Applied Artificial Intelligence, 17:631–660, 2003

ISSN: 0883-9514 print / 1087-6545 online
DOI: 10.1080/08839510390218058

Taylor & Francis
Taylor & Francis Group

OPTIMUM TOLERANCE DESIGN USING CONSTRAINT NETWORKS AND RELATIVE SENSITIVITY RATIO ALGORITHM

CHRISTOPHER C. YANG
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Hong Kong

V. N. ACHUTHA NAIKAN
Department of Industrial Engineering and Management,
Indian Institute of Technology, Kharagpur

Tolerance design plays a significant role in the relationship between the performance and the manufacture of a product. It is important to maintain valid tolerances when product design is constrained by the relationship between the dimensions of entities of a component and the functional requirement of the design. Increasing the tolerance may decrease the manufacturing cost, but will also worsen the performance. In this paper, an efficient algorithm is proposed for computerized optimal allocation of tolerance among the components of a complex assembly with a large number of constraints and entities. Basic concepts of hierarchical interval constraint networks have been used in combination with an iterative relative sensitivity analysis procedure for modeling and solving the tolerance allocation problem. The proposed algorithm can handle practically any number of constraints and entities with the required level of accuracy. It can also accept any type of cost-tolerance relationship for modeling. Examples have been discussed and the results of tolerance design obtained using the proposed method are presented.

INTRODUCTION

It is practically impossible to manufacture a component precisely with the required dimensions. Therefore, all design dimensions have specified tolerances. Manufacturing a component with a narrow tolerance band is more expensive

This project is supported by the Earmarked Grant for Research of the Research Grants Council of Hong Kong, CUHK 7031-99E.

Address correspondence to Dr. Christopher C. Yang, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong SAR. E-mail: yang@se.cuhk.edu.hk

when compared to a wider tolerance band. This is because a narrow tolerance band requires better material, machine tools, control mechanisms, workman skills, more processing time, measuring instruments, and involvement of management. Consequently, this will cost more when compared to wider band manufacturing tolerances. Wider band tolerances will be cheaper, but there will be a greater number of rejections during quality checks, assembly, and problems during operation. Moreover, the tolerance of the assembly obtained from components with wider tolerances may not meet all the required functional specifications. It is desirable to maintain valid tolerances when product design is constrained by the relationship between the dimensions of entities of a component and the functional requirement of the design. In this paper, a method is designed for allocation of assembly-level functional tolerances to the component-level manufacturing tolerances such that the total cost of manufacturing of the assembly is minimized. The basic principles of hierarchical interval constraint networks have been used for tolerance analysis and subsequently for deriving the necessary constraints of the optimization problem and eliminating the redundant constraints. The objective function of the optimization problem is minimization of the total manufacturing cost of the assembly. This function is obtained from the manufacturing cost functions of each entity. Illustrative examples are given in support of the proposed algorithm.

Literature on Tolerance Design

The relationship between the functional requirements and entities of the mechanical part can be derived and expressed as $F_1 = f(E_1, E_2, \ldots, E_n)$. Tolerance design consists of tolerance analysis and tolerance synthesis. In tolerance analysis, the goal is to ensure the tolerance of functional requirement tolerances are met given the entity tolerances. If the assigned functional requirement tolerances are not met, the tolerances for the entities need to be reassigned by tolerance synthesis. Therefore, in tolerance synthesis, the goal is to determine a set of feasible entity tolerances to achieve the functional requirement tolerances. The task of tolerance synthesis is more complicated because large numbers of entity tolerances are determined, based on the functional requirement tolerances, and minimization of manufacturing costs are a concern during the synthesis. Tolerance design has been the focus of many techniques such as tolerance analysis and synthesis, worst-case analysis, statistical analysis, design optimization, and constraint-based reasoning. Many of these techniques are restricted to either analysis or synthesis; only a few are applicable to both analysis and synthesis. Most of them approximate a nonlinear relationship between tolerances as a linear relationship for simpler computation and optimization. With this approximation, some of the essential characteristics of the tolerance relationships are often lost. Some of the most recent developments are discussed below.

Chen (2001) has proposed a neural network-based tolerance propagation algorithm for finding cost-tolerance relationships. The author claims that this will represent a better relationship when compared to the relationships usually developed by regression analysis. He has then used a simulated annealing algorithm for solving the models. Choi et al. (2000) also have suggested a complex search method for solving the tolerance optimization problem. In their analysis, the objective function is a convex function, which is taken as the sum of cost of manufacturing and Taguchi's quadratic loss function. Ji et al. (2000) have proposed a method based on the second-order fuzzy comprehensive evaluation and the model is solved by genetic algorithm. They have considered a reciprocal cost function for relating manufacturing cost and tolerance and the quality loss function has not been taken into account. Feng and Kusiak (2000) have used statistical design of experiments to solve the tolerance allocation problem. The objective of their study is to minimize the variation of tolerance stack-ups. The Monte Carlo simulation approach has been utilized for experimental analysis. Cho et al. (2000) have also proposed a method for combined optimization for robust and tolerance design. An experimental *response surface methodology* has been used for robust design, instead of the usual practice of using orthogonal arrays, linear graphs, and signal-to-noise ratio. Jeang (1999) and Jeang and Leu (1999) have also used the response surface methodology for optimal tolerance design by considering quality loss and machining cost simultaneously. The response variable is the total cost function. They have used Monte Carlo simulation for generating experimental data necessary for the analysis. Jeang (2001) has also proposed another model to determine the optimal values of design tolerances, process mean, and process tolerances. Moskowitz et al. (1999) have proposed a minimax cost model to determine tolerance allocations. This model can be used when the only information available is mean and variance of each design parameter (and the distribution) is not known.

Lu and Wilhelm (1991) and Wilhelm and Lu (1992) proposed a tolerance synthesis approach, CASCADE-T, which used a representation of the conditional tolerance relations that exist between features of a part. Conditional tolerances are automatically determined from functional requirements and shape information. A constraint propagation network is employed for tolerance computation. However, the tolerances are propagated in a random order. This technique may find one solution that satisfies the constraints but does not guarantee finding a feasible solution. In addition, minimization of manufacturing cost is not considered. In this paper, the interval constraint network is employed to model the relationship between the tolerances of entities and functional requirements; the iterative relative sensitivity analysis procedure is used to minimize the manufacturing cost during the propagation.

Literature on Interval Constraints

Constraint satisfaction problems (CSPs) are often formulated in artificial intelligence (AI) tasks. In CSPs, values are assigned to variables subject to a set of constraints. Constraint specification represents the relationships among the variables. A constraint network is a declarative structure that consists of nodes and arcs. The nodes represent the variables or the constraints. The arcs represent the relationship between the variables and the constraints. The variables are labeled by intervals or sets of possible values. The constraints include any type of mathematical operation or binary relation. Constraint propagation is utilized to perform inferences about quantities. For different types of variables and definitions of satisfaction in constraint satisfaction problems, different propagation techniques can be formulated. A thorough review of CSPs can be found in Tsang (1993). For tolerance design, the variables are labeled by intervals and the constraints are n-ary mathematical operations.

The constraint satisfaction problem was first formulated and investigated by Huffman (1971) and Clowes (1971) to solve line-labeling problems in computer vision. It was then investigated by other researchers for more advanced searching algorithms and other applications. Dechter and Pearl (1989; 1988) developed a method of generating heuristic advice to guide the order of value assignments based on sparseness in the constraint network and the simplicity of tree-structured CSPs. A backtrack search algorithm is utilized to search for one or all solutions that assign a value to each variable, which satisfies all the constraints. Mackworth and Freduer (1985) analyzed the time complexity of several node, arc, and path consistency algorithms in CSPs. The domains of the variables considered by Dechter and Pearl and Mackworth and Freduer are discrete, finite sets instead of real intervals. Ladkin and Reinefeld (1992) developed a technique to solve qualitative interval constraint problems. The constraints are binary relations on intervals instead of n-ary mathematical operations on intervals. Davis and Hyvonen's work is most closely related to the present study. The constraints in their interval constraint satisfaction problems (ICSPs) are n-ary mathematical operations and the intervals are real-valued intervals. Davis (1987) adapted the Waltz filtering algorithm for screening impossible values from the variable domain to solve the ICSPs. However, the Waltz filtering algorithm cannot determine global solutions in general. Hyvonen (1992) used the tolerance propagation approach, which combines the consistency techniques based on the topology of the constraint net with techniques of interval arithmetic, to solve the ICSPs. Although ICSPs have been investigated, optimization has not been considered.

In CSPs, all solutions determined by different techniques are considered equally good. However, in applications such as industrial scheduling and tolerance design, some solutions are considered better than the others. Assignment of different values to the same variables may satisfy the constraints but they acquire different costs. The constraint satisfaction optimization problems (CSOP) are an extension of CSP where the optimal solution is desired. CSOP is defined as follows:

Definition 1

A constraint satisfaction optimization problem is a quadruple (V, D, C, F), where V is a set of variables

D is a function mapping every variable in V to a set of possible values,

D: $V \rightarrow$ *a set of possible values*

C is a set of constraints on an arbitrary subset of variables in V

F is an optimization function mapping every solution to a numerical value

F: $S \rightarrow$ *a numerical value, S is the set of solutions*

TOLERANCE DESIGN USING CONSTRAINT NETWORKS

A careful and critical analysis of the literature has revealed that the tolerance allocation problems are not yet completely solved satisfactorily using conventional methods. The methods and models suggested by various authors can only handle certain types of simple problems. Solution of complex tolerance design problems with many types of cost functions, assembly constraints, and large number of entities requires further research and development of better methods. The current effort in this work is to develop an algorithm, which can solve the tolerance design of any complexity. Recent studies have shown that constraint networks can be used to simplify the complex tolerance design problems.

