A NEW ALGORITHM FOR DISCRETE TOLERANCE OPTIMIZATION

M. H. Gadallah & H.A. El Maraghy

Mechanical Engineering Department
McMaster University, Hamilton, Ontario, Canada, L8S 4L7

Abstract

A new algorithm has been developed which deals with the problem of least cost tolerance allocation with process selection. This algorithm uses the combinatorial nature of orthogonal arrays and experimental optimization techniques to allocate the magnitude of tolerance to each design dimension and select the corresponding manufacturing process. Interaction graphs are used to assign the dimensional tolerances to various orthogonal array structures. The proposed algorithm is capable of dealing with continuous and discrete cost functions as well as linear, nonlinear and multi-loop assembly functional requirements. Several examples are used to illustrate the effectiveness of the developed technique. Results indicate the superiority of the developed algorithm with those obtained using discrete, combinatorial, combined discrete and continuous and sequential quadratic programming.

Keywords

Discrete Optimization , Tolerance Allocation

1. Introduction

The problem of tolerance analysis and synthesis has attracted attention due to its importance for achieving parts functionality and assembly requirement. The classical tolerance problem is to minimize manufacturing cost subject to process constraints. It assumes that each dimension can be produced by one, and only one, process and is formulated as a continuous problem. With the introduction of discrete optimization techniques, the problem is formulated as a minimization of production costs with and without precision limits. Each design dimension can also be produced by one or more processes.

2. Literature Review

Speckhart [9] presented a workable analytical method for locating the optimum set of dimensional tolerances that minimize manufacturing costs and meet the imposed restraint conditions. A similar approach was presented by Spotts [7]. Wilde and Prentice [11] showed that the least–cost allocation of sure–fit machine tolerances for Speckhart's exponential cost model can be solved in closed form without numerical iteration and zero degree of difficulty. Later, Sutherland and Roth [8] presented design algorithms that account for the manufacturing cost and statistical manufacturing tolerance effects for function generating mechanisms. Ostwald and Huang [5] introduced a method for specifying independent functional tolerances using 'Zero – One ' algorithm.

Wu et al. [10] presented an evaluation of different algorithms for design tolerance analysis and synthesis. Lee and Woo [3] used a branch and bound algorithm to perform tolerance analysis by approximating the volume under the multi–variate probability density function constrained by nonlinear stack–up conditions.

Chase et al. [1] presented a discrete optimization scheme that deals with the combinatorics resulting from alternative manufacturing processes, ranges of dimensional tolerances and associated cost curves. Zhang and Wang [12] dealt with the same problem using a Simulated Annealing algorithm and the results were compared with the Sequential Quadratic Programming method (SQP).

3. Methodology

In this article, a new algorithm for tolerance allocation and optimum process selection is presented. The problem is viewed as a search in two domains: the first is tolerance allocation to satisfy the assembly functional requirement and the second is process–selection to minimize production cost. The search algorithm couples an inner array representing the tolerance selection domain and an outer array representing the process selection domain. The choice of different structures of orthogonal arrays has a tremendous impact on the resulting minimum production cost and the corresponding optimum tolerances. Each orthogonal array is represented by a search graph which assists the designer in the initial assignment phase. An example is used to illustrate the use of inner/outer orthogonal arrays to allocate tolerances and corresponding manufacturing processes.

4. Modeling Of Tolerance Allocation With Process Selection Using An Inner/Outer Orthogonal Arrays

Consider an assembly problem with two design dimensions X_1 and X_2 using an L9OA (9 experiments Orthogonal Array) as shown in fig. 1. In this case, X_1 and X_2 will have three design settings corresponding to the first, second and third levels. Therefore, (X_{11}, X_{12}, X_{13}) and (X_{21}, X_{22}, X_{23}) will correspond to levels 1, 2 and 3 of design dimension 1 and 2 respectively. For a complete evaluation of different structures of orthogonal arrays, the reader is referred to [2].

