

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_1 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 $y_{i1} + \dots + y_{is_i} = 1$, 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 \mathbf{R}_S for satisfying the following stack-up conditions:

$$\begin{aligned} F_1(\mathbf{X}) &= x_1 - x_2 - 0.1 \geq 0 \\ F_2(\mathbf{X}) &= -x_1 + x_2 + 0.2 \geq 0. \end{aligned} \tag{1}$$

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 \mathbf{R}_S and the rectangle specified by the tolerances. For example, the rectangle specified by (t_{12}, t_{24}) is entirely covered by \mathbf{R}_S , whereas a smaller fraction of the rectangle specified by (t_{12}, t_{21}) is covered by \mathbf{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 $F(\mathbf{X}) \geq 0$ represents a certain stack-up condition, where \mathbf{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:

$$\int_{F(\mathbf{X}) \geq 0} f(\mathbf{X}) d\mathbf{X} \quad (2)$$

where $f(\mathbf{X})$ is the multivariate probability density function (p.d.f.) of \mathbf{X} . $F(\mathbf{X})$, the function for stack-up condition, is nonlinear if non-rectilinear shapes and/or dimensions are in an engineering drawing. See, for example, $F_3(\mathbf{X})$ and $F_4(\mathbf{X})$ in Figure 4. (Those two conditions specify that the angular difference between θ_1 and θ_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 $F_3(\mathbf{X})$, 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 $F(\mathbf{X})$ 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

¹The two conditions $F_3(\mathbf{X}) \geq 0$ and $F_4(\mathbf{X}) \geq 0$ are derived by taking the following steps:

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

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:

$$\text{Min } \sum_{i=1}^n \sum_{k=1}^{s_i} c_{ik} y_{ik} \quad (3)$$

subject to

$$\beta_j \geq q_j, \quad \text{for } j=1, \dots, m \quad (3-1)$$

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

where $y_{ik} = 0$ or 1 for $k=1, \dots, s_i$ and $i=1, \dots, n$

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 β_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., $(t_{11}, t_{21}), (t_{11}, t_{22}), (t_{11}, t_{23}), (t_{11}, t_{24}), (t_{12}, t_{21}), (t_{12}, t_{22}), (t_{12}, t_{23})$ and (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 $\mathbf{X}=(x_1, \dots, x_n)^t$ 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 Σ , 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* $\mathbf{R}_{S_j} = \{ \mathbf{X} \mid F_j(\mathbf{X}) \geq 0 \}$ and a *fail region* $\mathbf{R}_{F_j} = \{ \mathbf{X} \mid F_j(\mathbf{X}) < 0 \}$. The safe region \mathbf{R}_S of satisfying all the stack-up conditions is the intersection of \mathbf{R}_{S_j} , $1 \leq j \leq m$. That is, $\mathbf{R}_S = \bigcap_{j=1}^m \mathbf{R}_{S_j}$. 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:

$$\int_{F_j(\mathbf{X}) \geq 0} \phi(\bar{\mathbf{X}}; \mathbf{V}) d\mathbf{X} \quad (4)$$

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 $F_j(\mathbf{X}) \geq 0$ 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:

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

subject to

(5)

$$\int_{F_j(\mathbf{X}) \geq 0} \phi(\bar{\mathbf{X}}; \mathbf{V}) d\mathbf{X} \geq 1 - \delta_j, \quad \text{for } j=1, \dots, m \quad (\text{reliability constraints})$$

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

$$\text{where } y_{ik} = 0 \text{ or } 1 \quad \text{for } k=1, \dots, s_i \text{ and } i=1, \dots, n$$

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

$$\text{Min} \quad c_{11}y_{11} + c_{12}y_{12} + c_{21}y_{21} + c_{22}y_{22} + c_{23}y_{23} + c_{24}y_{24}$$

subject to

$$\int_{F_1(\mathbf{X}) = x_1 - x_2 - 0.1 \geq 0} \phi(\bar{\mathbf{X}}; \mathbf{V}) d\mathbf{X} \geq 1 - \delta_1$$

$$\int_{F_2(\mathbf{X}) = -x_1 + x_2 + 0.2 \geq 0} \phi(\bar{\mathbf{X}}; \mathbf{V}) d\mathbf{X} \geq 1 - \delta_2$$

$$y_{11} + y_{12} = 1$$

$$y_{21} + y_{22} + y_{23} + y_{24} = 1$$

where $y_{11}, y_{12}, y_{21}, y_{22}, y_{23}, y_{24}$ are 0-1 variables and

δ_1, δ_2 are permissible dissatisfaction levels for $F_1(\mathbf{X}) \geq 0, F_2(\mathbf{X}) \geq 0$.

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 \mathbf{R}_{S_j} . Since \mathbf{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 β . (The solution scheme for computing β is also explained in Appendix A). Then, the probability of covering one side of the tangent hyperplane, $P(\mathbf{R}_{S_j})$, 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.

<Insert Figure 5>

Lemma 1. If the stack-up function $F_j(\mathbf{X})$ has a reliability index β_j , then

$$P(\mathbf{R}_{S_j}) \simeq \Phi(\beta_j). \quad (6)$$

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, $P(\mathbf{R}_{S_j}) = \Phi(\beta_j)$.

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

$$\begin{aligned} & \text{Min} \quad \sum_{i=1}^n \sum_{k=1}^{s_i} c_{ik} y_{ik} \\ & \text{subject to} \\ & \beta_j \geq \Phi^{-1}(1 - \delta_j), \quad \text{for } j=1, \dots, m \quad (\text{reliability constraints}) \\ & \sum_{k=1}^{s_i} y_{ik} = 1, \quad \text{for } i=1, \dots, n \quad (\text{selection constraints}) \\ & \text{where } y_{ik} = 0 \text{ or } 1 \quad \text{for } k=1, \dots, s_i \quad \text{and } i=1, \dots, n \end{aligned}$$

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 $g(\mathbf{X})$ is said to be monotone nonincreasing in every element x_i 's if $\mathbf{X}_1 \geq \mathbf{X}_2$ results in $g(\mathbf{X}_1) \leq g(\mathbf{X}_2)$.

