論文 한국공작기계학회지 제9권 제1호 2000. 2.
Journal of the Korean Society of Machine Tool Engineers

조립 수율을 고려한 공차할당 및 가공중심 결정

이 진 구*

Tolerance allotment with Design Centering considering Assembly Yield

Jinkoo Lee*

Abstract

The purpose of this research was developing an integrated way to solve two typical tolerance optimization problems, i.e. optimal tolerance allotment and design centering. A new problem definition, design centering-tolerance allotment problem (DCTA), was proposed here for the first time and solved. Genetic algorithm and coarse Monte Carlo simulation were used to solve the stochastic optimization problem. Optimal costs were compared with the costs from the previous optimization strategies. Significant cost reductions were achieved by DCTA scheme.

Key Words : Tolerance, Yield, Stochastic optimization, Design centering

1. Introduction

Tolerance in mechanical design represents the acceptable variation from the nominal value. In mechanical design, tolerances must be assigned in order to assure economical manufacturing cost and interchangeability of parts. In assemblies, the tolerances for individual dimensions have a strong effect on the functions and the interchangeability. Therefore tolerancing problems occur in the range from a clearance problem to the overall performance of complicated mechanisms. The sum of the individual dimensions(*stack-up*) in assemblies might not be in-spec even when the individual dimensions are in-spec. There are several approaches in assignment of tolerances, including worst case tolerancing, statistical tolerancing, and tolerance optimization. Worst case tolerancing requires that all dimensions in the tolerance range must satisfy the assembled product specifications(stack-up conditions). This guarantees that the assembly should work properly even in the worst case combination of dimensions. However, worst case tolerancing typically results in over-conservative tolerances, which increase manufacturing cost. In

* Department of Mechanical Engineering, Seoul National University of Technology

statistical tolerancing, the tolerances must be assigned to the dimensions such that the probability for the assembly to function properly is above some specified acceptable amount(*yield*). In this approach, the individual tolerances are represented as probability distributions. Therefore statistical tolerancing allows more reasonable tolerances and hence lower manufacturing cost. In tolerance optimization research, two types of problems are typically recognized: tolerance allotment problems[1] and design centering problems.[2] The allotment of tolerances is closely tied to the overall quality and cost of a product.[3] If the tolerances are too loose, the probability for an assembly to function acceptably(*yield*) will be low. On the other hand, if the tolerance is too tight, the manufacturing cost will become high. Thus tolerance allotment becomes an optimization problem to determine the optimal allotment of the tolerances under the constraints of the function requirements and acceptance probability(*specification yield*). In the design centering problem, nominal dimensions are changed in order to find the maximum yield with fixed tolerances. Therefore the design variables are the nominal dimensions in the design centering problem. Typical numerical optimization methods require analysis, perturbation, and reanalysis in an organized algorithmic fashion.

2. Tolerance Optimization Problems

2.1 Cost Function

The *cost function model* for tolerances is the mathematical representation of the manufacturing costs in terms of the tolerances. In most cases monotonically decreasing functions are selected as cost functions because of the monotonic behavior of manufacturing costs for tolerances. Two common models are the *reciprocal squared model* and the exponential model. In the reciprocal squared model, the cost function is represented as

$$C(t) = \frac{a}{t^2} + f \tag{1}$$

where a is a constant for variable manufacturing cost, and f is a constant for fixed manufacturing cost.

In multi-dimensional model, total manufacturing cost can be obtained by summing the individual costs for each dimension:

$$C(t) = \sum_{j=1}^{n}(t_j) \tag{2}$$

or in the standard deviation domain:

$$C(\sigma) = \sum_{j=1}^{n}(\sigma_j) \tag{3}$$

where t_j are tolerances and σ_j are standard variables.

2.2 Yield

In many tolerancing problems, individual dimensions are assumed to have normal distributions. This assumption provides a method to relate the tolerances to the degree of satisfaction. Therefore the dimension vector x is assigned to have the multivariate normal distribution. The mean value μ_j of a dimension x_j is given as nominal dimension in the design process. However the standard deviation σ_j is selected according to the degree of precision achieved with each manufacturing process. As a consequence, the standard deviations σ_j are a function of the tolerances t_j. In mass production, the tolerance of a single dimension is considered as sufficiently large if 99.73% of the dimensions are in spec. For normally distributed random variables, there is a 99.73% probability that the variable will take on a value within $\pm\ 3\sigma$ of the nominal value. Therefore, in common cases, σ_j is set to $t_j/6$. As a result, the characteristics of the distributions are determined if t_j is given.