Inner Array

Row	Column 1	Column 2
1	X_{11}	X_{21}
2	X_{11}	X_{22}
3	X_{11}	X_{23}
4	X_{12}	X_{21}
5	X_{12}	X_{22}
6	X_{12}	X_{23}
7	X_{13}	X_{21}
8	X_{13}	X_{22}
9	X_{13}	X_{23}

Outer Array

Row	Column 1	Column 2
1	P_{11}	P_{21}
2	P_{11}	P_{22}
3	P_{11}	P_{23}
4	P_{12}	P_{21}
5	P_{12}	P_{22}
6	P_{12}	P_{23}
7	P_{13}	P_{21}
8	P_{13}	P_{22}
9	P_{13}	P_{23}

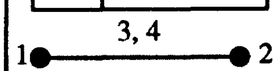

Fig. 1 An Inner–Outer Orthogonal Array and Corresponding Interaction Graph

The process–cost curves can be modeled using an outer orthogonal array. For instance, instead of assuming that each dimension corresponds to only one process, we will consider three process curves. In reality, each design dimension can be produced by one or more manufacturing processes. In addition, each process has precision limits. These two concerns should be included in the optimization model. In the outer array, P_{11}, P_{12} & P_{13} correspond to the cost of producing design dimension 1 using processes 1, 2 and 3 respectively. Similarly, P_{21}, P_{22} & P_{23} correspond to the cost of producing design dimension 2 using processes 1, 2 and 3 respectively.

This logic results in a combinatoric scheme for the tolerance allocation and process selection problem. Assume that the designer wishes to choose design levels X_{11} and X_{21} (first row using L9 OA), these two design dimensions can be produced using the following combinations of cost process curves: a) (P_{11}, P_{21}); b) (P_{11}, P_{22}); c) (P_{11}, P_{23}); d) (P_{12}, P_{21}); e) (P_{12}, P_{22}); f) (P_{12}, P_{23}); g) (P_{13}, P_{21}); h) (P_{13}, P_{22}) and i) i) (P_{13}, P_{23}) respectively. Therefore, the tolerance allocation and process selection domains are approximated using 9 experiments, in each domain, and 81 combinations.

5. Search Graph Techniques

The four interaction graphs used are presented in this section. Fig. 2 a shows an interaction graph for an L16 OA. This array is usually used for 2 and 3 design levels. Design dimensions are assigned to columns 2 (or 3), 4 (or 5), 8 (or 9) (three–level–designs) and columns 10,11,12 and 13 (two–level–designs) to ensure orthogonality and independence during the search. This is particularly useful for mixed two–three cost–process curves.

Fig. 2 b shows an interaction graph for an L27 OA. This array is used for three–level–designs with design dimensions assigned to columns 1, 2, 5, 8, 11, 14, 17 and 20 respectively. Fig. 2 c shows an interaction graph for an L64 OA. This array is used for two–level–designs with design dimensions assigned to columns 1, 2, 4, 8, 23, 16, 32, 45, 27, 42, 52, 14, 15, 28, 29, 30, 49, 51, 46, 35 and 39 respectively. Fig. 2 d shows an interaction graph for an L81 OA. This is used for three–level–design with design dimensions assigned to columns 1, 2, 5, 14, 26, 29, 35, 38, 9, 10, 12, 13, 18, 19, 21, 22, 24, 25, 33 and 34 respectively. The three design levels considered are: $\delta - \Delta$, δ, $\delta + \Delta$, where δ and Δ are the tolerance value and tolerance level difference respectively. At each iteration 16, 27, 64 and 81 design points are evaluated to approximate the design space using an L16 OA, L27 OA, L64 OA and L81 OA respectively. The assembly functional requirement is evaluated at each design point. The design space for the cost–process curves is approximated in a similar manner. The algorithm selects the minimum cost–process and the corresponding tolerance levels are chosen accordingly as the base point for the next iteration. This procedure continues until an optimum is reached. This problem formulation depends upon searching in two domains; however, the size of the combinatoric problem is much smaller than that encountered when using the usual exhaustive search techniques.

