Agile Inspection Planning Based on CAD-Data

Rami Musa[1], Robert H. Sturges[2] and F. Frank Chen[3]

Virginia Polytechnic Institute and State University
[1] rmusa@vt.edu
[2] sturges@vt.edu
[3] ffchen@vt.edu

ABSTRACT

Variations in components are inevitable and have to be dealt with efficiently. One way to reduce variation is by implementing efficient inspection plans that can minimize the total cost of the product. Manufacturers deal with this problem with many different strategies when minimal information is known about the product (newly launched products). In this paper, we develop a methodology for inspection planning for a new product based on CAD data and simulation. Decision variables in this formulation includes: how often to inspect a quality characteristic in a subassembly (probability of inspection), and the corrective action if the inspected quality characteristic is found to be out-of-tolerance. Inspection frequency as a decision variable was not proposed in this context before. Additionally, the proposed action plan is more realistic here than what has been proposed in the literature. Illustrative examples are presented and solved for demonstrative purposes and validation of our findings.

Keywords: Inspection Planning, Variation Reduction, Inspection Frequency, CAD Data

1. INTRODUCTION AND LITERATURE REVIEW

With the need for agile production for newly introduced and customized products, engineers are faced with the problem of developing sound inspection plans for new products. We can define inspection planning as the act of identifying what, when (how often), how to measure the subassemblies and the corrective action based on the measurement. We do not consider here the sensor allocation problem that aims at reducing product variation by inspecting the process items that make the parts, such as fixtures, cutting tool, etc. Rather, we look into what subassembly parts need to be inspected in order to minimize the total cost. In some sense, sensor allocation handles the causation of the variation in the parts. Azam et al. [1] modeled the problem of sensor allocation as a Knapsack problem, Lindsay and Bishop [9] used dynamic programming, Ding et al. [5] modeled it using the Stream of Variation Approach. Khan et al. [8] focused their work on the problem of allocating 3 sensors on a fixture unit that implements a 3-2-1 principle. They found that nearly 70% of the final product variability comes from fixture variation. On the other hand, Chen and Thornton [3] used Monte Carlo Simulation and a Simulated Annealing algorithm (SA) to solve the inspection planning problem for the parts. More discussion about their work is elaborated later in this paper. Greenstein and Rabinowitz [7] solved the problem statistically in two stages. The objective in their study was to fully inspect $K<n$ components in the first stage that “explain” the whole behavior of the n components. Their objective function is to minimize the cost of accepting a “bad” product, the cost of rejecting a “good product” and the cost of inspection. After that, they determine whether it is cheaper to inspect the rest of the batch or not. The authors assumed that the joint probability distribution function of the components is known apriori and that it is normally distributed. Moreover, they did not consider in the model any possible rework or scrap actions in their model and the specification limits were input information rather than being decision variables. Chen and Chung [4] introduced a model to determine the inspection precision and the optimal number of repeated measurements in order to maximize the net expected profit per item. The model is specifically applicable for the lower specification-limit quality characteristic; i.e. the specification has unbounded upper limit. The profit is modeled as the difference between the selling price and the following costs: inspection, production, and dissatisfying the customer. There is an assumption that all measurements are normally distributed and all items are completely inspected at least once because of inspection inaccuracy. Their model is mostly appropriate for industries where there is a need for repeated measurements because of known measurement errors and where the production is at late a stage of producing an item in the supply chain.

This work has been inspired by a work done by Chen and Thornton [3]. Their work can be used to develop inspection planning for existing products when there is enough variation data. However, when a new product is introduced, their approach has to be modified to integrate the approach with an alternative source of data, such as CAD Variation Analysis Software as we are proposing here. 3DCS and VSA are quality prediction simulation-based CAD packages. They primarily predict the variation of an assembly when variation information about the subassembly features is known. Another shortcoming in Chen and Thornton's [3] work is that the frequency of inspection was not considered as a possible decision variable in the problem because the throughput of the production system was not taken into consideration. As we will see later, sometime it is best to partially inspect a feature rather than fully inspecting it or not inspecting it at all. Additionally, the reaction decision based on inspection is not quite realistic because the optimal decision in Chen and Thornton's work could be to completely rework or completely scrap if an item is out of tolerance (*LL*, *UL*). We propose improving the model accordingly. The novelty of our work here is three-fold: (1) introducing the frequency of inspection as a possible optimizer, (2) creating a realistic action-based-upon-variation plan, and (3) using CAD data (simulated data) to develop an optimal inspection plan.

This paper is divided into five Sections and organized as follows: In this Section (Section 1), we started with introducing the need for inspection planning for new products, explained the originality of this work and presented a literature review on the topic. In Section 2, we introduce the problem definition and the total cost function we are optimizing. Next, we present the mathematical model and our methodology by mainly showing the details of the approach in flowcharts. We also raise the question whether the frequency of inspection has to be a decision variable in minimizing the total cost. Numerical examples are solved afterwards in Section 3 for illustrative and validation purposes for our findings. Finally, we close with concluding remarks and recommendations, and future research ideas in Sections 4 and 5; respectively.