The domain of dimensions is divided into a *safe region* and a *failure region* by inequalities. Those inequalities are the design functions(i.e. constraints on the sum dimensions). The intersection of the safe region and the *acceptable tolerance region* is referred to as the *reliable region*. The reliable region depends on the standard deviation σ_j of each dimension since the tolerance region varies with σ_j. An important concept called *yield* is computed as the probability of x being in reliable region. Let x_{iu} and x_{il} represent the upper and lower limits of an individual dimension x_i in an assembly. Then the yield is represented as

$$Y = \int_{x \in R_R} \phi(x) dx \qquad (4)$$

where R_R represents the reliable region.

Several approaches have been tried to calculate the yield in tolerancing problems, including the *Taylor series method*, the *Monte Carlo methods* and *the approximation using a reliability index*.

2.3 Optimal Tolerance Allotment Problem

The yield increases as the tolerances become tighter. However manufacturing costs also increase as the tolerances become tighter. The effect of individual tolerance improvement on the total manufacturing cost increase is different for each tolerance. Therefore, the allotment of individual tolerances, which minimizes the cost under the design functions and minimum acceptable yield constraints, becomes an optimization problem. This optimal tolerance allotment problem is defined as:

$$\text{minimize } C(t) \qquad (5)$$

$$\text{subject to } Pr(R_R) > Y_{spec}$$

where $C(t)$ represents cost function,
t represents tolerance vector,
Y_{spec} represents spec yield,
R_R represents reliable region,
and $Pr(R_R)$ represents estimated yield.

If the integration of the probability density function is performed by a simulation scheme, the optimal tolerance allotment problem becomes a stochastic optimization problem. Lee and Woo[1] solved this problem using domain approximation schemes. Lee and Johnson[4] solved the same problem as a stochastic optimization problem using genetic algorithm and coarse Monte Carlo simulation and demonstrated that new strategy gives the optimal solution for the original problem.

2.4 Design Centering Problem

In real manufacturing situations, the precision of a given manufacturing process to produce a dimension might be fixed by the environment. In a cutting process, cutting precision is determined depending on the selection of a machine. In these situations a machinist adjusts the fixture setting by turning the knob or displacing the jig slightly. In terms of statistics, the machinist tries to put the center of the manufacturing distribution on the center of the reliable region by shifting the mean in order to maximize the yield. The purpose of *design centering* can be stated as "choose the center dimensions x_c of the design variables so that yield Y be maximized for a given distribution $\phi(x)$". Therefore the optimization problem is defined as:

$$\text{minimize } Y(x_c) = \int_{x1l}^{x1h} \mathrm{L} \int_{xnl}^{xnh} q(x_1, \mathrm{L}, x_n) \\ \phi(x_1, \mathrm{L}, x_n; x_c) dx_1 \quad dx_n \qquad (6)$$

where $\phi(x_1, \mathrm{L}, x_n)$ is the multivariate normal probability density function,
$q(x_1, \mathrm{L}, x_n)$ is a test function which checks whether a stochastically selected point is in the reliable region or in the infeasible region and is defined as
$q(x_1, \mathrm{L}, x_n) = 1,\ if\ F_i(x_1, \mathrm{L}, x_n) > 0$ for all design functions 0, otherwise
x_{nl}, x_{nh} means lower and higher limits of the dimension x_n.

Lee[5] suggested new strategy to solve this problem in stochastic way. Remarkable increases in yields were observed in solving sample problems.

3. Design Centering-Tolerance allotment Problem

3.1 Problem Definition

The optimal tolerance allotment problem is a rigid modeling problem. It minimizes the cost by reducing the tolerances while fixing the distribution centers on the nominal dimensions. The fact that the tolerance can be changed means that the precision of manufacturing processes can be changed in order to meet tolerance requirements. In the design centering problem, the tolerance is fixed under the assumption that the manufacturing processes and the precision of the machines are determined already. In a more realistic manufacturing situation, both the precision of manufacturing processes and the fixture settings could be changed simultaneously to increase the yield and minimize the manufacturing cost. As discussed in the previous chapter, design centering plays a significant role in improving the yield when the tolerances are fixed. The main idea of the *design centering-tolerance allotment* (DCTA) problem is the solution of the tolerance allotment problem while changing the design centers, i.e. the solution is sought with a more flexible attitude. Design variables in DCTA problems are the center dimensions and the standard deviations. The problem can be described as

$$\text{minimize } C(\sigma) \tag{7}$$

subject to $Y(x_c,\ \sigma) > Y_{spec}$

where x_c is the vector of dimension centers,

σ is the vector of standard deviations,

and Y_{spec} Yspec is a spec yield.

