OPTIMUM SELECTION OF DISCRETE TOLERANCES

Woo-Jong Lee
Tony C. Woo

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report No. 87-34

December, 1987
OPTIMUM SELECTION OF DISCRETE TOLERANCES

WOO-JONG LEE
TONY C. WOO

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109–2117

December, 1987
ABSTRACT

Tolerance determination involves considerations from manufacturing, design, and assembly. Along with minimum cost and maximum functionality and interchangeability, the practice of tolerancing urges a process engineer to choose an appropriate manufacturing process as well. This situation is formalized by using a discrete model. For an optimum selection of tolerances from given tolerances of various manufacturing processes, minimization of manufacturing cost is achieved under the constraints of tolerance stack-ups.

A random variable and its standard deviation are assigned to a dimension and its tolerance. This probabilistic approach enables trade-off between performance and tolerance but it also suggests stochastic optimization. With the aid of a notion called the reliability index [8], tolerance selection is formulated as an integer programming problem. A branch and bound algorithm for ensuring an optimum selection is developed by exploiting the special structure of the constraints. To make the enumeration tree small, monotonic relations among the reliability index, cost, and tolerance are examined. The algorithm is tested with examples.

Keywords: Tolerancing, Combinatorial Optimization, Branch-and-Bound
1. INTRODUCTION

_Tolerance_ is the total amount by which a specific dimension in an engineering drawing is permitted to vary [1]. It is specified by the difference between the upper and the lower limits of a dimension. Once assigned, tolerance reflects the cost of manufacturing. Reduction of cost can be achieved by increasing tolerance but such an increase must be constrained by performance (functionality and interchangeability). A great deal of research [3-7,10,18,21,23] on tolerancing have been devoted to the resolution of this trade-off between cost and performance. However, they are concerned with tolerance analysis — that is, for a given set of tolerances, find the effect on cost or performance. It may be noted that tolerance is an input. If the particular combination of tolerances does not meet the criteria, some or all of the input have to be changed. And, the analysis procedure iterates. This potentially combinatorially complex situation of tolerance selection raises the concern for convergence. In an interactive mode [4-6,10,21], convergence depends on the “experience” of the user. However, because of the nature of input selection, there is no assurance of optimality. In a more resource abundant environment, Monte Carlo simulation [7,21] can be adopted. While, with simulation, the solution can be made arbitrarily close to the optimum, the amount of resource demanded can be arbitrarily large. The synthesis of tolerance, an alternative to manual data preparation and random number generation, remained as an intellectual and practical challenge. Recently, tolerance synthesis by non-linear optimization have been reported [11,14,15]. In them, tolerance is treated as a continuous variable such that the relationship between tolerance and cost takes the form of a continuous function as illustrated in Figure 1-(a).

<Insert Figure 1>

Tolerance can be thought of as a discrete variable. The eight classes of fit in the ANSI standard [1] is an example. For each class of fit, there is an associated
manufacturing process. While a process $p_i$ capable of producing a tight tolerance $t_1$ may be used to produce a less stringent tolerance $t_2$, its cost $c_1$, however, may be higher than $c_2$, that of another process $p_2$ capable of producing $t_2$. In this paper, a discrete tolerance model such as the one illustrated in Figure 1-(b) is adopted. That each process $p_i$ is the most efficient in tolerance and in cost is also understood. Because of the discrete nature of the tolerance-cost model, the problem of synthesizing tolerances becomes one of tolerance selection. This paper presents the development of a procedure that optimizes the selection.

Suppose $n$ tolerances are to be selected for the $n$ dimensions and each tolerance is to be selected from a set of $s_i$ manufacturing processes, $1 \leq i \leq n$. For each dimension $x_i$, the $s_i$ available manufacturing processes will be referred to as the PSMP$_i$ (Predetermined Set of Manufacturing Processes for the $i$-th dimension). Tolerance selection starts with the given PSMP$_i$ for $1 \leq i \leq n$. Only one manufacturing process is to be selected from each PSMP$_i$ and this condition will be referred to as the selection condition. In other words, the selection condition for dimension $x_i$ is described by using the 0-1 variables, $y_{ik}$, for $1 \leq k \leq s_i$ such that $y_{i1} + \ldots + y_{is_i} = 1$, where $y_{ik} = 1$ signifies that the $k$-th manufacturing process (tolerance) is chosen from PSMP$_i$.

As a criterion for selection, cost minimization is used. The cost minimization selection, however, is constrained by stack-up conditions for functionality and interchangeability. Consider Figure 2. Suppose the clearance between the two components in Figure 2-(a) is to be within limits $-0.1$ and $0.2$. Figure 2-(b) illustrates the area $R_S$ for satisfying the following stack-up conditions:

$$F_1(X) = x_1 - x_2 - 0.1 \geq 0$$

$$F_2(X) = -x_1 + x_2 + 0.2 \geq 0.$$  \hspace{1cm} (1)

Suppose the nominal dimensions for the dimensions $x_1$ and $x_2$ are 5.00 and 4.85, respectively. If two manufacturing processes are available for $x_1$ and four for $x_2$ as shown
in Figure 2−(c), the number of feasible solutions that satisfy the selection condition is eight and their relations with the stack-up conditions are shown in Figure 2−(d). Notice the overlap between \( R_S \) and the rectangle specified by the tolerances. For example, the rectangle specified by \( (t_{12}, t_{24}) \) is entirely covered by \( R_S \), whereas a smaller fraction of the rectangle specified by \( (t_{12}, t_{21}) \) is covered by \( R_S \). The fraction of coverage determines the level that the dimension within tolerance limits satisfies the stack-up conditions. Now, cost minimization leads to the selection of \( (t_{11}, t_{21}) \), but this selection gives the worst result − with the least fraction of coverage in the in-tolerance area. If a higher level of satisfaction of the stack-up conditions is deemed desirable, some of the manufacturing processes would have to be replaced by those with having smaller tolerances and the feasibility test for the stack-up conditions is iterated.