Hierarchical interval constraint networks can be used to represent the relationship between the highest (assembly) level functional requirements and the lowest level entities of any assembly (Yang et al. 1997). The functional requirements are decided by considering various factors, including customer feedback and market demand. Each functional requirement of the assembly can be represented as a mathematical expression (function) of a number of attributes. For example, volume of a cylindrical vessel is a function of the inner radius and the effective inner height of the vessel. In this case, volume is the functional requirement, whereas inner radius and height are the attributes. Attributes can also be described as functions of the mechanical dimensions of the associated parts entities. For example, the inner radius of a cylinder can be expressed as a function of the outer radius and the radial thickness. Similarly, the total height, base wall thickness, and the cylinder leg height are entities of the attribute inner height. These relationships can

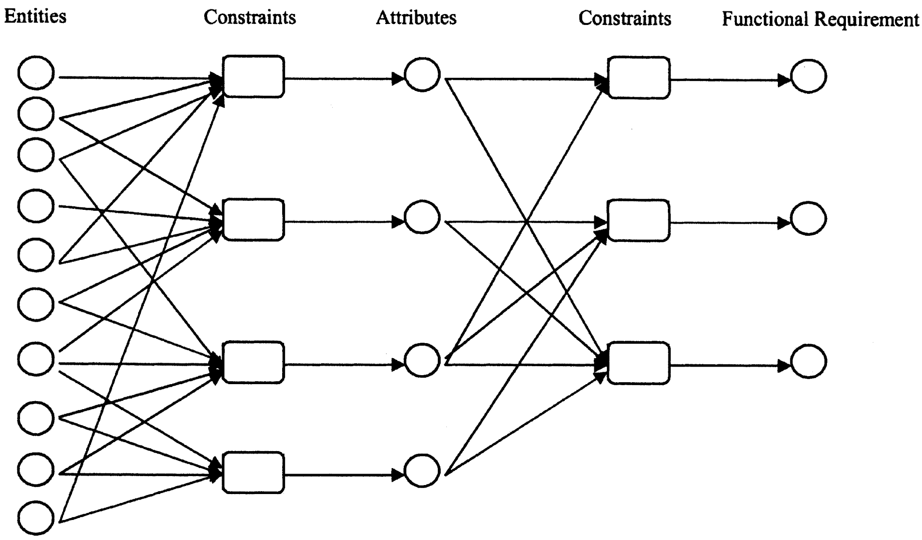

FIGURE 1. Representation of an assembly by an interval constraint network. (Direction of arrows indicate forward propagation. Opposite direction is backward propagation.)

be represented in a constraint network for tolerance analysis and synthesis. A contraint network is a diagram in which functional requirements, attributes, and entities are represented in circles called nodes of the diagram. The relationships between entities and attributes and between attributes and functional requirements are represented by rectangles. Arcs are used to connect nodes through the rectangles (constraints). Figure 1 illustrates an interval constraint network (ICN) designed for tolerance design.

The constraint functions represent the relationships between functional requirements and attributes and between attributes and entities. An interval constraint function is derived based on interval arithmetic. The interval arithmetic is an extension of the real arithmetic and it deals with close intervals $X = [x_{low}, x_{up}]$, representing $\{x|\ x_{low} \leq x \leq x_{up}\}$. Given an arithmetic operation, $\oplus$, $Z = X \oplus Y$ is defined as:

$$\begin{aligned} Z &= [z_{low}, z_{up}] = X \oplus Y = [x_{low}, x_{up}] \oplus [y_{low}, y_{up}] \\ &= \{x \oplus y | x_{low} \leq x \leq x_{up}, y_{low} \leq y \leq y_{up}\} \end{aligned}$$

Consistency of ICN for Tolerance Design

The satisfaction of a constraint networks can always be described in terms of consistency of variables or constraints. The definition of consistency depends on the applications of the constraint network. Hyvonen (1992) defined the satisfaction of interval constraint network in terms of the consistency of variables. The purpose of such definition is to refine the intervals of the variables in the interval constraint network as far as possible without losing possible exact solutions of the constraints. For tolerance design, it is desirable to have the computed interval using the input intervals and the interval constraint function to be a subset of the assigned output interval. For example, the tolerances propagated from the attributes to a functional requirement through the corresponding interval constraint function must be a subset of the tolerance of the functional requirement. $(f_k(A_i, \ldots A_{j,}, \ldots, A_n) \subseteq F_k)$. Therefore, the satisfaction of the interval constraint network for tolerance design is defined in terms of the constraint consistency.

The interval constraint function in ICN for tolerance design consists of multiple inputs and a single output function and is represented as a triple $C_i(U, k, f_i())$. U is the set of indexes for the input variables and k is the index of the output variable for the constraint $C_I, f_i()$ is the constraint function.

The definition of satisfaction of interval constraint network for tolerance design is as follows:

Definition 2
A constraint $C_i(U, k, f_i())$ is consistent if and only if $\bigcap_{j\in U} (\forall v_j \in V_j | V_j = v_j)$, $(\exists v_k | V_k = v_k)$ such that $C_i(U, k, f_i())$ is satisfied.

Definition 3
The ICN for tolerance design is satisfied if and only if all the constraints are consistent.

These definitions for consistency of constraints satisfaction of the ICN ensure that the tolerances assigned to the entities satisfy the requirements of the functional tolerance of the assembly. In other words, the computed tolerance of functional requirements based on the input tolerances and consistent constraints will be a subset of the actual functional tolerance requirements.

Forward and Backward Propagation of Tolerance

Tolerance propagation is carried out to update the intervals (tolerances) in the network to make the interval constraints consistent. There are two types of tolerance propagations: forward propagation (FP) and backward propagation (BP). In FP, tolerances of input variables are propagated through the constraints to obtain the tolerances of the output variables. FP will detect the consistency of various constraints in the network. On the other hand, in BP, tolerance of an output variable is propagated through the constraint expressions to distribute it among the multiple input variables (such that the constraints are made consistent); the minimization of the manufacturing costs is also considered in the propagation.

In this paper, FP is carried out to verify whether any of the functional tolerances of the assembly is/are already satisfied by the natural tolerances (readily available and most economical tolerance under the present process conditions) of the entities, i.e., *tolerance analysis*. This will provide a choice for either eliminating the already satisfied functional tolerance requirement and the corresponding entities (which do not affect other functional requirements) from further backward propagation or retaining them for BP. The first choice will reduce the dimension of the model and therefore will be easier to solve. It is also logically correct to accept the first option because natural tolerance is the most economical tolerance for any entity. (It is recommended that exercising the first choice be done carefully and only if it is appropriate to the process setup.) The FP is carried out with numerical values of the nominal and the natural tolerances of the entities. In BP, the tolerances of functional requirements are algebraically propagated through the constraint functions to derive the algebraic functions of tolerances. These functions will relate the increment of tolerances of assembly functions with that of the entities.

Tolerance Analysis by Forward Propagation

Forward propagation (FP), as described in Appendix A, propagates the numerical tolerances of entities through attributes to functions to determine if any of the constraints are consistent (satisfied). Entities, attributes, and functions for the satisfied constraint can be removed from further modeling. Those variables **affecting** the satisfied assembly function and **not affecting** the other functions may **only** be removed.

Algorithm for tolerance analysis by forward propagation is below:

SetF = $\{F_i,\ i = 1..n\}$	*/define SetF as a Set of all assembly functions/*
Call FP, Get $F_{FP} = \{F_{\lfloor nom\text{-}pp}, F_{\lfloor up\text{-}pp}, F_{\lfloor low\text{-}pp}, i = 1..n\}$	*/F_{PP} is the propagated values of F/*
Read $F_{req} = \{F_{\lfloor nom\text{-}req}, F_{\lfloor up\text{-}req}, F_{\lfloor low\text{-}req}, i = 1..n\}$	*/F_{req} is the required values of F/*
For $i = 1..n$ Do{	
IF $(F_{\lfloor nom\text{-}pp} = F_{\lfloor nom\text{-}req})$ AND $(F_{\lfloor up\text{-}pp} \leq F_{\lfloor up\text{-}req})$ AND $(F_{\lfloor low\text{-}pp} \geq F_{\lfloor low\text{-}req})$	
THEN $g = i$, SetF = $\{F_i,\ i = 1..n,\ i \neq g\}$	*/If C_i is satisfied, remove F_i from SetF to update it/*
}	
SetA = $\{A_j,\ j = 1..r\}$	*/define SetA as a Set of all attributes/*
For $j = 1..r$ Do {	
IF $F_g = f(A_j)$ AND $(F_i,\ i = 1..n,\ i \neq g) \neq f(A_j)$	*/if A_j affects only F_g, remove it from SetA/*
THEN $h = j$, SetA = $\{A_j,\ j = 1..r,\ j \neq h\}$	
}	
SetE = $\{E_k,\ k = 1..m\}$	*/Define SetE as a Set of all entities/*
For $k = 1..m$ Do {	
IF $A_L = f(E_k)$ AND $(A_j,\ j = 1..r,\ j \neq h) \neq f(E_k)$	*/if E_k affects only A_h, remove it from SetE/*
THEN $s = j$, SetE = $\{E_k,\ k = 1..m,\ k \neq s\}$	
}	
Write (SetF, SetA, Set E)	*/Remaining sets of functions, attributes, and entities/*

Propagate the Tolerance of Functional Requirements Backward Algebraically

After removing the entities, attributes, and functional requirements of the satisfied constraints, the algebraic relationships between the remaining entities and the remaining functional requirements are determined by backward propagation (BP).