Penalty function methods are used to change the constrained optimization problem as the unconstrained optimization problem. By using the method, problem (7) is converted as

$$\text{minimize } C(\sigma) + r\left\langle \int_{x\in R_R} \phi(xc,\ \sigma) - Yspec \right\rangle^2 \tag{8}$$

$$where\langle a \rangle = a, \text{ if } a \text{ is positive}$$
$$0, \text{ otherwise}$$

and r is penalty coefficient and positive.

The formulation looks similar to that of the tolerance allotment problem except that the center dimensions are added as design variables and the dimensionality of the problem is increased from n to 2n as a consequence.

3.2 Genetic Algorithms

The major steps of a genetic algorithm[6] are generating a new population from a current generation according to established adaptation rules. An original genetic algorithm is composed of three steps; Fitness proportionate reproduction, Cross-over and Mutation.

In DCTA problems, the center dimensions and their corresponding standard deviations are the design variables. Therefore the population structure of this problem is

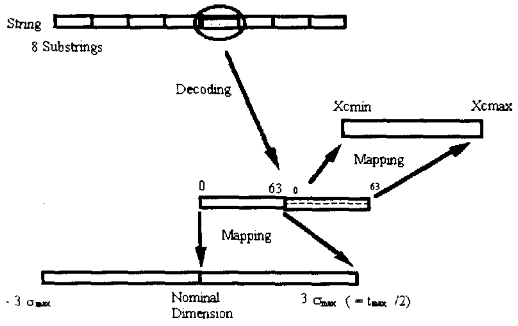

Fig. 1 Decoding and Mapping

$$\begin{aligned} x\sigma_{c1} &= \left((x_{c11},\ \sigma_{11}),\ (x_{c12},\ \sigma_{12}), \text{K}\ ,(x_{c1n},\ \sigma_{1n})\right) \\ x\sigma_{c2} &= \left((x_{c21},\ \sigma_{22}),\ (x_{c22},\ \sigma_{22}), \text{K}\ ,(x_{c2n},\ \sigma_{2n})\right) \\ &\vdots \\ x\sigma_{cp} &= \left((x_{cp1},\ \sigma_{p2}),\ (x_{cp2},\ \sigma_{p2}), \text{K}\ ,(x_{cpn},\ \sigma_{pn})\right) \end{aligned} \tag{9}$$

where p is the size of the population.

A substring is composed of the combination of a center dimension string and a standard deviation string. A center dimension and a standard deviation are each represented by six bit strings. Therefore a substring is a twelve bit binary string. When there are eight substrings corresponding to twelve dimensions, full string length is 96.

The decoding scheme involves two steps. At first the full string is divided into n substrings. Then the first part of a substring is mapped onto the center dimension of a variable. The precision of the center dimension discretization, π_c, is

$$\pi_c = \frac{x_{c\max} - x_{c\min}}{63} \tag{10}$$

Next the last part of the string is mapped onto the standard deviation domain of the variable. The precision of the tolerance discretization, π_σ, is

$$\pi_\sigma = \frac{3\sigma_{\max}}{63} \tag{11}$$

3.3 Monte Carlo Simulation

The sampling method of the original Monte Carlo simulation technique for evaluating probability density functions is based on the rejection method.[7] The rejection method

