Optimal Tolerance Allocation Based on Fuzzy Comprehensive Evaluation and Genetic Algorithm

S. Ji, X. Li, Y. Ma and H. Cai

The Advanced Machining Technology Center, Harbin Institute of Technology, Harbin, China

It is very important to know how to allocate tolerances economically for parts in a CAD/CAM system because this directly affects the machining costs of the parts. A new approach based on fuzzy comprehensive evaluation (FCE) and a genetic algorithm (GA) is presented to obtain a rational tolerance allocation for the parts. First, the current methods for tolerance allocation are reviewed in detail. Then, FCE is used to evaluate the machinability of a part; a new optimal model, which can fully exploit DFA (design for assembly) and DFM (design for manufacturing), is established by combining the functional sensitivity factors and machinability factors of parts. A genetic algorithm (GA) is developed and used to optimise the above model. Finally, an actual example is used to verify the feasibility of the above method; the computed result shows that the method can produce tolerance allocations economically and accurately.

Keywords: Fuzzy comprehensive evaluation; Genetic algorithm; Tolerance allocation

1. Introduction

In mechanical design, geometric and dimensional tolerances are used to specify the range within which a part geometry and size may vary while conforming to the functional requirements. Assigned tolerances have a direct effect not only on the machining costs but also on the product quality. Unnecessarily tight tolerances result in high production costs, yet the tolerances should ensure that the functional performance requirements of the products stay within a satisfactory range. Tolerances which are too loose can affect the product quality, and increase the scrap rate and production costs [1,2]. In general, designers allocate tolerances for parts based on their experience, and on handbooks and standards, which leads to some errors [3]. In recent years, computer-aided tolerancing design (CATD)

has become an important research direction in CAD systems and integrated CAD/CAM systems [4–7].

Tolerance allocation is one of the most important problems for CATD. Given a required tolerance for the assembly, the planner is first faced with the problem of how to allocate economically suitable tolerance values for the parts by considering the trade-off between functional requirements and machining costs. Many researchers considered tolerance allocation as an optimisation problem. The tolerance values of parts were taken as the control variables, and the machining costs were taken as the objective function to be minimised. The tolerance stack-up limits conditions were taken as constraints on the variables [8–10].

In this paper, the tolerance allocation is represented as an optimisation problem. A set of methods used is first reviewed in detail. Fuzzy comprehensive evaluation is used to evaluate the machinablity of parts, then a new mathematical model is established and solved using a genetic algorithm (GA). The tolerance allocation for an actual industrial assembly is produced by the above method, and the results show that the method can be used economically to design the tolerance values of parts.

2. Review: Methods for Tolerance Allocation

2.1 General Allocation Methods

When an assembly function requirement is given, the tolerance values of n parts must be solved. Because the given conditions are almost always insufficient, the tolerances are usually regarded as equal. The methods used often include the same tolerance method, the constant precision factor method, the same influence method and the proportional scaling method.

2.1.1 Same Tolerance Method

In the same tolerance method, all of the tolerances of parts are equal on the premise of satisfying the functional requirement of the assembly, that is,

---

Correspondence and offprint requests to: Dr Xiaoli Li, Department of Manufacturing Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong E-mail: mel50001@cityu.edu.hk

$$t_1 = t_2 = \ldots = t_n$$

where t_i denotes the tolerance value of part i.

2.1.2 Constant Precision Factor Method

The constant precision factor method is based on the rule of thumb that the tolerance of a part increases as the cubic root of the nominal size, thereby

$$t_i = P(d_i)^{1/3}$$

where $\sum t_i \leq J$, the precision factor P can be calculated as

$$P = \frac{J}{\sum d_i^{1/3}}$$

where J denotes the functional requirement on the assembly.

2.1.3 Same Influence Method

The assembly functional requirement is influenced by two factors. One is the tolerance t_i of each part, another is the functional sensitivity coefficient ξ_i. So the same influence method can be expressed as follows:

$$t_1\xi_1 = t_2\xi_2 =, \ldots, = t_n\xi_n$$

2.1.4 Proportional Scaling Method

The part tolerances are first determined using a database in the proportional scaling method. If the sum of the part tolerances exceeds the required assembly tolerance J, each part tolerance t_i is relatively reduced corresponding to dimension d_i. The general expression is:

$$\frac{d_1}{t_1} = \frac{d_2}{t_2} = \ldots = \frac{d_i}{t_i} = \ldots = \frac{d_n}{t_n}$$

where d_i is the dimension of part i, t_i is the tolerance allocated for d_i, and $\sum_{i=1}^{n} t_i \leq J$ (worst tolerance analysis), or $\sum_{i=1}^{n} t_i^2 \leq J^2$ (statistical analysis).

