Application of Hierarchical Interval Constraint Networks for Optimization of Tolerance Allocation

Christopher C. Yang
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, Hong Kong SAR

V.N.A. Naikan
Department of Industrial and Engineering Management
India Institute of Technology
Kharagpur, India

Abstract – In this paper, we propose an algorithm for optimal allocation of tolerances among components of a system by applying the basic principles of the hierarchical interval constraint networks. These networks consist of nodes and arcs. The nodes represent the component (or entity), attributes of component, and the system functions. Each arc represents a mathematical relationship (or constraint) between tolerances of the two corresponding nodes. A two-stage network is used to represent the tolerance allocation problem. Tolerance analysis is conducted by propagating the tolerances of entities or components forward through the constraint networks to calculate the tolerances of system functions. This is known as forward propagation of tolerance. The system functional tolerances can be propagated through the constraint network in the reverse direction for tolerance synthesis or allocation, known as backward propagation. The objective function of the problem is to minimize the total cost of manufacturing of all entities or components. Cost of manufacturing is expressed as a function of the entity tolerances. A Lagrange multiplier method is then applied to solve the non-linear optimization problem. An illustrative example is included.

INTRODUCTION

Literature on Tolerance Allocation

Tolerance calculations have been explored in several common design situations. For example, Foster and Spotts have developed formulas for calculating size position tolerances to achieve a desired class and grade of cylindrical fit between mating parts. However, optimization is not considered in the tolerance calculations. In addition, new formulae are required for each design. Fortini presented worst-case tolerance synthesis with a linear design equation in which the limits for the design variables and the tolerance variables are assumed and used to obtain the overall limits on the remaining tolerance variables. Parkinson utilized probabilistic methods to optimize the dimensional tolerance such that there is an acceptably low risk of the assembly failing to meet the specification, given the manufacturing costs. Hoffmann reduced the analysis of tolerance and process inaccuracies to the analysis of systems of linear inequalities. Michael and Siddall developed an optimal allocation of manufacturing tolerances by nonlinear optimization, using the minimization of material cost as the optimization criterion. Cagan et al. utilized a simulated annealing approach to allocate tolerances and the manufacturing processes in order to obtain the minimum cost. The cost function in the algorithm is based on the hourly costs of the machines and the production time for the processes. Soderberg proposed tolerance allocation by optimizing the quality and manufacturing cost. Wilhelm and Siddall

Lu have proposed a tolerance synthesis approach, CASCADE-T, which used a constraint network representation of the conditional tolerance relations that exist between features of a part. However, the solution is not consistent and optimization is not considered.

An algorithm is proposed in this paper for optimal allocation of tolerances among the components or entities. The basic principles of hierarchical interval constraint networks are used for tolerance analysis. The necessary constraints of the optimization problem are derived and subsequently eliminating the redundant constraints. The objective function of the optimization problem is the minimization of the total manufacturing cost of the assembly. An illustrative example is given in support of the proposed algorithm.

CONSTRANET NETWORKS FOR TOLERANCE DESIGN

A hierarchical Interval Constraint Network represents the relationship between the highest (assembly) level functional requirements and the lowest level entities of any assembly in the tolerance design problem. The functional requirement describes the functions of the design and the allowable tolerances. Each functional requirement of the assembly can be represented as a mathematical expression of a number of attributes. Attributes can also be described as functions of the mechanical dimensions of the associated parts entities. Figure 2 illustrates an interval constraint network (ICN), for a cylindrical vessel shown in Figure 1.

Consistency of ICN for Tolerance Design

In an interval constraint network for tolerance design, the constraint function is a multiple/single input(s) and single output and is represented as a triple, C(U,k,f()). U is the set of indexes for the input variables and k is the index of the output variable for the constraint C; f() is the constraint function. Following are the two definitions:

a. A constraint C(U,k,f()) is consistent if and only if
\[ \bigcap_{v_i \in V_i} \bigcup_{v_j \in V_j} (\exists v_i, v_j) \]

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

The consistency of constraints ensures that the tolerances assigned to the entities satisfy the functional requirements.