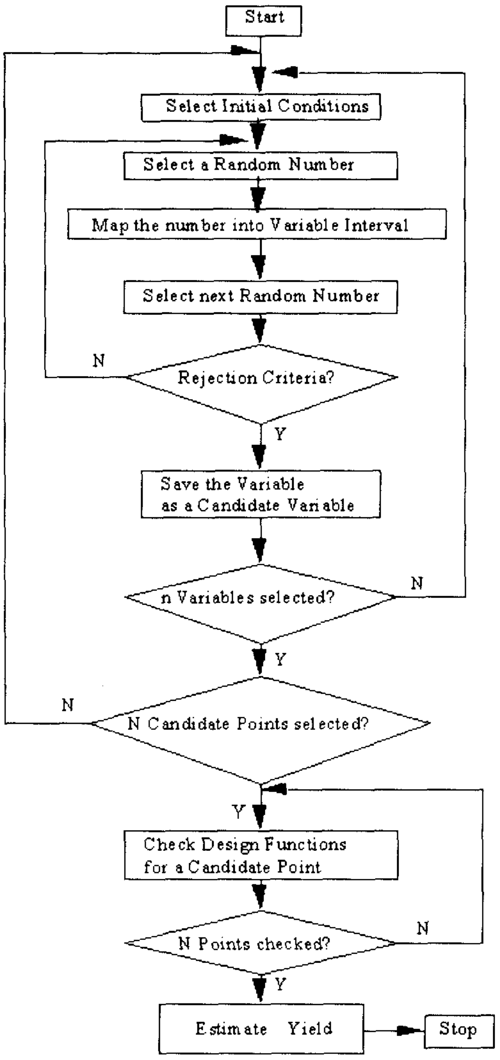
Start
Select Initial Conditions
Select a Random Number
Map the number into Variable Interval
Select next Random Number
Rejection Criteria?
N
Y
Save the Variable as a Candidate Variable
n Variables selected?
N
Y
N Candidate Points selected?
N
Y
Check Design Functions for a Candidate Point
N Points checked?
N
Y
Estimate Yield
Stop

Fig. 2 Monte Carlo Simulation for Yield Estimation

generates N sample points $x_1,\dots, x_N$ from the probability density function $\phi(x)$. After N sample points have been selected, the points are checked to see whether they are in the feasible region (i.e. satisfy design functions) or not. Then the estimated yield is

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{12}$$

where s_i is the number of points satisfying the design functions.

Because the standard deviation is a measure of the expected error of the estimation and the square root of the variance, the expected error is inversely proportional to the square root of the sample numbers N. The flowchart of the Monte Carlo simulation is shown in Figure 2.

As in the tolerance optimization problem, a function evaluation using simulation schemes would be a costly process in stochastic optimization problems. Therefore, if computation time can be saved by using approximate function evaluation schemes, more iterations can be performed in the given computation time. The outstanding performance of genetic algorithms in approximate function evaluation results from the genetic algorithms' nature of sampling, propagating, and reevaluating abilities along generations and population evolution. Even though a fitness of an individual string could decrease due to randomness of the crossover and mutation, the fitness of the population still increases as each generation proceeds. This observation makes genetic algorithms applicable to problems in which evaluation of candidate solutions can only be performed through Monte Carlo methods. This idea suggests that in some cases the overall efficiency of genetic algorithms may be improved by reducing the time spent on individual evaluations and increasing the number of generations performed.

3.4 DCTA Procedure

The procedure to solve the DCTA problem is presented next.

Step 1 Generate initial population. The initial population is a set of Np strings.
Strings are random combinations of 0 and 1.

Step 2 Divide a string into n substrings.

Step 3 Decode the first part of a substring.

Step 4 Map decoded value onto center dimension interval to get a candidate center dimension.

Step 5 Decode the last part of the string.

Step 6 Map decoded value onto tolerance interval.

Step 7 Repeat Step 3 through Step 6 until n substrings are decoded.

Step 8 Calculate the cost C for the tolerance vector.

Step 9 Estimate the yield using the Monte Carlo simulation.

Step 10 Evaluate the penalized cost C_p.

$C_p = C + (\text{penalty coefficient})(Y - Y_{spec})^2$, if $Y < Y_{spec}$
$C_p = C$, otherwise.

Step 11 Repeat Step 2 through Step 10 until all N_p strings are evaluated.

Step 12 Calculate average cost and set the maximum cost and the minimum cost.

Step 13 Rescale the fitnesses.

Step 14 Produce the population of the next generation by genetic algorithm: reproduction, crossover, and mutation.

Repeat Step 2 through Step 14 until the generation reaches the last generation to be evaluated.

3.5 Parameters

For genetic algorithms, the selection of parameters affects the behavior of the algorithm. DeJong[8] performed a study, which illustrates how the choices of the parameters affect the performance of genetic algorithms. For the test run, the parameters are selected as follows:

Population size	= 100
Cross-over Probability	= 0.7
Mutation Probability	= 0.005.