2. APPROACH

2.1. Problem

We propose an approach that can be used to develop inspection plans based on process capability data, CAD, simulation and optimization search techniques. The overall objective to be minimized was proposed in Chen and Thornton [3] and shown in Eqn. (1). It is the total costs of inspection (C_I), scrapping (C_S), reworking (C_R) and failure (C_F). It is important to mention at this point that the inspection, scrapping and rework are associated with subassembly quality characteristics, where the failure cost is associated with the final assembly. It is intuitive that when the failure cost is very high, then it is best to inspect everything. On the other hand, it is best not to inspect when the failure cost is negligible. We will elaborate on that more lately in the paper.

Minimize $$TC = C_I + C_S + C_R + C_F \tag{1}$$

Assuming normality for the contributing and concluding quality characteristics, the objective function in Eqn. (1) can be further decomposed as shown in Eqn. (2).

Minimize

$$TC = R\sum_{t=1}^{M} freq_t \left\{c_{It} + c_{Rt}P_{Rt}\left(LL_{S_t}, LL_{R_t}, UL_{S_t}, UL_{R_t}, \mu_t, \sigma_t\right) + c_{St}P_{St}\left(LL_{S_t}, UL_{S_t}, \mu_t, \sigma_t\right)\right\} + Q\sum_{a=1}^{N} c_{Fa}P_{Fa}\left(LL_a, UL_a, \mu_a, \sigma_a\right) \tag{2}$$

The quality characteristic of the final assembly (*a*) is given by:

$$q_a \sim N(\mu_a, \sigma_a) = f\left(freq_1, \dots, freq_M, \mu_1, \dots, \mu_M, \sigma_1, \dots, \sigma_M, LL_{S_1}, \dots, LL_{S_M}, LL_{R_1}, \dots, LL_{R_M}, UL_{S_1}, \dots, UL_{S_M}, UL_{R_1}, \dots, UL_{R_M}\right)$$

Where:

TC: Total cost

R: The number of items in each subassembly group before inspection

$freq_t$: Frequency of inspection for subassembly quality characteristic *t*

c_{It} : Inspection cost per subassembly quality characteristic *t*

c_{Rt} : Rework cost per subassembly quality characteristic *t*

P_{Rt} : Probability of reworking quality characteristic *t* (area under II and III in Fig. 1)

c_{St} : Scrap cost per subassembly for quality characteristic *t*

P_{st} : Probability of scrapping for quality characteristic t (area under I and IV in Fig. 1)

c_{Fa} : Failure cost per final assembly a

P_{Fa} : Probability of failure for the final assembly a

M: Number of subassembly quality characteristics
N: Number of final assembly quality characteristics
Q: Maximum number of items in a subassembly to be assembled, Q is less than R because the scrapped items will not be used for final assembly.

The cost function in Eqn. (2) cannot be easily dealt with analytically because there are no explicit forms to express the probabilities of rework, scrap and failure (P_{Rt},P_{St}, P_{Fa}). Although it is possible and yet tedious to fit functions that give probabilities for the given inputs (UL's and LL's), it is not reasonable to do so unless the functions are normally distributed and the functions that map the contributing quality characteristics to the final assembly quality characteristics are known. Further, it is unlikely for the resultant quality characteristic of the final assembly to be analytically determined although the contributing ones are normally distributed because of the reworks and scraps that are performed on the contributing quality characteristics. Therefore, we resort to Monte Carlo simulation to generalize the model as it was proposed by Chen and Thornton [3]. We assume that a final assembly is made of M subassemblies. We also assume that each subassembly contains a single quality characteristic (t); however the final assembly contains N quality characteristics.

Notice that we have introduced the *frequency* of inspecting a quality characteristic for a subassembly t ($freq_t$) as a portion of the objective function. Later in this paper, we study whether the frequency of inspection can be a decision variable or not. Moreover, we introduce two ranges of a tolerance (LL_S, LL_R, UL_S, UL_R) for each quality characteristic t. The reason for that is to generalize the action of rework and scrap compared to what is found in Chen and Thornton [3] where they proposed that a part has to be either reworked or scrapped if it is out-of-tolerance. What was proposed can be enhanced by introducing two additional decision variables for each subassembly t. In Fig. 1, when a subassembly t is in regions I or IV (so far from the nominal value), then we scrap it. However, if it is in regions II and III (not so far from the nominal value), then we rework it. Finally, if it is located otherwise we keep it. This is how we propose implementing a more realistic corrective plan.

It is practical to impose some functional constraints sometimes. If we know for a fact that it is impossible to rework an inspected item if a dimension is less or larger than a specified dimension, then we can impose a constraint that allows no rework by imposing a constraint that LL_S=LL_R or UL_R=UL_S. For instance; in hole-diameter inspection, we should impose the following constraint: UL_R=UL_S. On the other hand, in shaft-diameter inspection, we should impose the following constraint: LL_R=LL_S. Doing so reduces the search for the optimal solution as it is not feasible to rework an oversized hole or undersized shaft.

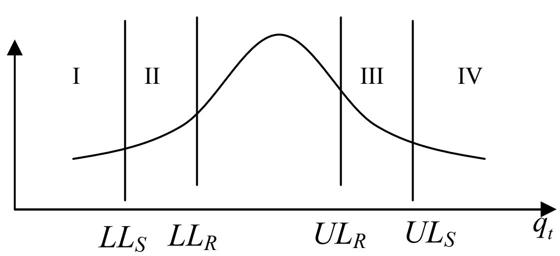

Fig. 1. QC for subassembly t, I & IV: Scrap; II & III: Rework.