In short, the above methods are very simple, but can only evaluate the tolerance value and corresponding precision grade of each part as a whole. These values are often used in the initial stage of tolerance allocation.

2.2 Minimising Costs Methods

The best rule for evaluating tolerance allocation is the machining costs rule, so the minimising costs methods have been important research topics. In these methods, the relationship between the machining costs and the tolerance of a part is expressed using mathematical formulae, and minimising the total machining costs is taken as an optimal objective under the constraints of the functional requirements. In the last 50 years, more than 10 models on the cost–tolerance relationship have been presented. They are shown in Table 1.

These models are all based on the empirical cost–tolerance data frequently used in production processes [20,21]. The model parameters are determined using the least-squares method based on these data. Owing to the lack of available production data about the costs–tolerance relationship, and the fact that the machining costs will change when the machining context changes, the applications of the methods are very limited.

2.3 Comprehensive Factor Method

The quantitative evaluation of the machinability of each part is called the comprehensive factor. Because machining costs are strongly related to the machinability, the relative machining costs of each part can be evaluated using its comprehensive factor. Usually, after the appropriate weight value of each factor (which has important effects on machining costs) is given, the comprehensive factor can be calculated by:

$$F_i = F_{i1}F_{i2} \ldots F_{ij} \ldots F_{im} \quad (i = 1, 2, \ldots, n;$$

$$j = 1, 2, \ldots, m) \quad P_i = F_i \Big/ \sum_{i=1}^{n} F_i$$

where i ($i=1 \ldots n$) denotes the number of parts, m is the number of factors relative to machinability, P_i is the percentage scaling which indicates the relative machinability of part i on the whole assembly. Generally, the factors that influence machining costs include machining methods, part materials, part geometrical structure, part size, etc. Because these factor values can only be satisfactorily determined by experts with abundant production knowledge and experience, the method is subjective.

2.4 Artificial Intelligent Method

Artificial intelligence techniques are currently used for CATD. Kopardekar and Anand [22] presented a neural network-based method for tolerance allocation, taking into account machinability and mean shifts. The neural network can predict individual part tolerances, and is shown in Fig. 1.

The advantages of the proposed method are:

1. The procedure does not need any assumption about distribution of the part dimensions, unlike statistical techniques.
2. This method can be extended to the assemblies of numbers of parts.

However, some disadvantages limit its application, such as:

1. The approach requires some data with known outputs.
2. The approach is not very efficient when the number of parts is small.

Dupinet [9] presented a new method that used fuzzy inference to evaluate the manufacturing difficulties of a part. All fuzzy rules were given by an expert and could be changed to accommodate the knowledge of other companies or other expert skills, so the method is very simple, but sometimes the experts have great difficulty in defining the fuzzy rules.

3. Tolerance Allocation Based on FCE and GA

The major drawback of the general allocation methods is that they do not take into account the machining costs. In the

Table 1. Cost–tolerance models presented.

Model	Mathematical expression	Reference
Exponential	$c(\delta) = a_0 e^{-a_i}$	[11, 19]
Reciprocal squared model (R-squared)	$c(\delta) = a_0/\delta^2$	[12, 19]
Reciprocal powers model (R-power)	$c(\delta) = a_0 \delta^{-a_i}$	[13, 19]
Reciprocal powers and exponential hybrid model (RP-E hybrid)	$c(\delta) = a_0 \delta^{-a_i} e^{-a_2\delta}$	[14, 19]
Reciprocal model (reciprocal)	$c(\delta) = a_0/\delta$	[15, 19]
Modified exponential model (M-exponent)	$c(\delta) = a_0 e^{-a_1(\delta - a_2)} + a_3 \ \delta_{min} \leq \delta \leq \delta_{max}$	[16, 19]
Discrete Model (discrete)		[17, 19]
Combined reciprocal powers and exponential model (combined RP-E)	$c(\delta) = a_0 + a_1 \delta^{-a_2} + a_3 e^{-a_4\delta}$	[18, 19]
Combined linear and exponential model (combined L-E)	$c(\delta) = a_0 + a_1 \delta + a_2 e^{-a_3\delta}$	[18, 19]
Cubic ploynomial (cubic-p)	$c(\delta) = a_0 + a_1 \delta + a_2 \delta^2 + a_3 \delta^3$	[18, 19]
Fourth-order polynomial (4th-p)	$c(\delta) = a_0 + a_1 \delta + a_2 \delta^2 + a_3 \delta^3 + a_4 \delta^4$	[18, 19]
Firth-order polynomial (5th-p)	$c(\delta) = a_0 + a_1 \delta + a_2 \delta^2 + a_3 \delta^3 + a_4 \delta^4 + a_5 \delta^5$	[18, 19]
Neural network model		[20]