Step 1. Propagate the algebraic tolerances of functional requirements to attributes for all assembly functions or only for those remaining after elimination in tolerance analysis. We get a system of algebraic expressions, each representing the incremental tolerance of assembly function in terms of the incremental tolerances of attributes.

For $i = 1..n, i \neq g$ Do
$F_i = f(A_j)$ /*Constraint C_i in terms of F_i and related A_js*/
$\Delta F_i = f(\Delta A_j)$ /*Taking the first order increment F_i*/

Step 2. Carry out the second level of BP (attributes to entities) for all attributes. We get another set of algebraic functions, each representing the incremental tolerance of attributes in terms of the incremental tolerances of entities.

For $j = 1..r, j \neq h$ Do
$A_j = f(E_k)$ /*Constraint C_j in terms of A_j, and related E_ks*/
$\Delta A_j = f(\Delta E_k)$ /*Taking the first order increment A_j*/

Step 3. Substitute the system of equations obtained in Step 2 into those in Step 1. We get a new system of equations, each representing the incremental tolerance of assembly functions in terms of the incremental tolerances of entities. Substitute the numerical value of the maximum required upper tolerance of all assembly level functions. (Observe the rule that if any positive increment of E_k results in monotonic decrease of F_i, then ΔE_k should be multiplied by (-1) to correctly represent in the maximum upper tolerance expression.)

For $i = 1..n, i \neq g$ Do
Read(ΔF_{iup})
$\Delta F_1 = f(\Delta E_k)$ /*Substituting ΔA_j obtained in Step 3 into Step 2*/
$f_1(\Delta E_k) = \Delta F_{iup}$ /*Substituting the numerical value of the required maximum positive tolerance of F_i*/

Step 4. Identify the redundant equation in the system of equations obtained in Step 3 and eliminate them. The remaining equations are the constraints of the optimization problem. If dth expression is redundant, update the constraint equations in Step 3 by eliminating it from the system of equations. Repeat this for other redundancies. The remaining equations are:

$$F_1(\Delta E_k) = \Delta F_{iup} \quad i = 1..n; j \neq d; i \neq g, \text{for all 'd' s and 'g' s}$$

The actual number of constraints to the optimizing problem equals the number of original assembly-level nodes (functions) minus the number of assembly functions satisfied by FP and the number of constraints eliminated due to redundancy.

Cost of Manufacturing and Tolerances

As discussed earlier, the cost of manufacturing increases when the tolerance bands are reduced. Several models for relating cost of manufacturing with required tolerance, expressed as functions of the tolerances, have been suggested in the literature. These include a linear model, a reciprocal model (cost of manufacturing is proportional to reciprocal of tolerance) (Chase and Greenwood 1988); a reciprocal-squared (Spotts 1973); a reciprocal-powered (Sutherland and Roth 1975); an exponential (Speckhart 1972); and a combined exponential-reciprocal powered (Michael and Siddali 1981; 1982), among others. While all these models are empirical and based on experiences, it has been observed that the reciprocal-squared and the exponential models are more frequently used than other models. Chen (2001) has proposed an algorithm using neural networks for deriving the cost-tolerance relationship. The proposed algorithm in this paper can handle any form of cost model. In the illustrative example, an exponential cost model has been used for the cost function. The general form of exponential cost model is as given below:

Exponential cost tolerance relation is:

$$CM(\Delta E_k) = a + b^* \exp(-c\Delta E_k)$$

where ΔE_k corresponds to the tolerance of kth part (or entity), a, b, c are constants for a part (estimates from cost-tolerance data), and $CM(\Delta E_k)$ corresponds to the cost of manufacturing of kth entity with tolerance ΔE_k.

The objective function of the optimization problem is the total manufacturing cost function of the assembly. This is expressed as the sum of the cost functions of the individual entities. If any entity is eliminated in tolerance analysis, the cost function for that entity will not be included in the objective function. Moreover, if any particular entity is used more than once in an assembly, its cost function will be multiplied by the number of times it is used in the assembly. For example, suppose there are 4 similar drill holes in an assembly. The cost function of the drill holes should be multiplied by 4 while evaluating the total cost function of the assembly. All these cost functions will be in terms of the tolerances of individual entities. Mathematically, the objective function can be expressed as follows:

$$CM_{ASS} = \sum_{k=1,\ k\neq 0}^{m} N_k(CM_{E_k})$$

where CM_{ASS} equals the cost of manufacturing the assembly for required tolerances, CM_{E_k} equals the cost of manufacturing the kth entity, expressed as a function of its tolerance, and N_k equals the number of components with kth entity used in the assembly.

THE RELATIVE SENSITIVITY RATIO ALGORITHM

An efficient algorithm, which has been developed based on the principles of hierarchical interval constraint networks, is proposed in this paper for optimum tolerance design. The constraints and the objective functions of the assembly are derived by the procedure explained in the previous subsections. This will provide the mathematical models to represent the tolerance design problem. The procedure for solution of the rexponential problem is iterative, based on relative sensitivity ratio (RSR). RSR is defined as follows:

$$\text{relative sensitivity ratio} = \frac{\begin{array}{c}\text{sensitivity of cost of manufacturing}\\ \text{with respect to entity tolerance}\end{array}}{\begin{array}{c}\text{sensitivity of functional tolerance}\\ \text{with respect to entity tolerance}\end{array}}$$

Mathematically, relative sensitivity ratio for the ith assembly constraint function with respect to the kth entity, *RSR* (*i,k*), is derived as:

$$RSR(i,k) = \frac{\frac{\partial(CM_{ASS})}{\partial E_k}}{\frac{\partial(F_t)}{\partial E_k}}$$

where CM_{ASS} equals the cost of manufacturing of the assembly and

$$= \sum_{k=1}^{m} CM(\Delta E_k).$$

In this algorithm, all the entities are initially assumed to have tolerances equal to a very small number (say 0.0001 units). Then the RSR value is evaluated for each entity and the minimum value is found. The tolerance of that particular entity, having the minimum RSR value, is now increased by one small step (say 0.0001 units) and the functional tolerance requirements are tested. If the constraints are not satisfied, the procedure is repeated until all constraints are satisfied (within required accuracy). At any stage, if any one constraint is satisfied, it will be removed from the list of constraints and the present tolerance of entities of this constraint are stored as their optimum values. These entities may appear in other remaining constraints, but their tolerances will not change from these values. In each iteration, the tolerances are propagated through all unsatisfied constraint functions and entities. Selection of an entity for increasing its tolerance during the iteration is based on the RSR value. This is based on the logic that increasing tolerance of this particular entity should result in the highest decrease of overall cost of manufacturing and lowest increase of the tolerance of assembly-level functions. Details of the algorithm are discussed in the following sections.

The RSR Algorithm

Step 1. Define the accuracy required in solution.
(For example, the accuracy required for each entity tolerance is 0.0001 units.) accuracy = 0.0001

Step 2. Set the initial tolerance of each entity equal to the accuracy required.
For (k = 1..m), substitute ΔE_k = accuracy

Step 3. Evaluate RSR for all constraint functions with respect to all entities.
For (i = 1..n)
For (k = 1..m)

$$RSR(i,k) = \frac{\frac{\partial(CM_{ASS})}{\partial E_k}}{\frac{\partial(F_i)}{\partial E_k}}$$

Step 4. Evaluate the constraint functions for the present values of entity tolerances and evaluate the error.

For(i = 1 . . . n)
CF(i) = f(ΔE_k) */CF(i) is the propagated value of ΔF_i/*
Error(i) = CF(i) − f(i) */f(i) is the required tolerance of F_i/*

Step 5. (A) If any of the constraints are satisfied (or the error is within acceptable limits), remove this constraint from further consideration. The present values of the tolerances of those entities included in this constraint are stored as their optimum values. These entities may appear in other constraints, but their tolerances are now fixed.

For i = 1..n
If ((|Error(i)| <= ε)and(E_k is included in CF(i))) */ε is the maximum allowable error/*
$Eopt_{(k)} = \Delta E_k$ */$Eopt_{(k)}$ is the optimal ΔE_k/*
p(i) = p(i) + i; */p(i) is the list storing all the satisfied F_i/*
q(k) = q(k) + k */q(k) is the list storing all the satisfied E_k/*

(B) If all the constraints are satisfied or if all the entity tolerances are stored as optimum, stop the iteration. The present values of entity tolerances are the optimum.
If (p(i) = i) for all i, stop.

(C) For those unsatisfied constraints do the following:

(i) Evaluate RSR for unsatisfied constraints
For (i = 1..n)
For (k = 1..m)
If ((i $\neq$ p(i)) and (k $\neq$ q(k)))

$$RSR(i,k) = \frac{\frac{\partial(CM_{ASS})}{\partial E_k}}{\frac{\partial(F_i)}{\partial E_k}}$$

(ii) Find out the particular constraint and the particular entity (after eliminating the satisfied constraints and entities) for which the RSR is minimum.