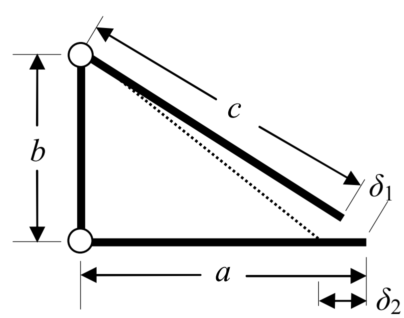

Fig. 2. Simple Assembly.

The number of decision variables for our problem so far is 5M; where M is the number of subassembly groups. The five decision variables for a subassembly are: $freq_t$, LL_S, LL_R, UL_R, UL_S. Suppose we have three quality characteristics in three subassemblies (a, b and c dimensions) that comprise final assembly requirements δ_1 and δ_2. Example for this case can be shown in Fig. 2. In this case, we are interested in mapping the relation between a, b and c with δ_1 and δ_2.

Sometimes, these mapping functions can be easily found as in the case of our example in Fig. 2. The mapping functions are shown in Eqn. (3). However, most of the time, it is hard to map these quality characteristics, therefore as a main part of our methodology; we are proposing using CAD data and then regression to evaluate the mapping functions.

$$\delta = f(a,b,c)$$
$$\delta_1 = f(a,b,c) = c - \sqrt{a^2 + b^2}$$
$$\delta_2 = f(a,b,c) = a - \sqrt{c^2 - b^2} \qquad (3)$$

For the example shown in Fig. 2, the inputs and the decision variables to conduct inspection planning search are:
Inputs: $UL_{\delta 1}$, $UL_{\delta 2}$, $LL_{\delta 1}$, $LL_{\delta 2}$, PDF_a, PDF_b, PDF_c, c_I, c_R, c_F, c_S (Note: c_I, c_R, c_F, c_S are inspection, rework, failure, and scrap costs of a single item and *PDF* stands for Probability Density Function).
Outputs (Decision variables): LLS_a, LLR_a, ULR_a, ULS_a, LLS_b, LLR_b, ULR_b, ULS_b, LLS_c, LLR_c, ULR_c, ULS_a, $freq_a$,$freq_b$, $freq_c$.

2.2. Methodology

In order to find the failure cost in Eqn. (2), we need to find a function that maps the input variation data to the outputs. Therefore; for a new product, we may not know the function that maps the subassembly with the assembly dimensions. We propose using variation analysis software such as 3DCS [6] or VSA to find out that function (refer to Figure 3). 3DCS and VSA are Monte Carlo simulation based software that analyzes the tolerance stackup for an assembly.

Since we cannot generally express the objective function explicitly, we propose using a Genetic Algorithm (GA). Our approach using a GA is summarized in a flowchart in Fig. 5. Fig. 6 further decomposes the cost estimation functions in the right hand half of Fig. 5. Details for these flowcharts are given in Sections 2.2.1 and 2.2.2. With frequency of measurements as the only decision variables, all the terms in the objective function in Eqn. (2) are linear (refer to Fig. 4). Hence, the total objective function will be linear. This means that the optimal inspection plan will be to either to fully inspect a subassembly (feature) *t* or not to measure it at all; i.e. $freq_t$=1 or 0. Refer to Fig. 4, we can say that the inspection, rework and scrap costs increase by increasing the inspection intensity. On the other hand, increasing the inspection intensity decreases the failure cost. Therefore, our inspection plan has to be a trade-off between those costs. We will discuss this in details in Section 2.3.

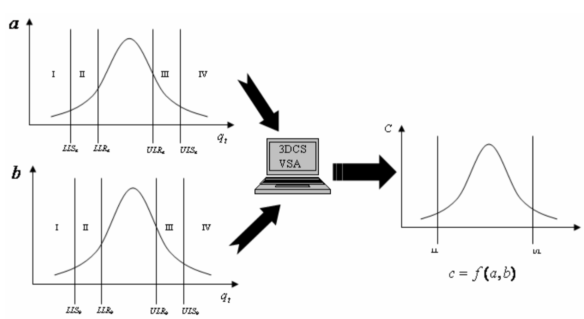

Fig. 3. Resultant Variation Data for the Final Assembly *c*. Note: The function that maps the input variation data (*a* and *b*) with the output variation data is evaluated by regression when the data are obtained from 3DCS or VSA.

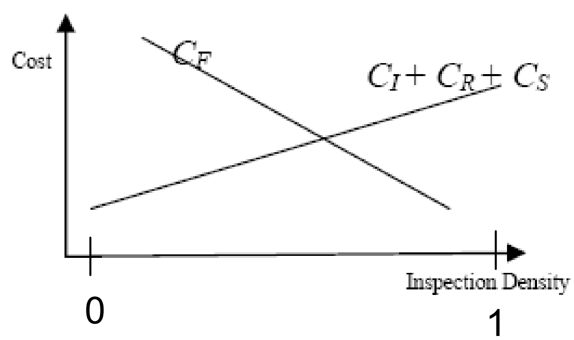

Fig. 4. Unconstrained objective function behavior (no yield constraint).

2.2.1. Genetic Algorithm (GA)

As is well-known, Genetic Algorithm is a meta –heuristic that mimics Darwin's evolutionary Theory. It has been proven to be an effective approach in solving difficult optimization problem. Some important terms in GA are: chromosome, gene, population, crossover, mutation, fitness, parent, offspring, and elitism. Solutions in GA are represented by a population of chromosomes (strings) of binary numbers (genes), e.g. 0110001. Our goal is to optimize (maximize or minimize) an objective function (we call it fitness). We can summarize the process for an unconstrained optimization problem with an example as follows:

(1) Generate Initial Population of Chromosomes (Solutions) - We usually begin with a population of randomly generated solutions. The size of population (P) impacts the convergence toward the optimal solution.