Fig. 1. Network architecture.

minimising costs method, the machining costs are taken into account, but it is difficult to determine the cost–tolerance relationship for every machining process. In fact, the designer can evaluate a non-uniform cost–tolerance relationship only according to previous experience. The general rule is that the machinability of a part determines its machining costs. The tolerance allocation of a part depends on its machinability.

Taking into account the advantages and disadvantages of the above methods, we propose a new method based on fuzzy comprehensive evaluation and a genetic algorithm (see Fig. 2).

3.1 Machinability Estimator Based on Fuzzy Comprehensive Evaluation

According to the design and machining criteria, the machinability of parts depends on the dimensions, the geometrical structure, the material machinability and the machining accu-

Fig. 2. The whole procedure of tolerance allocation using the new approach.

racy. Clearly, these are fuzzy factors. We use two-order fuzzy comprehensive evaluation to process the fuzzy factors. The method and theory of fuzzy comprehensive evaluation is referred to in [23,24].

3.1.1 Establishing the Fuzzy Factor Set

The factors of dimension size, geometrical structure, material machinability, and machining accuracy can be expressed as:

$$U = \{u_i, u_2, \ldots, u_m\} = \{DS, GS, MM, PA\} \quad (1)$$

where $m=4$, DS: dimension size, GS: geometrical structure, MM: material machinability, and PA: machining accuracy.

3.1.2 Grade of Factor

For evaluating accurately the value of each parameter, each factor is divided into different fuzzy subsets such that:

$$u_i = \{u_{i1}, u_{i2}, \ldots, u_{in_i}\} \quad (2)$$

where u_{ij} ($i = 1 \sim m$; $j = 1 \sim n_i$) denotes the jth grade of the ith factor, n_i represents the grade number of each factor (the detailed value is shown in Table 2).

3.1.3 Establishing the Evaluation Set

Since the range of machinability is between 0 .0 and 1.0, it can be divided into 10 equal levels, namely,

$$\varsigma = \{\varsigma_1, \varsigma_2, \ldots, \varsigma_k, \ldots, \varsigma_9, \varsigma_{10}\} = \{0.1, 0.2, \ldots, 0.9, 1.0\} \quad (3)$$

where ς is the fuzzy evaluation set.

3.1.4 First-Order Fuzzy Comprehensive Evaluation Matrix

Based on the experience of experts, the first-level fuzzy comprehensive evaluation matrix is determined below:

$$\mathbf{R}_1 = \begin{bmatrix} 0.0 & 0.0 & 0.2 & 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.4 & 1.0 & 0.4 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 1.0 & 0.4 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 1.0 & 0.8 \end{bmatrix}$$

$$\mathbf{R}_2 = \begin{bmatrix} 0.2 & 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.4 & 1.0 & 0.8 \end{bmatrix}$$

Table 2. Main effective factors and its grade division.

Factors		Grades of each factor			
		Grade 1	Grade 2	Grade 3	Grade 4
u_1	DS	~ 5mm	~ 25 mm	~ 75 mm	~ 120 mm
u_2	GS	Easy to manufacture	Difficult to manufacture	—	—
u_3	MM	Good	Medium	Poor	—
u_3	PA	Normal	Medium	Accuracy	—

Note: n_1=4, n_2=2, n_3=3, n_4=3.

$$\mathbf{R}_3 = \begin{bmatrix} 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.4 & 1.0 & 0.6 \end{bmatrix}$$

$$\mathbf{R}_4 = \begin{bmatrix} 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.4 & 1.0 & 0.4 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.4 & 1.0 & 0.6 \end{bmatrix}$$

3.1.5 Establishing the Weight Vector of each Grade

The weight vector is composed of membership degrees of the evaluation object to all grades in each grade of each factor, shown as follows:

$$\mathbf{A}_i = (a_{i1}, a_{i2}, \ldots, a_{in_i}) \tag{4}$$

where $a_{ij} = \mu_{ij} \Big/ \sum_{i=1}^{n_i} \mu_{ij}$ $(i = 1 \ldots 4; j = 1 \ldots n_i)$

Here, μ_{ij} is the membership degree of the evaluation object to the jth grade of the ith factor.

