more than one inspection is used, since the FAR decreases by product with any additional inspection, while the FRR increases additively (see Fig. 1).

Our evaluation of the algorithms developed in this paper in the biometric domain is based on 15 different types of biometric inspections as detailed in Table 7. The values specified in the table (Type I and Type II errors and unit inspection cost[12]) are based on mean values as published in publicly available resources [5,17,1,11,4], taking a $\pm 50\%$ interval as a feasible range.[13] In order to fit the conditions of different environments (business, sensitive facilities, top restricted facilities) we set the range of the cost of Type I errors at (100, 1000) (difference of one order of magnitude), and the range of the cost of Type II errors at (10000, 1000000) (difference of three orders of magnitude). The range for the percentage of impostors, q, which is equivalent to the a priori probability that an item is "non-conforming", was set to (1%, 20%). All cost values are on the same monetary scale. Naturally, as discussed above, multiple inspections of each type can be available as part of the problem input (since each technology might be offered by several vendors using different hardware and algorithms, differing in their accuracy parameter values and costs). Therefore, we assumed, for the purpose of this specific evaluation, that the number of different inspections available for each type was randomly drawn from the range 0–6, representing the potential number of vendors offering that specific biometric technology as a response to an RFI or RFQ.

We carried out 64,000 Monte-Carlo simulations, each based on a different problem instance. Each instance was generated by randomly drawing, for each type of inspection, the number of

Table 8
Test case results summary.

Measures	Optimal *BB*	Sort-based heuristics	
		SBBH	*SGAD*
% of non-optimal sequences	N/A	16.2%	62.5%
Calculation-time (ms)			
50%	1 247.5	30.2	2.5
mean	3 436.9	47.2	2.6
95%	12 953.9	145.9	4.8
99%	34 611.3	263.5	5.9
max	786 630.0	1 635.9	12.3
Deviation from the optimal profit (as % of *REV*)			
50%		0	0.0009%
Mean		0.000010%	0.0050%
95%		0.000001%	0.0207%
99%		0.000004%	0.0378%
Max		0.017000%	0.1493%

inspections available (by different vendors) as explained above. For each inspection, the unit inspection cost and the values of the two error types were drawn from the appropriate ranges, which appear in Table 7. Then, the *PEN*, *REV* and q values of that problem instance were drawn from the appropriate intervals. The summary of results is given in Table 8 (which is the equivalent of Table 3). The actual number of inspections available in the different problem instances ranged from 14 to 75 with a mean of 45 and a standard deviation of 2.8. According to Table 8 we can conclude that the *BB* algorithm is highly efficient in this domain, in terms of calculation-time and reaching the optimal solution in a reasonable amount of time (no more than 35 s for 99% of the cases). Still, there are rare instances in which the *BB* takes a substantial amount of time to run (a maximum of 786 s in this set of problems).[14] Therefore, the accuracy and calculation-time results obtained for the *SBBH* and *SGAD* heuristics are quite encouraging. The *SBBH* in this test case produced highly accurate results (less than 0.02% deviation and mostly zero), while its calculation-time was less than two seconds for all runs. The *SGAD* guarantees a very low calculation-time; however, it is slightly less accurate (yet, its maximum deviation was less than 0.15%). Overall, both heuristics offer a good tradeoff between calculation-time and accuracy in the tested domain.

[12] The cost of unit inspection is taken to be proportional to the reader's cost, as it is the main cost of the system (purchase and holding).

[13] To fit the MBSS with "reject upon first failure" policy, devices are assumed to be calibrated for Type I error lower than Type II error.

[14] When experimenting with an alternative set in which the number of vendors offering each inspection was drawn from the range 0–12 (i.e., an average of 90 available inspections per problem), some instances took more than 5 h to run).

7. Discussion and conclusions

This study presents optimal and heuristic methods for selecting and sequencing costly, unreliable inspections from a given set, and evaluates the effectiveness and computational efficiency of each method. The results of the thorough experimental evaluation provide strong evidence for the efficiency of the branch and bound algorithm (*BB*), in terms of calculation-time, for most problem instances tested. In particular, the algorithm's median calculation-time is of the same order of magnitude as the calculation-times of the tested heuristics. For some settings, however, this algorithm may require substantial calculation-time, potentially precluding its use in real-life applications. Our analysis shows that this latter problem can be mitigated by our ability to predict, for the vast majority of cases, whether or not the use of the *BB* algorithm is likely to involve a reasonable calculation-time. For those cases where the algorithm is suspected to require a substantial calculation-time, the *SBBH* heuristic, which offers relatively high accuracy, is recommended to be used instead. The upper bound used in our *BB* algorithm facilitates many applications. For example, one may set a limit over the maximum deviation allowed from the optimal expected profit. This way, the algorithm can terminate once the distance between the current best solution and current maximal upper bound is smaller than the limitation defined, substantially reducing the calculation-time.