(2) Fitness Evaluation - Evaluate the fitness for each chromosome, rank and the sort them accordingly. Fitness evaluation is usually determined through an explicit function. Sometime, fitness has to be estimated through simulation as it is in our case here (refer to Fig. 6).

(3) Selection Process - We select $P/2$ pair of chromosomes (parents) in order to generate P offsprings, which means that each pair generates two offsprings. The fittest chromosomes will have more chance to be selected to be paired in the next generation. This can be achieved using the rank selection approach by associating each chromosome with the following number: $P_n = (P-n+1)/\sum_{i=1}^{P} i$, where P is total number of chromosomes and n is the chromosome rank.

(4) Crossover Process – Each selected pair of chromosomes from the previous step will generate two offsprings. Each offspring shares genes from its parents. This is done by slicing the chromosomes and crossing over their genes. This crossover location is randomly generated to be uniformly distributed between 0 and P.

(5) Mutation Process – With a particular probability (relatively very small, typically 1 to 5%), there is a chance for each gene in the generated offspring to be swapped from a pool of all possible values.

(6) Fitness Evaluation of the Offspring Generation – The fitness values for each of the P generated offspring solutions are found.

(7) Generate the New Generation of Solution - Select the fittest P chromosomes from the parents and offsprings.

(8) If stopping criteria is achieved, stop; otherwise go to step 2.

What you see in Fig. 5 is the same procedure that we introduced here for GA but with the following two modifications:

(1) Since the decision variables are continuous variables, we created a discrete pool of gene values from which a value can be selected. For instance, *frequency* can take a value between 0 and 1. Hence, we created a function (its name is Discritizer function in the flowchart) that can match a randomly generated variable between 0 and 1 with the pool of gene values that we selected. In this paper, we generated three pools (sets): (1) 0 or 1, (2) [0.0, 0.1, 0.2… 1.0], and (3) [0.00, 0.05, 0.10, 0.15… 1.00]. Apparently, finer pools (sets) can generate more accurate solution but they are more computationally expensive. On the other hand, we discritized the specification limits differently (UL_R, UL_S, LL_R, LL_S). These decision variables were given sigma numbers; i.e. $UL_{Rt}=3$ means $\mu_i+3\sigma_i$ value for a QC t.

(2) For each quality characteristic, t; in order for a string (chromosome) to be feasible, we forced the constraint that $UL_R \geq UL_S$ and $LL_s \geq LL_R$. Moreover, we imposed the constraint of minimum yield (as it is shown in Fig. 6) by giving the total cost *infinity* value when the minimum yield constraint is not satisfied.

2.2.2. *Cost Estimation*

Fig. 6 summarizes how we estimate the cost (fitness) by using the simulation. This is a modified approach to what Chen and Thornton [3] proposed. The thick arrows in the Figure connect the external data (such as CAD data, Chromosome, etc) to the internal functions. The flowchart is divided into two components: inspection and final assembly. It starts off through design decision for all possible subassembly QC (X's) contributors on the final requirements (Y's) and by assuming the behavior of those chosen X's (upper left corner of Fig. 5). After that, enough simulations are run through 3DCS or VSA to find the Y's that are associated with the generated X's. The Y's are then mapped with the X's through regression. This fitted function will be needed in the final assembly part of the simulation. To find a cost, we generate a random variable that represents the t^{th} subassembly according to the known *PDF* behavior; X_{tr}. We put the generated QC into the inspection process according to a given frequency. If the generated value (X_{tr}) was found to be in region I or IV, then we scrap it (send nothing to be assembled by giving the X_{tr} a value of zero) and we add that cost. However, if it is located in 2 or 3, then we rework it to the near nominal value and we add that cost. If the part was not inspected, then we send it right away to the final assembly batch. Only the non-inspected and the reworked parts are sent to the final assembly batch. After checking all the input subassembly QC ($t=1…M$) for all the given parts ($r=1…R$), we reach to a point where we have different number of subassemblies because of the scrap procedure. Suppose there are 2 subassembly groups (a and b) of size 100 for each one of them. If 50/0 parts of subassembly groups a/b were scrapped, this means that only 50% of subassembly b will be utilized and 100% of the resultant a's will be utilized. At that point, we can say that the yield reduced from 100% to 50% because of the inspection. The number of parts for this example will be 50 ($Q=50$ in the final inspection simulation). Q can be given as follows:

$$Q = R - \max\left(\sum_{r=1}^{R} step(XX_{tr}), \forall t = 1,...,M\right) \quad (4)$$

$$step(y) = \begin{cases} 1, y > 0 \\ 0, otherwise \end{cases}$$

In the final assembly part of the simulation, we take a set of input *X*'s and find the associated *X*'s according to the function found after fitting the CAD data. Alternatively, we can feed the *X*'s data we collected to the CAD model to predict the *y*. If the part (*Y*) is found to be within specifications (*LL* and *UL*), then we proceed with the next part. However, if it does not fall within the specification limits, then we fail it and we add the failure cost. After examining all the *Q* parts, we estimate the yield based on that. We notice that the final yield is dependent on the scrap and failure rates, as follows:

$$YRT = \frac{Q - Q_F}{R} \quad (5)$$

Where:

