Tolerance Synthesis by Constraint Propagation

Christopher C. Yang and Jason Wong
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

Abstract

Optimizing the manufacturing cost is critical in tolerance synthesis. Decreasing the tolerance range will improve performance but will also increase the manufacturing costs. It is desirable to optimize the tolerance range under such constraints of product design as the relationship between the dimensions of entities of a component and the functional requirement of the design. We propose a constraint-based reasoning mechanism to synthesize a new set of tolerances to satisfy the functional requirements of a product and minimizing the manufacturing cost during the tolerance propagation in interval constraint networks by genetic algorithm.

A hierarchical interval constraint network and interval propagation techniques for automatic tolerance synthesis are presented in this paper. The nodes in interval constraint networks represent the entities, the attributes, and the functional requirements of the mechanical parameters or the constraint functions. The arcs represent the relationships between the entities, the attributes, the functional requirements and the constraint functions. In tolerance synthesis, given the functional requirement tolerances, the goal is to synthesize a new set of entity tolerances. Backward propagation technique is developed for tolerance synthesis. In backward propagation, the minimization of the manufacturing cost is developed based on genetic algorithm.

The goal of the proposed investigation is to develop a constraint-based reasoning technique for tolerance design, which plays an important role in the relationship between performances and manufacturing cost of a product. Such technique should be able to satisfy all the constraints in the tolerance design and minimize the manufacturing cost.

1. Introduction

Since it is practically impossible to manufacture a component exactly with the required dimensions, all design dimensions are provided with specified tolerances. It is difficult to manufacture a component with narrow tolerance band compared to a wider tolerance band. Narrow tolerance band require better material, machine tool, control mechanism, workman skills, extra processing time, measuring instruments and involvement of management. This will definitely cost more compared to wider band manufacturing tolerances. Wider band tolerances will be cheaper, but there will be more 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 the required functional specifications. Therefore, a method is to be designed for distribution of assembly level functional tolerances to the component level manufacturing tolerances such that the total cost of manufacturing of the assembly is minimized. Tolerance propagation has been proposed in this paper for optimal allocation of tolerances among the components or entities. The basic principles of hierarchical interval constraint networks have been used for tolerance analysis and then for deriving the necessary constraints of the optimisation problem. Genetic algorithm is utilized in backward propagation for optimising the manufacturing cost.

For a given design of a mechanical part, a relationship can be derived for the functional requirement in terms of the entities. This relationship can be expressed as: \( Y = f(X_1, X_2, ..., X_n) \) where \( Y \) is the functional requirement and \( X_i \) is the \( i \)th entity. \( n \) is the number of entities that are related by the equation to the corresponding functional requirement.

In tolerance analysis, the entity tolerances, \( X_1, X_2, ..., X_n \), are given. The goal is to ensure that the functional requirement tolerance, \( Y \), is met. The tolerances \( X_i \) and \( Y \), are the range of acceptable values for, \( X_i \) and \( Y \), respectively. If the assigned functional requirement tolerances are not met, the tolerances for the entities need to be reassigned by tolerance synthesis in order to achieve the functional requirements. In tolerance synthesis, the functional requirement tolerance, \( Y \), is given. The goal is to determine a set of feasible entity tolerances, \( X_1, X_2, ..., X_n \), to fulfill the functional requirement. Figure 1 gives the concept and relationship of tolerance analysis and synthesis.
1.1 Constraint Networks

Constraint satisfaction problems (CSPs) are often formulated in 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. The mathematical operations can be multiple inputs single output (MISO) or single input single output (SISO). 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. For tolerance design, the variables are labeled by intervals and the constraints are n-ary mathematical operations.

For each mechanical design, the relationship between the highest level, functional requirement, and the lowest level, entity, can be represented by a hierarchical network. The functional requirement describes the functions of the design and the requirement to satisfy these functions. Each functional requirement can be described as a function (constraint) in terms of attributes. An attribute is also described as a function in terms of the mechanical part's entities. These relationships are described as a hierarchical interval constraint network as shown in Figure 2 in our approach.