For (i = 1..n)
For (k = 1..m)
If ((i $\neq$ p(i)) and (k $\neq$ q(k))
and RSR(i,k) < RSR_{min})
RSR_{min} = RSR(i,k)
imin, kmin: indices of i
and k corresponding to RSR_{min}

(iii) Increase the tolerance of this entity by adding a value equal to accuracy to its present value. It will result in maximum reduction in cost and minimum increase in functional tolerances.

$$\Delta E_{kmin} = \Delta E_k + \text{accuracy}$$

(iv) Evaluate the error for unsatisfied constraints. Go back to 5(A)

Error(imin) = CF(imin) − f(imin)
Go to 5(A)

Advantages of the Prosposed RSR Algorithm

The proposed algorithm has many advantages over the existing models for tolerance design. The existing models do not guarantee optimum solution to all types of tolerance design problems, especially when the problem is complex. Most of these models can handle only specific cases of tolerance design problems, that is, those that are problem specific. There are restrictions on the types of cost functions, number of assembly functions, and entities to use these models. These models do not ensure any level of required accuracy in the final solution.

The proposed algorithm in this paper is a general-purpose optimization model for solving tolerance design problems for assemblies. It can handle any complex problem (with any number of entities, attributes, and assembly function requirements) and any type of cost-function relating cost of manufacturing with entity tolerance. It can also solve nonlinear problems with a combination of different cost functions for entities. Required level of accuracy or the permitted error can be stated at the beginning of the program and the algorithm assures any level of accuracy in the final solution. The proposed algorithm is very powerful, but simple and easy to program.

Many existing models define the objective functions as a sum of cost of manufacturing tolerances and the Taguchi's loss function. Obviously, this function will be a convex function having a minimum value for a certain combination of entities' tolerances. However, this combination of entities' tolerances will not satisfy the assembly-level function requirements. That means such models only find a combination of entities' tolerances to minimize the total cost without satisfying constraints.

In the proposed model, the objective function is defined as a sum of cost functions for all entities. The Taguchi's loss function is not included in the objective function. We know that the tolerance requirements of assembly functions are the constraints of the problem. For an assembly, the Taguchi's loss function is a result of these constraints. These are decided by the design engineer, so there is no need to include this in the objective function. In other words, the assembly function tolerance requirements restrict the Taguchi's loss function for an assembly, Therefore, it can be concluded that the proposed model generates an optimum combination of entities' tolerances, which will minimize the cost of manufacturing tolerance of the assembly and will satisfy all the assembly function tolerance requirements.

EXAMPLES

In this section, the tank problem and the movable double-bearing assembly are used as examples to illustrate the relative sensitivity ratio algorithm for interval constraint networks in the application of tolerance analysis and synthesis.

Example 1: Tank Problem

The tank as illustrated in Figure 2 is made of two cylinders. The functional requirements are the volume and the thickness as labeled in

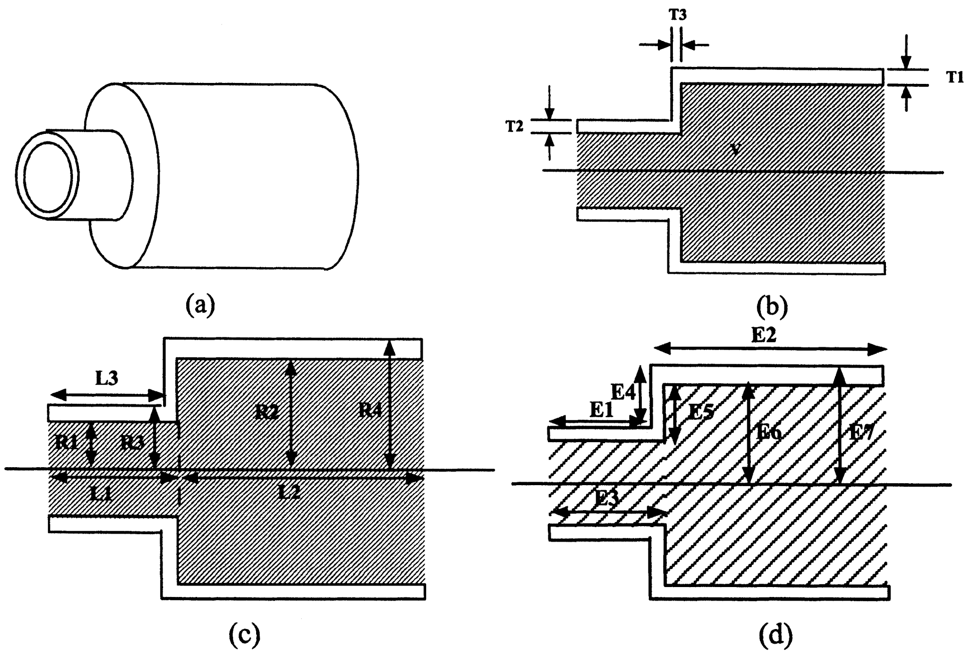

FIGURE 2. (a) A tank, (b) the labels of functional requirement, (c) the labels of attributes of the cylinders, and (d) the labels of the measurable entities.

Figure 2(b). The attributes are the inner, outer radius, inner length, and outer lengths as labeled in Figure 2(c). The entities are several measurable lengths and radius as shown in Figure 2(d). The corresponding interval constraint network is given in Figure 3 and the constraint functions are given in Table 1.

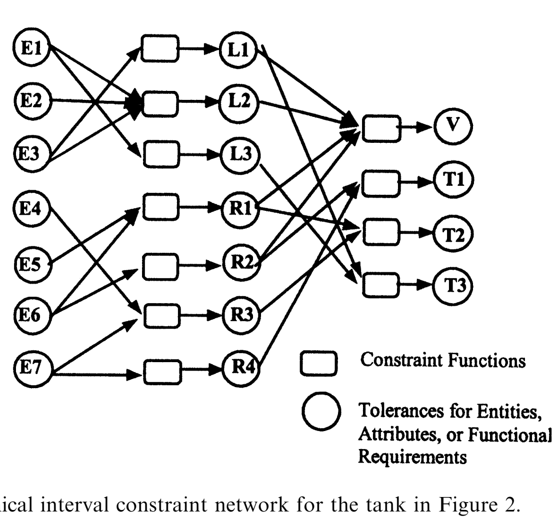

FIGURE 3. Hierarchical interval constraint network for the tank in Figure 2.

TABLE 1 Constraint Functions for the Interval Constraint Network in Figure 3

Constraint functions between entities and attributes	Constraint functions between attributes and functional requirements
L1 = E3	$V = \pi \times R1 \times R1 \times L1 + \pi \times R2 \times R2 \times L2$
L2 = E1 + E2 − E3	T1 = R4 − R2
L3 = E1	T2 = R3 − R1
R1 = E6 − E5	T1 = LI − L3
R2 = E6	
R3 = E7 − E4	
R4 = E7	

Forward Propagation

Table 2 shows the initial assignment of tolerances and the result of forward propagations for tolerance analysis. Using forward propagation, it has been found that the tolerance of the functional requirement of volume is

TABLE 2 The Tolerances of the Entities, Attributes, and Functional Requirements, initially, after Forward Propagation

		Natural tolerances	Forward propagation	Constraint satisfaction
Entities	E1	[94 mm, 96 mm]	[94 mm, 96 mm]	
	E2	[204 mm, 206 mm]	[204 mm, 206 mm]	*
	E3	[99 mm, 101 mm]	[99 mm, 101 mm]	
	E4	[49 mm, 51 mm]	[49 mm, 51 mm]	
	E5	[49 mm, 51 mm]	[49 mm, 51 mm]	
	E6	[189 mm, 191 mm]	[189 mm, 191 mm]	
	E7	[199 mm, 201 mm]	[199 mm, 201 mm]	
Attributes	L1	[100 mm, 100 mm]	[99 mm, 101 mm]	
	L2	[200 mm, 200 mm]	[197 mm, 203 mm]	**
	L3	[95 mm, 95 mm]	[94 mm, 96 mm]	
	R1	[140 mm, 140 mm]	[138 mm, 142 mm]	
	R2	[190 mm, 190 mm]	[189 mm, 191 mm]	
	R3	[150 mm, 150 mm]	[148 mm, 152 mm]	
	R4	[200 mm, 200 mm]	[199 mm, 201 mm]	
Functional Requirements	V	$[2.8 \times 10^7 mm^3, 3.0 \times 10^7 mm^3]$	$[2.8 \times 10^7 mm^3, 3.0 \times 10^7 mm^3]$	***Satisfied
	T1	[9 mm, 11 mm]	[8 mm, 12 mm]	Not Satisfied (BP is Required)
	T2	[9 mm, 11 mm]	[6 mm, 14 mm]	Not Satisfied (BP is Required)
	T3	[4.5 mm, 5.5 mm]	[3 mm, 7 mm]	Not Satisfied (BP is Required)

***V is satisfied with the present level of tolerances.
**L_2 is the only attribute affecting V, but not others. So L_2 is not considered for BP.
*E_2 is the only entity affecting L_2, but not others. The present tolerances of E_2 are its natural tolerances and therefore the most economic. Therefore, E_2 is not considered for BP.

satisfied, but the tolerances of three other functional requirements, thickness, are not satisfied.