YRT : Rolled Yield Throughput

Q : Maximum number of items in a subassembly to be assembled

R : Number of subassembly parts before inspection

Q_F : Number of failed subassembly parts after the final assembly, $Q_F = iCountFailed$ (Fig. 6)

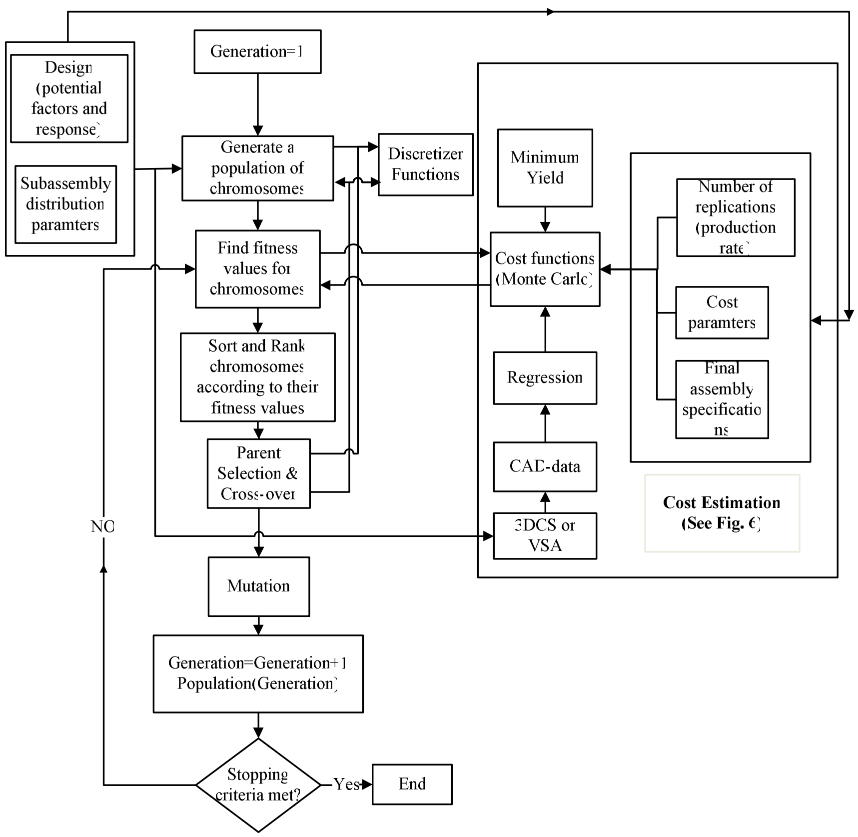

Fig. 5. New Product Inspection Planning Flowchart using a Genetic Algorithm (GA).

In order to impose the throughput constraint, then if the found yield was below the minimum yield then we give the total cost a value of *infinity* so the solution can be excluded later in the GA.

2.3. Is *frequency* an optimizer?

Up to this point, we have found that the frequency of inspecting a subassembly t has to be always 0 or 1 when the objective function is unconstrained and the subassembly QC is normally distributed; which means it is optimal to fully inspect a quality characteristic or not to measure it at all. In the following remark, we show that frequency can be a significant decision variable that minimizes the total cost when we consider yield as a constraint. We also verify this point in Section 3 by solving two numerical examples.

Remark

When all decision variables other than *frequency* are known, for a subassembly $t \in M$ that is normally distributed:

(1) When the effect of failure cost is more dominant than the other costs (inspection, rework and scrap), the optimal plan will be to fully inspect the subassembly t (i.e. $freq_t$=1).

(2) When the effect of failure cost is equal to the effect of other cost items, then fully, partially inspecting a subassembly or not inspecting it at all (inspecting it at any frequency between 0 and 1) will have a similar total cost.

(3) When the effect of inspection, rework and scrap cost is more dominant than the failure cost:

3.1. If the yield constraint $(yield \geq y_{\min})$ is not active, the optimal plan will be to not to inspect the subassembly t (i.e. $freq_t$=0).

3.2. If the yield constraint $(yield \geq y_{\min})$ is active, the optimal plan will be to partially or completely inspect the subassembly t ($freq_t$>0).

