International Journal of Engineering, Science and Mathematics
Vol. 7 Issue4, April 2018(special issue NCIME)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

MACHINING TOLERANCE ALLOCATION THROUGH TOLERANCE CHARTING USING A DIFFERENTIAL EVOLUTION (DE) TECHNIQUE

[1]Dr. M.Thilak, [2]V. Senthilkumar
[1]Associate Professor, [2] Assistant Professor, Department of Mechanical Engineering, TRP Engineering College, Trichy

Abstract

Tolerance plays a key role in product design and manufacturing. Tolerance allocation impact on manufacturing cost and product quality. The main purpose of tolerance allocation is to produce product components with least machining cost possible, while meeting all functional requirements of the product. In this work, tolerance charting is used to determine the working dimension and tolerances at the lowest cost without violating the blueprint dimensions. A simple dimensional chains identification approach is also proposed. A non-linear optimization model is formulated with the manufacturing cost as the objective function and blue print tolerances, blue print dimensions, stock removal and process capability as constraints. The nonlinear, constraint tolerance allocation problem has been solved using Differential Evolution (DE)

Keywords: Tolerance Allocation, Tolerance charting, Differential Evolution.

1. Introduction:

Tolerances are fundamental to both design and manufacturing. Perfect parts are difficult to manufacture in large quantifies. Tolerances are the allowable variation specified on the size, form and geometry of parts. Tolerance design is an experience-based discipline involving precision and cost considerations. A loose (large) tolerance invariably leads to a loss in the functionality while a tight tolerance (small zone) becomes very expensive. Optimal tolerance allocation involves the minimization of the manufacturing cost subject to constraints on the assembly. Dealing with tolerance is inevitable at all stages of product development from product specifications to the fabrication of the product. In the product design stage, tolerances have to be allocated to the product. The maximum amount of tolerances that can be allocated is restricted by the specifications and functionality of the product. The tolerances allocated to the product have a direct impact on functionality and cost of that product. The continuous effort to minimize manufacturing cost and maximize functionality has promoted much research to be carried out in the area of tolerance allocation. In process planning of machined parts, a tolerance chart is used to determine the optimal allocation of working dimensions and working tolerances such that the designed blue print dimensions and tolerances can be obeyed to achieve certain objectives. Typically, in a tolerance charting method, a predefined machining sequence is used for allocating the working dimensions and tolerances. The manual methods for tolerance charting was proposed by Eary and Johnson [1] and Wade [2]. The blue print dimensions and tolerances can be expressed in terms of manufacturing and setting dimensions [3], which involves allocation of tolerances and redistribution of residual tolerances in each blueprint. Gibson [4] et., al presented a model capable of determining an optimum set dimensions and tolerances at minimum manufacturing cost, in which they minimized the cost of scrap. The resultant dimensions and tolerances during tolerance charting can be derived using a matrix approach [5]. Ngoi and Fang [6] presented a simplified approach for deriving the working dimensions between surfaces of work piece during tolerance charting. Ngoi and M.S.M.Seng[7] presented a simple dimensional chains identification method. Speckhart[8],Spotty[9]and Sutherland[10] introduced cost tolerance model which is an approach to tolerance optimization problem. Based upon the general characteristic of a production cost tolerance data curve, they proposed the exponential model, reciprocal model and reciprocal power model. The tolerance optimization method has been improved since then by many researchers. Michal and Siddall [11] solved optimal design problem with both design parameters and their tolerances and introduced the power and exponential hybrid model.

In this paper, tolerance charting is used to determine the working dimensions and tolerances at the lower cost without violating the blueprint dimensions. A simple method for deriving the working dimensions and tolerances during tolerance charting is also proposed. A non- linear optimization model is formulated with the manufacturing cost as the objective function and blue print tolerances, blue print dimensions, stock removal and process capability as constraints. The non-linear, constraint tolerance allocation problem has been solved using Differential Evolution (DE)

International Journal of Engineering, Science and Mathematics
Vol. 7 Issue4, April 2018(special issue NCIME)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

2. Methodology

The work piece shown in Figure1 is used as an example to illustrate the proposed method. The analysis focused only on the linear dimensions, by omitting all diametrical dimensions. To facilitate the identification of surfaces, each surface is labeled and all cuts are represented with working dimensions D1 to D9 and working tolerances T1 to T9.