<Insert Figure 2>

Tolerance selection is formulated here as a combinatorial optimization problem by treating cost minimization as the objective function and stack-up conditions and selection conditions as the constraints. The overall procedure for tolerance selection is outlined in Figure 3.

<Insert Figure 3>

Concept in probability is used. Since tolerance implies randomness, a random variable and its standard deviation are respectively associated with a dimension and its tolerance. The probabilistic approach enables the partial satisfaction of the stack-up conditions. By permitting a fraction of the assemblies to be "out-of-spec," a reduction in cost may be achieved by selectively increasing some of the tolerances. This approach is considered to be advantageous over the deterministic approach [3,5,14] for the 100% "in-spec" case only. Suppose an inequality \( F(X) \geq 0 \) represents a certain stack-up condition, where \( X \) is a random vector composed of dimensions. The level of satisfaction of the stack-up condition can be described by the following multiple integral:
\[ \int_{F(X) \geq 0} f(X) \, dX \]  

(2)

where \( f(X) \) is the multivariate probability density function (p.d.f.) of \( X \). \( F(X) \), the function for stack-up condition, is nonlinear if non-rectilinear shapes and/or dimensions are in an engineering drawing. See, for example, \( F_3(X) \) and \( F_4(X) \) in Figure 4. (Those two conditions specify that the angular difference between \( \theta_1 \) and \( \theta_2 \) is to be within \( \pm (\pi/180) \) radians.)\(^1\) Now, suppose the stack-up conditions are linear. Tolerance selection then can be formulated as a pure 0–1 linear integer programming problem because of the following property: under the assumption of independence, the variance of a linear function can be expressed as the linear sum of the variances of the constituting dimensions. Hence, binary tree enumeration such as Balas’ 0–1 algorithm [2] can be used for tolerance selection [9,16,23]. Unfortunately, such is not the case in general as engineering drawings are not always rectilinear. Furthermore, there is no rule for expressing the variance of a nonlinear function, such as \( F_3(X) \), in terms of variances of its constituting dimensions.

<Insert Figure 4>

To compute the multiple integral of equation (2), simulation or numerical methods may be considered. But, since it is an inner loop in the overall algorithm for tolerance selection (as illustrated in Figure 3), a fast approximation is sought. In this paper, the nonlinear function \( F(X) \) is approximated by a hyperplane. To retain accuracy, the area that is probabilistically the densest is preserved by computing the distance from the hyperplane to the origin of standardized coordinate system. A notion called the “reliability

\[ \begin{align*}
- (\pi/180) & \leq \theta_1 - \theta_2 \leq \pi/180 \\
\tan(- (\pi/180)) & \leq \tan(\theta_1 - \theta_2) \leq \tan(\pi/180) \\
\tan \theta_1 - \tan \theta_2 & \\
- \tan (\pi/180) & \leq \frac{\tan(\theta_1 - \theta_2)}{1 + \tan \theta_1 \tan \theta_2} \leq \tan(\pi/180) \\
- \tan (\pi/180) & \leq \frac{(x_8 - x_7)(x_2 - x_3) - (x_6 - x_5)(x_{10} - x_9)}{(x_{10} - x_9)(x_2 - x_3) + (x_8 - x_7)(x_6 - x_5)} \leq \tan(\pi/180)
\end{align*} \]

\(^1\)The two conditions \( F_3(X) \geq 0 \) and \( F_4(X) \geq 0 \) are derived by taking the following steps:
index” introduced by Hasofer and Lind [8] is used for computing this distance.

With the aid of the reliability index for computing (2), the tolerance selection problem can be formulated as the following nonlinear 0–1 integer programming problem:

\[
\text{Minimize } \sum_{i=1}^{n} \sum_{k=1}^{s_i} c_{ik} y_{ik} \quad (3)
\]

subject to

\[
\beta_j \geq q_j, \quad \text{for } j=1, \ldots, m \quad (3-1)
\]

\[
\sum_{k=1}^{s_i} y_{ik} = 1, \quad \text{for } i=1, \ldots, n \quad (3-2)
\]

where \( y_{ik} = 0 \) or \( 1 \) for \( k=1, \ldots, s_i \) and \( i=1, \ldots, n \)

where \( c_{ik} \) is the cost incurred when using the \( k \)-th manufacturing process for dimension \( x_i \).

The constraint (3–1) states that the \( j \)-th stack-up condition, whose satisfaction level is approximated through the reliability index \( \beta_j \), should be satisfied with at least the given probability \( q_j \). The constraint (3–2) describes the selection rule such that only one manufacturing process is selected from the given PSMP\(_i\). The notation is summarized in Table 1.

<Insert Table 1>

An algorithm for (3) that comes to mind first is the binary tree enumeration since the formulation takes the form of a pure 0–1 integer programming problem. The number of solutions, however, is of \( O(2^N) \), where \( N \) is the total number of manufacturing processes. If each PSMP\(_i\) for \( 1 \leq i \leq n \) has \( s_i \) elements (manufacturing processes), then \( N = \sum_{i=1}^{n} s_i \). For example, if two and four different manufacturing processes are considered for the dimensions \( x_1 \) and \( x_2 \), then \( s_1 = 2, s_2 = 4 \) and \( N = 6 \). Tolerances \( (t_{11}, t_{12}) \) and \( (t_{21}, t_{22}, t_{23}, t_{24}) \) are associated with the dimensions \( x_1 \) and \( x_2 \), respectively. The number of solutions for the binary tree enumeration becomes \( O(2^6) \) though only eight solutions, i.e., \( (t_{11}, t_{21}), (t_{11}, t_{22}), (t_{11}, t_{23}), (t_{11}, t_{24}), (t_{12}, t_{21}), (t_{12}, t_{22}), (t_{12}, t_{23}), (t_{12}, t_{24}) \) are actually feasible to constraint (3–2). A branch-and-bound algorithm is developed in this paper by
exploiting the special structure of the constraint (3–2), the so called “multiple choice
constraint.”