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, viz.,
Forward Propagation (FP) and Backward Propagation (BP). In FP, tolerances of input variables are propagated through the constraints to obtain the tolerance of the output variable. FP will detect the consistency of various constraints in the network, which is known as tolerance analysis. 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), which is known as tolerance synthesis. Algorithms for forward propagation and backward propagation without considering manufacturing costs are available in [3][4].

In this paper, the FP is carried out to verify if any of the functional tolerances of the assembly is/are already satisfied by 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. Optimization is introduced in the backward propagation. This is further clarified in subsequent sections.

Tolerance and Cost

Several models relating cost of manufacturing with tolerance are available in the literature. These include a linear model, reciprocal, reciprocal squared, reciprocal powered, exponential, and combined exponential-reciprocal power [6]. The proposed algorithm in this paper can handle any form of cost model.

PROPOSED ALGORITHM

An algorithm is proposed to derive constraints, objective functions and solution using Lagrange multipliers.

Derivation of Constraint Functions

The optimization problem will have as many constraints as the number of functional requirements of the assembly. Redundant and satisfied constraints can be removed while optimizing. Following steps are proposed.

Step 1: Carry out a FP of numerical tolerances of entities through attributes to functions. Find whether any of the constraints is consistent. 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 Step 1 is:

```
SetF=[Fi, i=1..n]
Call FP, Get Fpp = [Finom-pp, Fup-pp, Filow-pp, i=1..n]
Read Freq = [Finom-req, Fup-req, Filow-req, i=1..n]
For i=1..n Do {
    IF (Finom-pp=Finom-req) AND (Fup-pp>Fup-req)
        AND (Filow-pp>Filow-req) THEN g=i
}
SetF=[Fi, i=1..n, i≠g]
SetA=[Aj, j=1..r]
For j=1..r Do {
    IF (Fi is dependent on Aj) AND ((Fi = Freq) are
        independent of Aj)) THEN h=j
}
SetA=[Aj, j=1..r, j≠h]
```

SetE=[Ek, k=1..m]
For k=1..m Do {
    IF (Ah is dependent on Ek) AND ((Aj, j=1..r, j≠h) are
        independent of E_k)) THEN s=j
}
SetE=[Ek, k=1..m, k≠s]
Write (SetF, SetA, SetE)

Step 2: Carry out the first level of BP (functions to attributes) for all assembly functions. Propagate the algebraic tolerances and not the numerical tolerances.

```
For i=1..n, i≠g Do {
    Fi = f(Aj)
    ∆Fi = f(∆Aj)
}
```

Step 3: Carry out the second level of BP (attributes to entities) for all attributes by considering step 1. 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≠h Do {
    Aj = f(Ek)
    ∆Aj = f(∆Ek)
}
```

Step 4: Substitute the system of equations obtained in step 3 into those in step 2. Substitute the numerical value of the maximum required upper tolerance of all assembly level functions.

```
For i=1..n, i≠g Do {
    Fi = ∆Fip
    Fiup = fi(∆Ej)
    filow = fi(∆Ek)
}
```

Step 5: Find out the presence of any redundant equation in the system of equations obtained in step 4 above and eliminate them. The remaining equations are:

```
F(∆E_k) = ∆Fip
```

The actual number of constraints to the optimization problem is equal to the number of original assembly level nodes (functions) minus the number of assembly functions satisfied by FP minus the number of constraints eliminated due to redundancy.

Derivation of the Objective Function

The objective function of the optimization problem is the total manufacturing cost function of the assembly. This is expressed as sum of the cost functions of the individual entities. If any entity is eliminated in step 1, the cost function for that entity will not be included in the objective function. Mathematically, the objective function can be expressed as follows:

