Journal of the Operational Research Society (2003) 54, 605–613 0160-5682/03 $25.00

www.palgrave-journals.com/jors

Economic optimization in a fixed sequence of unreliable inspections

T Avinadav and T Raz*

Tel Aviv University, Ramat Aviv, Israel

Given a fixed sequence of unreliable inspection operations with known costs and inspection error probabilities of two types (classifying good items as defective and *vice versa*), we develop a model for selecting the set of inspections that should be activated in order to minimize expected total costs (inspection and penalties). We present an efficient branch and bound algorithm for finding the optimal solution, and two variations of a greedy heuristic that can be applied jointly to provide very good solutions at a O(n^2) computational complexity. The conclusions are backed by a factorial experiment that included 1440 problem instances.

Journal of the Operational Research Society (2003) **54,** 605–613. doi:10.1057/palgrave.jors.2601543

Keywords:

Introduction

Notwithstanding the modern emphasis on defect prevention and process improvement, inspection of the output of production or service activities is still an effective way of preventing defects from reaching the subsequent production stages or, eventually, the customer. Basically, inspection can be conceived of as the classification of each item into one of two mutually exclusive categories: either it is 'good', that is, fully conforms to specifications and expectations, or it is 'defective', that is, contains at least one defect, or deviation from specifications. Inspection processes and technologies have vastly improved over the last decades, but still in many cases they are not fully reliable. There are two types of inspection errors: classification of a good item as defective (Type 1), and classification of a defective item as good (Type 2). The correctness of the final disposition decision can be improved by taking into account the results of several inspections carried out on the same item. Here we need to address the issue of which inspections should be carried out, out of a given set of possible candidates. This issue is particularly relevant when we consider the costs, which include, in addition to the inspection costs, the cost resulting from incorrect decisions: rejection of good items and acceptance of defective ones. This subject has received a fair amount of attention in the research literature.

Raz and Thomas[1] presented a model based on a 'branch-and-bound' algorithm to determine the cost-optimal order of a set of inspections that are subject to errors of various types and costs while meeting a given average outgoing quality level. Raz and Bricker[2] expanded that model to include constraints on the percentage of good units among those rejected. Later on, Raz and Bricker[3] developed heuristic and optimal solutions for the variable inspection problem, whereas the next inspection to he carried out depends on the outcomes of the inspections performed so far. This work was expanded in Raz and Bricker[4] to account for situations where all inspections selected must be carried out on all units, and when their order may be fixed or variable.

Duffuaa and Raouf[5] and Raouf *et al*[6] presented a model for the optimal ordering of multiple inspections to minimize total cost. Liou *et al*[7] found an analytical solution for the model of Raz and Thomas[1] under certain conditions by modifying slightly the cost model. Their solution provides the number of inspectors and their optimal sequence in several cases when all inspectors have the same inspection errors and the same sampling fraction. For other cases, they suggest solutions based on the rules developed by Raouf *et al.*[6]

Yum and McDowell[8] developed a binary programming model that allows for repair, replacement or salvage of defective units, with disposition costs varying accordingly. Viswanadham *et al*[9] proposed heuristics based on a genetic algorithm and on simulated annealing for locating inspections in production systems in order to minimize cost. The model of Shaoxiang and Lambrecht[10] maximizes profit while allowing for repeated inspection of the product features.

In this paper, we consider the problem of selecting, out of a set of available inspection operations, the subset that will maximize the expected profit per item produced, while accounting for both direct inspection costs and for the costs caused by inspection errors. The available inspections are subject to both types of errors, and the order in which they

Correspondence: T Raz, University Campus, PO Box 39010, Ramat Aviv, Tel Aviv 69978, Israel.

can be carried out is already determined. The paper is organized as follows. In the next section, we present the assumptions of the model and the notation that will be used. Then, we formulate the objective function to be minimized, and present a branch and bound algorithm for finding the optimal solution. Since the optimization problem has non-polynomial computational complexity, we propose a greed heuristic and report the results of an experiment aimed at assessing its performance as well as the performance of the branch and bound algorithm. We conclude with a discussion of some practical implications of this work and with directions for future expansion.

Assumptions and notation

There is a set of n available inspections, each with known and constant parameters: unit inspection costs, and misclassification probabilities (Types 1 and 2 errors). The order in which the inspections are to be carried out is already determined by technological, logistics or administrative constraints. We wish to select one or more inspections out of this set (if it has cost justification) to carry out on each and every item produced at a certain production stage. If more than one inspection is selected, then they are all carried out according to the predetermined order. If an item is classified as good, then it is submitted to the next inspection in the sequence or, if it is the last inspection in the sequence, it is delivered to the customer. If the item is classified by a certain inspection as defective, then no more inspections are performed on it, and the item exits the system and is disposed of as if it were defective, without any salvage value. The development of the model is based on the following assumptions:

1. The results of a given inspection are independent of those of other inspections in the sequence.
2. The probability distribution function of any given item being good or defective is independent of that of the other items.
3. Each available inspection can appear at most once in the sequence.