It is noted that the computation time for the branch-and-bound algorithm can be
\( O(\prod_{i=1}^{n}s_i) \). Hence, making the enumeration tree smaller is essential. The monotonic
relations among cost, performance and tolerance — as tolerance decreases, cost and
performance increase — enable efficient pruning and fathoming operations in the
algorithm. Monotonicity is also used to simplify the lower bound computation and the
feasibility test in each iteration. Furthermore, to facilitate fathoming by cost, a heuristic
procedure requiring \( O(N) \) feasibility tests is given to provide an initial incumbent value.
Finally, the practical run time of the algorithm (about 3 CPU seconds) is illustrated by one
of examples (for 12 dimension variables, each with up to 5 manufacturing processes).

2. FORMULATION

A random vector \( \mathbf{X} = (x_1, \ldots, x_n)^t \) is used to represent the \( n \) dimensions in an
engineering drawing to capture the randomness of manufacturing processes. Mean and
standard deviation vectors for \( \mathbf{X} \), denoted by \( \bar{\mathbf{X}} \) and \( \Sigma \), are associated with nominal
dimensions and tolerances, respectively. Dimension variables \( x_i \)'s are assumed to be
independent with each other and to follow the normal distribution [13].

Stack-up conditions are assumed to be given and are represented by the functional
form \( F_j(\mathbf{X}) \) for \( 1 \leq j \leq m \), that are inequalities. Each inequality divides the dimensional
space into a safe region \( R_{S_j} = \{ \mathbf{X} \mid F_j(\mathbf{X}) \geq 0 \} \) and a fail region \( R_{F_j} = \{ \mathbf{X} \mid F_j(\mathbf{X}) < 0 \} \). The
safe region \( R_S \) of satisfying all the stack-up conditions is the intersection of \( R_{S_j} \), \( 1 \leq j \leq m \).
That is, \( R_S = \bigcap_{j=1}^{m} R_{S_j} \). Figure 2–(b) illustrates the safe region specified by the two
stack-up conditions of (1). The probability of satisfying a stack-up condition \( F_j(\mathbf{X}) \geq 0 \) is
then described as:
\[
\int_{F_j(X) \geq 0} \phi(\overline{X}; V) \, dX
\]

The objective of tolerance selection is to select tolerances (standard deviations) by minimizing the manufacturing cost. The selected tolerances are constrained to satisfy the stack-up condition \( F_j(X) \geq 0 \) at a desired probability level \( 1 - \delta_j \). These constraints are referred to as reliability constraints. Also, by the nature of tolerance selection, only one tolerance should be selected from a given PSMP. These selection constraints are referred to as selection constraints. Then, the problem can be formulated as:

Min \[ \sum_{i=1}^{n} \sum_{k=1}^{s_i} c_{ik}y_{ik} \]

subject to

\[ \int_{F_j(X) \geq 0} \phi(\overline{X}; V) \, dX \geq 1 - \delta_j, \quad \text{for} \ j = 1, \ldots, m \quad \text{(reliability constraints)} \]

\[ \sum_{k=1}^{s_i} y_{ik} = 1, \quad \text{for} \ i = 1, \ldots, n \quad \text{(selection constraints)} \]

where \( y_{ik} = 0 \) or \( 1 \) for \( k = 1, \ldots, s_i \) and \( i = 1, \ldots, n \)

As an example, consider Figure 2 for which \( s_1 = 2 \) and \( s_2 = 4 \). Tolerance selection is then formulated as follows:

Min \[ c_{11}y_{11} + c_{12}y_{12} + c_{21}y_{21} + c_{22}y_{22} + c_{23}y_{23} + c_{24}y_{24} \]

subject to

\[ \int_{F_1(X) = x_1 - x_2 - 0.1 \geq 0} \phi(\overline{X}; V) \, dX \geq 1 - \delta_1 \]

\[ \int_{F_2(X) = -x_1 + x_2 + 0.2 \geq 0} \phi(\overline{X}; V) \, dX \geq 1 - \delta_2 \]

\[ y_{11} + y_{12} = 1 \]

\[ y_{21} + y_{22} + y_{23} + y_{24} = 1 \]

where \( y_{11}, y_{12}, y_{21}, y_{22}, y_{23}, y_{24} \) are 0-1 variables and
\( \delta_1, \delta_2 \) are permissible dissatisfaction levels for \( F_1(X) \geq 0, F_2(X) \geq 0 \).

When solving problem (5), two difficulties arise. First, the reliability constraints of (5) involve a multiple integral bounded by nonlinear functions. Evaluating it numerically is time consuming. Second, the selection constraints of (5) involve a large number of solutions, i.e., \( \prod_{i=1}^{n} s_i \). The first problem is dealt with in the remainder of this section while the second problem is treated in the next section.

Observe that the integral of the reliability constraints of (5) is to be taken in the safe region \( R_{S_j} \). Since \( R_{S_j} \) is to be integrated under the multivariate normal p.d.f., the accuracy of an approximation for (4) depends on the preservation of the probabilistically densest area. Suppose the dimension variables are standardized (to be described in Appendix A) such that they, denoted by \( z_i \)'s, follow the standard normal distribution. Then, the vicinity of the origin in the standardized coordinate system (standard system, for short) is probabilistically the densest and the density decreases exponentially from the origin.

Linearization is performed by taking the tangent hyperplane at a point on the stack-up function having the minimum distance from the origin in the standard system. This minimum distance is called as reliability index [8] and is denoted by \( \beta \). (The solution scheme for computing \( \beta \) is also explained in Appendix A). Then, the probability of covering one side of the tangent hyperplane, \( P(R_{S_j}) \), can be computed based on the univariate normal distribution, because rotational symmetry is preserved in the standard system as illustrated in Figure 5. This is summarized in Lemma 1.

\[ \text{<Insert Figure 5>}
\]

**Lemma 1.** If the stack-up function \( F_j(X) \) has a reliability index \( \beta_j \), then

\[ P(R_{S_j}) = \Phi(\beta_j). \quad (6) \]

The accuracy of (6) depends on the curvature of the stack-up function. As long as the
radius of the curvature at the expansion point is large compared to the reliability index as in most practical cases, this approximation is quite accurate [12]. For a linear stack-up function, \( P(R_{S_j}) = \Phi(\beta_j) \).

