

A Genetic Algorithm for Multiple Inspections with Multiple Objectives

Patrick R. McMullen

School of Business, Wake Forest University, Winston-Salem, USA
Email: mcmullpr@wfu.edu

Received August 28, 2013; revised September 28, 2013; accepted October 5, 2013

Copyright © 2013 Patrick R. McMullen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

This research presents a genetic algorithm to address the problem where multiple inspections are done to test conformity of multiple product characteristics. The genetic algorithm is employed to find an inspection plan where the multiple inspections are carried out, motivated to optimize two objectives: minimization of the total cost associated with the inspection; and maximization of probability of accepting conforming units. The genetic algorithm includes a constraint to induce variety into the characteristics being tested, so that the inspections are not dominated by “specialized” product characteristics. The resulting solutions are compared to optimal solutions, and it is determined that formidable solutions are found via the Genetic Algorithm approach.

Keywords: Component; Formatting; Style; Styling; Insert

1. Introduction

An important aspect of the quality assurance function in manufacturing is deciding how to implement inspection stations for multiple-stage manufacturing. The general consequence of sub-optimal implementation of multiple inspection stations is essentially twofold: first, 100% inspection and/or too many inspection stations become expensive; and second, too few inspection stations will likely result in non-conforming units being sent to the customer, resulting very unpleasant circumstances for all stakeholders involved [1].

Another force complicating this issue is the situation where inspection errors are possible [2-7]. In this context, there are essentially two types of inspection errors that are possible: the first type is where a conforming unit is incorrectly classified as non-conforming; the second type is where a non-conforming unit is incorrectly classified as conforming. The former is referred to as a Type I error, while the latter is referred to as a Type II error. Both of these errors are costly, but it is typical for a Type II error to have a high per unit cost as compared to a Type I error, because of the residual problems of shipping a non-conforming unit to the customer.

Yet another force complicating this issue is the situation where the items requiring inspection have many different attributes or characteristics that require inspection

[8-12]. An electronic assembly is a good example of this—many different features of an electronic assembly may require inspection. It is also possible that some of these specific features may require inspection multiple times, due to the subsequent value adding activities that occur during the assembly and/or manufacturing. For example, the power supply of a desktop computer may require inspection or conformance testing before and after the power supply is connected to the system board.

Additionally, some attributes of a manufactured or assembled item may not require testing at all. An example of this might be an electrical connection in an assembly. The testing of this characteristic may be considered unnecessary due to the consistent reliability of the connection. In circumstances such as this, it might be appropriate to direct inspection resources to more critical attributes of the manufactured item.

It is also important to note that for each type of attribute requiring inspection, there may be differing inspection costs, and differing probabilities of conformities and differing probabilities of acceptance. The consequence of this is that determining the cost of an inspection plan, along with determining the probability of conforming units being accepted through all stages of inspection via the inspection plan is far from trivial.

In the face of the complicating forces detailed above, it

is desired to design an inspection plan in a minimal-cost fashion. This cost is an aggregation of inspection costs, total cost of rejecting conforming units and the total cost of accepting non-conforming units. It is desired to find an inspection plan such that the “best” overall aggregation of these entities is found. At the same time, however, constraints need to apply such that “cheaper” attribute inspections do not dominate the inspection plan. In other words, it is imperative to mandate that the inspection plan is not dominated by attribute inspections that will “drive down” the total cost. A minimum number of unique attribute inspections are needed specified to prevent dominance.

Additionally, cost is not the only element that could be considered when seeking to find an optimal inspection plan [13]. While cost is an important metric of an inspection plan (perhaps the most important), there are other entities that could be considered. One important entity of an inspection plan, aside from cost, is maximizing the probability of conforming items that “survive” the inspection process. That is, maximizing the number of units that are accepted after the final inspection station. In layman’s terms, it is desired to have as many finished units as possible sent to the customer. While the cost of rejecting conforming units is included in the cost function, it is still desired to maximize the number of conforming units that “survive” the inspection process, so as to maximize long-term revenue.

As was the case for minimizing cost, it is also important to constrain the number of unique items that comprise the inspection plan when seeking to maximize the number of units that “survive” inspection through the final inspection station. This is necessary so as to avoid specific attributes being repeatedly inspected that result in high probabilities of “surviving” inspection through the last station.

As such, the present research effort simultaneously considers two objectives for optimization: minimization of cost associated with the inspection plan, and maximization of the expected value of conforming units being accepted. While these multiple objectives are pursued, there does exist a constraint imposing a minimal number of unique characteristics present in the inspection plan, so as to avoid certain characteristics from dominating the resultant inspection plan. This is considered a unique feature of this research effort—the authors are not aware of other efforts that give consideration to this issue.

Finding an optimal solution to problems of this type is quite difficult due to the combinatorial nature of the problem. If there are m unique product attributes that could be tested, and n tests that are required, there can be, in the absence of constraints, as many as m^n possible inspection plans. Because of this combinatorial difficulty, larger problem instances cannot be solved to guaran-

teeoptimality. Instead, some sort of heuristic is needed to obtain a solution to large problems. For this research effort, a Genetic Algorithm is employed to find desirable inspection plans, in terms of the aforementioned objectives, for several test problems. A genetic algorithm was chosen as the heuristic because of this ability to create and manipulate sequences—for all intents and purposes, and the details of an inspection plan can easily be thought of as a “sequence”.

The following sections will detail the computational aspects of this problem, provide a Genetic Algorithm methodology to solve the problem, employ the Genetic Algorithm on a set of test problems, compare the Genetic Algorithm results to optimality, and offer concluding comments and a general perspective on the state of this problem.