The following notation will be used throughout the development:

$i=$ the inspection index ($i=1,\ldots,n$)
$B_i=$ the cost of inspecting one item by inspection $i (B_i>0)$
$REV=$ the revenue for supplying a good item to the customer
$PEN=$ the penalty for supplying a defective item to the customer. The penalty is in addition to the loss of revenue that would have been generated if the item were good
$\phi_i=$ probability of a Type 1 error in inspection i ($i=1,\ldots,n$)
$\theta_i=$ probability of a Type 2 error in inspection i ($i=1,\ldots,n$)
$q_0=q=$ probability of an item being defective
$p_0=1-q=$ probability of an item being good
$q_i=$ probability of an item being defective and classified as good by the first i inspections in the sequence
$p_i=$ probability of an item being good and classified as good by the first i inspections in the sequence

Given that the order of the inspections is fixed, it is advantageous to view the inspection sequence as including all the n available inspections, some of them being active and the rest inactive. An active inspection is one that is actually carried out on the items produced, and entails the corresponding costs and error probabilities. An inactive inspection is one that is not performed; as such it incurs no cost and rejects no items. Thus, we can define the decision variable for our model, x_i as follows:

$$x_i = \begin{cases} 1 & \text{if inspection } i \text{ is active} \\ 0 & \text{if inspection } i \text{ is inactive.} \end{cases}$$

Objective function

If inspection i is inactive, then its effective unit cost is 0; its effective probability of Type 1 error is 0, and its effective probability of Type 2 error is 1. The effective cost and error probabilities of any inspection, either active or inactive, can be stated mathematically as follows.

$$\text{Effective cost}: \; B_i \cdot x_i = \begin{cases} B_i & \text{if } x_i = 1 \\ 0 & \text{if } x_i = 0 \end{cases} \quad (1)$$

Effective probability of Type 1 error :

$$\phi_i \cdot x_i = \begin{cases} \phi_i & \text{if } x_i = 1 \\ 0 & \text{if } x_i = 0 \end{cases} \quad (2)$$

Effective probability of Type 2 error :

$$\theta_i \cdot x_i + 1 - x_i = \begin{cases} \theta_i & \text{if } x_i = 1 \\ 1 & \text{if } x_i = 0 \end{cases} \quad (3)$$

The principle of operation of the inspection sequence is that items proceed from one inspection to the next only if they are classified as good. Accordingly, the probability that an item will be good and will be classified as good by the first i ($i=1,\ldots,n$) inspections is equal to the probability that the item is good multiplied by the product of the probabilities that each of the first i inspections will not commit a Type 1 error and classify a defective item as good. Using the notation introduced in (1)–(3). we obtain the following:

$$p_i = (l-q) \cdot (1-\phi_1 \cdot x_1) \cdot (1-\phi_2 x_2) \cdots (1-\phi_i x_i) \quad (4)$$

Similarly, the probability that an item will be defective and classified as good by the first $i\,(i=1,\ldots,n)$ inspections is equal to the probability that the item is defective multiplied by the product of the probabilities that each of the first i inspections will commit a Type 2 error and classify the item as good:

$$q_i = q\cdot(\theta_1 x_1 + 1 - x_1)\cdot(\theta_2 x_2 + 1 - x_2)\cdots(\theta_i x_i + 1 - x_i) \tag{5}$$

The objective is to maximize the expected profit from the operation. The profit term excludes the production costs of the items, since these are already sunk costs. Profit, or net revenue, consists of the following three terms:

1. The expected revenue from supplying good items to the customer, which is the product of the revenue (REV) and the probability of supplying a good item to the customer (p_n).
2. The expected penalty for supplying defective items to the customer, which is the product of the value of the financial damage (PEN) and the probability of supplying the customer a defective item (q_n).
3. The expected cost of quality inspection per item, which is the sum across all the inspections ($i=1,\ldots,n$) of the product of the cost of inspecting the item in inspection i ($B_i\,x_i$, according to Equation (1)) and the probability that the item (good or defective) will reach inspection i, which is equal to $P_{i-1}+q_{i-1}$.