Based on Lemma 1, formulation (5) can be converted into the following deterministic optimization problem:

\[
\text{Min } \sum_{i=1}^{n} \sum_{k=1}^{s_i} c_{ik} y_{ik}
\]

subject to

\[
\beta_j \geq \Phi^{-1}(1-\delta_j), \quad \text{for } j=1,\ldots,m \quad (\text{reliability constraints})
\]

\[
\sum_{k=1}^{s_i} y_{ik} = 1, \quad \text{for } i=1,\ldots,n \quad (\text{selection constraints})
\]

where \( y_{ik} = 0 \) or 1 for \( k=1,\ldots,s_i \) and \( i=1,\ldots,n \)

The value \( \Phi^{-1}(1-\delta_j) \) in the reliability constraints can be looked up in the standard normal distribution table.

3. ALGORITHM

A branch-and-bound algorithm for efficient tree enumeration is developed in this section. To make the enumeration tree small, monotonic relations among the reliability index, cost and tolerance are exploited.

3-1. Monotonicity

A function \( g(X) \) is said to be monotone nonincreasing in every element \( x_i \)'s if \( X_1 \geq X_2 \) results in \( g(X_1) \leq g(X_2) \).

The monotonicity between tolerance and cost is generally understood as: the more tolerance the less cost. That is, \( \partial C_i(\sigma_i) / \partial \sigma_i < 0 \) where \( C_i(\sigma_i) \) is the cost function in terms of the standard deviation (i.e., tolerance). Indeed, most tolerance-cost models [19,20,22]
exhibit this relationship. Without loss of generality, this strict nonincreasing monotonicity is described by taking the following ordering:

\[ c_{i1} < c_{i2} < c_{i3} \ldots < c_{is_i} \quad \text{for } i = 1, \ldots, n \]
\[ \sigma_{i1} > \sigma_{i2} > \sigma_{i3} \ldots > \sigma_{is_i} \quad \text{for } i = 1, \ldots, n \]

The monotonicity between tolerance and the reliability index is less obvious. But it can be established based on the following observation: If a certain standard deviation is reduced, the distance from the origin of the standard system to a point on the stack-up function becomes greater than or equal to the distance before the reduction. This observation can be generalized as follows:

**Lemma 2.** Suppose \( d_{j,0} \) is the distance in the standard system from a point \( Z_0 \) on the \( j \)-th stack-up function to the origin. Then, \( \partial d_{j,0} / \partial \sigma_1 \leq 0 \) for \( 1 \leq i \leq n \) and \( 1 \leq j \leq m \).

**Proof.** The proof is given in Appendix B. \( \bullet \)

From this lemma, it follows immediately that the minimum distance \( \beta_j \) also decreases with the decrease of the standard deviations. That is,

\[ \partial \beta_j / \partial \sigma_1 \leq 0 \quad (7) \]

Inequality (7) reconfirms the trade-off between tolerance and performance: tolerance represented by standard deviation has an inverse relationship with performance implied by the reliability index.

3-2. Branch-and-Bound

The enumeration tree used in this paper is depicted in Figure 6. To denote a partial solution, a \( 1 \times n \) row vector \( Y \) is used. For example, \( Y = (2, s_2, 0, 0, \ldots, 0) \) means that for dimension \( x_1 \) the second manufacturing process is selected from PSMP\(_1\) and for dimension \( x_2 \) the \( s_2 \)-th manufacturing process from PSMP\(_2\). The value 0 in \( Y \) shows that the decision for the corresponding dimension has not been made yet. Notice that, by using the vector \( Y \), the selection constraints can be handled implicitly. That is, if it is decided that a
dimension is to be produced by a certain manufacturing process, the remaining manufacturing processes in the same selection constraint (i.e., the same PSMP) are set to 0.

<Insert Figure 6>

The following branch-and-bound procedure TOL-B&B provides a way of tracing the enumeration tree. Here, the terminology of branch-and-bound [17] is assumed. A flowchart is given in Figure 7.

<Insert Figure 7>

**Procedure TOL-B&B**

Begin

Step 1. Initialize the candidate list to consist of partial solution vectors 
\((1,0,\ldots,0),(s_1,0,\ldots,0)\), and set the incumbent value \(C^*\) to an arbitrary large number. Set the incumbent solution, denoted by a \(1 \times n\) row vector \(C\), to 
\((0,0,\ldots,0)\), where \((0,0,\ldots,0)\) signifies that no decision has been made for any dimension.

Step 2. Go to Step 3 if the candidate list is not empty. Otherwise, stop and the current incumbent solution is then optimum except the case in which the incumbent value is an initial value set in Step 1. (The exceptional case happens only when there is no feasible solution.)

Step 3. Select a candidate problem from the candidate list. The selection rule is LIFO (last-in-first-out). The candidate list is maintained in a stack.

Step 4. Calculate the lower bound \(C_L\) for the candidate problem. Suppose the current partial solution for the candidate problem is \(Y=(i_1,i_2,\ldots,i_r,0,\ldots,0)\).

Then, the lower bound \(C_L\) for \(Y\) is 
\[c_{i_1} + \cdots + c_{i_r} + \sum_{j=r+1}^{n} c_{j}.\]

Step 5. (Fathoming Criterion 1) If \(C_L \geq C^*\), go to Step 2.

Step 6. (Fathoming Criterion 2) Modify \(Y=(i_1,i_2,\ldots,i_r,0,0,\ldots,0)\) with \(Y^*=(i_1,i_2,\ldots,i_r,s_{r+1},\ldots,s_n)\). Check if this \(Y^*\) is feasible. If it is not feasible, go to Step 9.

Step 7. (Fathoming Criterion 3) If \(r=n\) (terminal node), go to Step 8. Otherwise, go to Step 10.