2. The Multiple-Stage Sequencing Problem

Prior to describing the multiple-stage sequencing problem, a few relevant terms are defined in **Table 1**. Given these terms, the multiple-stage inspection model is detailed in **Figure 1**.

Table 1. Terms used to describe multiple-stage sequencing problem.

Term	Description
N	units entering inspection station #1
n	number of inspections performed
m	number of unique product attributes
i	index for inspections ($i = 1, 2, \dots, n$)
j	index for attributes ($j = 1, 2, \dots, m$)
n_i	number of items entering inspection station i
c_j	cost to inspect a unit involving attribute j
x_i	The i^{th} attribute of the current inspection plan
q_j	probability of attribute j being non-conforming
α_j	probability of a type I error for attribute j
β_j	probability of a type II error for attribute j
A	per unit cost of rejecting a conforming item
B	per unit cost of accepting a non-conforming item
$N(A G)$	number of conforming units accepted by the final inspection station
$N(A B)$	number of non-conforming units accepted by the final inspection station
IC	inspection cost of plan
TIC	type I error cost
$TIIC$	type II error cost
TC	total cost of plan
Z	objective function value

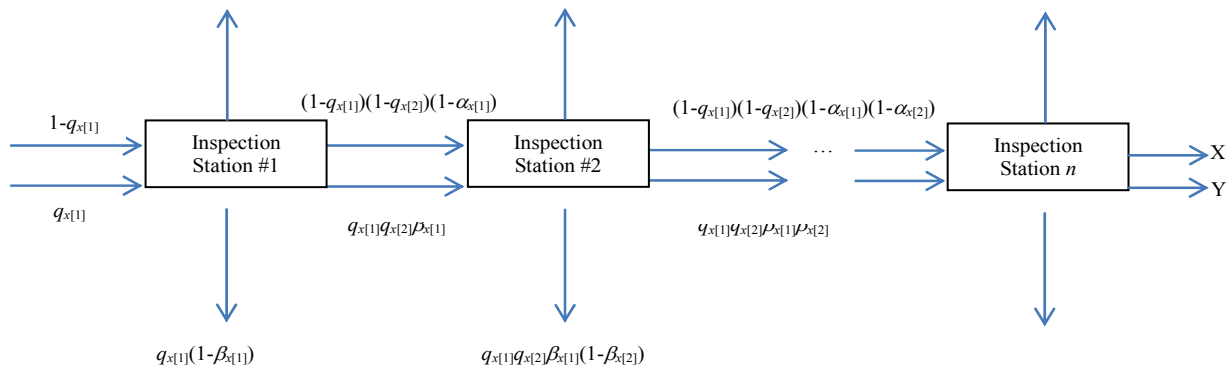


Figure 1. Layout of inspection stages.

The upward pointing arrows denote conforming items that are rejected (Type I errors). The downward pointing arrows denote non-conforming items that are rejected. The entity denoted by “X” represents the proportion of conforming units that are accepted. When multiplied by the number of units entering the first inspection station, we are provided with the expected value of conforming units that are accepted through the n^{th} inspection station—an objective function value of interest. Mathematically, this is as follows:

$$N(A|G) = N \prod_{i=1}^n (1 - q_{x_i}) \prod_{i=1}^n (1 - \alpha_{x_i}) \quad (1)$$

The entity denoted by “Y” represents the proportion of non-conforming units that are accepted (Type II errors). When multiplied by N , the number of units entering the first inspection station, it represents the expected value of non-conforming units accepted at the n^{th} inspection station. Mathematically, this is as follows:

$$N(A|B) = N \prod_{i=1}^n q_{x_i} \beta_{x_i} \quad (2)$$

The other objective function value of interest is the cost associated with the inspection plan. The cost is comprised of three components: the cost of inspecting items at each of the n inspection stations; the number of conforming units rejected at the n inspection stations (treated here as an opportunity loss); and the number of non-conforming units that are accepted at the n^{th} inspection station and sent to the customer.

The cost of inspecting items through all n inspection stations is the total number of units inspected through all stations, multiplied by the inspection cost at that particular station. Mathematically, this is as follows:

$$IC = \sum_{i=1}^n c_{x_i} n_{i-1} \left[(1 - q_{x_i})(1 - \alpha_{x_{i-1}}) + q_{x_i} \beta_{x_{i-1}} \right] \quad (3)$$

In this formula, n_0 is initialized to N , while α_0 is initialized to 0, and β_0 is initialized to 1. In essence, the inspection cost at station i is a function of the number of

units surviving inspection through the prior $(i - 1)$ inspection station. As a matter of full disclosure, n_i is as follows:

$$n_i = n_{i-1} \left[(1 - q_{x_i})(1 - \alpha_{x_{i-1}}) + q_{x_i} \beta_{x_{i-1}} \right] \quad (4)$$

The total cost of rejecting units that should be accepted is as follows:

$$TIC = A \sum_{i=1}^n n_i (1 - q_{x_i}) \alpha_{x_i} \quad (5)$$

The cost of accepting non-conforming units at the n^{th} inspection station is as follows, which is an expansion of Equation (2):

$$THIC = BN \prod_{i=1}^n q_{x_i} \beta_{x_i} \quad (6)$$

The total cost, then, of the inspection plan is the sum of the inspection cost, the Type I error cost, and the Type II error cost. Mathematically, this is as follows:

$$TC = IC + TIC + THIC \quad (7)$$

In terms of the terms defined in **Table 1**, the total cost is as follows:

$$TC = \sum_{i=1}^n c_{x_i} n_{i-1} \left[(1 - q_{x_i})(1 - \alpha_{x_{i-1}}) + q_{x_i} \beta_{x_{i-1}} \right] + A \sum_{i=1}^n n_i (1 - q_{x_i}) \alpha_{x_i} + BN \prod_{i=1}^n q_{x_i} \beta_{x_i} \quad (8)$$

As stated earlier, the intent for this research is to optimize two objectives: maximization of $N(A|G)$ and minimization of TC . To do this, a ratio involving the two objectives can be constructed:

$$Z = TC / N(A|G) \quad (9)$$

Given that it is intended to minimize TC and maximize $N(A|G)$, it is intended to minimize the Z .