Backward Propagation

Referring to Table 1, the following constraint equations relate atributes and functions:

$$T_1 = R_4 - R_2$$
$$T_2 = R_3 - R_1$$
$$T_3 = L_1 - L_3$$

To propagate the tolerances backward, we have to derive the tolerance relationship between functions and attributes. For this purpose, we take first-order derivatives (increments) of the corresponding constraint equations. We get the following tolerance equations:

$$\Delta T_1 = \Delta R_4 - \Delta R_2$$
$$\Delta T_2 = \Delta R_3 - \Delta R_1$$
$$\Delta T_3 = \Delta L_1 - \Delta L_3$$

The following constraint equations are retrieved from Table 1 to relate entities with attributes:

$$L_1 = E_3$$
$$L_3 = E_1$$
$$R_1 = E_6 - E_5$$
$$R_2 = E_6$$
$$R_3 = E_7 - E_4$$
$$R_4 = E_7$$

Similarly, we get the following tolerance equations:

$$\Delta L_1 = \Delta E_3$$
$$\Delta L_3 = \Delta E_1$$
$$\Delta R_1 = \Delta E_6 - \Delta E_5$$
$$\Delta R_2 = \Delta E_6$$
$$\Delta R_3 = \Delta E_7 - \Delta E_4$$
$$\Delta R_4 = \Delta E_7$$

Substituting the tolerance function of the attributes to the tolerance function of functional requirements, we obtain:

$$\Delta T_1 = \Delta E_7 - \Delta E_6$$
$$\Delta T_2 = \Delta E_7 - \Delta E_4 - \Delta E_6 + \Delta E_5$$
$$\Delta T_3 = \Delta E_3 - \Delta E_1$$

Referring to the functional requirements specified earlier, we get the required maximum tolerances of T_1, T_2, and T_3 as follows:

$$\Delta T_1 = T_{1up} - T_{1nom} = 1.0\,mm$$
$$\Delta T_2 = T_{2up} - T_{2nom} = 1.0\,mm$$
$$\Delta T_3 = T_{3up} - T_{3nom} = 0.5\,mm$$

Substituting the maximum allowable tolerance values of the functions and removing the redundant equations, we get the following constraint for the optimization problem:

$$\Delta E_4 + \Delta E_5 + \Delta E_6 + \Delta E_7 = 1.0\,mm$$
$$\Delta E_1 + \Delta E_3 = 0.5\,mm$$

Manufacturing Cost Functions

As mentioned earlier our algorithm is able to handle any kind of cost model. For this particular example, we adopt the cost functions with reciprocal square relationships for illustration purposes. That is, the cost of manufacturing is inversely proportional to square of the entity tolerances.

$$CM(\Delta E_k) = c + \frac{d}{(\Delta E_k)^2}$$

where c and d are constants for a given entity and ΔE_k is the tolerance of k^{th} entity.

For this example, the constants, c and d for entities E_1 to E_7 are given in Table 3.

Total cost of manufacturing is:

$$CM_{ASS} = 497 + \frac{10}{(\Delta E_1)^2} + \frac{15}{(\Delta E_3)^2} + \frac{16}{(\Delta E_4)^2} + \frac{18}{(\Delta E_5)^2} + \frac{20}{(\Delta E_6)^2} + \frac{10}{(\Delta E_7)^2}$$

The constraint satisfaction optimization problem is therefore:

$$\text{Minimize } CM_{ASS} = 497 + \frac{10}{(\Delta E_1)^2} + \frac{15}{(\Delta E_3)^2} + \frac{16}{(\Delta E_4)^2} + \frac{18}{(\Delta E_5)^2} + \frac{20}{(\Delta E_6)^2} + \frac{10}{(\Delta E_7)^2}$$

TABLE 3 Values of Constants for the Manufacturing Cost Functions of E_1 to E_7

	E_1	E_2	E_3	E_4	E_5	E_6	E_7
c	58	112	87	43	67	75	55
d	10	12	15	16	18	20	10

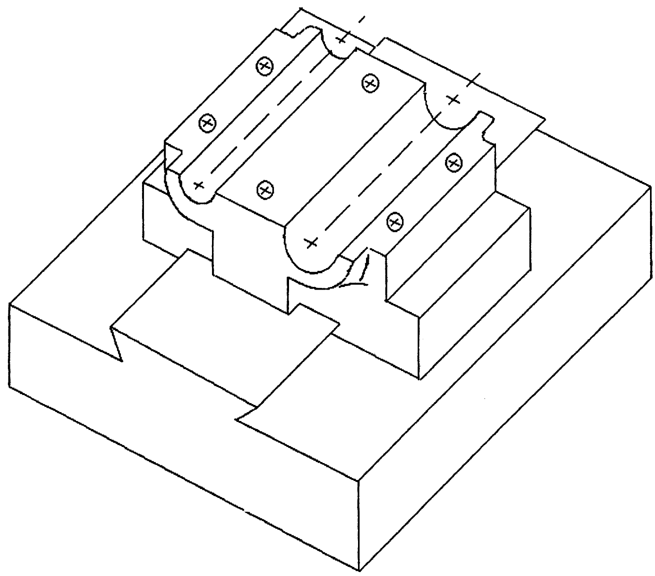

FIGURE 4. Movable double-bearing assembly (shows the sliding motion of bearing base on V-rail).

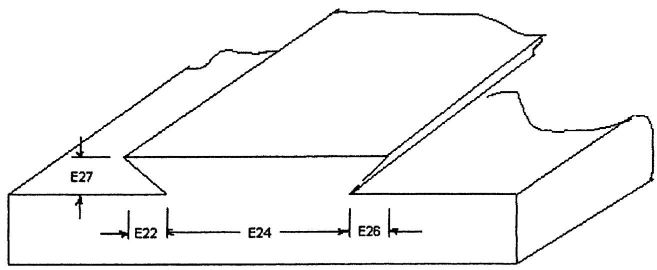

FIGURE 5. V-guide rail for sliding the bearing base.

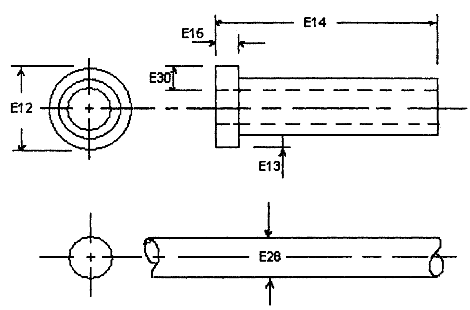

FIGURE 6. Bush 1 and Shaft 1. Bush 2 and Shaft 2 are similar to these. The dimensions of Bush 2 are E16, E17, E18, E19 and E31 The diameter of Shaft 2 is E29.

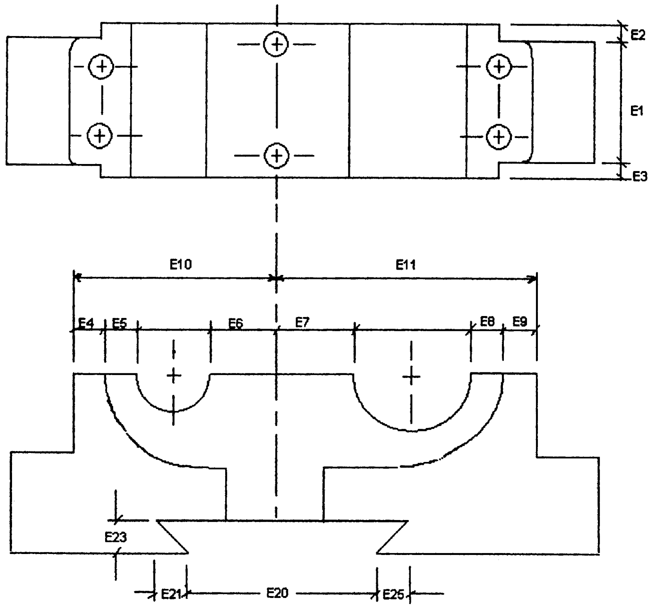

FIGURE 7. Bearing holder base.

$$\text{subject to:}\quad \Delta E_4 + \Delta E_5 + \Delta E_6 + \Delta E_7 \leq 1.0 \text{ mm}$$
$$\Delta E_1 + \Delta E_3 \leq 0.5 \text{ mm}$$

Solution using RSR Algorithm

The optimization problem is solved by the proposed RSR algorithm also. The starting tolerances for all the entities were equal to 0.0001 units and the required accuracy was assumed equal to 0.0001 for each constraint. The following is the results:

$\Delta E_1 = 0.2332$ $\Delta E_2 = 1.0000$ $\Delta E_3 = 0.2669$
$\Delta E_4 = 0.2517$ $\Delta E_5 = 0.2618$ $\Delta E_6 = 0.2712$
$\Delta E_7 = 0.2153$

Example 2: Movable Double-Bearing Assembly

The main parts of the movable double-bearing assembly are base rail, bearing base, bush bearings, and shafts. The bearing base can hold two parallel bushes and shafts. It can slide on the base rail groove, parallel to the axis of the shafts. This is a much more complicated problem compared to the tank problem discussed previously. There are 9 functional requirements, 17 attributes, and 31 entities in this assembly. Figure 4, 5, 6, and 7 show various