The first term represents revenue and consequently is positive. The two other terms reflect costs and are negative. Combining the three gives the following expression for the expected profit per item:

$$\mathrm{E}[Profit] = p_n\cdot REV - q_n\cdot PEN - \sum_{i=1}^{n}[p_{i-1} + q_{i-1}]\cdot B_i\cdot x_i \tag{6}$$

Substituting (4) and (5) into (6), we obtain the following expression for the objective function:

$$\begin{aligned}\mathrm{E}[Profit] =&(1-q)\cdot\prod_{i=1}^{n}(1-\phi_i\cdot x_i)\cdot REV\\ &- q\cdot\prod_{i=1}^{n}(\theta_i\cdot x_i + 1 - x_i)\cdot PEN\\ &-\sum_{i=1}^{n}\left[(1-q)\cdot\prod_{j=1}^{i-1}(1-\phi_j\cdot x_j)\right.\\ &\left.+q\cdot\prod_{j=1}^{i-1}(\theta_j\cdot x_j + 1 - x_j)\right]\cdot B_i\cdot x_i\end{aligned} \tag{7}$$

Optimal solution

Finding the values of x that maximize the expression in (7) is a combinatorial problem that has no structural constraints besides that the decision variables are binary. Its computational complexity is $O(2^n)$. The problem can be formulated as the following binary programming problem:

$$\begin{aligned}\max:\Big\{\mathrm{E}[Profit] =&(1-q)\cdot\prod_{t=i}^{n}(1-\phi_i\cdot x_i)\cdot REV\\ &- q\cdot\prod_{i=1}^{n}(\theta_i\cdot x_i + 1 - x_i)\cdot PEN\\ &-\sum_{i=1}^{n}\left[(1-q)\cdot\prod_{i=1}^{t-1}(1-\phi_j\cdot x_i)\right.\\ &\left.+q\cdot\prod_{j-1}^{i-1}(\theta_j\cdot x_i + 1 - x_i)\right]\cdot B_i\cdot x_i\Big\}\\ s.t.:&\ x_i = \{0,1\}\end{aligned} \tag{8}$$

This problem can be solved by using a general algorithm for solving binary programming problems. However, the objective function is quite complicated, and might be difficult to adapt to the format required by a general optimization program. Instead, we propose to exploit the specific structure of the problem by applying the branch-and-bound approach to find the optimal solution.

A branch and bound algorithm

The elements of the branch-and-bound algorithm for this problem are introduced and defined next, followed by a description of the algorithm.

Node: A vector of length n whose elements are the decision variables $(x_1x_2\ldots x_n)$. Each decision variable can be in one of two states: determined (meaning that its value has been set to either '0' or '1'), or undetermined, denoted by '?', meaning that its value has not been set yet. We restrict the structure of the vectors so that all the undetermined elements appear at the end. Thus, the vector consists of two parts: elements 1 through m, which are all determined, and elements $m+1$ through n, which are all undetermined, with $1\leqslant m<n$. The node represents a subset of the solution space corresponding to all the solutions, where the first m inspections are either active or inactive as determined, and the other inspections may be active or inactive. Thus, a node with a vector with m determined elements represents a subset that contains 2^{n-m} possible solutions.

Basic sequence: A specific sequence contained in the subset of the solution space defined by the node. The basic sequence is the sequence obtained by setting all the undetermined elements of the node vector equal to '0'. In other words, the basic sequence is that which includes only the inspections that have been determined. Obviously, the basic sequence is a feasible solution to the problem.

Empty node: A node with a vector with all its elements undetermined (*?…??*)

Full node: A node with a vector with all its elements determined. A full node represents a single feasible solution: theoretically there are 2^n full nodes.

Valid node: A node where the last determined element, x_m, is equal to 1. This concept is used to create an effective branching process.

Branching is the process of creating two or more child nodes out of a given node. The child node inherits the determined elements of the parent. It seems intuitive to create two children out of each node, corresponding to $x_{m+1}=0$ and 1. We prefer to use a different approach that seems more effective. From a node consisting of m determined elements and $n-m$ undetermined elements, we create $n-m$ child nodes. Each child node is obtained by setting one of the undetermined elements say .x_k, $m+1 \leqslant k \leqslant n$, equal to '1', setting all previously undetermined elements x_{m-1} through x_{k-1} equal to '0', and leaving elements x_{k+1} through x_n undetermined, as they were. For example, the node '$x_1x_2\ldots x_n$????' will be branched into the following child nodes: '$x_1x_2\ldots x_m$1???', '$x_1x_2\ldots x_m$01??', '$x_1x_2\ldots x_m$001?', '$x_1x_2\ldots x_m$0001'. Clearly, every node obtained by branching in this way is valid.

For every child node we calculate two quantities: the value of its basic sequence, and an upper bound on the values of the sequences included in the subset of the solution space defined by the node. The basic sequence, which was defined earlier as the sequence consisting of only the inspection whose state has been determined, is evaluated using the expression on the right-hand side of (7).