3. Finding a Desirable Inspection Plan

As stated earlier, there are m item characteristics that can

be tested and n tests to be performed. This means that there can be many possible inspection plans. Specifically, in the absence of constraints regarding unique attributes to be inspected, there can be as many as m^n unique inspection plans. Again, this upper bound assumes no limitations on the number of times a specific attribute can be inspected. In the presence of constraints dictating a limitation on the number of times a unique attribute can be inspected, there can be as few as $m!/(m - n)!$ possible inspection plans. This lower bound assumes the extreme case where each attribute can be inspected one time at most ($U = n$).

Even at the lower bound, the number of unique possible inspection plans is combinatorial in nature, implying that a guaranteed optimal solution is difficult to obtain. Integer and/or nonlinear programming approaches can be used to address problems of this variety, but they also have complicating characteristics [6,7,11]. The same can be said for dynamic programming approaches [14,15]. Because of this, search heuristics become a necessary course of action to find inspection plans desirable with respect to the objective functions.

Genetic Algorithms are well-suited for a problem of this type. Genetic Algorithms have the ability to construct and manipulate sequences motivated by an objective function. In this context, an inspection plan can be thought of as a sequence—a list of length n , where each of the n elements can take on one of m possible values. Through the Genetic Algorithm trademarks of selection, crossover and mutation, a population of inspection plans can be obtained which will, hopefully, result in inspection plans with desirable objective function values. The interested reader is referred to the work of Goldberg [16] and that of Michalewicz and Fogel [17] for a detailed description of the inner-workings of Genetic Algorithms, along with many informative examples.

Prior to describing the specific Genetic Algorithm used here, some specific definitions are stated in **Table 2**.

3.1. Genetic Algorithm Process

The following sub-sections will describe the details of

Table 2. Terms used to describe genetic algorithm.

Term	Description
U	minimum number of unique attributes to be inspected
Pop	population size
$P(S_j)$	probability of selecting inspection plan j
p	crossover position
$P(M)$	probability of a mutation
Gen	number of generations

the Genetic Algorithm used to find inspection plans motivated to optimize the objectives of minimum cost and maximization of $N(A|G)$.

3.1.1. Initialization

A population of feasible solutions (inspection plans) is randomly generated. Specifically, attributes to be inspected are randomly selected in the $[1, m]$ interval for all n inspections needed. This random generation of a feasible solution is repeated until the number of unique attributes to be inspected is at least U . The random generation of feasible solutions is repeated in this fashion Pop times.

3.1.2. Selection

For each of the Pop solutions, the objective function value is determined via Equation (9). All objective function values (Z) in Pop are given the index j ($j = 1, 2, \dots, Pop$), the maximum value is referred to as $\max(Z)$, and is used to determine the probability of selecting inspection plan j , $P(S_j)$, via the following:

$$P(S_j) = \frac{(1 + \max(Z) - Z_j)}{\sum_{j=1}^{Pop} (1 + \max(Z) - Z_j)}, \forall j \quad (10)$$

This relationship is constructed such that the solutions with the lowest objective function value (value of Z), have the highest probability of being selected, so that its characteristics can be “passed on” to the next generation. Monte Carlo simulation is used to select all Pop members of the new population.

3.1.3. Crossover

The members of the population are put into $Pop/2$ couples. Members 1 and 2 comprise the first couple, members 3 and 4 comprise the second couple, and so on. For each couple, crossover is performed, by randomly selecting a crossover point, p , on the $[2, m - 1]$ interval. At that point, the attributes to be inspected from the p^{th} position to the m^{th} position are swapped for each couple. To illustrate this, an example is provided, where m is 8, and n is 8. Before crossover, the two inspection plans are coupled, where $p = 3$.

$\overline{ABCDEFGHIH}$
 $\overline{BBAGCHDE}$

The underlined section denotes the attributes to be inspected marked for crossover. After crossover, the following inspection plans result:

$\overline{ABAGCHDE}$
 $\overline{BBCDEFGH}$

For purposes of full disclosure, the first inspection plan prior to crossover shows eight unique items for inspection, while the second shows seven unique attributes

to be inspected. After crossover, the first inspection plan shows seven unique attributes to be inspected, while the second also shows seven. This particular example does not apply any constraint regarding a minimum number of unique attributes to be inspected.

For each couple, crossover is repeated until the number of unique attributes to be inspected for each inspection plan is at least U .

3.1.4. Mutation

After crossover, each inspection plan is considered for mutation. Mutation for each inspection plan will occur with a probability of $P(M)$. In the event that an inspection plan will be mutated, an attribute to be inspected is randomly selected, with all attributes of the inspection plan being equally likely for selection. This selected attribute will be modified to a different attribute, and this different attribute is selected randomly. For example, consider the inspection plan selected for mutation, where the fourth-positioned attribute is targeted for mutation:

ABAGCHDE

After mutation, the following inspection plan is possible, where attribute “G” was randomly changed to “C.”

ABACCHDE

As was the case with crossover, mutation will occur to a selected inspection plan until the number of unique items in the inspection plan is at least U .

3.1.5. Iteration and Completion

A single iteration has been complete—a new generation has been selected, crossed-over and mutated. The solutions comprising this new generation have their objective function values re-calculated. It is determined whether or not the new generation contains a solution having the best objective function value (that is, a solution with the lowest value of Z).