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:

$$\begin{aligned} c_{i1} < c_{i2} < c_{i3} \cdots < c_{is_i} & \text{ for } i=1, \dots, n \\ \sigma_{i1} > \sigma_{i2} > \sigma_{i3} \cdots > \sigma_{is_i} & \text{ for } i=1, \dots, n \end{aligned}$$

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, $\partial d_{j,0} / \partial \sigma_i \leq 0$ for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Proof. The proof is given in Appendix B. •

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

$$\partial \beta_j / \partial \sigma_i \leq 0 \quad (7)$$

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 \mathbf{Y} is used. For example, $\mathbf{Y}=(2, s_2, 0, 0, \dots, 0)$ means that for dimension x_1 the second manufacturing process is selected from PSMP₁ and for dimension x_2 the s_2 -th manufacturing process from PSMP₂. The value 0 in \mathbf{Y} shows that the decision for the corresponding dimension has not been made yet. Notice that, by using the vector \mathbf{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 $(1,0,\dots,0), \dots, (s_1,0,\dots,0)$, 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 $(0,0,\dots,0)$, where $(0,0,\dots,0)$ 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 $Y = (i_1, i_2, \dots, i_r, 0, \dots, 0)$.
Then, the lower bound C_L for Y is $c_{1i_1} + \dots + c_{ri_r} + \sum_{j=r+1}^n c_{j1}$.
- Step 5. (Fathoming Criterion 1) If $C_L \geq C^*$, go to Step 2.
- Step 6. (Fathoming Criterion 2) Modify $Y = (i_1, i_2, \dots, i_r, 0, 0, \dots, 0)$ with $Y^* = (i_1, i_2, \dots, i_r, s_{r+1}, \dots, s_n)$. 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 $\sum_{j=1}^n c_{jj}$. 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 $(i_1, i_2, \dots, i_{r-1}, 1, 0, \dots, 0)$ and $(i_1, i_2, \dots, i_{r-1}, 2, 0, \dots, 0)$ can be deleted. Go to Step 2.

Step 10. Separate the candidate problem and add its offsprings to the candidate list. The offsprings of \mathbf{Y} are $(i_1, i_2, \dots, i_r, 1, 0, \dots, 0)$, $(i_1, i_2, \dots, i_r, 2, 0, \dots, 0)$, \dots , $(i_1, i_2, \dots, i_r, s_{r+1}, 0, \dots, 0)$. 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 $\mathbf{Y}=(i_1, \dots, i_r, 0, \dots, 0)$ is equivalent to the cost for the selection $(i_1, \dots, i_r, 1, \dots, 1)$. Similarly, in Step 6, to check for the feasibility of \mathbf{Y} , $\mathbf{Y}^*=(i_1, \dots, i_r, s_{r+1}, \dots, s_n)$ 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 $(i_1, \dots, i_r, 0, \dots, 0)$, 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 $Y^i = (s_1, \dots, s_n)$ 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 $Y^{i+1} = (1, 1, \dots, 1, 1)$, then stop since there is no feasible solution. Otherwise, $Y^{i+1} := Y^i$.
- Step 6. stop-check:= stop-check + 1.
- Step 7. If current-flag = n, then current-flag:=1. Otherwise, current-flag:= current-flag + 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 \mathbf{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 σ_{ik} are used to compute

β_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 θ_1 and θ_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. $F_3(\mathbf{X})$ and $F_4(\mathbf{X})$ are nonlinear and the others are linear.

The nominal dimensions for the 12 dimensions are given as $\bar{\mathbf{X}} = (50.0, 40.00125, 20.05, 9.9985, 9.9985, 30.00, 10.00, 30.00, 10.05, 30.00, 40.00, 50.00)^t$. 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., $\delta_1 = \dots = \delta_4 = 1 - (0.95)^{1/6}$. 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, 2, 1, 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 $F_j(\mathbf{X})$

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

- [1] *ISO System of Limits and Fits, General Tolerances and Deviations*, R286-1962, American National Standards Institute, New York (1962).
- [2] Balas, E., "Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research*, **13**, 517-546 (1965).
- [3] Balling, R.J., Free, J.C. and Parkinson, A.R., "Consideration of Worst-Case Manufacturing Tolerances in Design Optimization," *Trans. of ASME, Journal of Mechanisms, Transmissions and Automation in Design*, **108**, 438-441 (Dec. 1986).
- [4] Bjorke, O., *Computer Aided Tolerancing*, Tapir Press, Norway (1978).
- [5] Dong, Z. and Soom, A., "Automatic Tolerance Analysis from a CAD Database," ASME Paper No. 86-DET-36 (1986)
- [6] Greenwood, W.H. and Chase, K.H., "A New Tolerance Analysis Method for Designers and Manufacturers," *Trans. of ASME, Journal of Engineering for Industry*, **109**, 112-116 (May 1987)
- [7] Grossman, D.D., "Monte-Carlo Simulation of Tolerancing in Discrete Parts Manufacturing and Assembly," Report No. STAN-CS-76-555, Computer Science Department, Stanford Univ. (May 1976)
- [8] Hasofer, A.M. and Lind, N.C., "Exact and Invariant Second-Moment Code Format," *ASCE J.Eng.Mech.Div.*, **100**, EM1, 111-121 (1974).
- [9] Huang, J. and Ostwald, P.F., "A Method for Optimal Tolerance Selection," ASME Paper No. 76-WA/DE-23 (1976).
- [10] Kirkpatrick, E.G., *Quality Control for Managers and Engineers*, John Wiley and Sons, Inc., New York (1970).
- [11] Lee, W.J. and Woo, T.C., "Tolerancing: Its Distribution, Analysis, and Synthesis," Technical Report 86-30, Dept. of Industrial and Operations Eng., Univ. of Michigan, Submitted to *Trans. of ASME, J. of Eng. for Industry* (1986)
- [12] Madsen, H.O., Krenk, S. and Lind, N.C., *Methods of Structural Safety*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1986).
- [13] Mansoor, E.M., "The Application of Probability to Tolerances Used in Engineering Designs," *Proc.Inst.Mech.Eng.*, **178**, 1, 29-51 (1963).
- [14] Michael, W. and Siddall, J.N., "The Optimization Problem with Optimal Tolerance Assignment and Full Acceptance," *Trans. of ASME, Journal of Mechanical Design*, **103**, 842-848 (Oct. 1981)
- [15] Michael, W. and Siddall, J.N., "The Optimal Tolerance Assignment with Less Than Full Acceptance," *Trans. of ASME, Journal of Mechanical Design*, **104**, 855-860 (Oct. 1982)
- [16] Monte, M.E. and Datseris, P., "Optimum Tolerance Selection for Minimum Manufacturing Cost and Other Design Criterion," ASME Paper No. 82-DET-35

(1982).