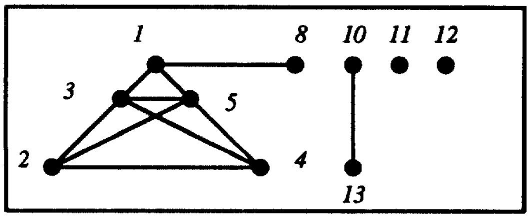

Fig. 2 a An Interaction Graph For An L16 OA

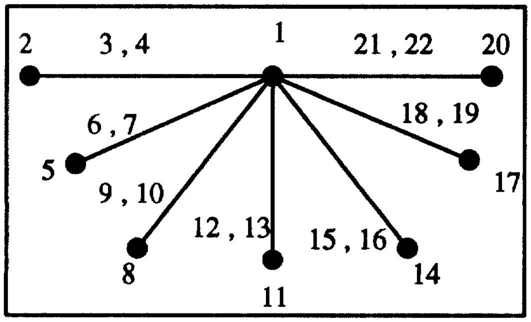

Fig. 2 b An Interaction Graph For An L27 OA

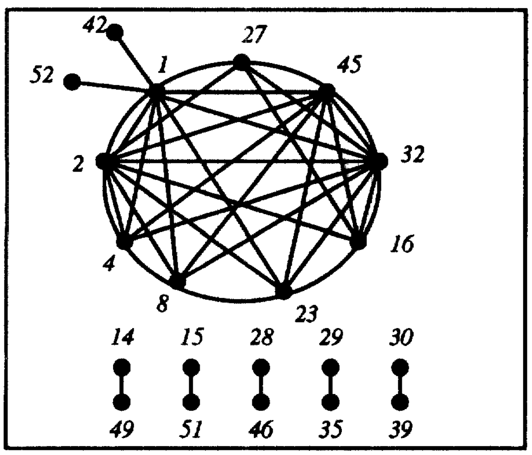

Fig. 2 c An Interaction Graph For An L64 OA

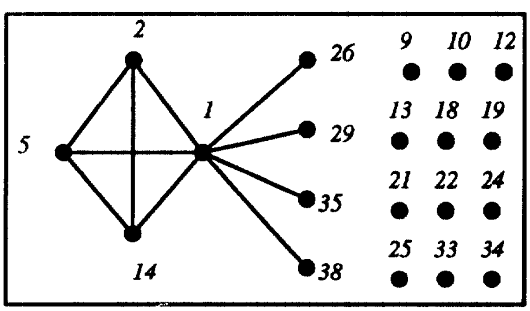

Fig. 2 d An Interaction Graph For An L81 OA

6. Tolerance Allocation With Process Selection Algorithm

1. Assign the design dimensions to a suitable orthogonal array. A three level orthogonal array is usually used to plan experimentation for optimum tolerance selection.

2. Assign process curves to a suitable orthogonal array. The choice of orthogonal array will depend on the number of process curves for each design dimension. For instance, if an assembly with two design dimensions produced using 2 and 3 process curves, a mixed two–three orthogonal arrays should be used. This allows varying a two–level design parameter with a three–level design parameter. In each case, confounding of either design design dimensions or process curves should not be allowed.

3. The search for tolerance allocation is constrained by: a) the range of process precision and b) overall assembly functional requirements. The search for minimum costs considers one process at a time for each design dimension and the cost must always be greater than zero.

4. An initial feasible starting point, (called seed), which has to satisfy assembly functional requirement as well as process precision limits is used.

5. Let an L9 OA be used to plan experimentation for both tolerance allocation (i = 1, 9) and process selection (j = 1,9).

for i = 1, 9, 1

for j = 1, 9, 1

$$\min f\ (\ P_{ij}\ ,\ \delta_{ij}\ )\ =$$

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} [\ P_{ij}\ .\ y(\ \delta_{ij}\ )\ ]$$

subject to:

$$\sum\ P_{ij}\ .\ \delta_{ij}\ \leq t_k \quad \text{worst case analysis}$$

$$\delta^{l} \leq\ \delta_{ij}\ \leq \delta^{u} \qquad \sum P_{ij}\ =\ 1$$

next j;

next i;

Where :

f = Total machining cost of individual tolerances.

δ_{ij} = Manufacturing tolerance of the ith component dimension produced by process j.

δ^{u} , δ^{l} = Upper and lower bounds on tolerance δ_{ij} .

n , m = Number of component dimensions and available processes respectively.

$y(\ \delta_{ij}\ )$ = Cost of producing tolerance δ on the ith component by process j.