This process is repeated Gen times, and at the end of Gen iterations, the “best” solution is returned.

4. Experimentation

4.1. Test Problems

To determine whether or not the Genetic Algorithm approach to this problem is effective, the approach is used on several test problems, solutions are obtained, and these solutions are compared to optimal solutions, where the optimal solutions were obtained via complete enumeration. The test problems come from four data sets, which have differing costs and probabilistic parameters. Their values are detailed in **Tables 3-6** below.

Several test problems are derived from each of these four input data sets, each exploiting feasible combinations of number of inspections to perform (n) for each inspection plan, along with at least three unique items to

Table 3. Input set 1.

Attribute	q_i	α_i	β_i	c_i
1	0.100	0.050	0.050	5
2	0.050	0.040	0.010	20
3	0.050	0.080	0.000	8
4	0.150	0.000	0.100	6
5	0.051	0.520	0.300	8
6	0.060	0.000	0.000	10

Table 4. Input set 2.

Attribute	q_i	α_i	β_i	c_i
1	0.0359	0.0281	0.0112	18.74
2	0.0552	0.0391	0.0120	19.13
3	0.0402	0.0241	0.0039	18.65
4	0.0353	0.0348	0.0066	9.61
5	0.0452	0.0198	0.0034	5.56
6	0.0139	0.0220	0.0044	14.89
8	0.0276	0.0320	0.0116	11.71

Table 5. Input set 3.

Attribute	q_i	α_i	β_i	c_i
1	0.0584	0.0380	0.0276	5.19
2	0.0326	0.0298	0.0300	2.72
3	0.0628	0.0210	0.0293	3.63
4	0.0095	0.0178	0.0299	8.83
5	0.0367	0.0176	0.0259	2.01
6	0.0200	0.0343	0.0279	4.52
7	0.0188	0.0167	0.0265	9.43
8	0.0275	0.0373	0.0250	3.32
9	0.0284	0.0329	0.0264	3.30
10	0.0200	0.0308	0.0273	5.45

Table 6. Input set 4.

Attribute	q_i	α_i	β_i	c_i
1	0.0605	0.0522	0.0223	7.06
2	0.0791	0.0066	0.0205	8.87
3	0.1019	0.0074	0.0490	18.46
4	0.0980	0.0189	0.0146	20.47
5	0.0784	0.0168	0.0128	22.02
6	0.1102	0.0517	0.0187	7.63
7	0.1018	0.0243	0.0198	19.35
8	0.0799	0.0382	0.0357	10.05
9	0.0135	0.0329	0.0418	4.97
10	0.0797	0.0119	0.0361	24.23
11	0.0559	0.0524	0.0480	12.88
12	0.0595	0.0358	0.0425	10.81

be inspected for each inspection plan ($U \geq 3$).

Each feasible combination of number of inspection attributes (m) and number of inspections to perform (n) for at least three unique attributes to inspect yields ninety test problems.

4.2. Comparison to Optimal

Each of these ninety problems is solved to optimality via complete enumeration of all feasible inspection plans. For each problem, three performance measures are captured: the minimum cost (TC); the maximum number of conforming units accepted at the final inspection station ($N(A|G)$); and the minimum objective function value (Z). For each problem, these three optimal values are compared to the values obtained via the Genetic Algorithm.

Formal hypothesis testing is performed to see if there are any significant differences between the results obtained via enumeration and the results obtained via the Genetic Algorithm.

Specifically, the relative inferiority of the objective function value (Z) of the Genetic Algorithm to the optimal value is obtained via the following:

$$\text{Relative Inferiority}(Z) = (Z_{GA} - Z_{OPT}) / Z_{OPT} \quad (11)$$

The Relative Inferiority value is determined for all ninety test problems, and the mean is used as the sample mean for the following hypotheses:

$$H_0 : \mu_Z = 0; H_A : \mu_Z > 0 \text{ (H}_1\text{)}$$

For total cost associated with the inspection plan, the same logic applies for Relative Inferiority:

$$\text{Relative Inferiority}(TC) = (TC_{GA} - TC_{OPT}) / TC_{OPT} \quad (12)$$

Hypotheses are constructed in a similar fashion as above

$$H_0 : \mu_Z = 0; H_A : \mu_Z > 0 \text{ (H}_2\text{)}$$

Finally, for the number of conforming units accepted through the n th inspection station, we have the following Relative Inferiority measure:

$$\begin{aligned} &\text{Relative Inferiority}(N(A|G)) \\ &= -\left(N(A|G)_{GA} - N(A|G)_{OPT}\right) / N(A|G)_{OPT} \end{aligned} \quad (13)$$

The relevant hypotheses for this question are con-

structed as follows:

$$H_0 : \mu_{N(A|G)} = 0; H_A : \mu_{N(A|G)} > 0 \text{ (H}_3\text{)}$$

4.3. Computational Experience

Both the Genetic Algorithm approach described above and the enumeration approach were programmed using the C++ programming language, compiled via the Cygwin compiler, and run on Dual Intel Pentium E2180 processors with a clock speed of 2.00 GHz. The operating system was Windows Vista.

Computational efficiency was always a priority. The Genetic Algorithm Parameters were as follows for the differing problem sets as shown in **Table 7**.