3.6 Experiment Schemes

The main purpose of the experiments is to explore how the robustness of the genetic algorithm can overcome the randomness of approximate Monte Carlo simulation in which extremely small number of sampling points are used. At the outset, it was assumed that the genetic algorithm would require more generations to reach a certain level of performance if the function evaluations were noisier. However the computation time for each iteration can be reduced by using the approximation strategy. Therefore the total computation time to reach a solution might be shorter when a coarse approximation is used. As discussed in section 3.3, the precision of the Monte Carlo simulation is inversely proportional to the square root of the sampling numbers. At the initial stage of the experiments, 100 sampling points were tried for the integration of an 8 dimensional multivariate probability density function. The precision of the estimation is 0.15 for one standard deviation. The algorithm converged to a good solution even for this noisy evaluation. The number of sampling points was gradually decreased until the algorithm would no longer converge to the solution. As the next step, various sampling numbers were tried in order to explore the relations between the computation efforts, the approximation, and the performance.

4. Results

A linear constraint problem is presented by modifying the Lee and Woo's problem.[1] New constraint set is

$$\begin{aligned}
F_1(x) &= -x_4 - x_5 + 5.005 \\
F_2(x) &= x_2 - x_1 - x_8 + x_7 - 0.0003 \\
F_3(x) &= x_7 - x_6 - x_3 + x_2 + 0.001 \\
F_4(x) &= x_4 - x_3 - x_6 - 0.0003 \\
F_5(x) &= x_4 + x_5 - 4.985 \\
F_6(x) &= -x_2 + x_1 + x_8 - x_7 + 0.0071 \\
F_7(x) &= -x_7 + x_6 + x_3 - x_2 - 0.005 \\
F_8(x) &= -x_4 + x_3 + x_6 + 0.0071.
\end{aligned}$$

Design functions represent the clearance condition for assembly. The reciprocal squared cost function model, equation (1), was modified and used to define total manufacturing cost as

$$C = \frac{a_i}{(6\sigma_i)^{bi}}(10^{-3}) \tag{13}$$

The coefficients in equation (13) were set as: $a_1 = a_2 = 1.0$, $a_3 = a_4 = 1.5$, $a_5 = 0.8$, $a_6 = 0.9$, $a_7 = 0.8$, and $a_8 = 0.6$; and $b_1 = 2.0$, $b_2 = 1.8$, b3 = 1.7, $b_4 = 2.0$, $b_5 = 3.0$, $b_6 = 2.0$, and $b_7 = b_8 = 1.9$. The specification yield is 95%.

Figure 4 shows the convergence of the algorithm with respect to computation effort. The algorithm converged faster when a smaller number of sampling points was used. The computational complexity of an algorithm can be represented by various measurements such as: the number of arithmetic operations, the number of function evaluations, the number of iterations, or computation time. The time complexity of the computation in this algorithm is the sum of the time spent for the Monte Carlo simulation and the time spent for the genetic algorithm. Simple experiments

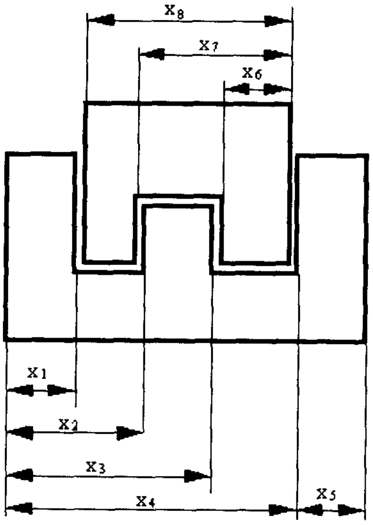

Fig. 3 Linear Constraint Problem

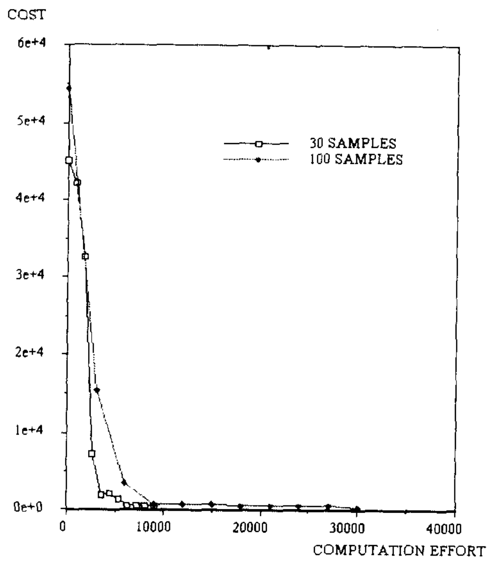

Fig. 4 Convergence w.r.t. Computation Effort

were performed to compare the computation time for the Monte Carlo simulation and the genetic algorithm. In performing the algorithm with 100 sampling points and 100 generations, the CPU time spent for the Monte Carlo simulation and the genetic algorithm was 216.2 seconds and 5.8 seconds each on a Pentium PC. Therefore the computation time for the genetic algorithm is very small compared to the time associated with the Monte Carlo simulation. Because the computation time for the Monte Carlo simulation is exactly proportional to the number of the sampling points, the computation effort is defined as the generation number multiplied by the sampling points number and can be used as a measurement of the time complexity. In solving this problem, averages of 171 were required for 10 sampling points. Therefore the computation complexity is 1,710 in this case.