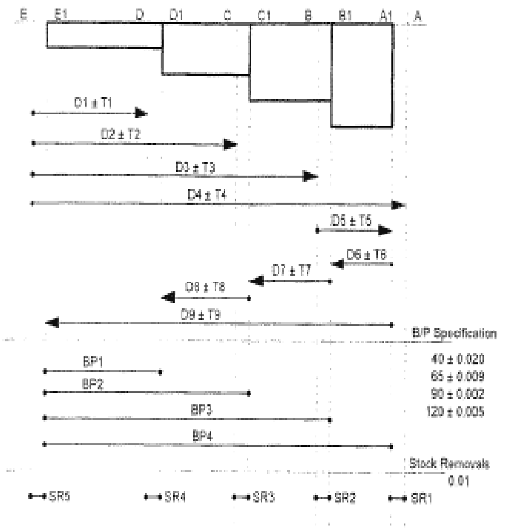

Figure1.The work piece

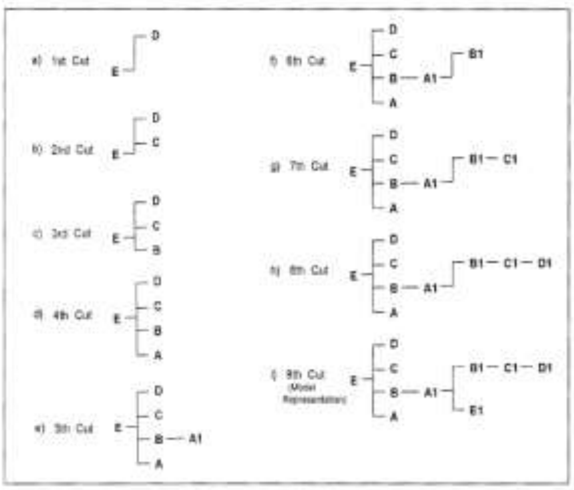

Figure 2. Model Construction

2.1 Dimensional Chain Identification:

The dimensional chain can be identified by constructing model for the working dimensions. The model construction for the given work piece as shown in Figure 1 can be represented with a very simple model. It begins with the first cut and proceeds chronologically to the last cut. Figure 2 illustrates the sequential construction of this model. For BP1, which begins with E1 and ends at D1, it will have to bypass EIA1, A1B1, B1C1 and C1D1. Since E1A1 is in the same direction as the above convention, a positive sign is associated with it. On the other hand, sinceA1B1, B1C1 and C1D1 run in the opposite direction to the above convention, negative signs are assigned to their links. Hence, BP1 = + E1A1 - AIB1 - B1C1 –C1D1.However, when equating the above to tolerance, negative signs are inappropriate as the main concern is on the tolerance aggregate sum between any two surfaces.

International Journal of Engineering, Science and Mathematics
Vol. 7 Issue4, April 2018(special issue NCIME)
ISSN: 2320-0294 Impact Factor: 6.765
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

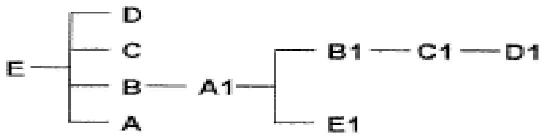

Figure3. The model for determining dimensional chains

The associated working dimensions/tolerances between any two surfaces can be determined by the following dimensional chain.

$$E \rightarrow E1 \rightarrow D \rightarrow D1 \rightarrow C \rightarrow C1 \rightarrow B \rightarrow B1 \rightarrow A1 \rightarrow A$$

For instance, when determining the dimensional chains for blueprint 1 (BP2), which extend from surface E1 to D1, one will have to begin its trace from E1 to C1 and not C1 to El. Having established this, one can begin its trace with reference to the model in Fig 3.

3. Problem Statement:

The problem considered is a stepped bar as shown in figure1. It can be machined by the single machining process namely turning process. The blueprint dimensions along with tolerances given that BP1= 40 ± 0.020, BP2= 65 ± 0.009, BP3= 90 ± 0.002, BP4= 120 ± 0.005. The blue print dimensions given here are only linear dimensions, not the diametrical requirements.

3.1 Objective function:

For working tolerances

$$\text{Min Cost} = \sum_{i=1}^{n} (A * \exp^{B*T_i} + C/ T_i^{D} + E/ T_i + F)$$

where A ,B,C,D,E and F are the coefficients of the cost-tolerance relationship

T_i is the working tolerance for the i^{th} operation and

n is the number of operations

For working dimensions:

$$\text{Max} = \sum_{i=1}^{n} D_i$$

where D_i is the working dimension for the i^{th} operation and

n is the number of operations

3.2 Constraints:

The constraints can be derived from the tolerance charting

(a) Equating working dimensions to the blue print dimensions:

BP1: E1A1- A1B1-B1C1-C1D1 = 40
D9-D6-D7-D8 =40

BP2: E1A1- A1B1-B1C1 =65
D9-D6-D7=65

BP3: E1A1- A1B1 =90
D9-D6 =90

BP4: E1A1 =120
D9 =120

(b) Equating working dimensions to the blue print tolerances:

BP1: E1A1+ A1B1+B1C1+C1D1 $\leq$ 0.020
T9+T6+T7+T8 $\leq$ 0.020

BP2: E1A1+A1B1+B1C1 $\leq$0.009
T9+T6+T7$\leq$0.009

BP3: E1A1- A1B1 $\leq$0.002
T9+T6 $\leq$ 0.002

BP4: E1A1 $\leq$0.005
T9 $\leq$ 0.005

(c) Equating working dimensions to the stock removal requirements:

SR1: -A1B1-BE+EA = 0.01
-D5-D3+D4 = 0.01

SR2: BA1-A1B1 = 0.01
D5-D6= 0.01

SR3: -CE+EB+BA1-A1B1-B1C1 = 0.01
D2+D3+D5-D6-D7 = 0.01

SR4: -DE+EB+BA1-A1B1-B1C1-C1D1 = 0.01
-D1+D3+D5-D6-D7-D8 = 0.01

SR5: EB+BA1-A1E1 = 0.01
D3+D5-D9=0.01

(d) Equating working dimensions to the process capabilities constraints:

Lower tolerance limit of operation i $\leq T_i \leq$ Upper tolerance limit of operation i

$$0.0005 \leq T_i \leq 0.023$$

where T_i is the working tolerance for the i^{th} operation.