6. The process continues until an optimum is reached.

7. Examples

The described algorithm has been implemented and verified using 8 examples A–I taken from Chase et al.[1]. Table 1a shows the layout assignment for the 8 problems. The number of dimensions, number of processes, number of iterations and optimum costs using the Orthogonal–based–algorithm are given in table 1b. Since the same problems have been solved using other search techniques, the optimum cost, number of possible combination and CPU(time) are given in table 2 for comparison. These search techniques include discrete methods (Balas Zero–One and combinatorial) and continuous methods (Sequential Quadratic Programming) and combined discrete and continuous methods. The algorithm developed in this paper will be referred to as 'Orthogonal–based algorithm'. In problems A–I, the optimum costs obtained using the Orthogonal–based algorithm are less than those obtained using Balas Zero–One, Combinatorial methods and combined discrete and continuous methods.

In problem E with 3 assembly loop equations, the optimum cost obtained is almost the same as those using the Balas Zero–One and combinatorial methods. The number of possible combinations vary from 256 to 5184 depending on the size of inner and outer orthogonal arrays. The trial combinations performed using the Orthogonal–based–algorithm are about 12% of those performed using the Balas algorithm. The CPU (time) used by the Orthogonal–based agorithm is generally less than that used by Balas Zero One but higher than the time used by the combinatorial methods. In problems H and I, the Balas algorithm failed to obtain a solution in a reasonable time.

The optimum costs obtained using Orthogonal–based algorithm are always higher than those obtained using the Sequential Quadratic techniques (SQP). This is due to the ability of SQP algorithm to split each design dimension between two or more processes. For instance, a dimension is processed 80% using process 1 and 20% using process 2. Clearly, this is worthless from manufacturing point of view as pointed out in [1,13].

The Orthogonal–based algorithm performs very well for problems with multiple assembly loop equations. For instance, problem E is solved using two global optimization search algorithms: the exhaustive and Univariate methods. Both methods failed to obtain a solution since they can not handle multiple assembly loops.

8. Conclusion

1. The formulation based on an inner/outer orthogonal arrays is capable of dealing with the problem of tolerance allocation and optimum process selection.

2. For all problems tested, the Orthogonal–based algorithm was able to provide solutions better than Balas, combinatorial methods (discrete), Sequential Quadratic Programming and combined discrete and continuous methods

3. The Orthogonal–based algorithm is capable of dealing with multi–loop assembly problems. The exhaustive and univariate search techniques encountered difficulties during the search and the cost of solution became unreasonably high.

4. The graph associated with each orthogonal array makes the layout assignment of either design dimensions or process–cost–curves systematic. Therefore,the Orthogonal–based algorithm can be used as an additional design tools for the problem of tolerance allocation with process selection.

References

[1]Chase K.W., Greenwood W.H., Loosli B.G. and Hauglund L.F., " Least Cost Tolerance Allocation For Mechanical Assemblies With Auto mated Process Selection", Manufacturing Review Vol. 3 , no. 1 , 1990,pp. 49 – 59.

[2]G. Taguchi System Of Experimental Design , Vol. 1 and 2 Unipub , American Supplier Institute , 1987.

[3]Lee W. and Woo T.C. "Optimum Selection Of Discrete Tolerances ", Journal Of Mechanisms , Transmission and Automation In Design, Vol. 111 , 1989, pp. 243–251.

[4]Michael, W. ,Siddall J.N. , "The Optimal Tolerance Assignment With Less Than Full Acceptance", Journal Of Mechanical Design , Vol. 104 , 1982, pp. 852 – 860.

[5]Ostwald , P.F. " A Method For Optimal Tolerance Selection ", Journal Of Engineering For Industry , 1977, pp. 558 – 565.

[6]Peters , J. " Tolerancing The Components Of An Assembly For Minimum Cost ", Journal Of Engineering For Industry , 1970, pp. 677 – 682.

[7]Spotts , M.F. " Allocation Of Tolerances To Minimize Cost Of Assembly ", Journal Of Engineering For Industry , 1973 , pp. 762 – 764.

[8]Sutherland , G.H. and Roth , B. " Mechanism Design: Accounting For Manufacturing Tolerances and Costs In Function Generating Problems ", Journal Of Engineering For Industry, 1975 , pp. 283 – 286.