2. Satisfaction of Interval Constraint Networks for Tolerance Design

The application of the interval constraint networks determines the definition of satisfaction. In ICSP, according to Hyvonen, the satisfaction of the interval constraint network is defined as follows:

A variable, \( V_i \), is consistent if and only if
\[
\forall (v_i \in V_i \mid V_i = v_i), \exists (v_1 \in V_1, \ldots, v_{i-1} \in V_{i-1}, v_{i+1} \in V_{i+1}, \ldots, v_n \in V_n \mid V_1 = v_1, \ldots, V_{i-1} = v_{i-1}, V_{i+1} = v_{i+1}, \ldots, V_n = v_n)
\]
such that all constraints are satisfied where \( V_i \) is an interval, \( V_i \) is a variable, and \( v_i \) is an value.

The constraint network is satisfied if and only if all variables are consistent.

Such properties of the consistency of a variable as described in ICSP are not appropriate for tolerance design. Based on the application of tolerance design, the definition of consistency focus on constraints. In a constraint network for tolerance design, the constraint is multiple/single inputs and single output (MISO or SISO) and is represented as a triple, \( C_i(U,k,f()) \). \( U \) is the set of indexes for the input variables and \( k \) is the index of the output variable for the constraint \( C_i \).

**Definition 1:**
A constraint, \( C_i(U,k,f()) \), is consistent if and only if
\[
\bigcap_{j \in U} (\forall v_j \in V_j \mid V_j = v_j) \land (\exists v_k \in V_k \mid V_k = v_k)
\]
such that \( C_i(U,k) \) is satisfied.

where \( U \) is the set of indexes for the input variables and \( k \) is the index of the output variable for the constraint \( C_i \).

**Definition 2:**
The interval constraint network for tolerance design is satisfied if and only if all of the constraints are consistent.

3. Manufacturing Cost

As discussed in previous sections, cost of manufacturing increases when we reduce the tolerance bands. 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, reciprocal (cost of manufacturing is proportional to reciprocal of tolerance) [1], reciprocal squared [6], reciprocal powered [7], exponential [5], and combined exponential-reciprocal power [3,4] among others. While all these models are empirical and based on experiences, it has been noticed that the reciprocal squared and the exponential models are more frequently used than other models. Chen [2] 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, we have used reciprocal square model for the cost function.

\[ MC_{x_i}(\Delta x_i) = A_i + \frac{B_i}{(\Delta x_i)^2} \]

where \( \Delta x_i \) is the tolerance of \( x_i \).

4. Tolerance Propagation

Tolerance propagation is utilized to update the intervals in the network to make the interval constraints consistent. Tolerance can be propagated from the input intervals of a constraint to the single output interval, which is known as forward propagation. Tolerance can also be propagated from the single output interval of a constraint to multiple input intervals, known as backward propagation. The forward and backward propagation techniques for tolerance design are developed based on Definitions 1 and 2. Given a constraint with constraint function \( X_k = f(X_1, X_2, ..., X_n) \) with input intervals, \( X_1, X_2, ..., X_n \) and output interval, \( X_k \), if the constraint is not consistent (\( F(X_1, X_2, ..., X_n) \) is not a subset of \( X_k \)), either \( X_k \) must be relaxed (widened) or one or more of the input intervals must be tightened (narrowed). \( X_k \) is relaxed by propagating \( X_1, X_2, ..., X_n \) forward. \( X_1, X_2, ..., X_n \) are tightened by propagating \( X_k \) backward.