The above nonlinear, constraint tolerance allocation problem has been solved using simple Differential Evolution (DE).

4. DIFFERENTIAL EVOLUTION (DE):

Price and Storn [12]) is an improved version of a genetic algorithm for faster optimization. Unlike simple GAs that use binary coding for representing problem parameters, DE uses the real coding of floating point numbers. Among the DE's advantages are its simple structure, ease of use, speed, and robustness. The simple adaptive scheme used by DE ensures that the mutation increments are automatically scaled to the correct magnitude. Similarly, DE uses a non-uniform crossover in that the parameter values of the child vector are inherited in unequal proportions from the parent vectors. For reproduction, DE uses a tournament selection, where the child vector competes against one of its parents. The overall structure of the DE algorithm resembles that of most other population based searches. The parallel version of DE maintains two arrays, each of which holds a population of NP, D-dimensional, real-valued vectors. The primary array holds the current vector population, while the secondary array accumulates vectors that are selected for the next generation. In each generation, NP competitions are held to determine the composition of the next generation. Every pair of vectors (X_a, X_b) defines a vector differential, $X_a - X_b$. When X_a and X_b are chosen randomly, their weighted differential is used to perturb another randomly chosen vector Xc. This process can be mathematically written as $X^1= X_c +F (X_a - X_b)$: The scaling factor F is a user-supplied constant in the range (0< F<1.2). The optimal value of F for most of the functions lies in the range from 0.4 to 1.0 (Price and Storn [12]). Then, in every generation, each primary array vector, Xi is targeted for crossover with a vector like $X^1= X_c +F (X_a - X_b)$ to produce a trial vector X_t. Thus, the trial vector is the child of two parents, a noisy random vector, and the target vector against which it must compete.

The non-uniform crossover is used with a crossover constant CR, which is in the range 0<CR<1. The CR actually represents the probability that the child vector inherits the parameter values from the noisy random vector. When CR=1, for example, every trial vector parameter is certain to come from $X^1= X_c +F (X_a - X_b)$. If, on the other hand, CR=0, all but one trial vector parameter comes from the target vector. To ensure that X_t differs from X_i by at least one parameter, the final trial vector parameter always comes from the noisy random vector, even when CR=0. Then, the cost of the trial vector is compared with that of the target vector, and the vector that has the lowest cost of the two would survive for the next generation. In all, just three factors control evolution under DE: the population size, NP; the weight applied to the random differential, F; and the crossover constant, CR. The general convention used is DE/x/y/z. DE stands for differential evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for the perturbation of x, and z stands for the type of crossover being used (exp: exponential; bin: binomial). Thus, the working algorithm outlined above is the strategy of DE, i.e., DE/rand/1/bin. Hence, the perturbation can be either in the best vector of the previous generation or in any randomly chosen vector. Similarly for perturbation, either a single or two vector differences can be used. For perturbation with a single vector difference, out of the three distinct randomly chosen vectors, the weighted vector differential of any two vectors is added to the third one. Similarly for perturbation with two vector differences, five distinct vectors, other than the target vector, are chosen randomly from the current population. Out of these, the weighted vector difference of each pair of any four vectors is added to the fifth one for perturbation. In binomial crossover, the crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the CR value.

4.1 Pseudo code for DE

The pseudo code for DE used in the present study is given below:

– Choose a seed for the random number generator.

– Initialize the values of D, NP, CR, F, and MAXGEN (maximum generation).

– Initialize all of the vectors of the population randomly. The variables are normalized within the bounds. Hence, generate a random number between 0 and 1 for all of the design variables for initialization:

for i=1 to NP

{for j=1 to D

$X_{i,\,j}$=lower bound+random number*(upper bound - lower bound)}

– All of the vectors generated should satisfy the constraints. A penalty function approach, i.e., penalizing the vector by giving it a large value, is followed only for those vectors which do not satisfy the constraints.

– Evaluate the objective function of each vector. Here,the value of the objective function to be minimized is calculated by a separate function defunct. objective():

for i=1 to NP

C_i=defunct .objective ()

– Find out the vector with the minimum objective value,i.e., the best vector so far:

C_{min}=C1 and best=1

for i=2 to NP

{ if (C_i<C_{min})

then C_{min}=C_i and best=i}

– