한국 CAD/CAM 학회 논문집
제 2 권 제 4 호 1997년 12월 pp. 215-221

Design Centering by Genetic Algorithm and Coarse Simulation

Jinkoo Lee*

ABSTRACT

A new approach in solving design centering problem is presented. Like most stochastic optimization problems, optimal design centering problems have intrinsic difficulties in multivariate intergration of probability density functions. In order to avoid to avoid those difficulties, genetic algorithm and very coarse Monte Carlo simulation are used in this research. The new algorithm performs robustly while producing improved yields. This result implies that the combination of robust optimization methods and approximated simulation schemes would give promising ways for many stochastic optimizations which are inappropriate for mathematical programming.

Key words : Design centering, Stochastic optimization, Monte Carlo simulation, Genetic algorithm, Yield

1. Introduction

In tolerance optimization research, two problem types are typically recognized: *tolerance allotment* problems and *design centering* problems. The allotment of tolerances is closely tied to the overall quality and cost of a product. 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 (*spec yield*). On the other hand, the precision of a given manufacturing process to produce a dimension might be fixed by the environment in real manufacturing situations. 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 carefully. 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. If there exist many dimensions and complicated constraints, determining the optimal centering position by the intuition becomes virtually impossible.

*정회원, 서울산업대학교 기계공학과

2. Problem Definition

2.1 Yield

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_i of each dimension since the tolerance region varies with σ_i. 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

$$Y = \int_{x_{1l}}^{x_{1u}} \cdots \int_{x_{nl}}^{x_{nu}} q(x_1, \cdots, x_n)\, \phi(x_1, \cdots, x_n)\, dx_1 \cdots dx_n \quad (1)$$

where $\phi(x_1, \ldots, x_n)$ is the multivariate normal pro-

bability density function, and $q(x_1, ..., x_n)$ 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 follows:

$q(x_1, \cdots, x_n)=1$, if $F_i(x_1, \cdots, x_n)>0$ for all design functions

$q(x_1, \cdots, x_n)=0$, otherwise

On the other hand, in more condensed form, the yield is

$$Y = \int_{x \in R_R} \phi(x)\, dx \qquad (2)$$

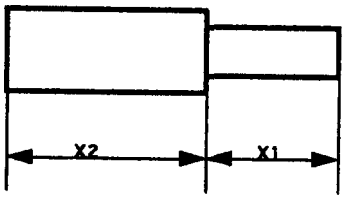
where R_R represents the reliable region.

As discussed earlier it is not easy to compute the yield Y from the set of tolerances (standard deviations) that constitute the multivariate normal probability density function $\phi(x)$ in equation (2) when the dimensionality becomes high.

Methods for calculating yield (i.e. multidimensional integration) can usually be classified as either *approximate methods* or *stochastic methods* (e.g. Monte Carlo simulation[1]). Several approaches have been tried to calculate the yield in tolerancing problems, including the *Taylor series method*[2], the *Monte Carlo methods*[3] and the *approximation using a reliability index*[4]. Other approximate methods try to calculate the yield by modifying the tolerance domain. In problems with low dimensions, original tolerance domains are approximated as simple regions such as spheres, cubics, or simplices[5-7]. Other approaches for approximating discrete distributions have been also tried[8,9].

2.2 Design centering problem

Let us consider the process to cut a large number of shafts as shown in Fig. 1. A lathe is used to cut the parts. In cutting procedures the worker sets the fixture and knows the precision of the machine. The machine can cut x_1 and x_2 in 0.012 and 0.018 tolerance range each with 99.73% reliability (i.e.$\pm$ 3σ range). If the dimensions are assumed to have normal distributions, the standard deviations for x_1 and x_2 are 0.002 and 0.003 each. In order to increase the 'in-spec' rate of the shafts, he might try to find the best fixture setting by intuition. Clearly



$1.998 < x_1 < 2.004$

$2.998 < x_2 < 3.005$

$4.997 < x_1 + x_2 < 5.007$

Fig. 1. Cutting a shaft.

the in-spec rate increases when the settings for x_1 and x_2 are on 2.0005 and 3.001 respectively compared to settings of 2 and 3. 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 yields by Monte Carlo simulation for each fixture setting are 56.08% for (2, 3) and 62.96% for (2.0005, 3.001). If there exist many dimensions and complicated constraints, determining the optimal centering position by the intuition becomes virtually impossible.

Fig. 3 shows the change of the design center increasing 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 $\phi(x)$". Since the standard deviations of

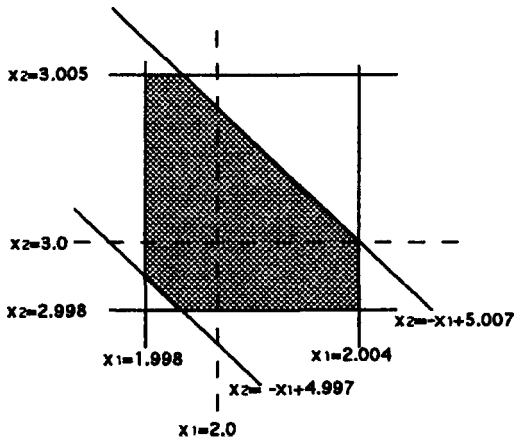


Fig. 2. Reliable region for the shaft.

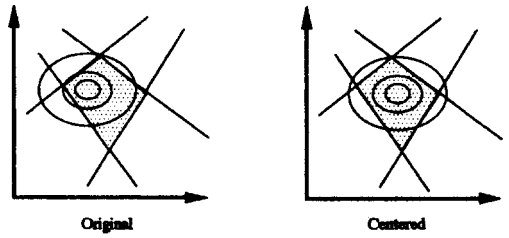


Fig. 3. Design centering for the assembly.

the distributions are known and fixed in this problem, the design centering problem can be defined to have two phases, namely;

a) determining the feasible region R_R and a design center, and

b) evaluating the yield for the design center.

Therefore the optimization problem is defined as:

$$\text{maximize } Y(x_c)= \int_{x_{l1}}^{x_{u1}} \cdots \int_{x_{ln}}^{x_{un}} q(x_1, \cdots, x_n)\, \phi(x_1, \cdots, x_n; x_c)\, dx_1, \cdots, dx_n \quad (3)$$

where $\phi(x_1, \cdots, x_n)$ is the multivariate normal probability density function, and $q(x_1, \cdots, x_n)$ 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

$q(x_1, \cdots, x_n)=1$, if $F_i(x_1, \cdots, x_n) > 0$ for all design functions

$q(x_1, \cdots, x_n) = 0$, otherwise

or

$$\text{maximize } \int_{x_{l1}}^{x_{u1}} \cdots \int_{x_{ln}}^{x_{un}} \phi(x_1, \cdots, x_n; x_c)\, dx_1, \cdots, dx_n \quad (4)$$

subject to $F_i(x_1, x_2, \cdots, x_n; x_c)>0$ for all design functions.

Therefore the variables x_i's are stochastic, whereas x_c is a vector of deterministic variables.

3. Basis of New Approach

3.1 Genetic algorithm

The basic structure processed by the genetic algorithm[10] is the *string*. The strings in artificial systems are analogous to chromosomes in biological systems. In natural systems, the *chromosome* (or set of chromosomes) is a prescription of the biological unit. In artificial systems, the string is a description of the parameter set. A chromosome is composed of genes which take on the number of values called *alleles*. A *gene* is also identified by its position on the chromosome called its *locus*. In most cases, a particular string is represented in its binary form.

Genetic algorithms generate one population of individuals at a time. As time proceeds, new generations are born and older generations die away creating constantly changing populations. Genetic algorithms generate a sequence of string populations in the same manner. Therefore the population is the body of knowledge containing a multitude of notions and ranking of those notions for task performance. Genetic algorithms can keep a diversity of population by adapting themselves.

The major steps of a genetic algorithm are generating a new population from a current generation according to established adaptation rules. An original genetic algorithm is composed of three steps:

1. Fitness proportionate reproduction
2. Cross-over
3. Mutation

3.2 Monte Carlo simulation

The sampling method of the original Monte Carlo simulation technique for evaluating probability density functions is based on the rejection meth od[11]. The rejection method generates N sample points $x_1, \cdots, x_N$ from the probability density function $\phi(x)$. For a one dimensional probability density function $\phi(x)$, calculate the maximum value ϕ_{max}. Two random numbers, r_1 and r_2, are generated uniformly between 0 and 1. Then the selected dimension is mapped onto a real selection domain by

$$x = r_1(u - l) + l \quad (5)$$

where u and l are the upper limit and the lower limit of the domain.

If

$$r_2 \le \frac{\phi(x)}{\phi_{max}} \quad (6)$$

the point x is accepted as a candidate point. Oth-

erwise, repeat the selection again until N sample points are selected. 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 $\hat{Y}$ is

$$\hat{Y} = \frac{1}{N}\sum_{i=1}^{N} s(x_i) \simeq \frac{1}{N}\sum_{i=1}^{N} s_i \tag{7}$$

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

In order to examine the relation between the precision of the simulation and the number of the sampling points, let us assume N_t trials were performed. Then the average $\hat{Y}$ of estimated yield $\hat{Y}_i$ is

$$\bar{Y} = \frac{1}{N_t}\sum_{i=1}^{N_t} \hat{Y}_i \tag{8}$$

Then the mean of $\bar{Y}$ is

$$E(\bar{Y}) = \mu_Y$$

where μ_Y is the population mean, and the variance is

$$e[(\bar{Y} - \mu_Y)^2] = \frac{\sigma_Y^2}{N_t} \tag{9}$$

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.

3.3 Integration of genetic algorithm and monte carlo simulation

In tolerance optimization problems, evaluation of the yield has been the critical point in the optimization processes. Accurate Monte Carlo simulation is not feasible due to the high computation cost because many function evaluations are typically required by numerical optimization schemes. A reasonable idea can be raised by carefully considering the nature of genetic algorithms and the difficulties of the yield estimation. Very coarse Monte Carlo simulation in evaluating multivariate integration yield could be used if genetic algorithm robustly guides the search. Similar approach showed good performance in optimal tolerance allotment problems(12) which determine a set of tolerances minimizing total manufacturing cost with the constraints of function requirements and minimum yield.

Each iteration of the proposed procedure is composed of two steps. At the first step, a set of variables is selected as the initial population. Then the yield is evaluated by Monte Carlo simulation in the analysis step. As a synthesis step, genetic algorithm perturbs the population using the three steps: reproduction, cross-over, and mutation.

4. Problem Modeling

In the design centering problem, the objective is to maximize the yield. Because the original idea of the genetic algorithm is the maximization of the fitness function, the fitness proportionate reproduction scheme can be used without any fitness inversion. However, in order to prevent the algorithm from premature convergence and stalling, the modified linear fitness scaling scheme is used.

The design variables are center dimensions in the design centering problem. Therefore the population structure of this problem is:

$$\begin{aligned} x_{c1} &= (xc_{11}, xc_{12}, \cdots, x_{c1n}) \\ x_{c2} &= (x_{c21}, x_{c22}, \cdots, x_{c2n}) \\ &\vdots \\ x_{cp} &= (x_{cp1}, x_{cp2}, \ldots, x_{cpn}) \end{aligned} \tag{10}$$

where p is the size of the population.

A string is composed of n substrings. Each substring represents the candidate design centers for each dimension. A substring is mapped onto the design center interval $[x_{cmin}, x_{cmax}]$. When a string is composed of 8 substrings which have binary string of length 6, the full length of the string is 48. Then the precision of the discretization, π, is

$$\pi = \frac{x_{cmax} - x_{cmin}}{63}$$

The flowchart of the tolerance centering is

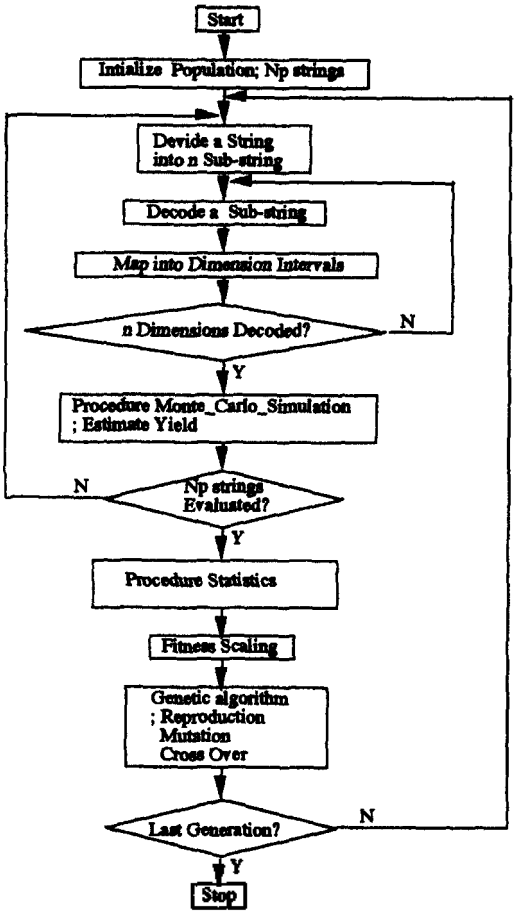
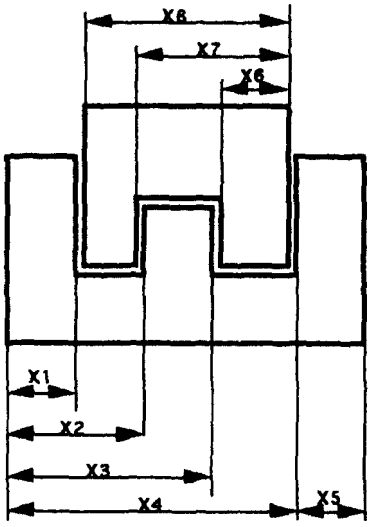