5. Experimental Results

5.1. Presentation of Results

Tables 8-11 show the results from the ninety test problems obtained via both enumeration and by the Genetic Algorithm. The results are presented by each combination of attributes to inspect (m) and the minimum number of unique attributes to inspect (U) for each input data set. Each m and U cell combination in the table shows the results obtained via enumeration on the first line, with the results obtained via the Genetic Algorithm on the second line. The first entry on each line is the minimum total cost (TC) found for all feasible inspection plans, the second entry on each line is the maximum $N(A|G)$ value found for each feasible inspection plan, while the third entry on each line is the minimum ratio (Z) found for each feasible inspection plan. The fourth entry on the first line, in boldface, is the number of feasible inspection plans—this value is obtained via complete enumeration.

To better understand these tables, an example is provided. Specifically, **Table 9(a)**, for the cell detailing at least six unique items ($U \geq 6$) and eight inspections to be performed ($n = 8$) for eight possible attributes ($m = 8$), the details are shown below in **Table 12** in a more decompressed fashion.

For this particular problem, there are 6,531,840 feasible solutions. Comparing the results obtained via Enumeration with the results obtained via the Genetic

Table 7. Genetic algorithm parameters.

Problem Set	Population Size (<i>Pop</i>)	Probability of Mutation (<i>P(M)</i>)	Generations (<i>Gen</i>)
Input 1	64	3.00%	10,000
Input 2	64	3.00%	15,000
Input 3	96	3.00%	20,000
Input 4	128	3.00%	25,000

Table 8. Results from input1 data set ($m = 6$).

Unique Items	6 Tests ($n = 6$)	5 Tests ($n = 5$)
6 ($U \geq 6$)	\$84464.66/26.26%/3218.63/ 720	
	\$84464.66/26.26%/3218.63	
5 ($U \geq 5$)	\$50418.97/52.42%/1054.90/ 11,520	\$54594.33/55.77%/979.49/ 720
	\$50418.97/52.42%/1054.90	\$54594.33/55.77%/979.49
4 ($U \geq 4$)	\$37291.12/57.30%/833.46/ 34,920	\$37841.33/60.96%/722.07/ 4320
	\$37291.12/57.30%/833.46	\$37841.33/60.96%/722.07
3 ($U \geq 3$)	\$30758.00/62.67%/705.50/ 45,720	\$29481.52/66.24%/575.89/ 7320
	\$31238.29/62.27%/711.54	\$29481.52/66.24%/575.89

Table 9. (a) Results from input2 data set, 8 tests and 7 tests ($m = 8$); (b) Results from input2 data set, 6 tests and 5 tests ($m = 8$).

(a)		
Unique Items	8 Tests ($n = 8$)	7 Tests ($n = 7$)
8 ($U \geq 8$)	\$108304.29/60.04%/1803.79/ 40,320	
	\$108612.80/60.04%/1809.28	
7 ($U \geq 7$)	\$95029.27/64.46%/1542.01/ 1,169,280	\$93570.91/66.09%/1436.78/ 40,320
	\$96741.80/64.46%/1571.92	\$94206.38/66.09%/1452.76
6 ($U \geq 6$)	\$83076.84/67.87%/1330.29/ 6,531,840	\$81015.21/69.21%/1232.56/ 463,680
	\$83645.80/67.86%/1339.40	\$82086.96/69.20%/1244.64
5 ($U \geq 5$)	\$71380.65/71.31%/1124.94/ 13,587,840	\$68788.28/72.51%/132.55/ 1,404,480
	\$71380.65/70.77%/1124.94	\$69016.32/72.40%/1035.97
4 ($U \geq 4$)	\$60362.87/74.95%/934.54/ 16,445,520	\$57279.70/76.11%/845.69/ 1,992,480
	\$60362.87/73.33%/934.54	\$57631.01/75.62%/851.11
3 ($U \geq 3$)	\$52268.95/78.79%/818.15/ 16,770,096	\$48829.58/79.96%/729.32/ 2,093,616
	\$52268.95/73.83%/818.15	\$48829.58/75.57%/729.32
(b)		
Unique Items	6 Tests ($n = 6$)	5 Tests ($n = 5$)
6 ($U \geq 6$)	\$78960.93/70.96%/1135.74/ 20,160	
	\$79181.18/70.96%/1138.91	
5 ($U \geq 5$)	\$66187.48/73.94%/943.94/ 120,960	\$63,164.80/75.81%/851.43/ 6720
	\$66081.94/73.92%/942.44	\$63,545.21/75.74%/856.89
4 ($U \geq 4$)	\$54184.03/77.40%/761.95/ 230,160	\$50,691.87/78.92%/677.28/ 23,520
	\$54083.74/76.45%/760.54	\$50,811.80/78.87%/678.88
3 ($U \geq 3$)	\$45376.62/81.20%/646.32/ 260,400	\$41,586.50/82.57%/564.17/ 31,920
	\$45271.11/80.02%/644.96	\$41,486.21/82.08%/562.81

Table 10. (a) Results from input3 data set, 10 tests and 9 tests ($m = 10$); (b) Results from input3 data set, 8 tests and 7 tests ($m = 10$); (c) Results from input3 data set, 6 tests and 5 tests ($m = 10$).