```
CM_{ASS} = \sum_{i,j=k} N_i \cdot CM_{E_i}
```

where

- \( CM_{ASS} \): Cost of Manufacturing the Assembly for required tolerances
- \( CM_{E_i} \): Cost of manufacturing the \( k \)-th Entity, expressed as a function of its tolerance.
Solving the Optimization Problem

In this section, we present the procedure for solving the optimization problem. The objective function and the constraint equations derived in previous sections may be linear or non-linear. Therefore, we follow a general method for non-linear optimization using Lagrange multipliers to obtain a closed form solution. Mathematically, the optimization problem can be stated as follows:

Minimize \( CM_{\text{ASS}} = \sum_{k \in \mathbb{K}} N_k (CM_{E_k}) \)

Subjected to \( f(\Delta E_k) = \Delta F_{\text{up}} \); \( i=1..n; i \neq d; i \neq g \)

Lagrange Multiplier Method

In this method, we combine the objective function and the constraint equations to get a single equation known as Lagrange Function.

\[ L = \text{ObjectiveFunction} - \sum_{i=1}^{n} \lambda_i \left( \text{Constraint Equation} \right) \]

where \( \lambda_i \) : Lagrange Multiplier for constraint equation \( i \).

Take the first order derivatives of \( L \) with respect to the tolerance of each entity and with respect to each \( \lambda_i \) and equate them to zero. Solution of these simultaneous equations will give the optimum tolerance allocation of the assembly. For the tolerance optimization problem, we get the following Lagrange function:

\[ L = \sum_{k \in \mathbb{K}} N_k (CM_{E_k}) + \sum_{i=1}^{n} \lambda_i (f(\Delta E_i) - \Delta F_{\text{up}}) \]

EXAMPLE

A tank as shown in Figure 2 is to be designed with optimum tolerance allocation. Functional requirements, their allowable tolerances, and the cost functions are given in Table 1 and Table 2.

Functional Requirements: The effective inner volume and the wall thickness at different sections of the cylinder are the assembly level functional requirements as shown in Table 1.

Table 1: Assembly Function requirements

<table>
<thead>
<tr>
<th>Assembly Functions</th>
<th>Notations</th>
<th>Nominal Dimensions</th>
<th>Allowable Tolerances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inner Volume</td>
<td>V</td>
<td>2.9*10^7 mm³</td>
<td>0.1*10^7 mm³</td>
</tr>
<tr>
<td>Thickness</td>
<td>T₁</td>
<td>10 mm</td>
<td>1 mm</td>
</tr>
<tr>
<td>Thickness</td>
<td>T₂</td>
<td>10 mm</td>
<td>1 mm</td>
</tr>
<tr>
<td>Thickness</td>
<td>T₃</td>
<td>5 mm</td>
<td>0.5 mm</td>
</tr>
</tbody>
</table>

Cost of Manufacturing and Entity Tolerance: The cost of manufacturing of each entity is related with its tolerance by an inverse square rule as stated below:

\[ CM = a + \frac{b}{(\text{tolerance})^2} \]

The values of ‘a’ and ‘b’ are given in Table 2.

Table 2: Manufacturing cost functions

<table>
<thead>
<tr>
<th>Entity</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>E₁</td>
<td>58.0</td>
<td>10.0</td>
</tr>
<tr>
<td>E₂</td>
<td>112.0</td>
<td>12.0</td>
</tr>
<tr>
<td>E₃</td>
<td>87.0</td>
<td>15.0</td>
</tr>
<tr>
<td>E₄</td>
<td>43.0</td>
<td>16.0</td>
</tr>
<tr>
<td>E₅</td>
<td>67.0</td>
<td>18.0</td>
</tr>
<tr>
<td>E₆</td>
<td>75.0</td>
<td>20.0</td>
</tr>
<tr>
<td>E₇</td>
<td>55.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Constraint Functions: Constraint functions between entities and attributes and those between attributes and assembly requirements are obtained from Figure 2. These are given in Table 3.

Table 3: Constraint Functions for the Hierarchical Interval constraint Network

<table>
<thead>
<tr>
<th>Constraint Functions</th>
<th>Constraint Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>between Entities and Attributes</td>
<td>between Attributes and Functional requirements</td>
</tr>
<tr>
<td>L₁ = E₃; L₂ = E₁ + E₂ – E₃</td>