3.1.6 First-Order Fuzzy Comprehensive Evaluation

When the fuzzy comprehensive evaluation is made for every grade of the ith factor, the first-order fuzzy comprehensive set is obtained by:

$$\mathbf{B}_i = \mathbf{A}_i \circ \mathbf{R}_i = (a_{i1}, a_{i2}, \ldots, a_{in_i}) \circ$$

$$\begin{bmatrix} r_{i11}, & r_{i12}, & \ldots, & r_{i1p} \\ r_{i21}, & r_{i22}, & \ldots, & r_{i2p} \\ \ldots\ldots\ldots\ldots\ldots \\ r_{in_i1}, & r_{in_i2}, & \ldots, & r_{in_ip} \end{bmatrix} = (b_{i1}, b_{i2}, \ldots, b_{ip}) \tag{5}$$

where p=10. Membership degrees are determined directly by the experts, or by membership functions.

Here the composition operator $M(\bullet, +)$, which can take into account the effects of all factors, but can also contain all the information of an individual factor, is expressed by

$$b_{ik} = \sum_{j=1}^{n_i} a_{ij}r_{ijk} \quad (i = \{1,2,\ldots,4\}; k = \{1,2,\ldots,10\}) \tag{6}$$

Then, the first-order fuzzy comprehensive evaluation matrix can be written as

$$\mathbf{R} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_m \end{bmatrix} = \begin{bmatrix} b_{11}, & b_{12}, & \ldots, & b_{1p} \\ b_{21}, & b_{22}, & \ldots, & b_{2p} \\ \vdots & & \vdots \\ b_{m1}, & b_{m2}, & \ldots, & b_{mp} \end{bmatrix} \tag{7}$$

where m=4 and p=10.

3.1.7 Determining the Weight Vector of Factors

After obtaining the first-order fuzzy comprehensive evaluation matrix, we have to determine the weight vector of factors, which indicates the degrees of importance of factors to the evaluation object. It can be written as

$$\mathbf{A} = (a_1, a_2, \ldots, a_m) \tag{8}$$

When assuming that DS and GS are more important than MM and PA, we write Eq. (8) as $\mathbf{A} = (0.3, 0.3, 0.25, 0.15)$.

3.1.8 Second-Order Fuzzy Comprehensive Evaluation

Finally, we make the second-order fuzzy comprehensive evaluation. Fuzzy set $\mathbf{B}$ can be calculated by

$$\mathbf{B} = \mathbf{A} \circ \mathbf{R} = (a_1, a_2, \ldots, a_m) \circ \begin{bmatrix} b_{11}, & b_{12}, & \ldots, & b_{1p} \\ b_{21}, & b_{22}, & \ldots, & b_{2p} \\ \vdots & & & \vdots \\ b_{m1}, & b_{m2}, & \ldots, & b_{mp} \end{bmatrix} \tag{9}$$

$$= (b_1, b_2, \ldots, b_p)$$

where m=4 and p=10.

3.1.9 Determining Parameter

Generally, the weighted averaging method is used to obtain the accuracy of the evaluation object. It is shown as follows

$$s = \sum_{p=1}^{10} b_p s_p \Big/ \sum_{k=1}^{n} b_p \tag{10}$$

According to the above steps, the machinability, which is very important for tolerance allocation, can be determined.

3.2 Modelling of Tolerance Allocation

Machininability can be determined based on the FCE method. The assembly response function is assumed as

$$tol_{asm} = g(tol) = tol_0 + \xi_1 tol_1 + \xi_2 tol_2 + \ldots + \xi_n tol_n \quad (11)$$

where

$$\xi_i = \partial g/\partial(tol_i) \quad (12)$$

where, tol_{asm} denotes the assembly function requirement in tolerance design, $g(\cdot)$ is the assembly response function, tol_i represents the tolerance value of the ith corresponding part, tol_0 is constant, ξ_i, which reflects the degree of importance of each design tolerance on an assembly, is the assembly sensitivity coefficient of the ith part.