Fig. 4. Flowchart of design centering.



$F_1(x) = -x_4 - x_5 + 5.005$

$F_2(x) = x_2 - x_1 - x_8 + x_7 - 0.0003$

$F_3(x) = x_7 - x_6 - x_3 + x_2 + 0.001$

$F_4(x) = x_4 - x_3 - x_6 - 0.0003$

$F_5(x) = x_4 + x_5 - 4.985$

$F_6(x) = -x_2 + x_1 + x_8 - x_7 + 0.0071$

$F_7(x) = -x_7 + x_6 + x_3 - x_2 - 0.005$

$F_8(x) = -x_4 + x_3 + x_6 + 0.0071$

Fig. 5. Example for design centering.

shown in Fig. 4.

5. Results and Analysis

An example problem is selected and modified from Lee and Woo's paper[5]. The shape of the assembly is shown in Fig. 5. Design functions represent the clearance conditions for the assembly.

The tolerance for each dimension is set as follows; t_1=0.0040, t_2=0.0023, t_3=0.0025, t_4=0.0053, t_5=0.0143, t_6=0.0021, t_7=0.0015 and t_8=0.0020. The yield for the given tolerance set and nominal dimensions is calculated by the Monte Carlo simulation with the design functions. The calculated yield for original nominal dimension is 57.17% (simulation 100,000 times). The results of a test run are given in Table 1. For the test run, the number of sampling points was 30 and the generation number was 150. The maximized yield is 79.11%. Therefore the yield has increased about 22% by adjusting the centers.

Fig. 6 shows the convergence of the algorithm for 30 sampling points and 100 sampling points. At the initial stage, the yields are extremely small due to the random selection of the candidate dimensions. As the generation proceeds, the yields approach to maximum value. As expected, the algorithm converges in fewer generations when larg-

Table 1. The results design centering

Variable	Tolerance	Nominal Dimension	Design Center
x_1	0.0040	1.0	0.99981
x_2	0.0023	2.0	2.00143
x_3	0.0025	3.0	3.00085
x_4	0.0053	4.0	4.00194
x_5	0.0143	1.0	0.99433
x_6	0.0021	0.998	0.99810
x_7	0.0015	2.0	2.00012
x_8	0.0020	2.998	2.99873
Yield		57.17%	79.71%

er sampling numbers are used for the Monte Carlo simulation. 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 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

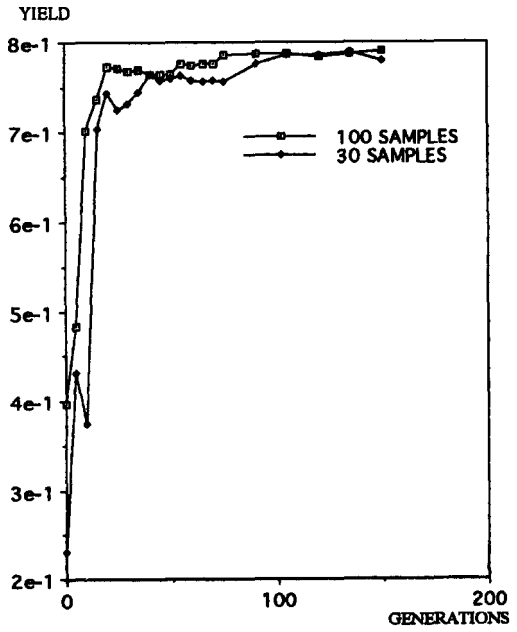


Fig. 6. Convergence w.r.t. Generations.