- [17] Murty, K.G., "Linear and Combinatorial Programming," John Wiley and Sons, Inc., New York, 437-480 (1976).
- [18] Parkinson, D.B., "The Application of Reliability Methods to Tolerancing," *Trans. of ASME, Journal of Mechanical Design*, **104**, 612-618 (July 1982)
- [19] Peat, A.P., *Cost Reduction Charts for Designers and Production Engineers*, The Machining Publishing Co. Ltd., Brighton (1968).
- [20] Spotts, M.F., "Allocation of Tolerances to Minimize Cost of Assembly," *Trans. of ASME, Journal of Engineering for Industry*, 762-764 (August 1973).
- [21] Turner, J.U., "Tolerances in Computer-Aided Geometric Design," Ph.D. dissertation, Rensselaer Polytechnic Institute, Troy, New York (1987).
- [22] Wilde, D. and Prentice, E., "Minimum Exponential Cost Allocation of Sure-Fit Tolerances," ASME Paper No. 75-DET-93 (1975).
- [23] Wilde, D., "Simplifying Discrete Tolerance Assignment," ASME Paper No. 75-DET-106 (1975).

APPENDICES

Appendix A. Reliability Index

The Euclidean distance of a point \mathbf{Z} in the standard system to the origin is $(\mathbf{Z}^t \mathbf{Z})^{1/2}$.

The standardized vector \mathbf{Z} is obtained by the transformation $z_i = \frac{x_i - \bar{x}_i}{\sigma_i}$ for $1 \leq i \leq n$.

Hence, $(\mathbf{Z}^t \mathbf{Z})^{1/2} = \{(\mathbf{X} - \bar{\mathbf{X}})^t \mathbf{V}^{-1} (\mathbf{X} - \bar{\mathbf{X}})\}^{1/2}$. The reliability index can be obtained by solving the following problem:

$$\begin{aligned} & \text{Min } \beta^2 = (\mathbf{X} - \bar{\mathbf{X}})^t \mathbf{V}^{-1} (\mathbf{X} - \bar{\mathbf{X}}) \\ \text{subject to } & \mathbf{F}(\mathbf{X}) \leq 0 \end{aligned} \quad (8)$$

Here, the objective function is expressed as β^2 instead of β since the positive definiteness of the covariance matrix \mathbf{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:

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

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

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 \mathbf{Z}_0 can be expressed in terms of the original vector \mathbf{X} by $\left\{(\mathbf{X}_0 - \bar{\mathbf{X}})^t \mathbf{V}^{-1} (\mathbf{X}_0 - \bar{\mathbf{X}})\right\}^{1/2}$, where \mathbf{X}_0 is for the point \mathbf{Z}_0 before standardization. Note that this distance expression corresponds to the equation of an ellipsoid with center $\bar{\mathbf{X}}$, 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 σ_i decreases, the distance $d_{j,0}$ increases.

Table 1. Notation

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

Table 2. Cost and Standard Deviations for Example 2

Di- men- sion	Process									
	1		2		3		4		5	
	Cost	SD*	Cost	SD*	Cost	SD*	Cost	SD*	Cost	SD*
1	20.0	35.0	23.0	30.0	29.0	25.0	-	-	-	-
2	2.0	2200.0	5.0	2000.0	11.0	1800.0	-	-	-	-
3	13.0	97.0	16.0	96.0	22.0	95.0	-	-	-	-
4	13.0	110.0	16.0	105.0	22.0	100.0	-	-	-	-
5	35.0	3.3	45.0	3.2	65.0	3.1	-	-	-	-
6	35.0	4.1	41.0	4.0	53.0	3.0	-	-	-	-
7	32.0	3.0	39.0	2.9	53.0	2.8	-	-	-	-
8	32.0	2.2	38.0	2.1	50.0	2.0	-	-	-	-
9	2.0	2100.0	5.0	2050.0	11.0	2000.0	-	-	-	-
10	4.0	136.0	6.0	134.0	10.0	132.0	18.0	130.00	-	-
11	10.0	97.0	12.0	96.0	16.0	95.0	24.0	94.00	-	-
12	16.0	31.0	18.0	30.0	22.0	29.0	30.0	28.00	46.0	27.0

* standard deviation (unit: 10^{-4})

Table 3. Results for Example 2

	Data	see Table 2
Input	Size ($\sum_{i=1}^n s_i$)	40
	No. of Selections ($\prod_{i=1}^n s_i$) ^(a)	1,574,640
	Run Statistics	iterations in TOL-B&B ^(b) = 2553 (3267 ^(f)) fathoming by cost ^(c) = 1271 (1719 ^(f)) feasibility checks ^(d) = 1282 (1548 ^(f)) fathoming by feasibility = 460 (490 ^(f)) incumbent changes = 9 (73 ^(f))
	Efficiency of TOL-B&B ^(e)	616.8 (482.0 ^(f))
	Run Time	3.076 (3.884 ^(f)) CPU seconds
Output	Optimum Selection	(3,2,1,3,2,2,2,1,1,2,2,1)
	Optimum Cost	262.00

Note: (b) = (c) + (d)