(a)		
Unique Items	10 Tests ($n = 10$)	9 Tests ($n = 9$)
10 ($U \geq 10$)	\$40465.48/55.29%/731.88/ 488,880	
	\$40870.17/55.29%/737.29	
9 ($U \geq 9$)	\$35491.57/59.83/648.65/ 22,715,280	\$34465.71/60.93%/601.74/ 3,628,800
	\$35957.54/59.51%/657.36	\$34642.23/60.93%/604.82
8 ($U \geq 8$)	\$30712.13/64.18%/569.57/ 210,133,944	\$29801.06/65.08%/529.18/ 68,947,200
	\$30712.13/64.18%/569.57	\$30534.13/65.08%/531.37
7 ($U \geq 7$)	\$28270.43/67.71%/499.33/ 706,758,648	\$27059.92/68.51%/456.39/ 348,364,800
	\$28270.43/65.68%/499.33	\$27220.64/68.47%/458.76
6 ($U \geq 6$)	\$25765.24/71.30%/446.09/ 1,196,888,433	\$24292.98/72.06%/404.17/ 748,440,000
	\$25765.24/67.99%/446.09	\$24348.19/69.08%/405.09
5 ($U \geq 5$)	\$23466.13/75.06%/397.87/ 1,383,381,140	\$22033.24/75.82%/360.80/ 958,638,240
	\$23466.13/69.02%/397.87	\$22172.48/71.58%/363.08
4 ($U \geq 4$)	\$21795.04/78.49%/353.87/ 1,409,014,434	\$20447.51/79.27%/319.71/ 997,799,040
	\$21795.04/71.29%/353.87	\$20469.30/71.93%/320.07
3 ($U \geq 3$)	\$20153.99/82.06%/318.31/ 1,410,057,744	\$18810.58/82.86%/286.15/ 999,977,040
	\$20153.99/72.98%/318.31	\$18842.15/73.85%/286.63
(b)		
Unique Items	8 Tests ($n = 8$)	7 Tests ($n = 7$)
8 ($U \geq 8$)	\$28974.20/66.27%/492.42/ 1,814,400	
	\$29209.10/66.27%/496.42	
7 ($U \geq 7$)	\$26000.93/69.46%/420.19/ 18,748,800	\$24950.37/70.74%/385.79/ 604,800
	\$26000.93/69.43%/420.19	\$25015.17/70.72%/386.94
6 ($U \geq 6$)	\$23168.06/72.91%/370.56/ 58,968,000	\$21901.82/73.93%/336.03/ 3,780,000
	\$23168.06/71.07%/370.59	\$21904.49/73.84%/336.07
5 ($U \geq 5$)	\$20816.65/76.63%/328.01/ 90,720,000	\$19435.17/77.54%/294.41/ 8,013,600
	\$20816.85/73.29%/328.01	\$19435.17/75.08%/294.41
4 ($U \geq 4$)	\$19170.73/80.07%/288.61/ 99,283,040	\$17735.24/80.93%/256.80/ 9,777,600
	\$19170.73/75.47%/288.61	\$17737.84/76.12%/256.80
3 ($U \geq 3$)	\$17372.96/83.68%/254.51/ 99,988,560	\$15977.95/84.53%/225.37/ 9,994,320
	\$17372.96/75.96%/254.51	\$15977.95/79.25%/225.37
(c)		
Unique Items	6 Tests ($n = 6$)	5 Tests ($n = 5$)
6 ($U \geq 6$)	\$20615.41/75.29%/302.15/ 151,200	
	\$20615.41/75.27%/302.15	
5 ($U \geq 5$)	\$18056.15/78.62%/262.38/ 604,800	\$16666.12/86.44%/231.36.23/ 30,240
	\$18056.15/77.97%/262.38	\$16666.12/79.48%/231.36
4 ($U \geq 4$)	\$16275.69/81.88%/226.66/ 932,400	\$14890.26/83.02%/198.92/ 80,640
	\$16275.69/79.42%/226.66	\$14890.26/82.08%/198.92
3 ($U \geq 3$)	\$14513.67/85.43%/197.00/ 997,200	\$13012.28/86.44%/169.78/ 98,640
	\$14513.67/81.93%/197.00	\$13012.28/83.93%/169.78

Table 11. (a) Results from input4 data set, 10 tests and 9 tests ($m = 12$); (b) Results from input4 data set, 8 tests and 7 tests ($m = 12$); (c) Results from input4 data set, 6 tests and 5 tests ($m = 12$).