Step 8. Update the incumbent solution \(C\) with \(Y\) and the incumbent value \(C^*\) with \(\sum_{j=1}^{n} c_{j}\). Go to Step 2.
Step 9. Prune the left siblings of the current partial solution from the candidate list. For example, if \( i_r = 3 \), the candidates \((i_1, i_2, \ldots, i_{r-1}, 1, 0, \ldots, 0)\) and \((i_1, i_2, \ldots, i_{r-1}, 2, 0, \ldots, 0)\) can be deleted. Go to Step 2.

Step 10. Separate the candidate problem and add its offsprings to the candidate list. The offsprings of \( Y \) are \((i_1, i_2, \ldots, i_r, 1, 0, \ldots, 0)\), \((i_1, i_2, \ldots, i_r, 2, 0, \ldots, 0)\), \ldots, \((i_1, i_2, \ldots, i_r, s_{r+1}, 0, \ldots, 0)\). Go to Step 2.

\end

Steps 4 and 6 are based on the monotonic relations among the reliability index, cost and tolerance. In Step 4, the cost ordering and its strict nonincreasing monotonicity is used to compute the lower bound such that \( C_L \) for the partial solution \( Y = (i_1, \ldots, i_r, 0, \ldots, 0) \) is equivalent to the cost for the selection \((i_1, \ldots, i_r, 1, \ldots, 1)\). Similarly, in Step 6, to check for the feasibility of \( Y \), \( Y^* = (i_1, \ldots, i_r, s_{r+1}, \ldots, s_n) \) is tested based on the nonincreasing monotonicity of the reliability index of (7).

Now, to save computation time, the following observations are made.

**Observation 1.** If the current node is the right-most sibling, its partial solution for feasibility test is the same as that of its parent.

From this observation, the feasibility test in Step 6 can be omitted if the candidate problem is the right-most offspring. In a similar manner, the computation for the lower bound in Step 4 can be expedited with the following observation.

**Observation 2.** If the current node is the left-most sibling, its lower bound in cost is the same as that of its parent.

Hence, the computation in Step 4 can be omitted also if the candidate problem is the left-most offspring. More generally, this computation can be done in the following way:

**Observation 3.** Provided that the partial solution is \((i_1, \ldots, i_r, 0, \ldots, 0)\), the \( j \)-th left offspring of the current partial solution has the following lower bound in cost:
\[ C_L \text{ of the } j\text{-th offspring} = C_L \text{ of the parent} + c_{r+1,j} - c_{r+1,1}. \]

Note that Observation 2 is a special case of Observation 3, since \( c_{r+1,j} = c_{r+1,1} \).

The pruning operation of Step 9, which contributes to making the enumeration tree smaller, is based on the following observation:

**Observation 4.** For siblings having the same parent, the lower bound in cost and the feasibility of the sibling to the right are greater than those of the sibling to the left.

Lastly, to facilitate fathoming by cost in Step 5, the following heuristic procedure TOL-INITIAL is used for generating a reasonable initial incumbent value \( C^* \).

**Procedure TOL-INITIAL**

Begin

Step 1. Start with \( Y^i=(s_1,\ldots,s_n) \) where \( i=0 \). Set the integer variables “stop-check” and “current-flag” to 1. The variable stop-check is used to check the stop of the circular search of feasible solution. And, the current-flag indicates the dimension being considered.

Step 2. If the manufacturing process index for the dimension indicated by the current-flag is the first one, go to Step 6.

Step 3. Change the manufacturing process index of the dimension indicated by the current-flag to the manufacturing process index one lower than the current one. Let this changed solution be \( Y^{i+1} \).

Step 4. Test if \( Y^{i+1} \) is feasible. If feasible, stop-check:=0 and go to Step 7. Otherwise, go to Step 5.

Step 5. If \( Y^{i+1}=(1,\ldots,1,1) \), then stop since there is no feasible solution. Otherwise, \( Y^{i+1}:=Y^i \).

Step 6. stop-check:= stop-check + 1.


Step 8. If stop-check = n+1, then \( C:=Y^{i+1} \) and stop. Otherwise, go to Step 2.

end;
Procedure TOL-INITIAL starts with the manufacturing process having the highest cost. It reduces the cost through circular search until cost reduction is impossible because of infeasibility. TOL-INITIAL involves O(N) feasibility tests in the worst case. In the example to follow, it is assumed that in Step 1 of TOL-B&B, the initial solution C is now obtained from TOL-INITIAL. Since TOL-INITIAL checks for the existence of feasible solutions in Step 5, the use of TOL-INITIAL simplifies Step 2 of TOL-B&B as follows: if candidate list is empty, stop and the incumbent solution is optimum.

4. EXAMPLES

Algorithm TOL-B&B has been programmed in PASCAL and run on the University of Michigan IBM 3090-400/VM under the MTS operating system. Two examples are given. Example 1, based on the simple assembly of Figure 2-(a) with two dimension variables and two linear stack-up conditions, traces the flow of the algorithm. In particular, effective pruning by TOL-B&B is illustrated. Example 2 is based on the assembly in Figure 4, with twelve dimension variables, each with three to five selections for a total of 1,574,640 possible solutions.

Example 1.

Consider the assembly in Figure 2-(a) in which stack-up conditions are specified by the two inequalities of (1). These two conditions define the safe region $R_S$ in the relevant positive dimension space as shown in Figure 2-(b). Suppose each stack-up condition of (1) is to be satisfied with a $97.468\% \ (= \sqrt{0.95})$ confidence level. Then, $\delta_1 = \delta_2 = 1 - \sqrt{0.95}$. Here, two and four different manufacturing processes are respectively considered for the dimensions $x_1$ and $x_2$, i.e., $s_1 = 2$, $s_2 = 4$, and $N = 6$. The corresponding tolerance-cost model is illustrated in Figure 2-(c). The discrete tolerances $t_{ik}$ as shown in Figure 2-(c) are assumed to signify $99.73\%$ two-sided confidence intervals. That is, $t_{ik}$ is the same as $\pm 3\sigma_{ik}$ such that $\sigma_{ik} = t_{ik}/6$. Note that these standard deviations $\sigma_{ik}$ are used to compute
\( \beta_j \)'s in the reliability constraints of (3-1). The dimensions \( x_1 \) and \( x_2 \) are assumed to have the nominal dimension values 5.00 and 4.85 respectively, i.e., \((\bar{x}_1, \bar{x}_2) = (5.00, 4.85)\). The relationship between tolerances and the nominal dimensions is illustrated in Figure 2-(d).