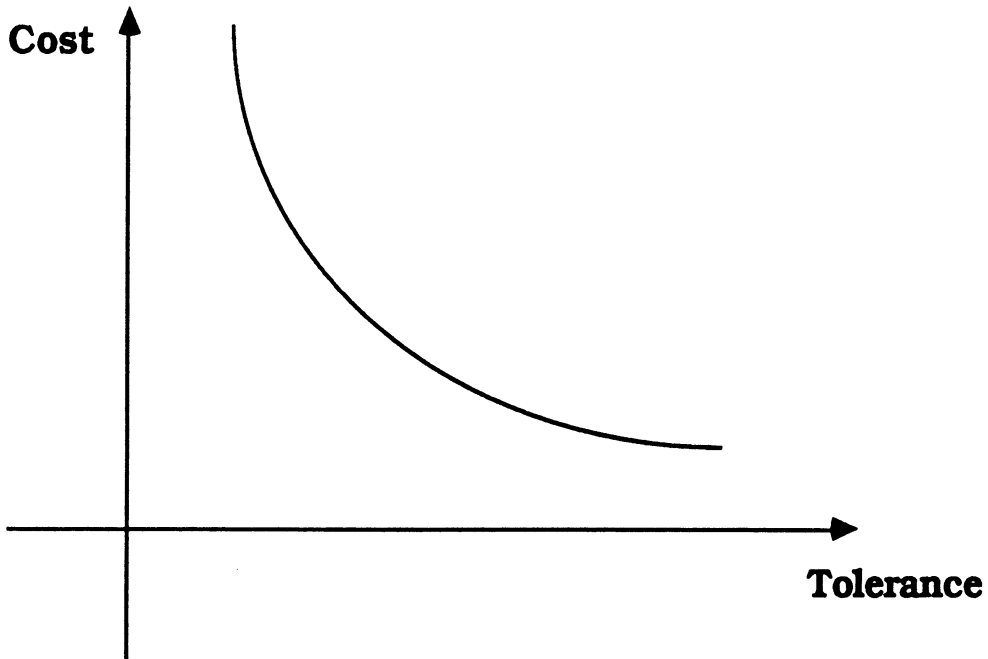
(e) = (a) / (b)

(f): result obtained without TOL-INITIAL

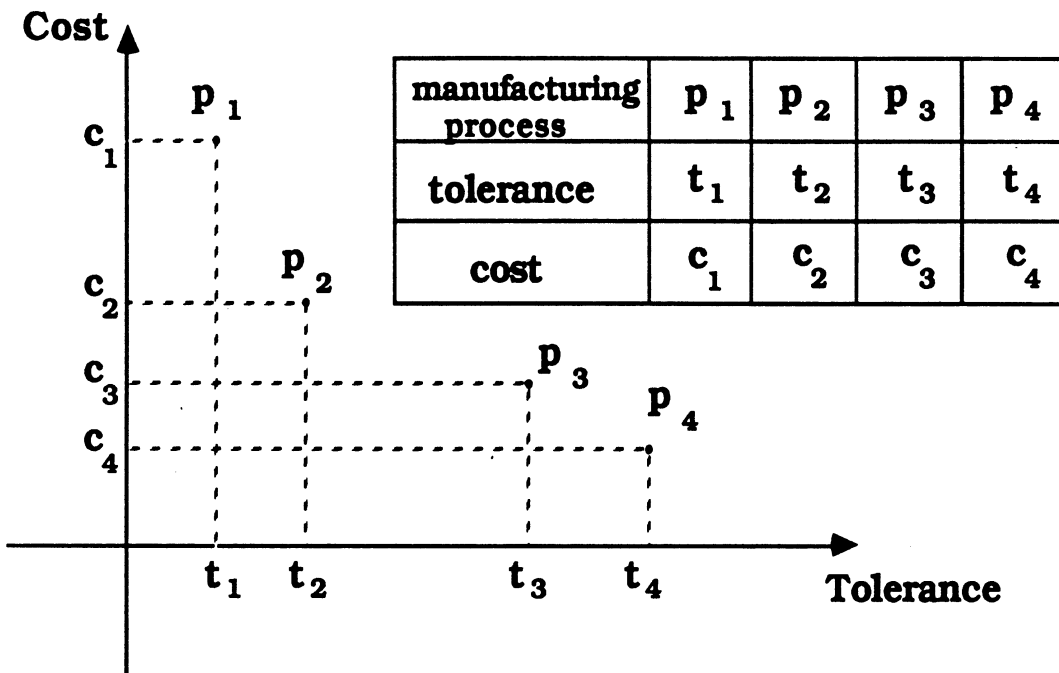
Table 4. Satisfaction Level at Optimum in Example 2

Stack-Up Condition	Reliability Index (β_j)	Satisfaction Level in % ($\Phi(\beta_j)$)	Over-Satisfaction Level in % ($\Phi(\beta_j) - 0.991488$)
1	2.38697	99.151	0.002
2	2.38618	99.149	0.000
3	2.39801	99.176	0.027
4	2.39556	99.170	0.022
5	2.51101	99.398	0.249
6	2.51101	99.398	0.249

Note: The desired reliability index corresponding to 99.1488% is 2.38617
since $\Phi^{-1}(1-0.991488) = 2.38617$



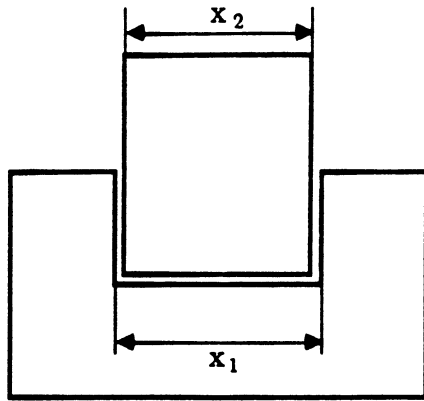
(a) Continuous Model



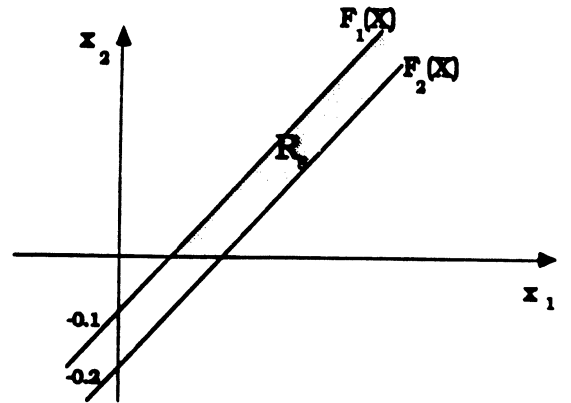
(b) Discrete Model

Figure 1. Tolerance-Cost Relationship

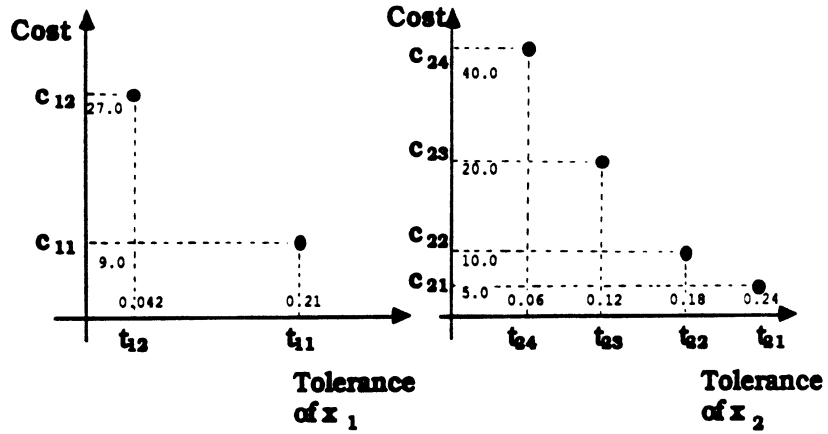
2-(a)