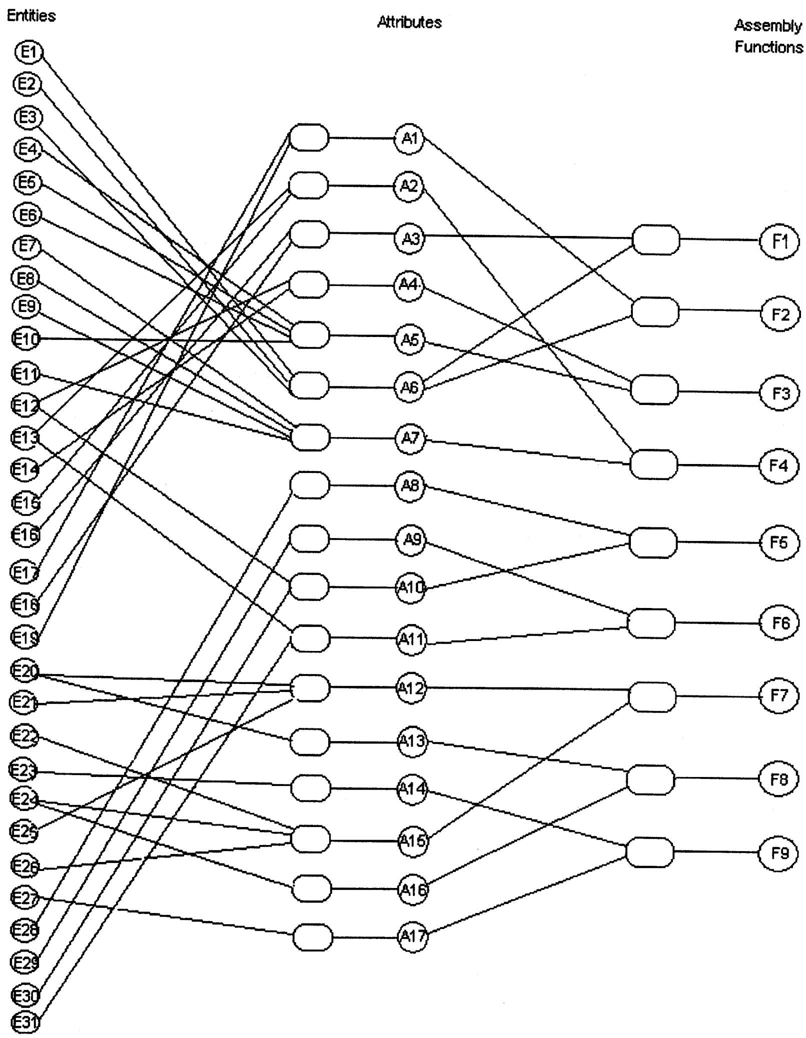

FIGURE 8. Hierarchical interval constraint network for movable double-bearing assembly.

parts and entities. Figure 8 shows the interval constraint network of the movable double-bearing assembly problem.

Table 4 shows the various assembly functions required for smooth operation of the movable double-bearing assembly. The notations used for assembly functions are also shown in this table. Table 5 shows the various attributes of different parts of the assembly with their notations. The list of entities is shown in Table 6. There are 9 assembly functions, 17 attributes, and 31 entities in this problem. The constraint functions between entities and attributes and those between attributes and assembly functions are shown in Table 7. The nominal dimensions and their required tolerances are specified

TABLE 4 Assembly Functions

Sl No.	Description	Function required	Notation
1	Relative Length of Base & Bush 1	Minimum	F_1
2	Relative Length of Base & Bush 2	Minimum	F_2
3	Fit Between Base & Bush 1	Interference Locational Fit	F_3
4	Fit Between Base & Bush 2	Interference Locational Fit	F_4
5	Fit Between Bush 1 & Shaft 1	Clearance Locational Fit	F_5
6	Fit Between Bush 2 & Shaft 2	Clearance Locational Fit	F_6
7	Upper Rail Fit	Running & Sliding Fit	F_7
8	Lower Rail Fit	Running & Sliding Fit	F_8
9	Height Difference Between Base and Support Rails	Minimum	F_9

TABLE 5 Attributes

Sl. No.	Part	Description	Notation
1	Base	Inner Diameter of Hole 1	A_5
2		Inner Diameter of Hole 2	A_7
3		Width of Base	A_6
4		Upper Width of Rail	A_{12}
5		Lower Width of Rail	A_{13}
6		Height of Rail	A_{14}
7	Bush 1	Length of Bush 1	A_3
8		Outer Diameter of Bush 1	A_4
9		Inner Diameter of Bush 1	A_{10}
10	Bush 2	Length of Bush 2	A_1
11		Outer Diameter of Bush 2	A_2
12		Inner Diameter of Bush 2	A_{11}
13	Support Rail	Upper Width of Rail	A_{15}
14		Lower Width of Rail	A_{16}
15		Height of Rail	A_{17}
16	Shaft 1	Diameter of Shaft 1	A_8
17	Shaft 2	Diameter of Shaft 2	A_9

in Table 8. These are the numerical values of functional requirements. The nominal dimensions and their initial tolerances of all entities are specified in Table 9. Table 9 also shows the values of a, b, and c for the exponential cost-tolerance equations.

Forward Propagation

The nominal values and initial tolerances of the entities are propagated forward through the first stage of the constraint network to the attributes. Thus, we get the nominal values and tolerances of all the attributes. These are shown in Table 10. Thereafter, the nominal values and tolerances of attributes obtained in the first stage of forward propagation are propagated to the assembly functions through the second stage of the network. These are shown in Table 11.

TABLE 6 List of Entities

Parts	Entities
Base:	$E_1,E_2,E_3,E_4,E_5,E_6,E_7,E_8,E_9,E_{10},E_{11},E_{20},E_{21},E_{22},E_{23}$
Bush 1:	$E_{12},E_{13},E_{14},E_{15}$
Bush 2:	$E_{16},E_{17},E_{18},E_{19}$
Support Rail:	$E_{24},E_{25},E_{26},E_{27}$
Shaft 1:	E_{28}
Shaft 2:	E_{29}

TABLE 7 Constraint Functions

Constraint functions between entities and attributes	Constraint functions between attributes and functional requirements	
$F_1 = A_6 - A_3$	$A_1 = E_{17} - E_{19}$;	$A_2 = E_{13} - 2\ E_{15}$
$F_2 = A_6 - A_1$	$A_3 = E_{16} - E_{18}$;	$A_4 = E_{12} - 2\ E_{14}$
$F_3 = A_4 - A_5$	$A_5 = E_{10} - E_4 - E_5 - E_6$;	$A_6 = E_1 + E_2 + E_3$
$F_4 = A_2 - A_7$	$A_7 = E_{11} - E_7 - E_8 - E_9$;	$A_8 = E_{28}$
$F_5 = A_{10} - A_8$	$A_9 = E_{29}$;	$A_{10} = E_{12} - 2\ E_{30}$
$F_6 = A_{11} - A_9$	$A_{11} = E_{13} - 2\ E_{31}$;	$A_{12} = E_{20} + E_{21} + E_{25}$
$F_7 = A_{12} - A_{15}$	$A_{13} = E_{20}$;	$A_{14} = E_{23}$
$F_8 = A_{13} - A_{16}$	$A_{15} = E_{22} + E_{24} + E_{26}$;	$A_{16} = E_{24}$
$F_9 = A_{14} - A_{17}$	$A_{17} = E_{27}$	

TABLE 8 Functional Requirements

Function	Notation	Nominal	Minimum	Maximum
Relative Length of Base & Bush 1	F_1	0.0000	−0.0065	0.0065
Relative Length of Base & Bush 2	F_2	0.0000	−0.0065	0.0065
Interference Between Base & Bush 1	F_3	0.0056	0.0035	0.0077
Interference Between Base & Bush 2	F_4	0.0056	0.0035	0.0077
Clearance Between Bush 1 & Shaft 1	F_5	0.0025	0.0021	0.0029
Clearance Between Bush 2 & Shaft 2	F_6	0.0025	0.0021	0.0029
Clearance of Upper Rails	F_7	0.0032	0.0030	0.0034
Clearance of Lower Rails	F_8	0.0018	0.0015	0.0021
Height Difference Between Base and Support Rails	F_9	0.0007	0.0005	0.0009

Comparison of Table 8 with Table 11 shows that none of the constraints is satisfied with the present level of tolerance of entities. Therefore, all the constraints and entities are considered for backward propagation.