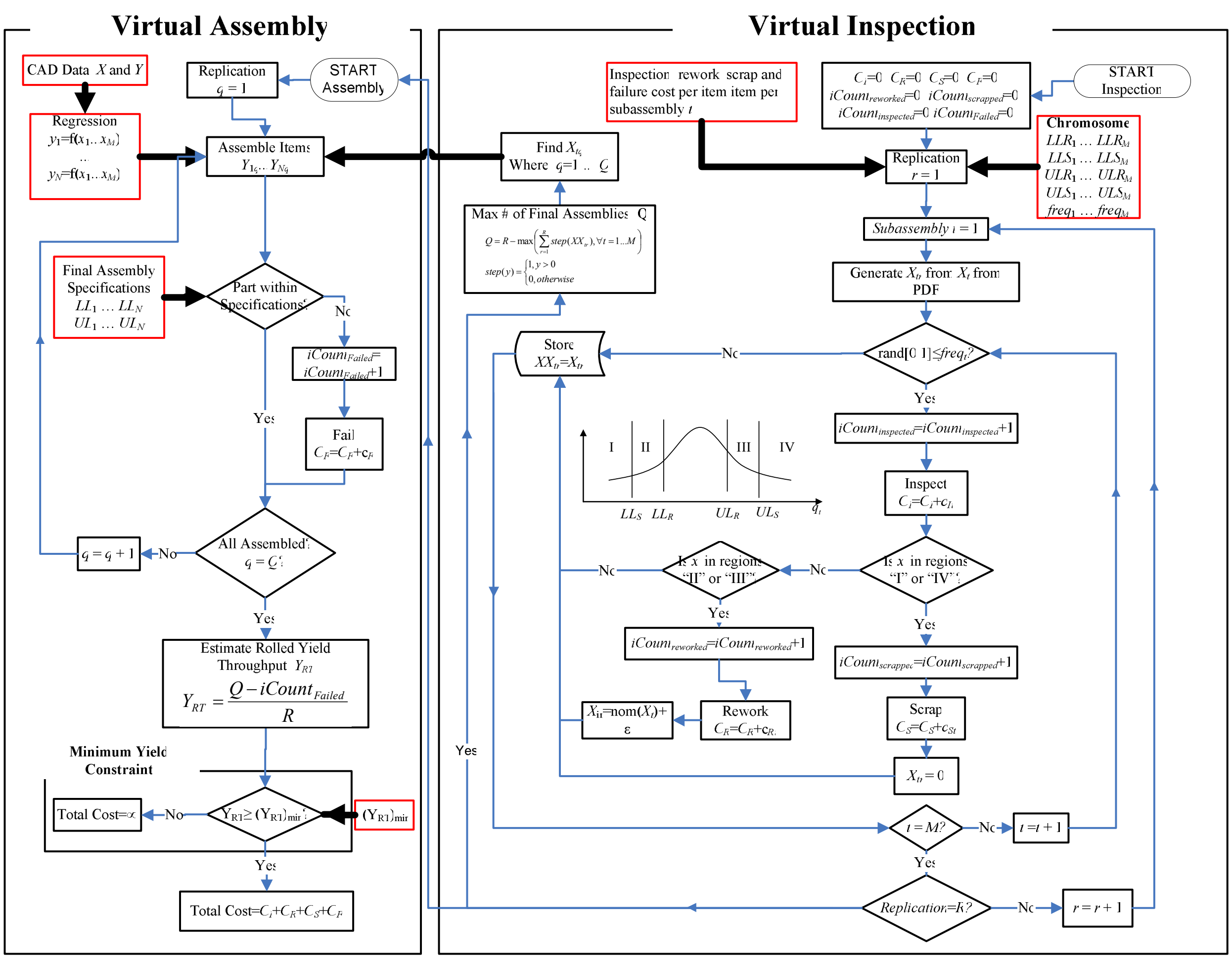

Fig. 6. Cost Determination using Monte Carlo Simulation.

3. ILLUSTRATIVE EXAMPLES

In this Section, we solve two numerical examples using our proposed approach. We want to develop inspection plans for a product that has four input quality characteristics (*X*'s, *M*=4) and three final quality characteristics (*Y*'s, *N*=3). The behavior of *Y*'s when *X*'s change can be mapped by simulating the product using 3DCS or VSA. After that, we can relate the *X*'s with the *Y*'s by using regression. The objective is to determine the optimal frequencies and action specifications (rework and scrap limits) that give the minimum total cost. We also consider that we are constrained with a minimum yield. We did not run the CAD simulation package because it is not capable in providing the *Y*'s that are associated with the generated *X*'s. Therefore, we generated the *X*'s to be all U(0,1). The Y's were generated according to the following functions: $y_1 = 100 - 50x_1 + x_2 - x_3 + 50x_4$, $y_2 = 35 - x_1 + x_2 - x_3 + 12x_4$, $y_3 = 2.5 - x_1 + x_2 + 2x_4$.

Example 1: Final Assembly Specifications: Lower Limits for the *N* QC's are respectively: 80,35 and 3.5 and the Upper Limits are respectively: 120,42, and 5. *R*=1000 (Initial number of items in a subassembly), Minimum Yield = 45%, Maximum Number of GA Replication = 1000, c_I = \$10 (cost of inspecting a single QC); c_R = \$30 (cost of reworking a single QC); c_S = \$30 (cost of scrapping a single QC); c_F = \$0 (cost of failing a single final assembly), Population Size: 6 Chromosomes, Mutation Rate: 5%.

In order to solve the problem using a Genetic Algorithm, we discretized the frequency. Tab. 2 shows the solutions when we consider different discrete frequency sets. We know from our experience with the data that the fourth quality characteristic has more significance than the other quality characteristics. This makes it intuitive to expect to inspect that subassembly more often than the others as you can see in the Table. The Table suggests that by refining the frequency set, we will get less total cost because the frequency is an optimizer in this case. Notice that the optimal frequency for the fourth QC is 40%, which did not change when we further refined the frequency set from set 2 to 3. This is also in agreement with the remark because the failure cost here is much less than the other cost parameters. The other optimal decision variables in that case are shown in Tab. 3. Figs. 7 show the solution (maximum, minimum, and average objective function for GA population) over the progress of GA replications for the three sets. As part of our approach, we imposed the minimum throughput constraint by giving a plan that leads to less than the specified throughput a cost of *infinity*. The discrete behavior of the cost functions in Figs. 7 is because we have those infinity costs that are not plotted.