2-(b)



2-(c)



2-(d)

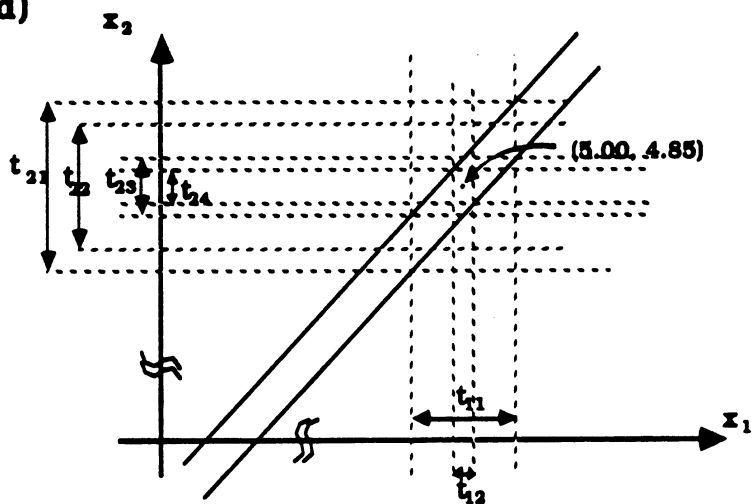


Figure 2. Clearance Condition of Mating Parts