The upper bound of the node is calculated, taking into account the positive contributions of all the inspections that can still be activated, and ignoring their negative contributions. The positive contribution results from the reduction in the expected penalty for accepting defective units. The negative contributions consist of the additional inspection costs and the increase in lost revenue due to rejection of good items. In other words, the upper bound is calculated by assuming that all the undetermined inspections will be carried out at zero cost and without incurring a Type 1 error. For a node with m inspections whose state has been determined and $n-m$ undetermined inspections ($x_1\ldots x_m?\ldots?$), the upper bound UB{} is calculated according to Equation (9).

$$\begin{aligned}\mathrm{UB}\{\mathrm{E}[Profit]_m\} =&(1-q)\cdot\prod_{i=1}^{m}(1-\phi_i\cdot x_i)\cdot REV\\&-q\prod_{i=1}^{m}(\theta_i\cdot x-1-x_i)\cdot\prod_{j=m+1}^{n}\theta_j\cdot PEN\\&-\sum_{i=1}^{m}\left[(1-q)\cdot\prod_{j=1}^{i-1}(1-\phi_j\cdot x_j)\right.\\&\left.+q\cdot\prod_{j=1}^{i-1}(\theta_i\cdot x_i+1-x_i)\right]\cdot B_i\cdot x_i\end{aligned} \quad (9)$$

The added component relating to E[Profit] is $\prod_{j=m+1}^{n}\theta_j$ which reduces *PEN*, as if all the undetermined inspections were active.

We start with the empty node and calculate its value, which, in effect, corresponds to the case where no inspections at all are activated. This becomes the current best feasible solution. Then we branch from that node, and for each child node we calculate its value and its upper bound. If the value of any of the newly created nodes is higher than that of the current best solution, then the basic sequence of that node becomes the current best. All the nodes with an upper bound value lower than that of the current best solution are closed. This cycle is repeated for all remaining open nodes, until no more branching is possible. At this point, the current best solution is identified as the optimal solution and the process stops. An evaluation of the efficiency of our branch and bound technique appears later on, following the description of the evaluation experiment.

Heuristic solution

The branch-and-bound technique does not reduce the theoretical computational complexity of the problem, which, in our case, is exponential in *n.* In this section, we present a heuristic technique for solving the inspection selection problem. The heuristic solution can be used as is or can serve as a good initial 'current best' solution if we wish to seek the optimum with branch and bound.

Our heuristic is of the 'greedy' type, whereas at each iteration we make the decision that provides the greatest immediate improvement of the objective function. Greedy heuristics have been applied to solve inspection systems design problems — see for instance Raz and Bricker.[3] There, the heuristic consists of adding the inspection that generates the greatest improvement in the value of the objective function. Here, we consider two variations:

1. Starting with the empty sequence, we activate, one at a time, the inspection that results in the greatest improvement.
2. Starting with the full sequence, we deactivate, one at a time, the inspection, which, if omitted from the sequence, results in the greatest improvement.

Both variations stop when the next step, either activation or deactivation according to the case, does not bring about an improvement in the objective function. For a set of n available inspections, there are at most n iterations, iteration k consisting of $n-k$ evaluations of the objective function. We propose to apply the two variations and to choose the better of the two resulting solutions.

The computational complexity of the heuristic is obtained as follows. In the first iteration, n inspections of the set are evaluated, in the second iteration the remaining $n-1$ inspections are evaluated, and in each successive iteration one more inspection is removed. The worst possible case occurs when we need to carry out all n iterations, which means that the number of calculations of the objective function is the sum of the following arithmetic progression:

$$S_n = n + (n+1) + (n-2) + \cdots + 1 = \tfrac{1}{2}\cdot n\cdot(n+1) = \tfrac{1}{2}\cdot n^2 + \tfrac{1}{2}\cdot n \quad (10)$$