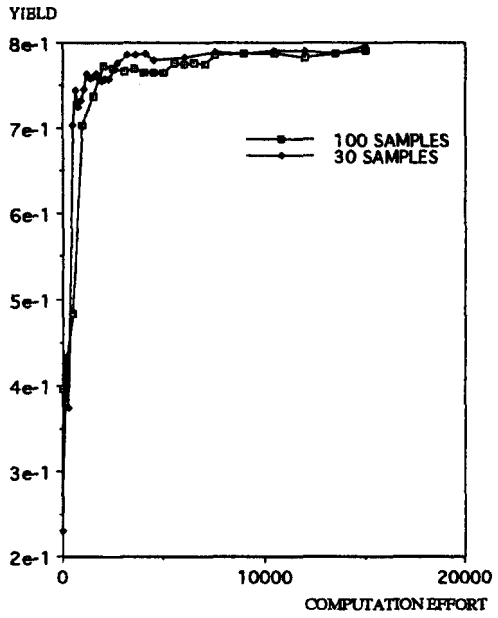


Fig. 7. Convergence w.r.t. Computation effort.

simulation and the genetic algorithm was 776 seconds and 9.2 seconds each on Apollo DN 5500 workstation. 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 this product can be used as a measurement of the time complexity. Fig. 7 shows the convergence of the algorithm in terms of the computation effort for each case. These two figures show that the algorithm converges faster when smaller number of sampling points are used even though it takes more generations.

6. Conclusions

A strategy which is presented in this paper can be utilized in appropriate manufacturing situations and will give good solutions in modest computing time. In fact, solving design centering problems is not trivial. The cost function modeling and the de-

termination of the design functions take substantial time and effort. Therefore the effort for modeling the problem would be much larger than that for solving the problem. Some domain approximation schemes have advantages in short computation time. However those algorithms often give non optimal solutions with deviated yields. Therefore, the results are inappropriate to be utilized in practical design tasks.

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.

Useful extensions of this research can be suggested for future work. Genetic algorithms have been studied as tools for AI (*artificial intelligence*) with remarkable success. In tolerancing problems, problem modeling procedures are as important as solution procedures. Shape tolerances would be represented by more complicated mathematical forms. Other design factors (i.e. material strength, endurance period, fatigue effect, or reliability) also could be considered as stochastic design variables in a design project. In those situations, problem modeling becomes complicated. Other extensions can be suggested from a practical view point. A well designed user interface will facilitate the use of the new algorithms. The algorithms would be more useful if the simulation for arbitrary non-normal distributions (e.g. gamma distribution) are implemented because the distributions in real world are not always normal.

References

1. Hammersley, J.M. and Handscomb, D.C., *Monte Carlo Methods*, Methuen, London, 1964.
2. Johnson, R.H., "How to Evaluate Assembly Tolerances", *Product Engineering*, 179-181, Jan. 1953.
3. Grossman, D.D., "Monte Carlo Simulation of Tolerancing in Discrete Parts Manufacturing and Assembly", Report No. STAN-CS-76-255, Computer Science Department, Stanford University, 1976.
4. Hasofer, A.M. and Lind, N.C., "Exact and Invarient Second-Moment Code Format", *Journal of Engineering Mechanics Division*, ASCE, Vol. 100, pp. 111-121, 1974.
5. Lee, W.J. and Woo, T.C., "Tolerances: Their Analysis and Synthesis", *Journal of Engineering for Industry*, Vol. 112, pp. 113-121, May. 1990.
6. Good, I.J. and Gaskins, R.A., "The Centroid method of numerical integration", *Numerical Mathematics*, Vol. 16, pp. 343-359, 1971.
7. 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.
8. Wets, R., "The distribution problem and its relations to other problems in stochastic programming", in: Stochastic Programming (e.d. M. A. H. Dempster), pp. 245-262, 1980.
9. Kall, P. and Stoyan, D., "Computational methods for solving two-stage stochastic linear programming problems with recourse including error bounds", *Mathematische Operationsforschung und Statististiks*, Vol. 13, pp. 431-447, 1982.
10. Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning*, Addison_Wesley Publishing Company, Inc., Readings, Massachusetts, 1989.
11. Siddall, J.N., Probabilities Engineering Design; Principles and Applications, Marcel Dekker, Inc., New York and Basel. pp. 151-160, 1983.
12. 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.

이 진 구

1983년 연세대학교 기계공학과 학사
1991년 University of Michigan 기계공학 석사
1992년 University of Michigan 기계공학 박사
1983년 - 1997년 대우전자 영상연구소 책임 연구원
1998년 ~현재 서울산업대학교 기계공학과
관심분야 : CAD/CAM/CAE, Design Optimization, Mechatronics system design