According to the experts' experience, the larger ξ_i is, the smaller is the corresponding tolerance allocated. Therefore, a comprehensive factor for tolerance allocation, Ψ_i, can be produced as follows:

$$\psi_i = \frac{\zeta_i}{\xi_i^2} = \frac{\zeta_i}{\left(\dfrac{\partial g}{\partial(tol_i)}\right)^2} \quad (13)$$

where ζ_i is the machinability of ith part obtained by the FCE. Referencing to the reciprocal model, a model of the optimal tolerance allocation can be expressed as follows:

$$\text{Min } C = f(tol) = C_0 + \sum_{i=1}^{n} \frac{\xi_i}{tol_i}$$

$$\text{Subject to: } l_i \leq tol_i \leq u_i, \quad 1 \leq i \leq n \quad (14)$$

$$l \leq tol_{asm} \leq u$$

where C denotes the total machining costs, $f(tol)$ is the function relationship of cost–tolerance, C_0 is the costs constant, $\mathbf{L} = (l_1, l_2, \ldots, l_n)$ and $\mathbf{U} = (u_1, u_2, \ldots, u_n)$ are constraint vectors for design tolerance of parts in an assembly, l and u represent the upper and lower limit of the assembly requirement.

3.3 Genetic Algorithm

A genetic algorithm, which is a recently developed heuristic optimisation strategy, has been used for global optimisation in a variety of research fields. A GA is based on the mechanics of natural selection and natural genetics and Darwinian survival of the fittest. Detailed discussion of the mechanisms of GA can be found in [25,26].

GA is different from traditional search methods encountered in engineering optimisation problems. GA works with a coding of the design variables as opposed to the variables themselves – continuity of parameter space is not a requirement. GA searches from a population of points, not a single point – parallel processing of points reduces the chance of being trapped into a local optimum. GA uses probabilistic transition rules, not deterministic transition rules, which leads to high-quality solutions, and GA requires only the objective function values, these minimal requirements result in a broad applicability of GA.

These important features of GA, such as the flexibility, global application, parallelism, simplicity, versatility, good problem solving capability, etc, make genetic algorithms very useful, and therefore popular. During the last decade, genetic algorithms have had increasing applications in a variety of

fields with promising results. Recently, some work has been successfully carried out using genetic algorithms for optimal engineering design problems.

The application of GA for tolerance optimisation allocation is given in the following sections.

3.3.1 Representation Scheme

In genetic algorithms, representation is an essential issue because the representation scheme links the real-world problem to the genetic algorithms and the genetic algorithms directly manipulate the coded representation of the problem. There are many kinds of representations, such as binary digit string representation, floating point representation, permutations of a list, etc. The selection of an appropriate representation depends on the characteristics of the search space.

Because machining precision is known in the tolerance allocation problem, the tolerance variable is considered as a discrete one in the feasible design domain. We adopt the binary digit string representation. If π_i is the machining precision of the tolerance design variable tol_i, then its string length λ_i is determined by the following inequity:

$$\lambda_i = \log_2 \left(\frac{tol_i^u - tol_i^l}{\pi_i} + 1 \right) \quad (15)$$

where tol_i^u and tol_i^l are upper and lower bounds of the ith tolerance design variable.

After the string length of each tolerance design variable has been determined, the length of the chromosome is computed from:

$$A = \sum_{i=1}^{n} \lambda_i \quad (16)$$

where n is the number of independent tolerance design variables.

3.3.2 Decoding

If the ith binary digit substring of a chromosome is decoded into an unsigned decimal integer I_i, then the physical value of the ith tolerance design variable tol_i is computed by:

$$tol_i = tol_i^l + I_i \pi_i \quad (17)$$

3.3.3 Fitness Function

Fitness is a quality value, which is a measure of the reproductive efficiency of living creatures according to the principle of survival of the fittest. In genetic algorithms, the fitness is used to allocate reproductive trials and thus is some measure of goodness to be maximised. This means that strings with higher fitness values will have a higher probability of being selected as parents.