The results of a test run are presented in Table 1. For the test run, the number of sampling points was 30 and the generation number was 300. The original problem was solved by Lee and Woo[1] using tolerance allotment and the minimum cost was about 1600. Lee and Johnson[4] solved same problem as a stochastic optimization problem using genetic algorithm and coarse Monte Carlo simulation and showed the minimum cost is 1500. However minimum

Table 1 The Result of DCTA

Variables	String 1	String 2	String 3
xc1	0.99929	0.99913	0.99913
xc2	1.99032	1.99032	1.99032
xc3	3.00214	3.00214	3.00214
xc4	3.99968	3.99905	3.99905
xc5	0.99770	0.99754	0.99770
xc6	0.99396	0.99396	0.99396
xc7	2.00905	2.00905	2.00873
xc8	2.99633	2.99681	2.99633
t1	0.00591	0.00486	0.00477
t2	0.00267	0.00210	0.00229
t3	0.00571	0.00229	0.00229
t4	0.00533	0.00419	0.00533
t5	0.01905	0.01905	0.01905
t6	0.00247	0.00285	0.00285
t7	0.00324	0.00324	0.00324
t8	0.00428	0.00428	0.00200
Cost	487.41 (509.46)*	528.89 (530.84)*	55.01
Yield	92.23%	94.47%	94.99%

xc; dimension centers, t; tolerances, string; candidate solutions
* penalized cost

cost by DCTA scheme is 550 while guaranteeing essentially the same yield. It can be recognized that the cost has been remarkably reduced (about 65%) by DCTA. This result shows that DCTA is shows great promise for problem modeling in realistic manufacturing situations.

5. Conclusion

In real manufacturing situations, both the precision of manufacturing processes and the fixture settings could be changed simultaneously to increase the yield and minimize the manufacturing cost. The main idea of the design centering-tolerance allotment (DCTA) problem is the solution of the tolerance allotment problem while changing the design centers, i.e. the solution is sought with a more flexible attitude. New problem type, design centering-tolerance allotment problem, was posed and solved with a remarkable cost reduction (about 65%) compared to conventional problem formulations. The DCTA problem formulation might be the most flexible way to solve tolerancing problems under the realistic manufacturing situation. This research suggests a new approach to solve other stochastic optimization problems. The combination of robust optimization methods and approximated simulation schemes would give promising results for many stochastic optimization problems, which are inappropriate for mathematical programming.

References

(1) Lee, W. J. and Woo, T. C., "Tolerances: Their Analysis and Synthesis", Journal of Engineering for Industry 112, pp. 113~121, May 1990.

(2) Director, S. W. and Hachtel, G. D., "The Simplicial Approximation Approach to Design Centering", IEEE Transactions on Circuits and Systems CAS-24, No. 7, pp. 363~372, July 1977.

(3) Bjϕrke, O., "Computer-Aided Tolerancing", ASME Press, New York, pp. 86~87,1989.

(4) Lee, J. and Johnson, G. E., "Optimal Tolerance Allotment using a Genetic Algorithm and Truncated Monte Carlo Simulation", Computer Aided Design, Vol.25, No.9, pp. 601~611, 1993.

(5) Lee, J., "Design Centering by Genetic Algorithm and Coarse simulation", Transactions of the Society of CAD/CAM Engineers, Vol.2, No.4, pp. 215~221, 1997.

(6) Goldberg, D. E., Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Publishing Company, Inc. Readings, Massachusetts, 1989.

(7) Siddall, J. N., Probabilities Engineering Design; Principles and Applications, Marcel Dekker, Inc. New York and Basel, pp. 151~160, 1983.

(8) DeJong, K. A., "An analysis of the behavior of the class of genetic adaptive systems" (Doctoral Dissertation, University of Michigan), Dissertation Abstracts International, 36(10), 5140B (University Microfilms No.76-9381), 1975.