<td>L₁ = E₁</td>
</tr>
<tr>
<td>L₃ = E₁</td>
<td>L₂ = E₁ + E₂ – E₃</td>
</tr>
<tr>
<td>R₁ = E₆ – E₅; R₂ = E₆</td>
<td>R₃ = E₇ – E₄; R₄ = E₇</td>
</tr>
<tr>
<td>V = πR₁²L₁ + πR₂²L₂</td>
<td>T₁ = R₁ – R₂</td>
</tr>
<tr>
<td>T₁ = R₄ – R₂</td>
<td>T₂ = R₁ – R₂</td>
</tr>
<tr>
<td>T₂ = R₃ – R₁</td>
<td>T₃ = L₁ – L₃</td>
</tr>
</tbody>
</table>

Forward Propagation: Forward propagation of tolerances is carried out in two stages as explained in previous sections using the constraint functions and the network. FP has revealed that \( V \) is satisfied with the present level of tolerances. \( L₂ \) is the only attribute affecting \( V \), but not other functions. So \( L₂ \) is not considered for BP. \( E₂ \) is the only entity affecting \( L₂ \), but not other attributes. Therefore, \( E₂ \) is not considered for BP.

Backward propagation: BP is also carried out in two stages to derive the constraints of the optimization problem. Using steps 2, 3, and 4 of the proposed algorithm, we get the following constraints.

\[ \Delta E₆ + \Delta E₇ = 1.0 \]  \( (1) \)
\[ \Delta E₄ + \Delta E₃ + \Delta E₆ + \Delta E₇ = 1.0 \]  \( (2) \)
\[ \Delta E₃ + \Delta E₅ = 0.5 \]  \( (3) \)

It is important to note that we are concerned with deterministic tolerance analysis and synthesis. Therefore, these relationships can be used for estimating the upper (maximum) tolerance limits in deterministic cases. This is a conservative approach and is known as stacking up of tolerances.

Removal of Redundant Constraint: Equation (1) is redundant because it is included in the equation (2). Therefore, the actual constraints are (2) and (3).
The Optimization Problem: The objective function is to minimize the total cost of manufacturing. The tolerance optimization problem can be stated as follows:

Minimize \( CM_{ASS} \)

\[
= 497 + \frac{10}{(\Delta E_1)^2} + \frac{15}{(\Delta E_2)^2} + \frac{16}{(\Delta E_3)^2} + \frac{18}{(\Delta E_4)^2} + \frac{20}{(\Delta E_5)^2} + \frac{10}{(\Delta E_6)^2} + \frac{10}{(\Delta E_7)^2}
\]

Subject to: \( \Delta E_1 + \Delta E_2 + \Delta E_3 + \Delta E_6 + \Delta E_7 = 1.0 \)

\( \Delta E_4 + \Delta E_5 = 0.5 \)

Solution: The Lagrange function for the above problem is obtained as follows:

\[
L = 497 + \frac{10}{(\Delta E_1)^2} + \frac{15}{(\Delta E_2)^2} + \frac{16}{(\Delta E_3)^2} + \frac{18}{(\Delta E_4)^2} + \frac{20}{(\Delta E_5)^2} + \frac{10}{(\Delta E_6)^2} + \frac{10}{(\Delta E_7)^2} - \lambda_1 (\Delta E_1 + \Delta E_2 + \Delta E_3 + \Delta E_6 + \Delta E_7 - 1.0) - \lambda_2 (\Delta E_1 + \Delta E_3 - 0.5)
\]

Take partial derivatives of the Lagrange function with respect to all \( \Delta E_i \) s and \( \lambda_i \) s and equate them to zero. We get eight non-linear equations with eight variables. Simultaneous solution of these equations provides the following results.

\[
\begin{align*}
\Delta E_1 & = 0.2331 \\
\Delta E_2 & = 0.2669 \\
\Delta E_3 & = 0.2517 \\
\Delta E_4 & = 0.2618 \\
\Delta E_5 & = 0.2712 \\
\Delta E_6 & = 0.2512 \\
\Delta E_7 & = 0.2152 \\
\lambda_1 & = -1578.441 \\
\lambda_2 & = -2005.667
\end{align*}
\]

CONCLUSIONS

An algorithm is proposed and illustrated for optimum allocation of tolerances. The concepts of Interval Constraint Networks consistency, forward and backward propagation of tolerances are very useful to derive the objective function as well as the constraints of the tolerance optimization problem. Majority of the tolerance optimization problems are non-linear and therefore Lagrange multiplier method is the most appropriate procedure for closed form solutions. The proposed algorithm guarantees a closed form solution to the optimum tolerance allocation problem of assemblies for any type of cost functions and constraints.

ACKNOWLEDGEMENT

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

REFERENCES