Set	Frequency Resolution	$freq^*_1$	$freq^*_2$	$freq^*_3$	$freq^*_4$	Total Cost $	Exec. Time [min]
1	[0,1]	0	0	0	1	16,300	26
2	[0,0.1,0.2…0.9,1]	0	0	0	0.4	8,330	26
3	[0,0.05,0.1…0.95,1]	0	0	0	0.4	8,210	26

Tab. 2. Inspection plans at selected frequency sets, Example 1.

Resolution	Decision Variables	X_1	X_2	X_3	X_4	Total Cost $
1: [0,1]	*LLR**	3.5	1	3	3.5	16,300
	*LLS**	3.5	4.5	5	4.5	
	*ULR**	1	1	1	1	
	*ULS**	5	2	3	3.5	
	*freq**	0	0	0	1	
2: [0,0.1,0.2…0.9,1]	*LLR**	2.5	4	0.5	4.5	8,330
	*LLS**	4.5	5	3	4.5	
	*ULR**	5	1.5	1	0.5	
	*ULS**	5	3	5	3	
	*freq**	0	0	0	0.4	
3: [0,0.05,0.1…0.95,1]	*LLR**	3.5	0.5	2	2	8,120
	*LLS**	5	2	3.5	5	
	*ULR**	1.5	2.5	2	0.5	
	*ULS**	3.5	3	3.5	3	
	*freq**	0	0	0	0.4	

Tab. 3. Example 1 optimal decision variables.

Example 2: The data used here are the same as the previous example except for the cost failure cost parameter and the minimum throughput requirement. In this example, we assume the failure cost to be $1 per failed final assembly; i.e. c_F = $1. We also examine the effect of changing the required throughput on the total cost and the required frequency of inspection of the forth subassembly. The results of the simulation are shown in Figure 8 and 9. Notice that when we changed the minimum required throughput from 0 to 40%, the cost was found to be constant ($630) and the optimal frequency for inspection of the forth subassembly was found to be 0. This behavior changes when we increase the throughput from 45% to 70%, where we see an increase in both the frequency of inspection and the total cost. We found that imposing a throughput higher than 70% would never lead to a feasible solution. Therefore, we do not show any results after a throughput of 70%. Moreover, we can notice that the optimal frequency of inspection and total cost changed considerably by increasing the failure cost from $0 to $1. The optimal frequency increased from 40% to 45% and the optimal cost increased from $8,120 to $9,300.

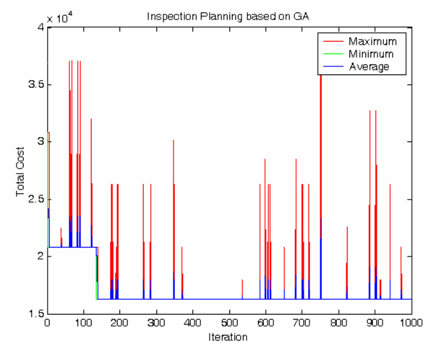

Fig. 7(a). Set 1 solution in Example 1 (C_F=$0)

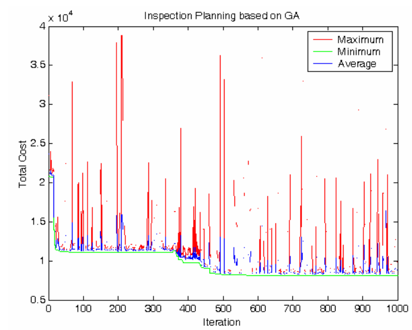

Fig. 7(b). Set 2 solution in Example 1 (C_F=$0)

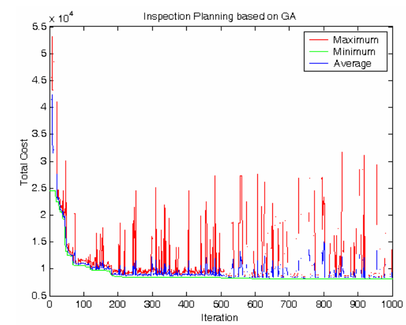

Fig. 7(c). Set 3 solution in Example 1 (C_F=$0)

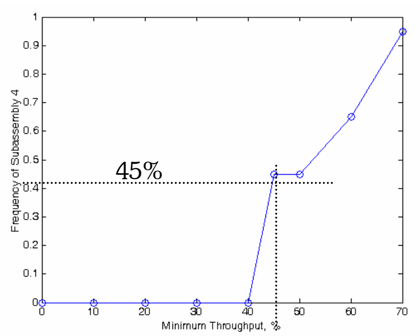

Fig. 8. Effect of minimum throughput on the optimal frequency of inspection for the forth subassembly (C_F=$1)

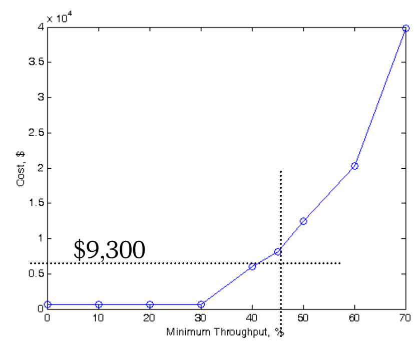

Fig. 9. Effect of minimum throughput on the final cost (C_F=$1)

4. SUMMARY AND CONCLUSIONS

In summary, we proposed an approach based on simulated data that we can obtain from CAD Variation Prediction packages and Genetic Algorithm. The approach is mainly designed to develop inspection plans for newly launched products which is quite challenging to design plans for. Moreover, we proposed a general and realistic action approach by introducing two tolerance limits with which keeping, reworking and scrapping an item are all possible. Frequency of inspection was also introduced as a possible decision variable that can further minimize the total cost. We can summarize our conclusions and findings as follows:

(1) We can notice that by refining the frequency set (Tab. 1), the optimal objective function is found to decrease because the optimal frequency values can be more accurately determined.