To obtain an initial solution for procedure TOL-B&B, procedure TOL-INITIAL is used. It finds an initial solution (2,3): the second manufacturing process with tolerance \( t_{12} \) is selected for dimension \( x_1 \) and the third manufacturing process with tolerance \( t_{23} \) is selected for dimension \( x_2 \). This initial solution incurring a cost of 47.0 is obtained in four iterations of TOL-INITIAL. These four iterations start respectively with solutions (2,4), (2,4), (2,3) and (2,3), and perform the feasibility test in Step 4 with solutions (1,4), (2,3), (1,3) and (2,2). The solution (2,3) and its cost of 47.0 are used as the initial incumbent solution and its incumbent value in Step 1 of procedure TOL-B&B.

TOL-B&B traverses the enumeration tree as illustrated in Figure 8. In that figure, the nodes indicated by solid circles are the nodes traversed by TOL-B&B while the dashed nodes are the non-traversed ones. The number in a traversed node represent the traversal sequence. This numbering shows that TOL-B&B is based on the depth-first search. When the procedure first visits node 1, the lower bound in cost for the partial solution (2,0) becomes 32.0. Since this lower bound is less than the incumbent value 47.0, the feasibility test is done with solution (2,4). It turns out to be feasible so that the next traversal node is node 2 having the solution (2,4). Nodes 2 and 3 are fathomed since their lower bounds in cost are not less than the incumbent value 47.0. Nodes 4 and 5 are fathomed because they are infeasible. It is noted here that TOL-B&B obtains the optimum solution with five iterations whereas the possible solutions for binary tree enumeration is 64 \((=2^6)\).

Figure 8 also shows how the monotonicity among tolerance, cost, and the reliability index works for partial enumeration. The pruning of node A is due to the monotonicity between tolerance and the reliability index: the reliability indices of the left-siblings are less than or equal to those of the current node. That is, node A is infeasible because of the infeasibility of node 4. In the case of node B, the solution for feasibility test is the same as
the solution of its parent, i.e., node 5, by Observation 1. Since node 5 turns out to be
infeasible, node B is also infeasible and this infeasibility makes possible the pruning of
node C as in the case of node A. In this example, the initial incumbent solution obtained
by TOL-INITIAL is the optimum.

<Insert Figure 8>

Example 2.

The assembly is shown in Figure 4. The first and second stack-up functions are for
specifying the vertical and the horizontal clearances between two parts, respectively. The
third and fourth stack-up conditions show the angular difference condition between angles
$\theta_1$ and $\theta_2$ as mentioned in the footnote in Section 1. The last two functions show that the
size difference of the horizontal lengths of two parts should be within 0.01. $F_3(X)$ and
$F_4(X)$ are nonlinear and the others are linear.

The nominal dimensions for the 12 dimensions are given as $X = (50.0, 40.00125, 20.05, 9.9985, 9.9985, 30.00, 10.00, 30.00, 10.05, 30.00, 40.00, 50.00)^T$. The other input
data for cost and standard deviations are listed in Table 2.

<Insert Table 2>

Suppose each of the six stack-up conditions is to be satisfied at $99.1488\%$
$= (0.95)^{1/6}$ level, i.e., $\delta_1 = \cdots = \delta_4 = 1 - (0.95)^{1/6}$. The output produced by TOL-B&B and its
statistics are given in Table 3. The initial incumbent solution by TOL-INITIAL is $2, 2, 2, 2, 2, 1, 1, 3, 4, 3$ and its corresponding incumbent value is 275. Starting with this
initial incumbent solution, the optimum solution of $(3, 2, 1, 3, 2, 2, 2, 1, 1, 2, 2, 1)$ is obtained in
2554 iterations which is 0.16% of the total number of possible selections 1,574,640. This
result (in 3.076 CPU seconds) is tabulated and compared to the one obtained without using
TOL-INITIAL (in 3.884 CPU seconds) in Table 3.

<Insert Table 3>

Under optimum selection, the reliability indices of the given stack-up functions $F_j(X)$,
for $1 \leq j \leq 6$ are shown in the second column of Table 4. This column shows that the reliability index obtained for each stack-up function is almost the same as the reliability index 2.38617 for the desired satisfaction level 99.1488%. By the definition of the reliability index, it corresponds to the minimum distance from each stack-up function to the origin in the standard system. This distance is used here to measure the satisfaction level covering the half-space safe region obtained by the hyperplane approximation of a stack-up function. By Lemma 1, these satisfaction levels are approximately equal to $\Phi(\beta_j)$ for $1 \leq j \leq 6$. Those are listed in the third column of Table 4 and their differences from the desired level of 99.1488% are shown in the fourth column of Table 4. The reason that the obtained reliability indices are not exactly the same as the desired reliability index is due to the discrete nature of tolerance selection. By Lemma 2 and inequality (7), the reliability index higher than the desired level needs a smaller tolerance because of the negative monotonicity of the reliability index to tolerance. The smaller tolerance, however, in turn incurs a high cost because of the negative monotonicity of cost to tolerance. Hence, the selection is optimized so as to reduce the gap between both sides of the reliability constraint of (3-1) to as small a value as possible.

<Insert Table 4>

5. SUMMARY

It is recognized that the arena of CAD, CAM and robotics is populated with “islands of technology.” While tolerancing may be construed as another island, it has the integrative element that bridge design, manufacturing, assembly and testing. The intent of this work is to fold the traditionally “down-stream” considerations such as functionality, interchangeability and cost to the design stage. Formulated probabilistically, the problem of automatic tolerancing permits trade-offs between these considerations.

The contribution of this work is in practice. As born by the practical considerations such as the association of tolerance to discrete manufacturing processes and the
observations on monotonicity hence the fathoming criteria, the branch-and-bound algorithm as implemented is suitable for near real-time tolerancing. In addition, the algorithm guarantees an optimum selection from given PSMPs under linear as well as nonlinear stack-up conditions.
REFERENCES


(1982).