4.1 Forward Propagation for a Single Constraint

The forward propagation is based on the constraint function such that the intervals of the input variables are propagated to the interval of the single output variable. If the interval propagated from the input intervals is not a subset of the output interval, the output interval is updated (relaxed) to the union of the propagated interval and the original assigned output interval, otherwise, the constraint is consistent and nothing is changed. The algorithm for forward propagation is given as:

**Forward Propagation** for constraint, \( C\{1,2, ..., n\}; k, f() \), \( FP(X_1, X_2, ..., X_n; X_k) \)

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

\[ x_{k\text{up}}' = f(x_1\varphi, ..., x_{n\varphi}) \]

where \( x_{i\varphi} = x_{i\text{up}} \) if \( X_k \) is monotonic increasing with respect to \( X_i \).

\[ x_{i\varphi} = x_{i\text{low}} \] if \( X_k \) is monotonic decreasing with respect to \( X_i \).

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

\[ x_{k\text{low}}' = f(x_1\kappa, ..., x_{n\kappa}) \]

where \( x_{i\kappa} = x_{i\text{up}} \) if \( X_k \) is monotonic increasing with respect to \( X_i \).

\[ x_{i\kappa} = x_{i\text{low}} \] if \( X_k \) is monotonic decreasing with respect to \( X_i \).

Relaxing the Output Tolerance

If \( x_{k\text{up}}' < x_{k\text{low}} \) or \( x_{k\text{low}}' > x_{k\text{up}}' \),

\[ \text{NO SOLUTION} \]

Otherwise,

\[ x_{k\text{up}} = x_{k\text{up}}' \] if \( x_{k\text{up}}' > x_{k\text{up}} \),

\[ x_{k\text{low}} = x_{k\text{low}}' \] if \( x_{k\text{low}}' < x_{k\text{low}} \).

4.2 Backward Propagation for a Single Constraint

The backward propagation is also based on the constraint function such that the interval of the output variable is propagated to one or more of the intervals of the input variables. If the constraint is not consistent, the output interval is propagated to the input intervals by tightening each of the input intervals. We integrate the manufacturing cost functions with the backward propagation in order to minimize the increase of cost when we tighten the tolerances.

4.2.1 Optimisation

Given the set of constraints, \( \{C_1, C_2, ..., C_n\} \), between two adjacent levels of variables, \( \{x_1, x_2, ..., x_m\} \) and \( \{y_1, y_2, ..., y_n\} \), the optimisation problem is formulated as follow subject to the constraints \( C_1, C_2, ..., C_n \):

\[ \text{Minimize} \sum_{i=1}^{m} MC_{x_i}(\Delta x_i) \]

The genetic algorithm is utilized to optimise the manufacturing costs of \( x_i \)s in the backward propagation from \( y_i \)s to \( x_i \)s.
Genetic Algorithm

Initialisation of Population

A chromosome represents the set of tolerances of variables in the lower lever of the propagations, $\Delta x_1$, $\Delta x_2$, ..., $\Delta x_m$. There are $m$ sets of genes in the chromosome, where each set representing a tolerance $\Delta x_i$. The higher number of genes ($M$) in a chromosome, the higher precisions of the tolerances are. Initially, $N$ numbers of chromosomes are generated randomly. For each chromosome, we compute the total manufacturing costs of all the variables in the lower lever.

The fitness of chromosome

$$\text{fitness} = 1/\sum_{i=1}^{m} MC_{x_i}(\Delta x_i)$$

Reproduction

Reproduction is the selection of a new population. A chromosome that has higher fitness value has a better chance of being selected. Each chromosome occupies a certain number of slots on a roulette wheel directly proportional to its fitness. Spinning the roulette wheel $N$ times, we select $N$ chromosomes for the new population. According to the genetic inheritance, the best chromosomes get more copies, the average stay even, and the worst die off.

Crossover and Mutation

There are two recombination operations, crossover and mutation. The crossover operates between a pair of chromosome and the mutation operates on a single chromosome.

Crossover: The probability of crossover, $P_c$, gives us the expected number $P_c N$ of chromosomes that should undergo the crossover operation. For each chromosome, we generate a random number, $X$, between 0 and 1. If $X$ is less than $P_c$, the chromosome is selected for crossover. For each pair of selected chromosomes, we generate a random number, $Y$, between 0 and $M-1$. $Y$ indicates the position of the crossing point. The coupled chromosomes exchange genes at the crossover point. If the crossover chromosome does not satisfy the constraints, it does not survive (it is eliminated).