Backward Propagation

The first stage of backward propagation is done from assembly functions to the attributes. The constraint functions for this are shown in Table 7. Equations for first stage BP can be obtained by taking the first-order

TABLE 9 Dimensions of Entities and Cost Functions

	Cost-Tolerance relation (exponential)					
Entity	a_k	b_k	c_k	Nominal	Minimum	Maximum
E_1	0.52	2.52	1432	3.0000	2.9950	3.0050
E_2	0.51	2.55	1443	0.5000	0.4996	0.5004
E_3	0.53	2.62	1428	0.5000	0.4994	0.5006
E_4	0.54	2.56	1436	0.5000	0.4992	0.5008
E_5	0.54	2.71	1430	0.5000	0.4993	0.5007
E_6	0.58	2.69	1442	1.0000	0.9990	1.0010
E_7	0.60	2.57	1441	1.0000	0.9990	1.0010
E_8	0.59	2.66	1429	0.5000	0.4991	0.5009
E_9	0.56	2.65	1418	0.5000	0.4994	0.5006
E_{10}	0.57	2.58	1434	3.5000	3.4995	3.5005
E_{11}	0.51	2.49	1422	4.0000	3.9994	4.0006
E_{12}	1.80	3.86	1224	2.0000	1.9985	2.0015
E_{13}	1.90	3.78	1234	2.5000	2.4988	2.5012
E_{14}	1.10	4.55	1126	0.2472	0.2469	0.2475
E_{15}	1.30	4.45	1146	0.2472	0.2469	0.2475
E_{16}	0.88	3.24	1184	4.5000	4.4985	4.5015
E_{17}	0.86	3.28	1196	4.5000	4.4985	4.5015
E_{18}	0.84	3.34	1208	0.5000	0.4986	0.5014
E_{19}	0.82	3.26	1192	0.5000	0.4986	0.5014
E_{20}	0.75	2.86	1368	4.0000	3.9985	4.0015
E_{21}	0.78	2.88	1376	0.5000	4.9995	0.5005
E_{22}	0.74	2.76	1372	0.4993	0.4992	0.4994
E_{23}	0.79	2.78	1370	0.5000	0.4995	0.5005
E_{24}	0.72	2.77	1362	3.9982	3.9981	3.9983
E_{25}	0.77	2.82	1358	0.5000	0.4995	0.5005
E_{26}	0.76	2.91	1380	0.4993	0.4992	0.4994
E_{27}	0.71	2.85	1378	0.4993	0.4992	0.4994
E_{28}	1.50	4.22	1155	1.1200	1.1196	1.1214
E_{29}	1.50	4.22	1164	1.4950	1.4934	1.4966
E_{30}	1.60	3.82	1242	0.4375	0.4365	0.4385
E_{31}	1.70	3.85	1238	0.5000	0.4990	0.5010

derivative of these constraints. We get the following equations for relating the tolerance of assembly functions with that of attributes:

$$\Delta F_1 = \Delta A_6 - \Delta A_3$$
$$\Delta F_2 = \Delta A_6 - \Delta A_1$$
$$\Delta F_3 = \Delta A_4 - \Delta A_5$$
$$\Delta F_4 = \Delta A_2 - \Delta A_7$$
$$\Delta F_5 = \Delta A_{10} - \Delta A_8$$
$$\Delta F_6 = \Delta A_{11} - \Delta A_9$$
$$\Delta F_7 = \Delta A_{12} - \Delta A_{15}$$
$$\Delta F_8 = \Delta A_{13} - \Delta A_{16}$$
$$\Delta F_9 = \Delta A_{14} - \Delta A_{17}$$

TABLE 10 Forward Propagation from Entities to Attributes

Description	Notation	Nominal	Minimum	Maximum
Inner Diameter of Hole 1	A_5	1.5000	1.4970	1.5030
Inner Diameter of Hole 2	A_7	2.0000	1.9969	2.0031
Width of Base	A_6	4.0000	3.9940	4.0060
Upper Width of Base Rail	A_{12}	5.0000	4.9975	5.0025
Lower Width of Base Rail	A_{13}	4.0000	3.9985	4.0015
Height of Base Rail	A_{14}	0.5000	0.4995	0.5005
Length of Bush 1	A_3	4.0000	3.9971	4.0029
Outer Diameter of Bush 1	A_4	1.5056	1.5035	1.5077
Inner Diameter of Bush 1	A_{10}	1.1250	1.1215	1.1285
Length of Bush 2	A_1	4.0000	3.9971	4.0029
Outer Diameter of Bush 2	A_2	2.0056	2.0038	2.0074
Inner Diameter of Bush 2	A_{11}	1.5000	1.4968	1.5032
Upper Width of Rail	A_{15}	4.9968	4.9965	4.9971
Lower Width of Rail	A_{16}	3.9982	3.9981	3.9983
Height of Rail	A_{17}	0.4993	0.4992	0.4994
Diameter of Shaft 1	A_8	1.1200	1.1195	1.1205
Diameter of Shaft 2	A_9	1.4950	1.4945	1.4955

Similarly, the second stage of BP is done from attributes to the entities. Using the constraint functions given in Table 8, we get the following equations for relating the tolerance of attributes with that of entities:

$$\Delta A_1 = \Delta E_{17} - \Delta E_{19}$$
$$\Delta A_2 = \Delta E_{13} - 2\ \Delta E_{15}$$
$$\Delta A_3 = \Delta E_{16} - \Delta E_{18}$$
$$\Delta A_4 = \Delta E_{12} - 2\ \Delta E_{14}$$
$$\Delta A_5 = \Delta E_{10} - \Delta E_4 - \Delta E_5 - \Delta E_6$$
$$\Delta A_6 = \Delta E_1 + \Delta E_2 + \Delta E_3$$
$$\Delta A_7 = \Delta E_{11} - \Delta A_7 - \Delta E_8 - \Delta E_9$$
$$\Delta A_8 = \Delta E_{28}$$
$$\Delta A_9 = \Delta E_{29}$$
$$\Delta A_{10} = \Delta E_{12} - 2\ \Delta E_{30}$$
$$\Delta A_{11} = \Delta E_{13} - 2\ \Delta E_{31}$$
$$\Delta A_{12} = \Delta E_{20} + \Delta E_{21} + \Delta E_{25}$$
$$\Delta A_{13} = \Delta E_{20}$$
$$\Delta A_{14} = \Delta E_{23}$$
$$\Delta A_{15} = \Delta E_{22} + \Delta E_{24} + \Delta E_{26}$$
$$\Delta A_{16} = \Delta E_{24}$$
$$\Delta A_{17} = \Delta E_{27}$$

TABLE 11 Forward Propagation from Attributes to Functions

Function	Notation	Nominal	Minimum	Maximum
Relative Length of Base & Bush 1	F_1	0.0000	−0.0089	0.0089
Relative Length of Base & Bush 2	F_2	0.0000	−0.0089	0.0089
Interference Between Base & Bush 1	F_3	0.0056	0.0005	0.0107
Interference Between Base & Bush 2	F_4	0.0056	0.0007	0.0105
Clearance Between Bush 1 & Shaft 1	F_5	0.0050	0.0010	0.0090
Clearance Between Bush 2 & Shaft 2	F_6	0.0050	0.0013	0.0087
Clearance of Upper Rails	F_7	0.0032	0.0004	0.0060
Clearance of Lower Rails	F_8	0.0018	0.0002	0.0034
Height Difference Between Base and Support Rails	F_9	0.0007	0.0001	0.0013

The results of the two stages of BP can be combined to get the constraint equations of the optimization problem. This is done by substituting the results of the second stage of BP into that of the first stage. Thus, we get the following constraints:

$$\begin{aligned}
\Delta F_1 &= \Delta E_1 + \Delta E_2 + \Delta E_3 - (\Delta E_{16} - \Delta E_{18}) \\
\Delta F_2 &= \Delta E_1 + \Delta E_2 + \Delta E_3 - (\Delta E_{17} - \Delta E_{19}) \\
\Delta F_3 &= \Delta E_{12} - 2\ \Delta E_{14} - (\Delta E_{10} - \Delta E_4 - \Delta E_5 - \Delta E_6) \\
\Delta F_4 &= \Delta E_{13} - 2\ \Delta E_{15} - (\Delta E_{11} - \Delta E_7 - \Delta E_8 - \Delta E_9) \\
\Delta F_5 &= \Delta E_{12} - 2\ \Delta E_{30} - (\Delta E_{28}) \\
\Delta F_6 &= \Delta E_{13} - 2\ \Delta E_{31} - (\Delta E_{29}) \\
\Delta F_7 &= \Delta E_{20} + \Delta E_{21} + \Delta E_{25} - (\Delta E_{22} + \Delta E_{24} + \Delta E_{26}) \\
\Delta F_8 &= \Delta E_{20} - (\Delta E_{24}) \\
\Delta F_9 &= \Delta E_{23} - (\Delta E_{27})
\end{aligned}$$

The upper limits of assembly-level functional tolerances are specified in Table 9. Substituting these values in the above equations and multiplying the coefficients of monotonically decreasing entity tolerances by −1, we get the following constraint equations:

$$\begin{aligned}
&\Delta E_1 + \Delta E_2 + \Delta E_3 + \Delta E_{16} + \Delta E_{18} &&= 0.0065 \\
&\Delta E_1 + \Delta E_2 + \Delta E_3 + \Delta E_{17} + \Delta E_{19} &&= 0.0065 \\
&\Delta E_{12} + 2\ \Delta E_{14} + \Delta E_{10} + \Delta E_4 + \Delta E_5 + \Delta E_6 &&= 0.0077 \\
&\Delta E_{13} + 2\ \Delta E_{15} + \Delta E_{11} + \Delta E_7 + \Delta E_8 + \Delta E_9 &&= 0.0077 \\
&\Delta E_{12} + 2\ \Delta E_{30} + \Delta E_{28} &&= 0.0029 \\
&\Delta E_{13} + 2\ \Delta E_{31} + \Delta E_{29} &&= 0.0029 \\
&\Delta E_{20} + \Delta E_{21} + \Delta E_{25} + \Delta E_{22} + \Delta E_{24} + \Delta E_{26} &&= 0.0034 \\
&\Delta E_{20} + \Delta E_{24} &&= 0.0021 \\
&\Delta E_{23} + \Delta E_{27} &&= 0.0009
\end{aligned}$$

Objective Function

The objective function of the problem is:

$$CM_{ASS} = \sum_{k=1}^{31} a_k + b_k \times \exp(-c\Delta E_k)$$

The values of $a_k, b_k,$ and c_k are given in Table 9.