This expression corresponds to $O(n^2)$

Experimental design

In order to assess the quality of the solutions obtained with the branch-and-bound technique and with the heuristics, we carried out an experiment with simulated data. We used a factorial design for the experiment, with seven factors and two or three levels for each factor. The factors corresponded to the parameters of the problem (number of inspections, costs and error probability parameters). All the factors had low and high levels, and some also had an intermediate level. The levels were chosen so that the sample would not only be representative of the population of practical problems, but would also exhibit one order of magnitude of variability. Due to technological and practical constraints very seldom does the number of inspections exceed ten. Thus, the number of available inspections was set at two levels: a low level of 8, which is a small number amenable to optimization by complete enumeration, and a high level of 16 which, while representing the range of the number of different inspections possible after a given process, does not require an unworkable amount of computing resources. The high level values of q, REV and PEN are greater than the low level values by a factor of 10. As for the values of the cost and error probabilities of the eight or 16 individual inspections in each problem, these were obtained from uniform distributions. For the inspection cost B_i the same distribution mean was used for the two factor levels, with a smaller variability ($\pm$20%) for the low level and a higher variability ($\pm$80%) for the high level. For the two factors corresponding to the error probabilities, the range of the uniform distribution was the same ($\pm$25%), but there the mean of the high level was ten times greater than that of the low level. For these two factors we also defined an intermediate level, where half of the individual inspection error probabilities came from the low level and half from the high level. The full details of the experimental design are summarized in Table 1.

Of the seven experimental factors defined, five have two levels and two have three levels, such that altogether $2^5\times 3^2=288$ different combinations of experimental factor levels are obtained. We took five observations for each combination, giving a total of $5\times 288=1440$ sample points from the population of possible problems. For each problem we found the optimal solution with the branch and bound technique, and the heuristic solutions. The analysis of the results is presented in the following two sections.

Performance of the branch and bound algorithm

Theoretically, the application of a branch and bound solution procedure does not reduce the computational complexity of the problem. However, from the practical perspective we would expect a significant reduction in the actual effort with respect to a full exhaustive search. In addition to the running time, an appropriate way of measuring computational effort is by considering the number of times that the objective function (8) was evaluated. For the exhaustive search, this is equal to the total number of solutions in the feasible space. In our experiment, that will be $2^8=256$ for $n=8$ and $2^{16}=65536$ for $n=16$.

As for the branch and bound, we should take into account that each node visited by the algorithm involves two calculations of similar effort: evaluation of the objective

Table 1 Details of the experimental design

Factor	*Low level*	*High level*	*Intermediate level*
n	$n=8$	$n=16$	Not applicable
q	$q=0.02$	$q=0.2$	Not applicable
Distribution of B_i	$B_i=10\pm20\%$. Individual values uniformly distributed $B_i\sim U(8,12)$	$B_i=10\pm80\%$. Individual values uniformly distributed $B_i\sim U(2,18)$	Not applicable
REV	$REV=10B$ (B=mean value of B_i)	$REV=100B$	Not applicable
PEN	$PEN=50B$	$PEN=500B$	Not applicable
Distribution of ϕ_t	$\phi_t=0.001\pm25\%$. Individual values uniformly distributed $\phi_t\sim U(0.00075, 0.00125)$	$\phi_t=0.01\pm25\%$. Individual values uniformly distributed $\phi_t\sim U(0.0075, 0.0125)$	Half the values ($i=1\ldots n/2$) from the low level and half ($i=n/2$–$1,\ldots,n$) from the high level
Distribution of θ_I	$\theta_I=0.025\pm25\%$. Individual values uniformly distributed $\theta_I\sim U(0.01875, 0.03125)$	$\theta_i=0.25\pm25\%$. Individual values uniformly distributed $\theta_i\sim U(0.1875, 0.3125)$	Half the values ($i=1\ldots n/2$) from the low level and half ($i=n/2$–$1,\ldots,n$) from the high level

function for the solution corresponding to the node, and evaluation of the upper bound, according to Equation (9). Thus, the computational effort required to find the optimal solution with the branch and bound is equal to twice the number of nodes visited, and in theory may exceed the computational effort required by exhaustive search.

Table 2 shows the average, median and maximum (worst case) number of calculations and running times for the problems in the experiment. Since we expect these statistics to be strongly affected by the size of the problem in terms of number of inspections available, we distinguish between $n=8$ and 16.

From Table 2 we can see that for the smaller problems ($n=8$) on the average, the branch and bound procedure required about 36% of the number of calculations that would have been needed for an exhaustive search, and about 26% of the computing time. For the larger problems ($n=16$) the ratio is even more favorable: only about 1% of the calculations and 0.8% of the time.

The advantage of the branch and bound is more evident when looking at the median of the sample measurements: 25% of the calculations and less than 1% of the time for $n=8$: 0.3% of both the calculations and the time for $n=16$. The relation between the mean and the median indicates that the majority of the observations were located at the low end of the scale, with very few observations in a thin, long tail to the right.

These findings are illustrated in Figures 1–4, which show the distribution of the number of calculations and the running times for the two problem sizes. The fact that the advantage of the branch and bound is more significant in the larger problems is also interesting. This finding is reinforced by the analysis of the worst cases. We can see that, for $n=8$, the worst case of the branch and bound algorithm required about 70% more calculations and about twice as much time as exhaustive search. However, for $n=16$, even the worst case was significantly better than exhaustive search, requiring only about 15% of the number of calculations and 17% of the time. We may conclude that our branch and bound algorithm is indeed efficient, and that its efficiency is more significant for problems of larger size.