Mutation: Mutation is performed on a bit-by-bit basis. The probability of mutation, $P_m$, gives us the expected number $P_m M N$. Every bit in all chromosomes of the whole population has an equal chance to undergo mutation. For each chromosome and for each bit within the chromosome, we generate a random number, $Z$, between 0 and 1. If $Z$ is less than $P_m$, we mutate the bit. Similar to the crossover operation, if the mutated chromosome does not satisfy the constraints, it does not survive (it is eliminated).

Convergence

After reproduction, crossover, and mutation, the new population is ready for the next generation. The evolution of the solution continues until the system converges. This occurs when the total of fitness for the whole population decreases less than a small value, $\delta$, for a few generations.

5. Example

A tank as shown in figure 3 is to be designed for optimum tolerance allocation. Following are the functional requirements of the tank.

Functional Requirements: (Nominal dimensions and tolerances)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Nominal Value</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume, $V$</td>
<td>$2.8 \times 10^7 - 3.0 \times 10^7$ mm$^3$</td>
<td>$2.9 \times 10^7$ mm$^3$</td>
</tr>
<tr>
<td>Thickness, $T_1$</td>
<td>$9 - 11$ mm</td>
<td>$10$ mm</td>
</tr>
<tr>
<td>Thickness, $T_2$</td>
<td>$9 - 11$ mm</td>
<td>$10$ mm</td>
</tr>
<tr>
<td>Thickness, $T_3$</td>
<td>$4.5 - 5.5$ mm</td>
<td>$5$ mm</td>
</tr>
</tbody>
</table>

The cost of manufacturing with required tolerance for each entity is as follows.

Cost of Manufacturing of each entity:

$$CM_{E_i} = 58 + \frac{10}{(\Delta E_i)^2}$$

$$CM_{E_i} = 112 + \frac{12}{(\Delta E_i)^2}$$

$$CM_{E_i} = 87 + \frac{15}{(\Delta E_i)^2}$$

$$CM_{E_i} = 43 + \frac{16}{(\Delta E_i)^2}$$

$$CM_{E_i} = 67 + \frac{18}{(\Delta E_i)^2}$$

$$CM_{E_i} = 75 + \frac{20}{(\Delta E_i)^2}$$

$$CM_{E_i} = 55 + \frac{10}{(\Delta E_i)^2}$$

Figure 4 shows the hierarchical interval constraint network for the tank problem as shown in Figure 3. The entities on the lowest level are labelled on Figure 3(d). The attributes in the middle level are labelled in Figure 3(c). The functional requirements in the highest level are labelled in Figure 3(b).
Figure 3. (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 4. Hierarchical interval constraint network for the tank in Figure 1

Table 1 shows the result of forward propagation. Given the natural tolerances of all the entities, the functional requirement tolerance of volume is satisfied, but the functional requirement tolerance of the thicknesses, $T_1$, $T_2$, and $T_3$, are not satisfied.