APPENDICES

Appendix A. Reliability Index

The Euclidean distance of a point $Z$ in the standard system to the origin is $(Z'Z)^{1/2}$. The standardized vector $Z$ is obtained by the transformation $z_i = \frac{x_i - \bar{x}_i}{\sigma_i}$ for $1 \leq i \leq n$. Hence, $(Z'Z)^{1/2} = \left\{ \frac{(X - \bar{X})'V^{-1}(X - \bar{X})}{\sigma_i} \right\}^{1/2}$. The reliability index can be obtained by solving the following problem:

$$\text{Min } \beta^2 = (X - \bar{X})'V^{-1}(X - \bar{X})$$

subject to

$$F(X) \leq 0$$

(8)

Here, the objective function is expressed as $\beta^2$ instead of $\beta$ since the positive definiteness of the covariance matrix $V$ always guarantees the same solution.

As a solution scheme for (8), the iterative method based on the Newton-Raphson method [12] is as follows:

$$X^{(k+1)} = \bar{X} + V \frac{\nabla F(X^{(k)}) (X^{(k)} - \bar{X})' \nabla F(X^{(k)}) - F(X^{(k)})}{\nabla F(X^{(k)})' V \nabla F(X^{(k)})}$$

where $X^{(k)}$ denotes the solution after the $k$-th iteration and $\nabla F(X)$ is the $n \times 1$ gradient vector of $F(X)$ at $X$. It is noted that this iterative method starts from the nominal dimension point $\bar{X}$. 
Appendix B. Monotonicity of the Reliability Index

As shown in Appendix A, the Euclidean distance $d_{j,0}$ from the origin of the standard system to a point $Z_0$ can be expressed in terms of the original vector $X$ by

$$\left( (X_0 - \bar{X})^T V^{-1} (X_0 - \bar{X}) \right)^{1/2},$$

where $X_0$ is for the point $Z_0$ before standardization. Note that this distance expression corresponds to the equation of an ellipsoid with center $\bar{X}$, the semi-axes of which are expressed as the product of $d_{j,0}$ and the standard deviations. Then,

$$d_{j,0}^2 = \sum_{i=1}^n \frac{(x_{0i} - \bar{x}_i)^2}{\sigma_i^2}$$

As a standard deviation $\sigma_i$ decreases, the distance $d_{j,0}$ increases.
Table 1. Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i = 1, \ldots, n)</td>
<td>dimension index</td>
</tr>
<tr>
<td>(j = 1, \ldots, m)</td>
<td>stack-up function index</td>
</tr>
<tr>
<td>(k = 1, \ldots, s_i)</td>
<td>manufacturing process index</td>
</tr>
<tr>
<td>(C)</td>
<td>incumbent solution in TOL-B&amp;B</td>
</tr>
<tr>
<td>(C^*)</td>
<td>incumbent value in TOL-B&amp;B</td>
</tr>
<tr>
<td>(c_{ik})</td>
<td>cost for dimension (i) and manufacturing process (k)</td>
</tr>
<tr>
<td>(C_L)</td>
<td>lower bound in cost for the partial solution (Y) in TOL-B&amp;B</td>
</tr>
<tr>
<td>(F_j(X))</td>
<td>the (j)-th stack-up function where (F_j(X) \geq 0) is the desired condition</td>
</tr>
<tr>
<td>(N)</td>
<td>total number of manufacturing processes, i.e., (s_1 + \cdots + s_n)</td>
</tr>
<tr>
<td>(q_j)</td>
<td>desired reliability of the (j)-th stack-up condition, i.e., (\Phi^{-1}(1 - \delta_j))</td>
</tr>
<tr>
<td>(R_{S_j}, R_{F_j})</td>
<td>safe and fail regions formed by (F_j(X))</td>
</tr>
<tr>
<td>(t_{ik})</td>
<td>tolerance of dimension (i) due to the use of manufacturing process (k)</td>
</tr>
<tr>
<td>(V)</td>
<td>(n \times n) diagonal variance matrix whose (i)-th diagonal element is (\sigma_i^2)</td>
</tr>
<tr>
<td>(X)</td>
<td>random dimension vector</td>
</tr>
<tr>
<td>(\bar{X})</td>
<td>mean vector of (X)</td>
</tr>
<tr>
<td>(Y)</td>
<td>partial solution in TOL-B&amp;B</td>
</tr>
<tr>
<td>(y_{ik})</td>
<td>0-1 decision variable for dimension (i) and manufacturing process (k)</td>
</tr>
<tr>
<td>(Z)</td>
<td>standardized dimension vector</td>
</tr>
<tr>
<td>(\beta_j)</td>
<td>reliability index for the (j)-th stack-up function</td>
</tr>
<tr>
<td>(\delta_j)</td>
<td>permissible dissatisfaction of the (j)-th stack-up condition</td>
</tr>
<tr>
<td>(\sigma_{ik})</td>
<td>standard deviation for dimension (i) and manufacturing process (k)</td>
</tr>
<tr>
<td>(\phi(\bar{X}; V))</td>
<td>multivariate normal p.d.f. having mean (\bar{X}) and variance (V)</td>
</tr>
<tr>
<td>(\Phi(\cdot))</td>
<td>cumulative standardized normal distribution function</td>
</tr>
<tr>
<td>(\Phi^{-1}(\cdot))</td>
<td>inverse cumulative standardized normal distribution function</td>
</tr>
</tbody>
</table>
Table 2. Cost and Standard Deviations for Example 2