(a)		
Unique Items	10 Tests ($n = 10$)	9 Tests ($n = 9$)
10 ($U \geq 10$)	\$80999.24/37.14%/2400.49/ 9,072,000	
	\$80999.24/37.12%/2400.49	
9 ($U \geq 9$)	\$71170.34/40.68%/2037.85/ 142,098,800	\$73754.08/41.90%/1923.63/ 13,334,800
	\$71170.34/40.65%/2037.85	\$73754.08/41.90%/1923.63
8 ($U \geq 8$)	\$66202.16/44.84%/1671.22/ 681,464,197	\$65787.55/45.82%/1606.71/ 133,503,802
	\$66202.16/44.84%/1671.22	\$66811.51/45.62%/1606.71
7 ($U \geq 7$)	\$58975.41/49.41%/1355.91/ 1,499,164,173	\$58079.48/50.28%/1291.14/ 442,364,786
	\$58975.41/49.27%/1355.91	\$58079.48/49.88%/1291.14
6 ($U \geq 6$)	\$54407.08/54.26%/1148.44/ 2,005,855,842	\$53048.18/55.12%/1081.55/ 737,622,797
	\$54407.08/54.26%/1148.44	\$53048.18/54.78%/1081.55
5 ($U \geq 5$)	\$51792.00/59.60%/974.33/ 2,133,046,519	\$49219.62/60.47%/909.06/ 848,602,360
	\$51792.00/59.60%/974.33	\$49219.62/60.47%/909.06
4 ($U \geq 4$)	\$48890.43/64.78%/837.93/ 2,144,749,728	\$45903.32/65.70%/771.68/ 864,134,501
	\$48890.43/64.78%/837.93	\$45903.32/65.70%/771.68
3 ($U \geq 3$)	\$46586.00/70.26%/732.87/ 2,145,057,599	\$44010.40/71.24%/666.68/ 864,807,380
	\$46586.00/70.26%/732.87	\$44010.40/71.24%/666.68
(b)		
Unique Items	8 Tests ($n = 8$)	7 Tests ($n = 7$)
8 ($U \geq 8$)	\$63309.83/47.20%/1481.74/ 19,958,400	
	\$64164.62/47.08%/1501.67	
7 ($U \geq 7$)	\$54760.29/51.38%/1,212.95/ 131,725,440	\$52531.92/52.92%/1100.06/ 3,991,680
	\$56002.74/51.11%/1214.23	\$52606.02/52.52%/1103.67
6 ($U \geq 6$)	\$49605.71/56.09%/1006.99/ 308,689,920	\$46555.07/57.32%/911.72/ 17,962,560
	\$50420.16/55.60%/1011.79	\$48595.21/56.92%/941.03
5 ($U \geq 5$)	\$46304.04/61.42%/841.29/ 408,481,920	\$43200.72/62.51%/774.38/ 31,268,160
	\$46977.25/61.32%/841.29	\$43208.83/61.46%/783.01
4 ($U \geq 4$)	\$42892.17/66.67%/710.34/ 428,689,800	\$39965.73/67.71%/649.06/ 35,426,160
	\$43062.89/66.67%/710.34	\$40406.85/67.60%/662.62
3 ($U \geq 3$)	\$40700.94/72.25%/602.77/ 429,964,920	\$37447.87/73.31%/544.04/ 35,823,480
	\$40700.94/72.25%/602.77	\$38209.54/73.31%/549.97
(c)		
Unique Items	6 Tests ($n = 6$)	5 Tests ($n = 5$)
6 ($U \geq 6$)	\$42254.10/59.04%/832.09/ 665,280	
	\$44254.10/58.61%/832.09	
5 ($U \geq 5$)	\$40555.13/63.87%/704.48/ 2,090,880	\$36361.25/65.79%/618.95/ 95,040
	\$40555.13/62.80%/704.48	\$36361.25/65.65%/618.95
4 ($U \geq 4$)	\$36762.93/68.91%/586.65/ 2,863,080	\$32782.30/70.41%/511.97/ 213,840
	\$37313.61/68.73%/596.41	\$33154.85/70.24%/517.79
3 ($U \geq 3$)	\$34046.74/74.46%/485.23/ 2,981,880	\$29948.18/75.78%/425.67/ 246,840
	\$34046.74/74.46%/488.46	\$30130.32/75.78%/426.29

Algorithm for this example problem, the last line of **Table 12** shows the Relative Inferiority of the Genetic Algorithm, where the Relative Inferiority metrics were introduced via Equations (11)-(13).

For the hypothesis tests ($H_1 - H_3$), a summary is detailed in **Table 13** below.

5.2. Discussion of Results

All three hypothesis tests show that the performance of the Genetic Algorithm is inferior to optimal results obtained via enumeration. As such, we are unable to claim that the Genetic Algorithm provides, from a statistical standpoint, optimal results.

However, the number of conforming units accepted through the n^{th} inspection station ($N(A|G)$) metric shows inferiority of 1.73% compared to optimal, whereas total cost of the inspection plan (TC) and the ratio (Z) both show inferiority of less than 0.50%. The authors therefore consider the performance of the Genetic Algorithm formidable.

It should be noted that of the three performance measures shown above, the relative inferiority of the objective function value (Z) show the least relative inferiority, or stated another way, shows the most formidable result. This should not come as a surprise, considering the genetic algorithm was singularly motivated to find the lowest possible value of Z .

6. Concluding Comments

This effort presented a Genetic Algorithm approach to addressing the multiple-attribute inspection problem when multiple objectives are a concern. The literature has seen little work done for multiple inspection problems when multiple product attributes require inspection. As such, the authors felt compelled to address this relevant problem. Additionally, most work done for multiple inspection problems focuses on the objective of cost

minimization. Given this, the authors also felt compelled to address maximization of conforming units surviving inspection in addition to cost minimization.

The authors felt compelled to address an objective in addition to cost minimization for the simple reason that cost minimization alone will favor all low-cost possibilities in terms of choosing attributes to be inspected as part of the inspection plan. The authors wanted to simultaneously consider a non-cost related performance measure, so that low-cost possibilities do not dominate the construction of the inspection plan. Maximization of conforming units surviving the final inspection station was considered a reasonable objective to pursue.

While the Genetic Algorithm approach did not provide optimal results, the results were still considered formidable, where the level of inferiority, in terms of the objective function (which considers both cost and conforming units accepted), obtained via the Genetic Algorithm, was less than 0.30%. The authors are comfortable in classifying this result as “near-optimal”.

Despite a formidable performance by the Genetic Algorithm, concerns remain. Past research has shown that heuristic performance, as compared to optimal performance, generally deteriorates to some degree as the problem size increases [18]. For this research, the largest problems encountered had been in excess of two billion feasible solutions. From a practical standpoint, it is difficult to enumerate all feasible solutions for problems larger than this. As such, gauging the performance of the Genetic Algorithm as compared to optimal performance is a challenge that remains. To compare this Genetic Algorithm’s performance to optimal performance for larger problems, improvements in CPU speed and memory capacity must come to fruition.

Another issue that requires mentioning relates to the objectives of cost minimization and maximization of conforming units surviving inspection. These two objectives were simultaneously considered via a ratio involving the two. This mathematical type of aggregation of the two objectives can certainly lead to a discussion regarding the best way to aggregate multiple objectives, and can start a debate on the best way to “scale” the objectives, “weight” the objectives and so on. This unresolved issue is considered as a future research opportunity.