Solution Using RSR Algorithm

The problem has now been solved using the proposed RSR algorithm. The initial values of all entity tolerances are assumed equal to 0.0001 units and the required accuracy is also assumed equal to 0.0001. Table 12 provides the results.

CONCLUSIONS

Tolerance design is essential in manufacturing, which plays an important role in relating performance to the design of a product. A good tolerance design mechanism should be able to determine a set of tolerances for the dimensioning of a component's entities, where the manufacturing costs are minimized, subject to the tolerance of the functional requirements. In this paper, we have adopted the hierarchical interval constraint networks for accurate and simple representation of an assembly and its optimal tolerance analysis and synthesis. Relative sensitivity ratio algorithm is an efficient technique for optimization of tolerance design. The RSR algorithm can be used for simple as well as complex assemblies with practically any number of entities and assembly functions. It can also handle any type of cost function and achieve any required accuracy in optimum allocation of tolerance. We have provided two illustrations to show the procedures of tolerance analysis and optimization in tolerance synthesis using interval constraint networks and RSR algorithm.

NOTATIONS

E, A, F:	Entity, attribute, or assembly function
ΔE, ΔA, ΔF:	Tolerances of E, A, F
i, j, k:	Indices for assembly functions, attributes and entities
i = 1 to n;	j = 1 to r; k = 1 to m

APPENDIX A

$$FP \text{ for constraint}, C(\{1,2,\ldots,n\},k,f()), FP(X_1,X_2,\ldots,X_n;X_k)$$

TABLE 12 Optimum Tolerances of Entities by RSR Algorithm

Entity	Optimum tolerance	Nominal dimension	Lower limit	Upper limit
E_1	0.0012	3.0000	2.9988	3.0012
E_2	0.0012	0.5000	0.4988	0.5012
E_3	0.0012	0.5000	0.4988	0.5012
E_4	0.0011	0.5000	0.4989	0.5011
E_5	0.0011	0.5000	0.4989	0.5011
E_6	0.0011	1.0000	0.9989	1.0011
E_7	0.0011	1.0000	0.9989	1.0011
E_8	0.0011	0.5000	0.4989	0.5011
E_9	0.0011	0.5000	0.4989	0.5011
E_{10}	0.0011	3.5000	3.4989	3.5011
E_{11}	0.0011	4.0000	3.9989	4.0011
E_{12}	0.0014	2.0000	1.9986	2.0014
E_{13}	0.0014	2.5000	2.4986	2.5014
E_{14}	0.0010	0.2472	0.2462	0.2482
E_{15}	0.0010	0.2472	0.2462	0.2482
E_{16}	0.0015	4.5000	4.4985	4.5015
E_{17}	0.0015	4.5000	4.4985	4.5015
E_{18}	0.0015	0.5000	0.4985	0.5015
E_{19}	0.0015	0.5000	0.4985	0.5015
E_{20}	0.0005	4.0000	3.9995	4.0005
E_{21}	0.0007	0.5000	0.4993	0.5007
E_{22}	0.0005	0.4993	0.4988	0.4998
E_{23}	0.0004	0.5000	0.4996	0.5004
E_{24}	0.0006	3.9982	3.9976	3.9988
E_{25}	0.0006	0.5000	0.4994	0.5006
E_{26}	0.0007	0.4993	0.4986	0.5000
E_{27}	0.0005	0.4993	0.4988	0.4998
E_{28}	0.0009	1.1200	1.1191	1.1209
E_{29}	0.0009	1.4950	1.4941	1.4959
E_{30}	0.0003	0.4375	0.4372	0.4378
E_{31}	0.0003	0.5000	0.4997	0.5003

Propagated from Input Tolerance to the Upper Limit of the Output Tolerance

$$xk'_{up} = f(x_{I\varphi}, \ldots, x_{n\varphi})$$

where $x_{i\varphi} = x_{i_{up}}$ if X_k is monotonic increasing with respect to X_i.

$x_{i\varphi} = x_{i_{low}}$ if X_k is monotonic decreasing with respect to X_i.

Propagated from Input Tolerance to the Lower Limit of the Output Tolerance

$$x'_{k_{low}} = f(x_{1\kappa} \ldots, x_{n\kappa})$$

where $x_{i_\kappa} = x_{i_{low}}$ if X_k is monotonic increasing with respect to X_i.
$x_{i_\kappa} = x_{i_{up}}$ if X_k is monotonic decreasing with respect to X_i.

REFERENCES

Chase, K.W., and W.H. Greenwood. 1988. Design issues in mechanical tolerance analysis *Manufacturing Review, ASME* 1(1):50–59.

Chen, M.-C. 2001. Tolerance synthesis by neural learning and non-linear programming. *International Journal of Production Economics* 70:55–65.

Cho, B. R., Y. J. Kim, D. L. Kimbler, and M. D. Phillips. 2000. An integrated and joint optimization procedure for robust and tolerance design. *International Journal of Production Research* 38(10):2309–2325.

Choi, H.-G. R., M.-H. Park, and E. Salisbury. 2000. Optimal tolerance allocation with loss functions. *Journal of Manufacturing Science and Engineering* 122(3):529–535.

Claralló, F. W., and C. Y. Christopher. 1997. Optimization of propagation in interval constraint networks for tolerance design. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* pages 1924–1929, Orlando, Florida, October 12–15, 1997.

Clowes, M. B. 1971. On seeing things. *Artifical Intelligence* 2:179–185.

Davis, E. 1987. Constraint propagation with interval lables. *Artifical Intelligence* 32:281–331.

Dechter, R., and J. Pearl. 1988. Network-based heuristics for constraint-satisfaction problem. *Artifical Intelligence* 34:1–38.

Dechter, R., and J. Pearl. 1989. Tree clustering for constraint networks. *Artifical Intelligence* 38:353–366.

Feng, C.-X., and A. Kusiak. 2000. Robust tolerance synthesis with the design of experiment approach. *Journal of Manufacturing Science and Engineering* 122(3):520–528.

Huffman, D. A. 1971. Impossible objects as nonsense sentence. *Machine Intelligence* 6:295–323.

Hyvonen, E. 1992. Constraint reasoning based on interval arithmatic: The tolerance propagation approach. *Artifical Intelligence* 58:71–112.

Jeang, A. 1999. Optimal tolerance design by response surface methodology. *International Journal of Production Research* 37(14):3275–3288.

Jeang, A. 2001. Combined parameter and tolerance design optimization with quality and cost. *International Journal of Production Research* 39(5):923–952.

Jeang, A., and E. Leu. 1999. Robust tolerance design by computer experiment. *International Journal of Production Research* 37(9):1949–1961.

Ji, S., L. Xiaoli, and R. Du. 2000. Tolerance synthesis using second-order fuzzy comprehensive evaluation and generic algorithm. *International Journal of Production Research* 38(15):3471–3483.

Ladkin, P. B., and A. Reinefeld. 1992. Effective solution of qualitive interval constraint problem. *Artifical Intelligence* 57:105–124.

Lu, S. C. Y., and R. Wilhelm. 1991. Automating tolerance synthesis: A framework and tool. *Journal of Manufacturing Systems* 10(4):279–296.

Mackworth, A. K., and E. C. Freuder. 1985. The complexity of some polynomial network consistency algorithms for constraint satisfaction problem. *Artifical Intelligence* 25:65–74.

Michael, W. and J. N. Siddali. 1981. The optimal tolerance assignment and full acceptance. *Journal of Mechanical Design, Transactions of ASME* 103:842–848.

Michael, W., and J. N. Siddali. 1982. The optimal tolerance assigment with less than full acceptance. *Journal of Mechanical Design, Transactions of ASME* 104:855–860.

Moskowitz, H., R. Plante and J. Duffy. 1999. The partial information case for multivariate tolerance design using scrap or rework cost. *International Journal of Production Research* 37(1):139–154.

Roy, U., and C. R. Liu, 1988. Tolerancing in computer aided design and manufacturing. *A review Technical Report TR-ERC 88-9* Schools of Engineering, Purdue University May 1988.

Speckhart, F. H. 1972. Calculation of tolerance based on a minimum cost appraoch. *Journal of Engineering for Industry, Transactions of ASME* 95: 447–453.

Spotts, M. F. 1973. Allocation of tolerances to minimize cost of assembly. *Journal of Engineering for Industry, Transactions of ASME* 97: 762–764.

Sutherland, G. H., and B. Roth. 1975. Mechanism design: Accounting for manufacturing tolerances and costs in function generating problems. *Journal of Engineering for industry, Transactions of ASME* 72:283–286.

Tsang, E. 1993. *Foundations of Constraint Satisfaction*. Essex: Academic Press.

Wilhelm, R. G., and S. C. Y. Lu. 1992. *Computer Methods for Tolerance Design*. Singapore: World Scientific Publishing Co.

Yang, C. C., M. M. Marefat, and F. W. Ciarallo. 1997. Tolerance analysis and synthesis by interval constraint network. In *Proceedings of the 1997 International Conference on Robotics and Automation* pages 2522–2527 Albuquerque, New Mexico, April 1997.

Yang, C. C., and M. H. Yang. 1997. Constraint networks: A survey. In. *Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, pages 271–287, Orlando, Florida, October 12–15, 1997.

Yang, C. C., M. M. Marefat, and F. W. Ciarallo. 2000. Interval constraint networks for tolerance analysis and synthesis. *Artifical Intelligence for Engineering Design, Analysis, and Manufacturing* 14(4):271–287.