[9]Speckhart F.H. " Calculation Of Tolerance Based On A Minimum Cost Approach ", Journal Of Engineering For Industry, 1972 , pp. 447 – 453.

[10]Wu Z., ElMaraghy W.H.,ElMaraghy H.A. , "Evaluation Of Cost Tolerance Algorithms For Design Tolerance Analysis And Synthesis" , ASME,Manufacturing Review,1988, pp. 168 – 179.

[11]Wilde , D. and Prentice , E. " Minimum Exponential Cost Allocation Of Sure Fit Tolerances ", Journal Of Engineering For Industry, 1975, pp. 1395 – 1398.

[12]Zhang C. and Wang H.P., "The Discrete Tolerance Optimization Problem" , Manufacturing Review Vol. 6, no. 1, 1993 , pp. 60–71.

A ■	A ●	B ■	B ●	C ■	C ●	D ■	D ●	E ■	E ●	F ■	F ●	H ■	H ●	I ■	I ●
1	2	1	1	1	12	1	1	1	1	1	6	1	1	1	1
2	10	2	2	2	1	2	2	2	2	2	7	2	2	2	2
3	11	3	4	3	2	3	4	3	12	3	8	3	5	3	5
4	12	4	5	4	3	4	6	4	14	4	9	4	9	4	9
		5	6	5	4	5	7	5	17	5	10	5	10	5	10
		6	12	6	5	6	12	6	3	6	11	6	12	6	12
				7	6	7	14	7	19	7	12	7	13	7	13
						8	17	8	4	8	13	8	14	8	14
										9	14	9	18	9	18
										10	15	10	19	10	19
										11	16	11	21	11	21
										12	17	12	22	12	22
														13	24
L16 OA		L16 OA		L16 OA		L16 OA		L36 OA		L64 OA		L81 OA		L81 OA	

■ Dimension Number

● Column Number

Table 1a Layout assignment for problems A–I

Problem	Number of components	Number of processes	Number of iterations	CPU (second)	Optimum cost ($)
A	4	10	23	4.30	22.2198
B	6	13	4	3.20	30.0650
C	7	15	3	3.10	19.1370
D	8	19	9	6.20	34.3426
E	8	20	67	38.30	5.1400
F	12	24	20	30.00	80.9461
H	12	36	24	40.50	59.6906
I	13	38	19	31.80	67.3659

Table 1b Efficiency & optimum costs for problems A–I using Orthogonal–based algorithm

Problem		Number Of Loops	Discrete Balas	Discrete Combinatorial	Continuous SQP	Combined	Orthogonal Based Algorithm
A		1	$ 25.00	$ 25.00	$ 20.90	$ 21.20	$ 22.2198
	Number Of Possible Combinations		205	36	335		256
	CPU time (sec)		0.95	0.00	23.05		4.30
B		1	$ 36.00	$ 36.00	$ 25.85	$ 35.00	$ 30.065
	Number Of Possible Combinations		681	96	512		1296
	CPU time (sec)		5.80	0.00	25.85		3.20
C		1	$ 31.00	$ 31.00	$ 18.05	$ 30.21	$ 19.137
	Number Of Possible Combinations		1,344	192	371		1296
	CPU time (sec)		15.97	0.030	18.05		3.10
D		1	$ 40.00	$ 40.00	$ 31.69	$ 34.90	$ 34.3426
	Number Of Possible Combinations		3,275	864	858		1296
	CPU time (sec)		53.68	0.320	31.69		6.20
E		3	$ 5.110	$ 5.110	$ 4.27	*****	$ 5.1403
	Number Of Possible Combinations		10,839	1296	931		1296
	CPU time (sec)		190.90	0.430	185.92		38.30
H		1	****	$ 77.00	$ 54.53	$ 74.54	$ 59.6906
	Number Of Possible Combinations		****	531,441	2,335		5184
	CPU time (sec)		****	213.23	400.35		40.50
I		1	****	$ 79.00	$ 56.05	$ 76.48	$ 67.3659
	Number Of Possible Combinations		****	1,062,882	2,540		5184
	CPU time (sec)		**** †	470.97	460.40		31.80

† signify that the search algorithm used could not reach a solution

Table 2 Orthogonal–based–algorithm vs. other tolerance allocation methods