Another future research opportunity relates to other objectives, aside from, or in addition to, the objectives addressed via this research effort. Cost minimization and maximization of conforming units surviving all inspection stations are not the only entities of importance. Even though the cost of sending a defective unit to the customer is factored into the total cost function, this concern could be directly addressed via an objective function component. Another entity that could be addressed is cost minimization on a per inspection basis. At present,

Table 12. Solution detail for example problem.

Approach	TC	$N(A G)$	Z
Enumeration	\$83076.84	67.87%	1330.29
Genetic Algorithm	\$83645.80	67.86%	1339.40
Relative Inferiority	0.68%	0.01%	0.68%

Table 13. Hypothesis test results for test problems.

Test	Metric	Sample Mean	Sample Std. Dev.	t-statistic	p-value
H_1	TC	0.4487%	0.8626%	4.9625	<0.0001
H_2	Z	0.2987%	0.5608%	5.0804	<0.0001
H_3	$N(A G)$	1.7229%	2.8572%	5.7572	<0.0001

the number of inspection stations to employ is pre-determined, or constrained. This need not be the case—a future research opportunity could explore cost minimization where the number of inspection stations need not be pre-determined, but be determined via the chosen search heuristic.

REFERENCES

- [1] A. Tzimerman and Y. T. Herer, “Off-Line Inspections under Inspection Errors,” *IIE Transactions*, Vol. 41, No. 7, 2009, pp. 626-641.
<http://dx.doi.org/10.1080/07408170802331250>
- [2] G. D. Eppen and G. Hurst, “Optimal Location of Inspection Stations in a Multistage Production Process,” *Management Science*, Vol. 20, No. 8, 1974, pp. 1194-1200.
<http://dx.doi.org/10.1287/mnsc.20.8.1194>
- [3] T. Raz and M. Kaspi, “Location and Sequencing of Imperfect Inspection Operations Serial Multi-Stage Production Systems,” *International Journal of Production Research*, Vol. 8, No. 3, 1991, pp. 1645-1659.
- [4] D. P. Ballou and H. L. Pazer, “The Impact of Inspector Fallibility on the Inspection Policy in Serial Production Systems,” *Management Science*, Vol. 28, No. 4, 1982, pp. 387-399.
- [5] T. Raz and M. U. Thomas, “A Method for Sequencing Inspection Activities Subject to Errors,” *IIE Transactions*, Vol. 15, No. 1, 1983, pp. 12-18.
<http://dx.doi.org/10.1080/05695558308974608>
- [6] T. Raz and D. Bricker, “Sequencing of Imperfect Inspection Operations Subject to Constraints on the Quality of Accepted and Rejected Units,” *International Journal of Production Research*, Vol. 25, No. 6, 1987, pp. 809-821.
<http://dx.doi.org/10.1080/00207548708919878>
- [7] T. Raz and D. Bricker, “Optimal and Heuristic Solutions to the Variable Inspection Policy Problem,” *Computers & Operations Research*, Vol. 18, No. 1, 1991, pp. 115-123.
[http://dx.doi.org/10.1016/0305-0548\(91\)90047-U](http://dx.doi.org/10.1016/0305-0548(91)90047-U)
- [8] J. W. Schmidt and G. K. Bennett, “Economic multiattribute Acceptance Sampling,” *AIIE Transactions*, Vol. 4, No. 3, 1972, pp. 194-199.
- [9] K. Tang, R. Plante and H. Moskowitz, “Multiattribute Bayesian Acceptance Sampling Plans under Non-Destructive Inspection,” *Management Science*, Vol. 32, No. 6, 1986, pp. 739-750.
<http://dx.doi.org/10.1287/mnsc.32.6.739>
- [10] K. Tang and J. Tang, “Design of Product Specifications for Multi-Characteristic Inspection,” *Management Science*, Vol. 35, No. 6, 1989, pp. 743-756.
<http://dx.doi.org/10.1287/mnsc.35.6.743>
- [11] L. L. Hau, “On the Optimality of a Simplified Multicharacteristic Component Inspection Model,” *IIE Transactions*, Vol. 20, No. 4, 1988, pp. 393-398.
- [12] C. Shaoxiang and M. Lambrecht, “The Optimal Frequency and Sequencing of Tests in the Inspection of Multicharacteristic Components,” *IIE Transactions*, Vol. 29, No. 12, 1997, pp. 1039-1049.
<http://dx.doi.org/10.1080/07408179708966422>
- [13] T. Avinadav and D. Sarne, “Sequencing Counts: A Combined Approach for Sequencing and Selecting Costly Unreliable Off-Line Inspections,” *Computers & Operations Research*, Vol. 39, No. 11, 2012, pp. 2488-2499.
<http://dx.doi.org/10.1016/j.cor.2011.12.018>
- [14] T. Raz and D. Bricker, “Sequencing of Inspection Operations Subject to Errors,” *European Journal of Operational Research*, Vol. 68, No. 2, 1993, pp. 251-264.
[http://dx.doi.org/10.1016/0377-2217\(93\)90307-9](http://dx.doi.org/10.1016/0377-2217(93)90307-9)
- [15] J. S. Park, M. H. Peters and K. Tang, “Optimal Inspection Policy in Sequential Screening,” *Management Science*, Vol. 37, No. 8, 1991, pp. 1058-1061.
<http://dx.doi.org/10.1287/mnsc.37.8.1058>
- [16] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning,” Addison-Wesley, Reading, 1989.
- [17] Z. Michalewicz and D. Fogel, “How to Solve It: Modern Heuristics,” Springer, New York, 2010.
- [18] P. R. McMullen, “JIT Mixed-Model Sequencing with Batching and Setup Considerations via Search Heuristics,” *International Journal of Production Research*, Vol. 48, No. 22, 2010, pp. 6559-6582.
<http://dx.doi.org/10.1080/00207540903321640>