Tolerance Selection

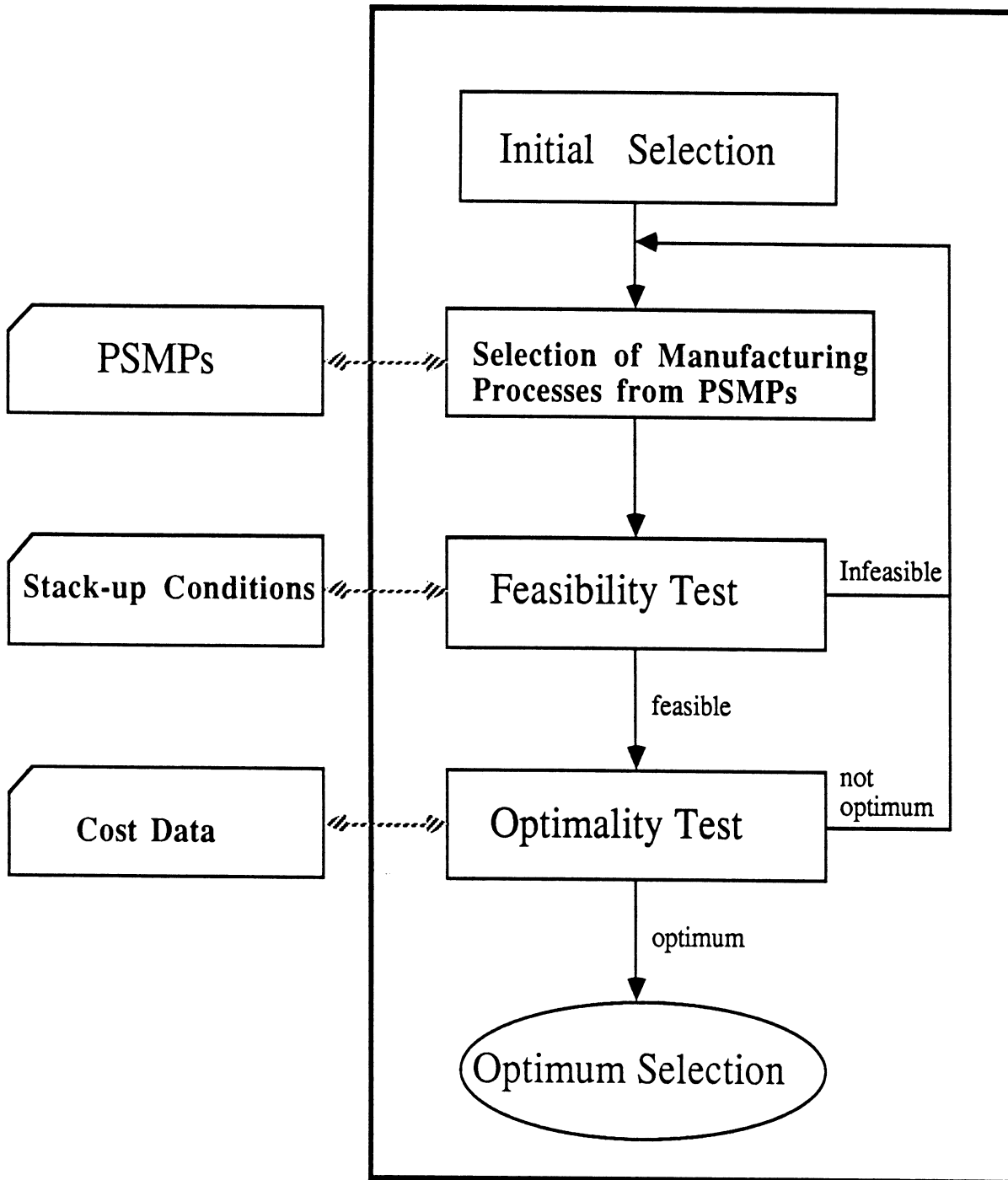
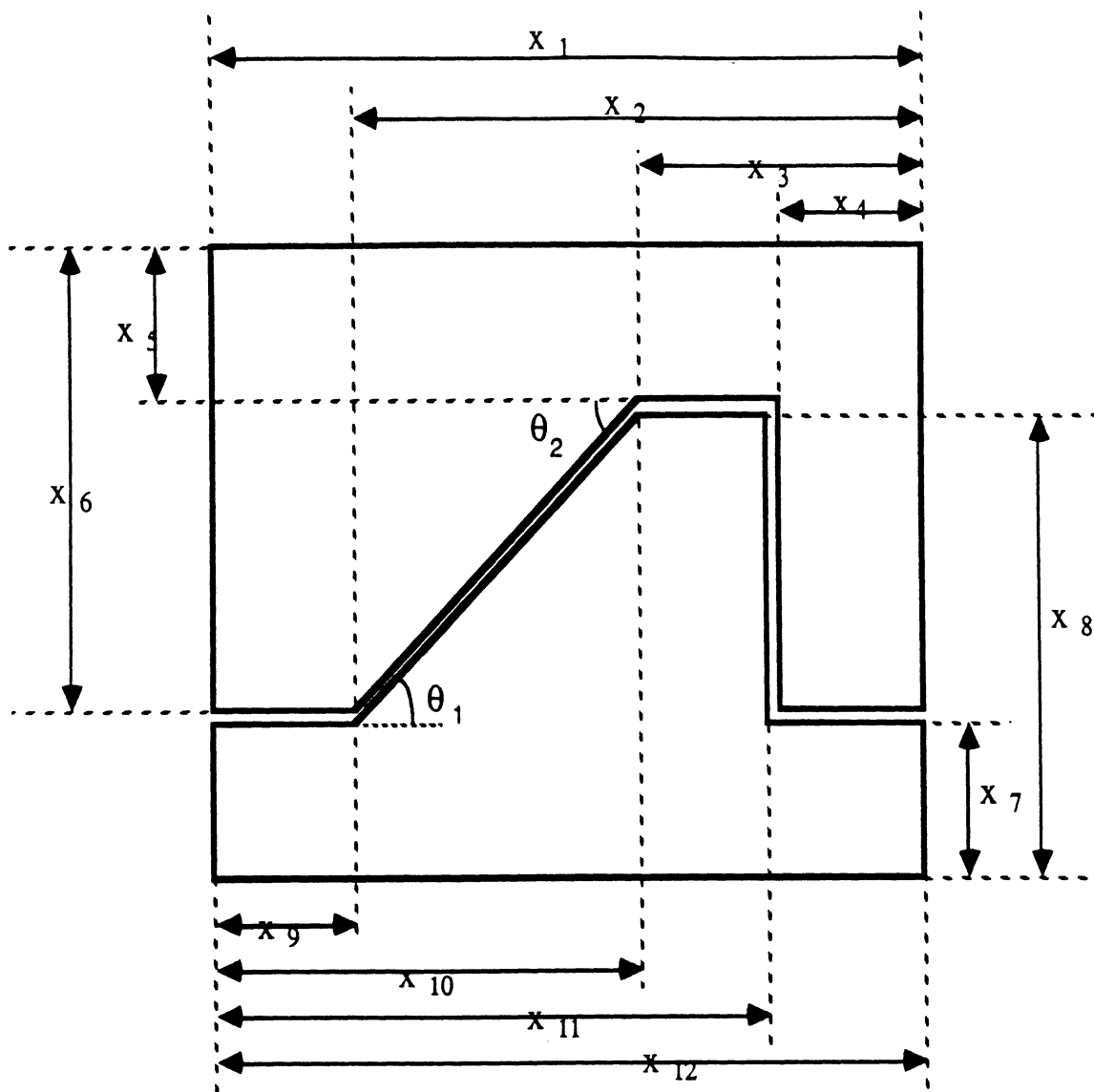