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Process</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>SD*</td>
<td>Cost</td>
<td>SD*</td>
<td>Cost</td>
<td>SD*</td>
</tr>
<tr>
<td>1</td>
<td>20.0</td>
<td>35.0</td>
<td>23.0</td>
<td>30.0</td>
<td>29.0</td>
<td>25.0</td>
</tr>
<tr>
<td>2</td>
<td>2.0</td>
<td>2200.0</td>
<td>5.0</td>
<td>2000.0</td>
<td>11.0</td>
<td>1800.0</td>
</tr>
<tr>
<td>3</td>
<td>13.0</td>
<td>97.0</td>
<td>16.0</td>
<td>96.0</td>
<td>22.0</td>
<td>95.0</td>
</tr>
<tr>
<td>4</td>
<td>13.0</td>
<td>110.0</td>
<td>16.0</td>
<td>105.0</td>
<td>22.0</td>
<td>100.0</td>
</tr>
<tr>
<td>5</td>
<td>35.0</td>
<td>3.3</td>
<td>45.0</td>
<td>3.2</td>
<td>65.0</td>
<td>3.1</td>
</tr>
<tr>
<td>6</td>
<td>35.0</td>
<td>4.1</td>
<td>41.0</td>
<td>4.0</td>
<td>53.0</td>
<td>3.0</td>
</tr>
<tr>
<td>7</td>
<td>32.0</td>
<td>3.0</td>
<td>39.0</td>
<td>2.9</td>
<td>53.0</td>
<td>2.8</td>
</tr>
<tr>
<td>8</td>
<td>32.0</td>
<td>2.2</td>
<td>38.0</td>
<td>2.1</td>
<td>50.0</td>
<td>2.0</td>
</tr>
<tr>
<td>9</td>
<td>2.0</td>
<td>2100.0</td>
<td>5.0</td>
<td>2050.0</td>
<td>11.0</td>
<td>2000.0</td>
</tr>
<tr>
<td>10</td>
<td>4.0</td>
<td>136.0</td>
<td>6.0</td>
<td>134.0</td>
<td>10.0</td>
<td>132.0</td>
</tr>
<tr>
<td>11</td>
<td>10.0</td>
<td>97.0</td>
<td>12.0</td>
<td>96.0</td>
<td>16.0</td>
<td>95.0</td>
</tr>
<tr>
<td>12</td>
<td>16.0</td>
<td>31.0</td>
<td>18.0</td>
<td>30.0</td>
<td>22.0</td>
<td>29.0</td>
</tr>
</tbody>
</table>

* standard deviation (unit: \(10^{-4}\))
<table>
<thead>
<tr>
<th>Input</th>
<th>Data</th>
<th>see Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size ( \sum_{i=1}^{n} s_i )</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>No. of Selections ( \prod_{i=1}^{n} s_i ) (^{(a)})</td>
<td>1,574,640</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Run Statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>iterations in TOL-B&amp;B (^{(b)})</td>
<td>= 2553 (3267(^{(f)}))</td>
</tr>
<tr>
<td>fathoming by cost (^{(c)})</td>
<td>= 1271 (1719(^{(f)}))</td>
</tr>
<tr>
<td>feasibility checks (^{(d)})</td>
<td>= 1282 (1548(^{(f)}))</td>
</tr>
<tr>
<td>fathoming by feasibility</td>
<td>= 460 (490(^{(f)}))</td>
</tr>
<tr>
<td>incumbent changes</td>
<td>= 9 (73(^{(f)}))</td>
</tr>
</tbody>
</table>

| Efficiency of TOL-B&B \(^{(e)}\) | 616.8 (482.0\(^{(f)}\)) |

| Run Time | 3.076 (3.884\(^{(f)}\)) CPU seconds |

<table>
<thead>
<tr>
<th>Output</th>
<th>Optimum Selection</th>
<th>(3,2,1,3,2,2,2,1,1,2,2,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimum Cost</td>
<td>262.00</td>
</tr>
</tbody>
</table>

Note: 
(b) = (c) + (d)  
(e) = (a) / (b)  
(f): result obtained without TOL-INITIAL
### Table 4. Satisfaction Level at Optimum in Example 2

<table>
<thead>
<tr>
<th>Stack-Up Condition</th>
<th>Reliability Index ($\beta_j$)</th>
<th>Satisfaction Level in % ($\Phi(\beta_j)$)</th>
<th>Over-Satisfaction Level in % ($\Phi(\beta_j) - 0.991488$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.38697</td>
<td>99.151</td>
<td>0.002</td>
</tr>
<tr>
<td>2</td>
<td>2.38618</td>
<td>99.149</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>2.39801</td>
<td>99.176</td>
<td>0.027</td>
</tr>
<tr>
<td>4</td>
<td>2.39556</td>
<td>99.170</td>
<td>0.022</td>
</tr>
<tr>
<td>5</td>
<td>2.51101</td>
<td>99.398</td>
<td>0.249</td>
</tr>
<tr>
<td>6</td>
<td>2.51101</td>
<td>99.398</td>
<td>0.249</td>
</tr>
</tbody>
</table>

**Note:** The desired reliability index corresponding to 99.1488% is 2.38617 since $\Phi^{-1}(1-0.991488) = 2.38617$
Figure 1. Tolerance-Cost Relationship
Figure 2. Clearance Condition of Mating Parts
Figure 3. Basic Scheme of Tolerance Selection
\begin{align*}
F_1(X) &= (x_6 - x_5) - (x_8 - x_7) \\
F_2(X) &= (x_3 - x_4) - (x_{11} - x_{10}) \\
F_3(X) &= (x_8 - x_7)(x_2 - x_3) - (x_6 - x_5)(x_{10} - x_9) + \tan(\pi / 180) \{ (x_{10} - x_9)(x_2 - x_3) + (x_8 - x_7)(x_6 - x_5) \} \\
F_4(X) &= (x_6 - x_5)(x_{10} - x_9) - (x_8 - x_7)(x_2 - x_3) + \tan(\pi / 180) \{ (x_{10} - x_9)(x_2 - x_3) + (x_8 - x_7)(x_6 - x_5) \} \\
F_5(X) &= -x_1 + x_{12} + 0.01 \\
F_6(X) &= x_1 - x_{12} + 0.01
\end{align*}

Figure 4. Example of Linear and Nonlinear Functions
Figure 5. Rotational Symmetry of Standard System
Figure 6. Enumeration Tree
Figure 7. Flow-Chart of Procedure TOL-B&B
Figure 8. Tree Traversal in Example 1