Table 1. Result of forward propagation

<table>
<thead>
<tr>
<th>Entities</th>
<th>Natural Tolerances</th>
<th>Forward Propagation</th>
<th>Constraint Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>[94mm, 96mm]</td>
<td>[94mm, 96mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E2</td>
<td>[204mm, 206mm]</td>
<td>[204mm, 206mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E3</td>
<td>[99mm, 101mm]</td>
<td>[99mm, 101mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E4</td>
<td>[49mm, 51mm]</td>
<td>[49mm, 51mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E5</td>
<td>[49mm, 51mm]</td>
<td>[49mm, 51mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E6</td>
<td>[189mm, 191mm]</td>
<td>[189mm, 191mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E7</td>
<td>[199mm, 201mm]</td>
<td>[199mm, 201mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L1</td>
<td>[100mm, 100mm]</td>
<td>[99mm, 101mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L2</td>
<td>[200mm, 200mm]</td>
<td>[197mm, 203mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L3</td>
<td>[95mm, 95mm]</td>
<td>[94mm, 96mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R1</td>
<td>[140mm, 140mm]</td>
<td>[138mm, 142mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R2</td>
<td>[190mm, 190mm]</td>
<td>[189mm, 191mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R3</td>
<td>[150mm, 150mm]</td>
<td>[148mm, 152mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R4</td>
<td>[200mm, 200mm]</td>
<td>[199mm, 201mm]</td>
<td>Satisfied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Natural Tolerances</th>
<th>Forward Propagation</th>
<th>Constraint Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>[94mm, 96mm]</td>
<td>[94mm, 96mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E2</td>
<td>[204mm, 206mm]</td>
<td>[204mm, 206mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E3</td>
<td>[99mm, 101mm]</td>
<td>[99mm, 101mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E4</td>
<td>[49mm, 51mm]</td>
<td>[49mm, 51mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E5</td>
<td>[49mm, 51mm]</td>
<td>[49mm, 51mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E6</td>
<td>[189mm, 191mm]</td>
<td>[189mm, 191mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>E7</td>
<td>[199mm, 201mm]</td>
<td>[199mm, 201mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L1</td>
<td>[100mm, 100mm]</td>
<td>[99mm, 101mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L2</td>
<td>[200mm, 200mm]</td>
<td>[197mm, 203mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>L3</td>
<td>[95mm, 95mm]</td>
<td>[94mm, 96mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R1</td>
<td>[140mm, 140mm]</td>
<td>[138mm, 142mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R2</td>
<td>[190mm, 190mm]</td>
<td>[189mm, 191mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R3</td>
<td>[150mm, 150mm]</td>
<td>[148mm, 152mm]</td>
<td>Satisfied</td>
</tr>
<tr>
<td>R4</td>
<td>[200mm, 200mm]</td>
<td>[199mm, 201mm]</td>
<td>Satisfied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Requirements</th>
<th>Natural Tolerances</th>
<th>Forward Propagation</th>
<th>Constraint Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>2.8x10^7 mm^3, 3.0x10^7 mm^3</td>
<td>2.8x10^7 mm^3, 3.0x10^7 mm^3</td>
<td>Satisfied</td>
</tr>
<tr>
<td>T1</td>
<td>[9mm, 11mm]</td>
<td>[8mm, 12mm]</td>
<td>Not Satisfied</td>
</tr>
<tr>
<td>T2</td>
<td>[9mm, 11mm]</td>
<td>[6mm, 14mm]</td>
<td>Not Satisfied</td>
</tr>
<tr>
<td>T3</td>
<td>[4.5mm, 5.5mm]</td>
<td>[3mm, 7mm]</td>
<td>Not Satisfied</td>
</tr>
</tbody>
</table>

Backward propagation is then utilized to propagate the tolerances of $T_1$, $T_2$, and $T_3$ to all the tolerances of attributes and entities that are affected by them and genetic algorithm is utilized to minimize their manufacturing costs.

The result of the optimisation in backward propagation is as follows:

$E1 = [94.7669 \text{ mm}, 95.2331 \text{ mm}]$

$E2 = [204 \text{ mm}, 206 \text{ mm}]$
E3 = [99.7331mm, 100.2669mm]
E4 = [49.7483mm, 50.251mm]
E5 = [49.7382mm, 50.2618mm]
E6 = [189.7288mm, 190.2712mm]
E7 = [199.7948mm, 200.2152]

6. Conclusion

Optimal tolerance design of assembly is essential for minimizing the cost of manufacturing, while meeting all its functional requirements. An algorithm is proposed and illustrated for this purpose. 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 optimisation problem. Majority of the tolerance optimisation problems are non-linear and therefore genetic algorithm is utilized. An example is provided to illustrate the propagation techniques and genetic algorithm.

7. Acknowledgement

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

8. References