Tolerance allocation is a constrained minimisation problem. If the objective function is also expressed by:

$$\text{Min} \quad f(tol) \quad (18)$$

$$\text{s.t.} \quad b_i(tol) \geq 0 \quad (i = 1, 2, \ldots, m)$$

Where $b_i(tol)$ is the constrained function, then it can be transformed to an unconstrained problem by the exterior penalty function method.

$$\text{Max}\quad F(tol) \tag{19}$$

where the fitness function F is computed by the following function:

$$F(tol) = G(tol)\, P(tol) \tag{20}$$

where

$$G(tol) = \begin{cases} \dfrac{K}{1 + (1.1)^{\beta f}} & (g \geq 0) \\[2mm] \dfrac{K}{1 + (0.9)^{\beta f}} & (g < 0) \end{cases} \tag{21}$$

$$P(tol) = \frac{1}{(1.1)^{\phi(tol)}} \tag{22}$$

$$\phi(tol) = \sum_{i=1}^{m} |b_i(tol)| \tag{23}$$

Where $P(tol)$ denotes the penalty function, $G(tol)$ is used to describe the quality of the solution, K and β are constant. In general, $K=1$, β is computed by:

$$\beta = \begin{cases} 1 & (f(tol) \geq 0) \\ -1 & (f(tol) < 0) \end{cases} \tag{24}$$

3.3.4 Linear-Fitness Scaling

Since the strings with higher fitness values have a higher probability of being selected as the parents, it is important to confine the allocation of selection to the best strings, especially in small population genetic algorithms. The fitness scaling can regulate and tune the number of times selected to prevent the domination of extraordinary individuals, and therefore it can prevent premature convergence. In addition, the fitness scaling can encourage a healthy competition among near equals when the population average fitness is close to the population best fitness.

A linear scaling formula is adopted with a normalisation process as follows:

$$F' = 1 - \frac{F_{max} - F}{F_{max} - F_{min}} \tag{25}$$

where F' is the scaled fitness, F is the raw fitness, F_{max} and F_{min} are maximum and minimum fitness, respectively.

3.3.5 Selection

The selection operator determines the set of individuals, which remain at the next generation. The roulette wheel selection scheme is used herein. This scheme is implemented as a linear search through a roulette wheel with each slice weighted in proportion to a scaled fitness value, the selection operator is obtained as follows:

1. Sum the fitness values of all the population members. The result is the total fitness F_{total}.
2. Generate a random number R between 0 and 1, then multiply F_{total} by R to generate an index number N:

$$N = R \times F_{total} \tag{26}$$

3. Return the first population member whose fitness, added to the fitness of the processing population members, is greater than or equal to N.

3.3.6 Crossover

Crossover is a primary operator in genetic algorithms and is the key to the success of genetic algorithms. We generate new individuals I'_a and I'_b from parent individuals I_a and I_b which are selected randomly from the population. They are divided into subparts at multiple points of crossover, and the new individuals are obtained by swapping them between these individuals.

3.3.7 Mutation

The mutation operator arbitrarily alters the gene value according to a predetermined probability. The mutation probability should be carefully prescribed. If the mutation probability is low, then the algorithm is often trapped at a local optimum. However, if the mutation probability is high, then the propagation of good schemata will be unduly hindered and the algorithms will degenerate to a random search method.

We use a new strategy to decide if the mutation operator is invoked. First, the number N of the same individual in a population is computed, then if N is greater than or equal to N_{same}, then the mutation operator is invoked, otherwise, the crossover operator is used. Here, N_{same} is a new control parameter for the algorithm and represents the permissible number of the same individuals in the population.

3.3.8 Memory Tool

We use memory tool (MT) to "remember" the best individual of a generation. When the best individual of the next generation is better than the one in MT, MT is updated using the new individual. Finally, the solution of GA is obtained from MT.

3.3.9 Parameters

For the tolerance allocation problem, the parameters are selected as follows:

Population size	= 100
Generation	= 500
Crossover probability	= 0.95
N_{same}	= 30

The procedure of the modified genetic algorithm is described as follows:

Start Program:
   a. Initialise (some essential parameters (GS, P_c, N_{same}, n));
   b. Initialise (MT);
   c. Generate (the first generation population randomly);
   d. Compute (their fitness);
   e. For i = 1 to GS do
     Begin
       If (the random number r is greater than P_c) continuous;
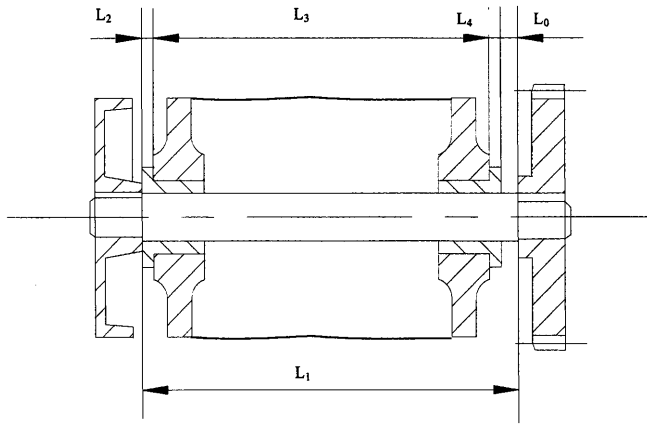   Else