(2) We also have verified our remark by solving the first example. The failure cost was negligible compared to the other cost factors. This makes it possible to partially inspect a quality characteristic, rather than not inspecting it or fully inspecting it.

(3) We can see from Figures 8 and 9 that in Example 2, increasing the required minimum throughput increases the optimal frequency of inspection and the total cost.

5. FUTURE WORK AND RECOMMENDATIONS

Some worthy future research and recommendations are summarized as follows:

(1) Improve the inspection plans when the actual data are collected over the time. A control system that can be fed back with the differences in the data can be used to dynamically modify the inspection plan.

(2) One part of our methodology is to map the inputs (tolerances) to the outputs (measures) through the CAD software (e.g. 3DCS). We would recommend providing such data (*X*'s vs. *Y*'s) more directly as an output file when running a simulation using the software. Alternatively, we can integrate the proposed algorithm with the CAD software.

(3) Referring to Fig. 5; after finishing the inspection part of the simulation, the assembler will receive different numbers of subassemblies. Consequently, the maximum number of the final assemblies will be the minimum number of subassemblies (assuming that the final assembly takes only one subassembly). Assembling items arbitrarily may result in stacking-up variations in the subassemblies. Benefiting from the fact that we have already inspected a large portion of the subassemblies, we can dynamically assign items together in order to reduce the final variation which will significantly reduce the failure rate and will increase the yield.

(4) The effects of changing different GA parameters could be very interesting to see how they can impact the computational accuracy. Some parameters that can be changed to see the effect on the solution are: population size, mutation rate, and implementing elitism.

(5) We used linear regression to fit (map) the *X*'s with *Y*'s (CAD data). A test for goodness of fit will be good to do so to make sure that our model is statistically valid. On the other hand, we can run a variable selection procedure that can optimally select the most contributing variables. One possible approach to implement is stepwise regression.

6. ACKNOWLEDGEMENT

We would like to thank Dimensional Control Systems (DCS) for providing us with their software.

7. REFERENCES

[1] Azam, M., Pattipati, K. R. and Patterson-Hine, A., Optimal Sensor Allocation for Fault Detection and Isolation, IEEE International Conference on Systems, Man and Cybernetics, The Hague, Netherlands, October 2004.

[2] Binglin, Z., Tinghu, Y. and Ren, H., A Genetic Algorithm for Diagnosis Problem Solving, in Proc. Int. Conf. SMC, 1993, vol.2 , pp. 404-408.

[3] Chen, T. J. and Thornton, A. C. 1999, Quantitative Selection of Inspection Plan, Design Lemma and Methodology Conference, ASME Design Technical Conferences, Las Vegas, NV.

[4] Chen, S.-L., Chung, K.-J., Selection of the Optimal Precision Level and Target Value for a Production Process: The Lower-Specification-Limit Case, *IIE Transactions*, 28, pp. 979-985, 1996.

[5] Ding, Y., Kim, P., Ceglarek, D. and Jin, J., Optimal Sensor Distribution for Variation Diagnosis in Multi-station Assembly Processes, *IEEE Transactions on Robotics and Automation*, Vol. 19(4), pp. 543-556, 2003.

[6] Dimensional Control System (3DCS), Tutorial Manual, 2005.

[7] Greenstein, E. and Rabinowitz, G, Double-Stage Inspection for Screening Multi-Characteristic Items, *IIE Transactions*, 29, pp. 1057-1061, 1997.

[8] Khan, A., Ceglarek, D., Ni, J., Sensor Location Optimization for Fault Diagnosis in Multi-Fixture Assembly Systems, *Trans. of ASME, Journal of Manufacturing Science and Engineering*, Vol. 120, No. 4, pp. 781-792, 1998, also in 1996 ASME International Mechanical Engineering Congress and Exposition, MED-4, pp. 473-484, Atlanta, GA.

[9] Lindsay, G. F. and Bishop, A. B., Allocation of screening inspection effort—a dynamic programming approach, *Manage. Sci*., vol. 10, pp. 342–352, 1964.

[10] Peng, Y. J. and Reggia, A., A connectionist model for diagnostic problem solving, *IEEE Transactions on SMC*, vol. 19, issue 2, pp.285-298, 1989.

[11] Peng, Y and Reggia, J., A Connectionist Model for Diagnostic Problem Solving, *IEEE Transactions on Systems, MAN, and Cybernetics*, Vol. 19, No. 2, 1989, pp. 285-298.

[12] Shakeri, M., Pattipati, K., Raghavan, V. and Patterson-Hine, A. ,Optimal and Near Optimal Algorithms for Multiple Fault Diagnosis with Unreliable Tests, *IEEE Transactions on SMC*, part C, vol. 28, no. 3, pp. 431-440, August 1998.

[13] Tongco, E. and Meldrum, D., Optimal Sensor Placement of Large Flexible Space Structures, *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 4, 1996, pp. 961-963.