Figure 3. Basic Scheme of Tolerance Selection



$$F_1(X) = (x_6 - x_5) - (x_8 - x_7)$$

$$F_2(X) = (x_3 - x_4) - (x_{11} - x_{10})$$

$$F_3(X) = (x_8 - x_7) * (x_2 - x_3) - (x_6 - x_5) * (x_{10} - x_9) + \tan(\pi / 180) * \{ (x_{10} - x_9) * (x_2 - x_3) + (x_8 - x_7) * (x_6 - x_5) \}$$

$$F_4(X) = (x_6 - x_5) * (x_{10} - x_9) - (x_8 - x_7) * (x_2 - x_3) + \tan(\pi / 180) * \{ (x_{10} - x_9) * (x_2 - x_3) + (x_8 - x_7) * (x_6 - x_5) \}$$

$$F_5(X) = -x_1 + x_{12} + 0.01$$

$$F_6(X) = x_1 - x_{12} + 0.01$$

Figure 4. Example of Linear and Nonlinear Functions

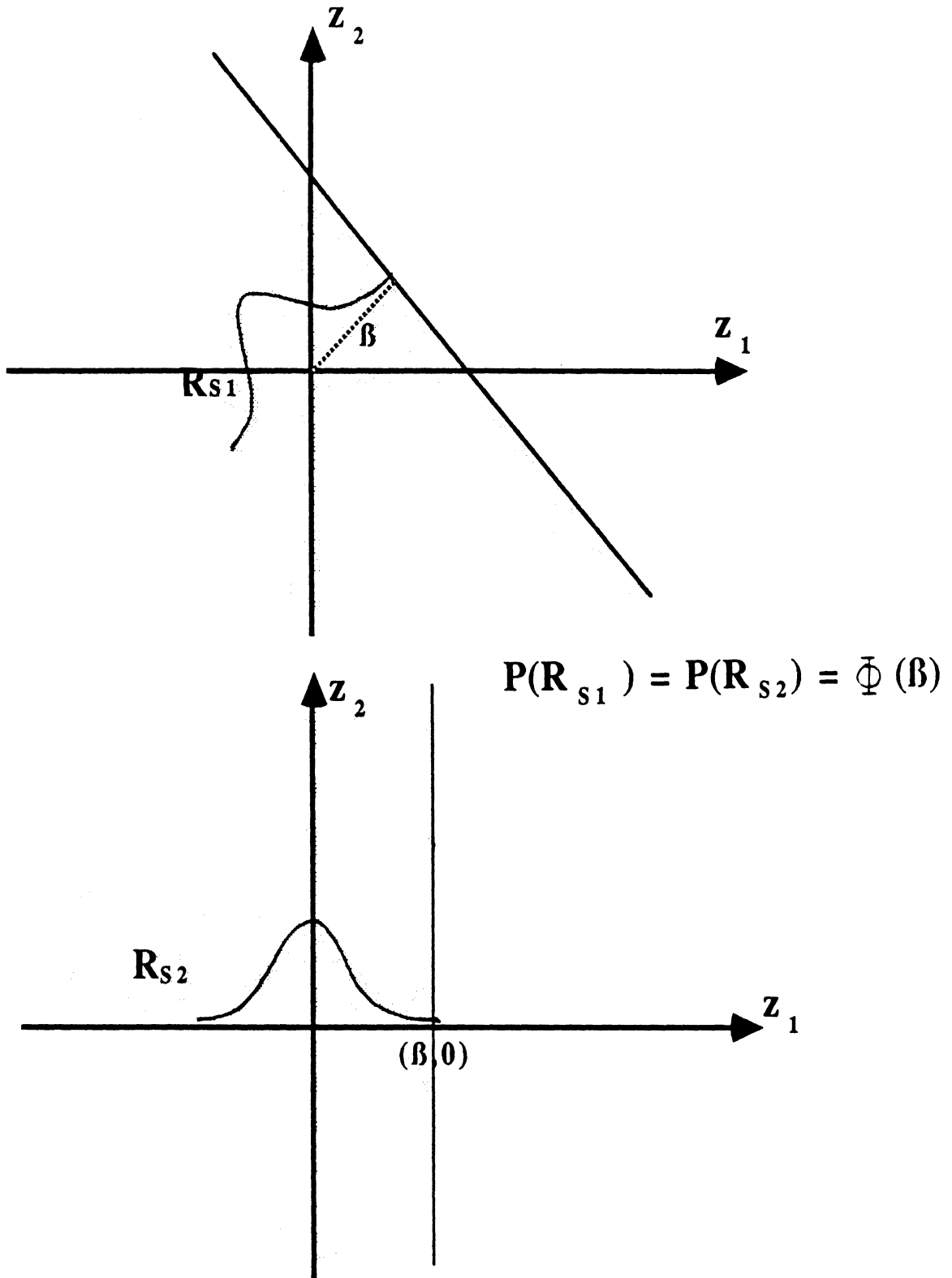


Figure 5. Rotational Symmetry of Standard System

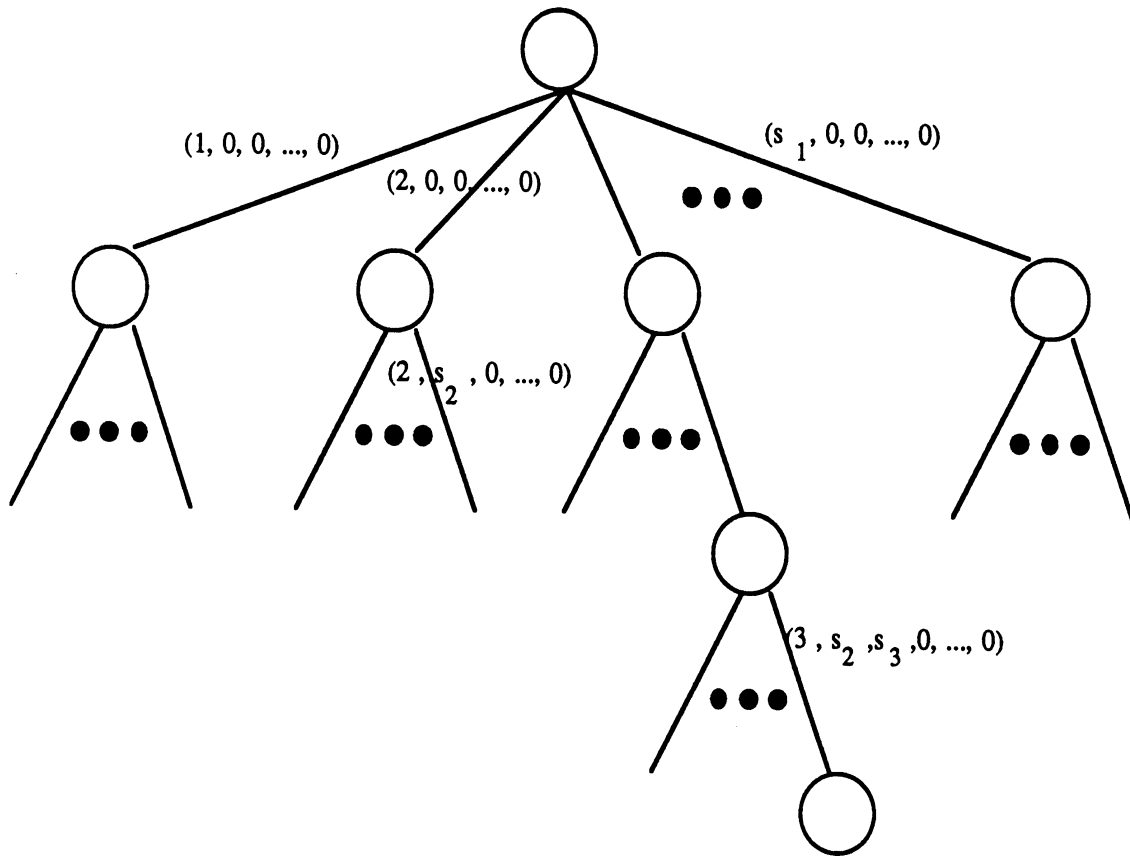


Figure 6. Enumeration Tree

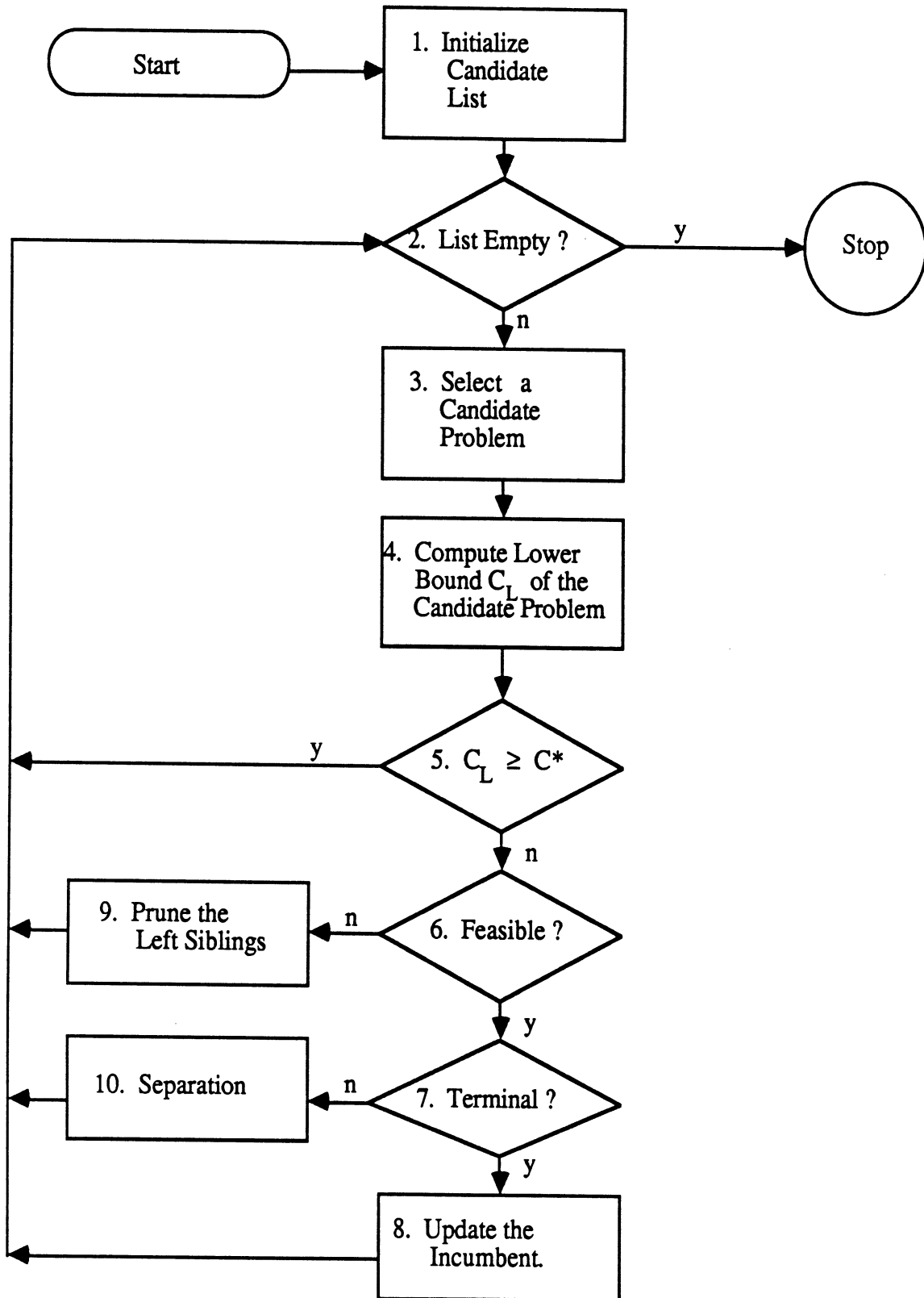


Figure 7. Flow-Chart of Procedure TOL-B&B

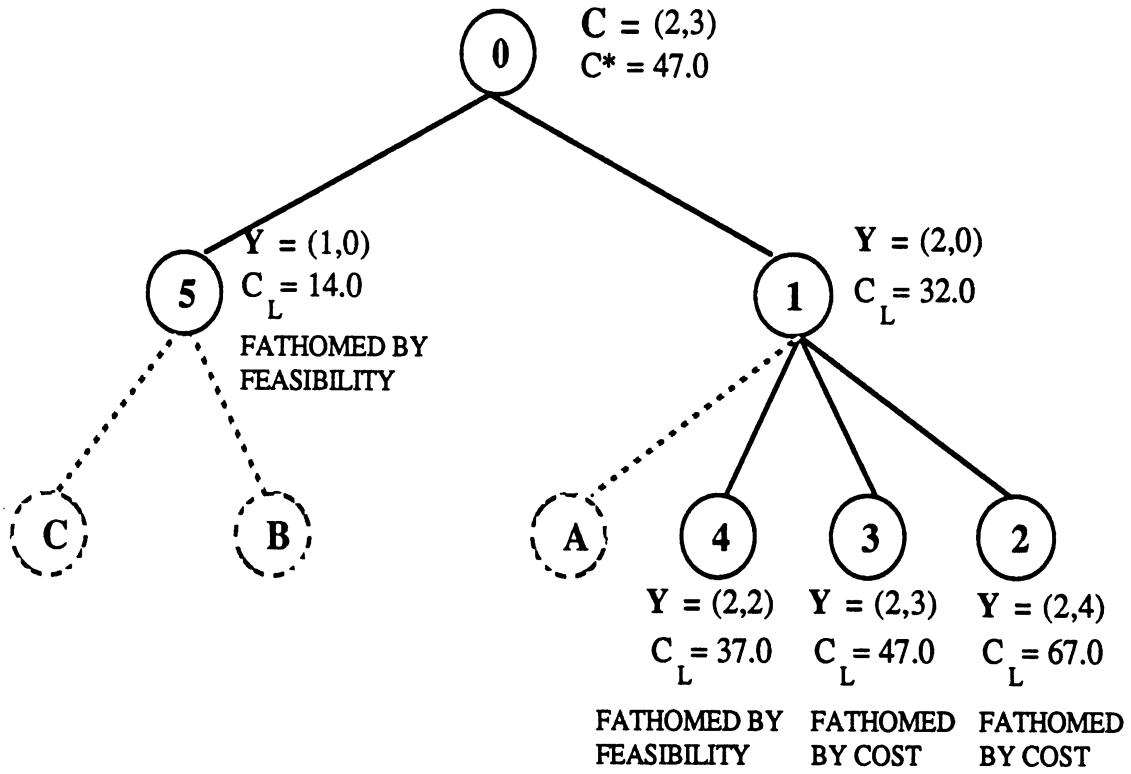


Figure 8. Tree Traversal in Example 1