Fig. 3. The driving device and its dimension loop chain.

```
      Calculate (the number of the same individual
      in population);
      If (n greater than or equal to N_same)
         Do (mutation operation);
      Else
         Do (crossover operation);
      Update (MT);
      Do (reproduction operation to product next generation);
   End
 f. Output (the individual of memory tool).
End (Program).
```

Where GS denotes the terminal generation number, n is the variable of counting, and P_c represents the crossover probability.

4. Example

In order to verify the proposed method, a driving device (see Fig. 3) is taken as an example. In Fig. 3, $L_1=160$ mm, $L_2=L_4=5$ mm, $L_3=150$ mm. Clearance L_0, which is the assembly function requirement, is needed to assemble the product without interference.

To solve the tolerance allocation problem, Tables 3–5 list the membership degrees of the four corresponding dimensions on the assembly, according to the known conditions.

Using the above method, the machinability vector of four corresponding part dimensions is obtained as follows:

$$\zeta = \{0.66, 0.41, 0.76, 0.41\}$$

Table 3. Membership degrees of length of the shaft shoulder.

Factor		Grades of each factor			
		I	II	III	IV
u_1	Dimension size	0.0	0.0	0.0	1.0
u_2	Geometrical structure	0.85	0.0	—	—
u_3	Material machinability	0.0	0.0	1.0	—
u_3	Process accuracy	0.5	0.5	0.0	—

Table 4. Membership degrees of thickness between end faces of left or right bearing.

Factor		Grades of each factor			
		I	II	III	IV
u_1	Dimension size	1.0	0.0	0.0	0.0
u_2	Geometrical structure	0.85	0.0	—	—
u_3	Material machinability	0.2	0.6	0.5	—
u_3	Process accuracy	0.5	0.5	0.0	—

Table 5. Membership degrees of thickness between end faces of carriers.

Factor		Grades of each factor			
		I	II	III	IV
u_1	Dimension size	0.0	0.0	0.0	1.0
u_2	Geometrical structure	0.5	0.85	—	—
u_3	Material machinability	0.0	0.9	0.5	—
u_3	Process accuracy	0.0	0.0	1.0	—

where, part 1 is the shaft shoulder; parts 2 and 4 are the left or right bearing, respectively; part 3 is the carrier. The result shows that the carrier is the most difficult to machine, and the left and right bearings are the easiest. This coincides with the practical machining case, so the approach is proved to be correct.

It is assumed that the two clearances are combined, the whole clearance is from 0.10 mm to 0.40 mm and its tolerance is 0.03 mm. The function equation is written as:

$$tol_{asm} = g(tol) = tol_1 - tol_2 - tol_3 - tol_4$$

Using Eq. (12), the vector of the function sensitivity factors is solved as follows:

$$\xi = \{1.0, -1.0, -1.0, -1.0\}$$

A comprehensive factor vector of tolerance allocation is obtained using Eq. (13):

$$\psi = \{0.66, 0.41, 0.74, 0.41\}$$

Therefore, a mathematical model for the tolerance allocation is established using Eq. (14):

$$\text{Min } C = C_0 + \frac{0.66}{tol_1} + \frac{0.41}{tol_2} + \frac{0.74}{tol_3} + \frac{0.41}{tol_4}$$

Subject to: $0.0 \leq tol_i \leq 0.30$, $1 \leq i \leq 4$

$$0.0 \leq tol_{asm} \leq 0.30$$

The tolerance vector computed by the improved GA is:

$$tol = \{0.8, 0.06, 0.10, 0.06\}$$

According to the principle that determines the bias of parts, the tolerance design result is:

$$L_1 = 160^{+0.00}_{-0.08}, \quad L_2 = 5^{+0.00}_{-0.06}, \quad L_3 = 150^{-0.18}_{-0.28}, \quad L_4 = 5^{+0.00}_{-0.06}$$

5. Summary

This paper presents a new approach based on fuzzy comprehensive evaluation and GA for tolerance allocation. In the tolerance allocation, the machinability, which depends on the fuzzy comprehensive evaluation and the function sensitivity factor, is considered, so the ideas of DFA and DFM are involved. The approach not only ensures the correctness of tolerance design, but also saves the machining costs. In addition, the improved GA is used to optimise the model based on DFA and DFM. The result of a detailed example shows that the method presented can obtain tolerance allocations with economically attainable accuracy.