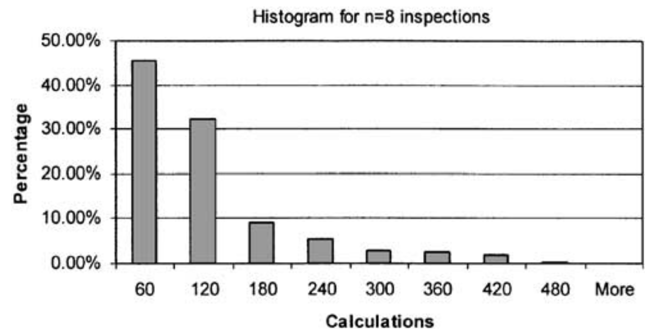

Figure 1 Distribution of the number of calculations for the branch and bound solution of the small problems ($n=8$).

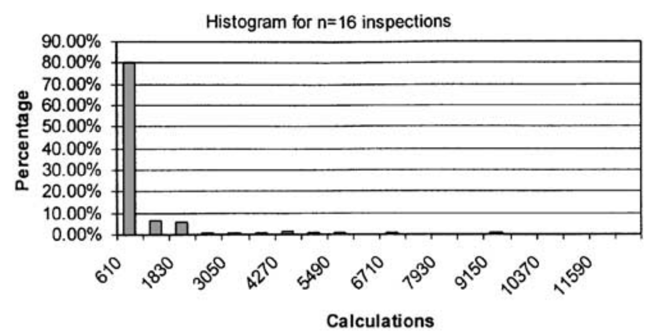

Figure 2 Distribution of the number of calculations for the branch-and-bound solution of the large problems ($n=16$).

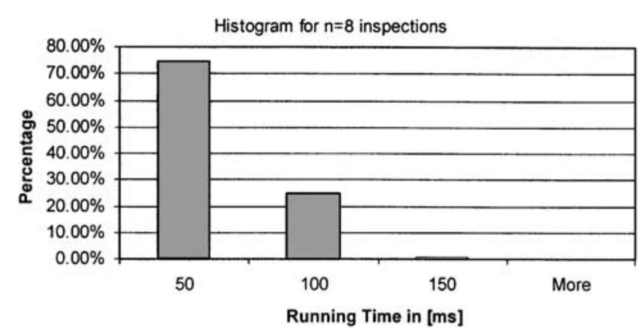

Figure 3 Distribution of the running time for the branch-and-bound solution of the small problems ($n=8$).

Table 2 Statistics on the number of calculations and running times with branch and bound

		$n=8$		$n=16$	
		Value	*% of exhaustive search*	*Value*	*% of exhaustive search*
Exhaustive search	Number of calculations	256		65536	
	Running time (ms)	54		41363	
Average	Number of calculations	92.7	36.2	668	1.0
	Running time (ms)	14	25.9	328	0.8
Median	Number of calculations	64	25.0	211	0.3
	Running time (ms)	<0.5*	<1	109	0.3
Maximum	Number of calculations	436	170.1	11130	15.2
	Running time (ms)	109	201.9	6077	17.0

*The computer function that measured running times rounded the results to the nearest millisecond.

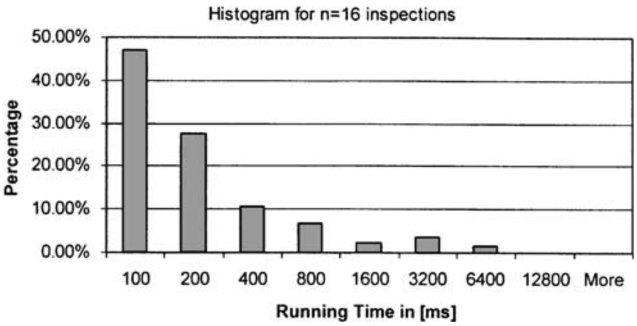

Figure 4 Distribution of the running time for the branch-and-bound solution of the large problems ($n = 16$).

Performance of the heuristics

Table 3 shows the average, median and maximum number of calculations and running times for the problems in the experiment. For $n = 8$ the average number of calculations as a percentage of the number for exhaustive search was 21.3%. In the larger problems ($n = 16$), the corresponding average percentage was only 0.27%, two orders of magnitude smaller. Of course, the reason for this dramatic change lies in the ratio of the respective computational complexities ($n^2/2^n$), which declines very rapidly as n increases. In the small problems ($n = 8$) there was a large difference in the running times. The median was less than half a millisecond and the maximum was equal to the exhaustive search (54 ms), which is 100 times larger. In the large problems ($n = 16$) the difference was much smaller. The average (102 ms) and the median (109 ms) were quite close to each other, while the maximum was 163 ms, which is just 1.6 times larger.