Despite its excellent performance in our experiments, the calculation-time of *SBBH* is theoretically exponential in the number of inspections available. As in the case of the *BB* algorithm, we can also predict when the *SBBH* is likely to require a relatively long execution time. In such cases we recommend the use of *SGAD*, which has a guaranteed quadratic calculation-time in the number of available inspections, at the cost of a mild compromise on optimality/accuracy. Overall, both *SBBH* and *SGAD* provide relatively good performance in terms of optimality (94.3% and 90.5% probability of reaching the optimal sequence, respectively) and relatively small normalized deviation from the optimal expected profit value (up to 1% and 3.2%, respectively, corresponding to 99% of the cases that resulted in non-optimal sequences). As evidenced from the analysis of the random-order branch and bound heuristic (*RBBH*), the order of inspections does affect the optimality and accuracy measures substantially. An interesting observation in this context is that the a priori ordering used by *SBBH* and *SGAD* substantially reduces calculation-time; however, at the same time it imposes a constraint that might preclude reaching the optimal sequence in some cases. Nevertheless, the accuracy obtained by the *SBBH* heuristic leads to the conclusion that in many cases the constrained order is indeed the optimal one for the selected inspections.

The *SGAD* heuristic takes the best of two sort-based greedy quadratic-time heuristics: ascending and descending. Use of the descending heuristic in inspections is uncommon. The advantage of this approach is that if the optimal sequence includes a large number of inspections, then this method is more likely to yield the optimal sequence than the ascending heuristic. The descending approach complements the ascending approach, where the initial tendency is to add inspections with low Type II error (even if these inspections are costly). In the descending sort-based heuristic the costly inspections are the first to be removed, because the penalty is already reduced by the combination of the cheap inspections (even if the error of each cheap inspection is relatively high). The *SGAD*'s solution can also be used to improve the performance of the *BB* algorithm, if used as an initial current best solution, enhancing the pruning process in early stages of the algorithm.

The current work and the analysis methodology can be used for the development of many real-life extensions of the model presented. For example, one may consider constraining the time by which a decision should be made whether to accept or reject the item, possibly forcing some of the inspections to be scheduled in parallel rather than sequentially. Other extensions may include constraining the maximal average percentage of delivered non-conforming items, complex penalty functions, and dependency between inspections' characteristics.

Acknowledgement

We thank Moshe Naaman for developing the infrastructure for running the experiments.