References

1. G. Yanxin, "Basic knowledge of interchangeability and measure techniques", Harbin Institute of Technology, 1992.
2. Song Shaoming, "The tolerance analysis and synthesis in CAD", PhD dissertation of HIT, September 1997.
3. A. Ballu and L. Matheu, "Analysis of dimensional and geometrical specifications: standards and models", Proceedings of 3rd CIRP Seminars on Computer Aided Tolerancing, pp. 157–170, 1993.
4. Z. Genbao, "The survey of computer aided tolerancing", China Mechanics Engineering, 7(5), pp. 47–50, 1996.
5. F. Hongfang and W. Zhaotong, "A novel mathematical model for tolerance design", Mechanical Engineering and Automation, 19(4), pp. 23–26, 1997.
6. L. Yusheng, W. Zhaotong and Y. Jiangxin, "A research of computer aided tolerance optimal design based on costs model", Zhejiang University Journal, 29(6), pp. 698–705, 1992.
7. J. Lee and G. E. Johnson, "Optimal tolerance allotment using a genetic algorithm and truncated Monte Carlo simulation", Computer-Aided Design, 25(9), pp. 601–611, 1993.
8. L. Chunpu, "Statistical tolerance and mechanical precision", Mechanical Engineering, 1990.
9. E. Dupinet, M. Balazinski and E. Caogala, "Tolerance allocation based on fuzzy logic and simulated annealing", Journal of Intelligent Manufacturing, 7, pp. 487–497, 1996.
10. S. Qi and Z. Quner, "Analysis of tolerance distribution method and machining costs", Aeronautical Machining Technology, 2, pp. 35–38, 1998.
11. F. H. Speckhart, "Calculation of tolerance based on a minimum costs approach", Journal of Engineering for Industry, 5, pp. 447–453, 1972.
12. M. F. Spotts, "Allocation of tolerance to minimize costs of assembly", Journal of Engineering for Industry, pp. 762–764, August 1973.
13. G. H. Sutherland and B. Roth, "Mechanism design: accounting for machining tolerances and costs in function generating problems", Journal of Engineering for Industry, 2, pp. 283–286, 1975.
14. W. Michael and J. N. Siddall, "Optimization problem with tolerance assignment and full acceptance", ASME Journal of Mechanical Design, 103, pp. 842–848, 1981.
15. K. W. Chase and W. H. Greedwood, "Design issues in mechanical tolerance analysis", Machining Review, 1(1), pp. 50–59, 1990.
16. Z. Dong and A. Soom, "Automatic optimal tolerance design for related dimension chains", ASME Machining Review, 3(4), pp. 262–271, December 1990.
17. W. J. Lee and T. C. Woo, "Optimum selection of discrete tolerances", Journal of Mechanisms, Transmissions, and Automation in Design, 111, pp. 243–251, 1989.
18. Z. Dong and W. Hu, "Optimal process sequence identification and optimal process tolerance assignment in computer-aided process planning", Computers in Industry, 17, pp. 19–32, 1991.
19. Z. Dong, W. Hu and D. Xue, "New production costs–tolerance models for tolerance synthesis", Journal of Engineering for Industry, 116, pp. 199–206, 1994.
20. Y. Jiangxin, G. Daqiang and W. Zhaotong, "Tolerance–costs model base on neural network", China Mechanical Engineering, 7(6), pp. 41–42, 1996.
21. H. E. Trucks and H. B. Smith, "Designing for economical production", Society of Machining Engineers, Dearborn, Michigan, 1976.
22. P. Kopardekar and S. Anand, "Tolerance allocation using neural networks", International Journal of Advanced Machining Technology, 10, pp. 269–276, 1995.
23. W. Caihua and S. Liantian, "The methodology of fuzziness", China Architecture Engineering, 1988.
24. C. Qinghong, "Fuzzy comprehensive evaluation of design variables in mechanical reliability", Fuzzy Techniques and Application(I), pp. 492–497, Beijing Aerospace University, 1994.
25. M. Pirlot, "General local search methods", European Journal of Operational Research, 92, pp. 493–511, 1996.
26. Z. Michalewica, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, 1998.