It appears that the relative advantage of the heuristics, similar to that of the branch and bound algorithm, becomes more significant as the problem size grows. These results suggest that for problems with more than 16 available inspections the performance of the heuristics will be even more attractive.

Finally, we carried out an analysis of variance on the sample results. The purpose was to find out if certain levels of some of the factors in the experiment were associated with significantly better or worse performance of the heuristic. The full factorial model with 43 degrees of freedom (including all main factors and all first level interactions) accounted for only $R^2 = 6.74\%$ with P-value of 0.0001 (due to 1396 degrees of freedom of the error term). The meaning of this low R^2 is that the percentage deviation of the heuristic solution from the optimal solution is virtually unaffected by the various levels of the experimental factors presented in the model. From this, we may conclude that the quality of the heuristic solution is excellent for two reasons: Not only is the average percentage deviation from the optimal solution very low on the average, but it is also insensitive to all the experimental factors and interactions among them. In other words, under a broad set of parameter values the heuristic will yield the optimum or very close to it. Accounting for the fact that the computational complexity of the heuristic is $O(n^2)$, while that of the original optimization problem is $O(2^n)$, we believe that it is a practical and highly efficient alternative to branch and bound optimization. In the following section, the percentages of optimal solution for the heuristics are exposed.

Comparison of the activating and deactivating heuristics

In this section, we compare and discuss the performance of the two variations of the greedy heuristic, activating (AH for short) and deactivating (DH). Table 4 shows the joint distribution of the solutions obtained in terms of their optimality. Looking at the first row and column, it can be seen the AH reached the optimal solution in 86.87% of the cases, while the DH reached the optimum in 93.34% of the cases. Thus, it appears that though they are both good, the deactivating heuristic is somewhat superior.

It is interesting to look at the joint distribution. In about 81% of the problems in the experiment, both heuristics reached the optimal solution. In about 18% of the cases, one heuristic succeeded in reaching the optimum while the other did not. This finding clearly indicates that the two heuristics complement each other: When one fails to yield the best solution, the other one is quite likely to do so. In fact, there

Table 3 Statistics on the number of calculations and running times with the combined heuristic

		$n = 8$		$n = 16$	
		Value	*% of exhaustive search*	*Value*	*% of exhaustive search*
Exhaustive search	Number of calculations	256		65536	
	Running time (ms)	54		41363	
Average	Number of calculations	54.6	21.3	175	0.27
	Running time (ms)	8.6	15.9	102	0.25
Median	Number of calculations	53	20.7	169	0.26
	Running time (ms)	<0.5*	<1	109	0.26
Maximum	Number of calculations	65	25.4	202	0.31
	Running time (ms)	54	100.0	163	0.39

*The computer function that measured running times rounded the results to the nearest millisecond.

Table 4 Percentage breakdown of solution optimality for the two heuristics

		Activating (AH)		
		Total (%)	*Optimal (%)*	*Non-optimal (%)*
Deactivating (DH)	Total	100	86.87	13.13
	Optimal	93.34	81.04	12.30
	Non-optimal	6.66	5.83	0.83

Table 5 Deviation as a percentage of the optimal solution for each heuristic

		Activating (%)	*Deactivating (%)*	*Combined (%)*
$n=8$	Average	0.12	0.04	0.00
	Median	0.00	0.00	0.00
	Maximum	9.82	5.63	0.13
$n=16$	Average	0.21	0.04	0.00
	Median	0.00	0.00	0.00
	Maximum	9.40	4.88	0.93

were very few instances (12 out of 1440 problems, or 0.83%) where both heuristics failed to reach the optimum. Since the computational complexity of both heuristics is low, $O(n^2)$, it is clear that they should be applied jointly. Of the two versions, the DH appears to be superior, as it contributed a share of optimal solutions (12.30%) more than twice as large as the AH (5.83%)

Table 5 provides additional support to this conclusion. The table shows, for each heuristic separately and for the two combined, the average percentage deviation from the optimal value of the objective function, the median and as well as the worst case for $n=8$ and 16 inspections. Here again we see that the DH is superior to the AH, in terms of both the average and the worst case. However, applying the two heuristics jointly improves performance by an order of magnitude, bringing the average deviation to 0.002% and the worst case to less than 1%.