References

[1] Hitachi's Finger Vein Technology a white paper. Your quest for the ideal biometric: is it in vain? August 2007 ⟨http://www.hitachi.eu/veinid/documents/veinidwhitepaper.pdf⟩.
[2] Avinadav T, Raz T. Economic optimization in a fixed sequence of unreliable inspections. Journal of the Operational Research Society 2003;54:605–13.
[3] Bendavid I, Herer YT. Economic optimization of off-line inspection in a process that also produces non-conforming units when in control and conforming units when out of control. European Journal of Operational Research 2009;195(1):139–55.
[4] The biometric consortium. Biometric vendors information; 2011 ⟨http://www.biometrics.org/vendors.php⟩.
[5] Inc. Biometric Technology. Biometric technical ASSESSMENT; August 2002 ⟨http://www.bioconsulting.com/Bio_Tech_Assessment.html⟩.
[6] Cohen R, Katzir L. Computational analysis and efficient algorithms for micro and macro OFDMA scheduling. In: INFOCOM. IEEE; 2008. p. 511–9.
[7] Ding J, Greenberg BS, Matsuo H. Repetitive testing strategies when the testing process is imperfect. Management Science 1998;44(10):1367–78.
[8] Duffuua SO, Pollock SM. Optimal sequencing of inspections of dependent characteristics. Journal of Optimization Theory and Applications 1999;103:557–65.
[9] Duffuua SO, Raouf A. An optimal sequence in multicharacteristic inspections. Journal of Optimization Theory and Applications 1990;67:79–86.
[10] Finkelshtein A, Herer YT, Raz T, Ben-Gal I. Economic optimization of off-line inspection in a process subject to failure and recovery. IIE Transactions 2005;37:995–1009.
[11] Özdemir MK. A framework for authentication of medical reports based on keystroke dynamics. Master's thesis; 2010.
[12] Jain A, Flynn P, Ross A. Handbook of biometrics. Secaucus, NJ, USA: Springer-Verlag New York, Inc.; 2010.
[13] Matyas V Jr, Riha Z. Biometric authentication systems. Technical report, Masaryk University; 2000.
[14] Kaspi M, Raz T. Optimal sequencing of production and inspection operations. International Journal of Production Research 1994;32(8):251–64.
[15] Liou MJ, Tseng ST, Lin TM. Optimal sequence of partial inspections subject to errors. International Journal of Production Economics 1994;33(1–3):109–19.
[16] Manivannan S, Padma E. Comparative and analysis of biometric systems. International Journal on Computer Science and Engineering 2011;3(5): 2156–62.
[17] Maszoros A, Bank Z, Czini L. Strengthening passwords by keystroke dynamics. In: IEEE international workshop on intelligent data acquisition and advanced computing systems: technology and applications; 2007. p. 574–7.
[18] Montgomery DC. Design and analysis of experiments. John Wiley & Sons; 2006.
[19] National Institute of Standards and Technology. The impact of calibration error in medical decision making (planning report 04-1). Technical report, Gaithersburg, MD; 2004.
[20] Orsag F. Biometric Security Systems: Speaker Recognition Technology. PhD thesis; 2004.
[21] Ross Quinlan J. C4.5: Programs for machine learning (Morgan Kaufmann Series in Machine Learning). Morgan Kaufmann, 1st ed.; January 1993.
[22] Raouf A, Jain JK, Sathe PT. A cost minimization model for multicharacteristic component inspection. IIE Transactions 1983;15:187–94.
[23] Raz T. A survey of models for allocating inspection effort in multistage production systems. Journal of Quality Technology 1986;18(4):239–47.
[24] Raz T, Bricker D. Sequencing of imperfect inspection operations subject to constraints on the quality of accepted and rejected units. International Journal of Production Research 1987;25(6):809–21.
[25] Raz T, Bricker D. Optimal and heuristic solutions to the variable inspection policy problem. Computers and Operation Research 1991;18(1): 115–23.
[26] Raz T, Bricker D. Sequencing of inspection operations subject to errors. European Journal of Operational Research 1993;68(2):251–64.
[27] Raz T, Kaspi M. Sequencing of imperfect inspection operations subject to constraints on the quality of accepted and rejected units. International Journal of Production Research 1991;29(8):1645–59.
[28] Raz T, Thomas M. A method for sequencing inspection activities subject to errors. IIE Transactions 1983;15:12–8.

[29] Shaoxiang C, Lambrecht M. The optimal frequency and sequencing of tests in the inspection of multicharactersitic components. IIE Transactions 1997;29: 1039–49.
[30] Sheu S, Chen Y, Wang W, Shin N. Economic optimization of off-line inspection with inspection errors. Journal of the Operational Research Society 2003;54(8):888–95.
[31] Tzimerman A, Herer Y. Off-line inspections under inspection errors. IIE Transactions 2009;41(16):626–41.
[32] Wang C. Economic off-line quality control strategy with two types of inspection errors. European Journal of Operational Research 2007;179(1):132–47.
[33] Wang WY, Sheu SH, Chen YC, Horng DJ. Economic optimization of off-line inspection with rework consideration. European Journal of Operational Research 2009;194(3):807–13.
[34] Yu HF, Yu WC. An optimal mixed policy of inspection and burn-in and the optimal production quantity. International Journal of Production Economics 2007;105(2):483–91.
[35] Yum BJ, McDowell ED. Optimal inspection policies in a serial productions systems including scrap rework and repair: an MILP approach. International Journal of Production Research 1987;25(10):1451–64.

Please cite this article as: Avinadav T, Sarne D. Sequencing counts: A combined approach for sequencing and selecting costly unreliable off-line inspections. Computers and Operations Research (2012), doi:10.1016/j.cor.2011.12.018