We believe that the better performance of the DH is the result of its ability to take advantage of synergies between two or more unreliable but relatively inexpensive inspections, which may function better than a single more reliable but more expensive inspection. If the optimal solution consists of a single activated inspection, then the AH, which starts with the empty sequence, will find it in the first iteration and will stop right after the second iteration. In contrast, the DH will require $n-1$ iterations in order to reach single-inspection solutions, and is not guaranteed to find the optimum. In these cases, the AH has a clear advantage.

However, if the optimal solution consists of activating two or more inspections, as is likely to be the case in most practical situations, then the advantage will be on the side of the DH. The reason for this is that by adding inspections one at a time, the AH may ignore combinations of two or more consecutive inspections, which if activated together contribute more to the objective function than the single inspection it selects for activation.

In mathematical terms, if we consider two inspections, i and $i+1$, with parameters $\{\phi_i, \theta_i, B_i\}$and $\{\phi_{i-1}, \theta_{i+1}, B_{i+1}\}$ their combined effect is that of an inspection with equivalent Type 1 error probability of $\phi_i+\phi_{i-1}-\phi_1\phi_2$, equivalent Type 2 error probability of $\theta_t\theta_{i-1}$ and equivalent unit inspection cost of $B_i+[(1-q')(1-\phi)+q'\theta_i]B_{i-1}$, whereas q' represents the fraction non-conforming after inspection $i-1$ in the sequence. This equivalent inspection is never considered by the AH, while it is part of the starting solution for the DH and consequently may be retained by it and eventually become part of the final solution. Of course, similar equivalence relationships can be developed for more than two inspections and for nonconsecutive inspections. In other words, the advantage of the DH stems from its ability to consider combinations of inspections that the AH never gets to evaluate, which is particularly helpful when the optimal solution includes multiple inspections.

We should mention that greedy algorithms based on removing rather than adding elements to the solution vector are relatively rare in the field of inspection selection. In fact, we are not aware of any similar deactivating heuristic having been mentioned in the inspection system design literature.

Concluding remarks

In this study, we developed and analyzed optimal and heuristic algorithms for selecting, out of a given set of inspections subject to errors, those that should be carried out in order to maximize the expected profit per item produced. The results of the fairly extensive factorial experiment suggest that even though the branch and bound algorithm that we proposed is efficient, it may not be required since the application of the combined heuristics provides excellent results under a wide range of problem parameters. For the designer of the quality inspection system, our results provide an easy to apply method for selecting which inspections should be performed in order to minimize the economic aspects of inspection costs and errors. For the researcher, the results point out the advantages of a variation of the greedy heuristic, namely the deactivating heuristics, which in this case demonstrated clear superiority in arriving at or close to the optimal solution.

The present study addressed an unconstrained situation, except for the implied constraints on the order of the inspections. The model could be extended to cases where there are constraints, such as a constraint on the average outgoing quality (AOQ), on the number of inspections or on the inspection budget. Other extension to this work could include having a penalty function that increases with the fraction defective, as well as inspection error probabilities that vary according to the incoming fraction nonconforming at the inspection station.

References

1 Raz T and Thomas MU (1983). A method for sequencing inspection activities subject to errors. *IIE Trans* **15**: 12–18.
2 Raz T and Bricker D (1987). Sequencing of imperfect inspection operations subject to constraints on the quality of accepted and rejected units. *Int J Prod Res* **25**: 809–821.
3 Raz T and Bricker D (1991). Optimal and heuristic solutions to the variable inspection policy problem. *Comput Opl Res* **18**: 115–123.
4 Raz T and Bricker D (1993). Sequencing of inspection operations subject to errors. *Eur J Opl Res* **68**: 251–264.
5 Duffuaa SO and Raouf A. (1990). An optimal sequence in multicharacteristics inspection. *J Optim Theory Appl* **67**: 79–86.
6 Raouf A, Jain JK and Sathe PT (1983). A cost minimization model for multicharacteristic component inspection. *IIE Trans* **15**: 187–194.
7 Liou M-J, Sheng-Tseng and Tsong-Ming Lin (1994). Optimal sequence of partial inspections subject to errors. *Int J Prod Econ* **33**: 109–119.
8 Yum Bong Jin and McDowell ED (1987). Optimal inspection policies in a serial production system including scrap rework and repair: an MILP approach. *Int J Prod Res* **25**: 1451–1464.
9 Viswanadham N, Sharma Shashi M and Taneja M (1996). Inspection allocation in manufacturing systems using stochastic search techniques. *IEEE Trans Systems Man Cybernet — Part A: Systems Humans* **26**: 222–230.
10 Shaoxiang C and Lambrecht M (1997). The optimal frequency and sequencing of tests in the inspection of multicharacteristic components. *IEEE Trans* **29**: 1039–1049.

Received November 2001;
accepted November 2002
after one revision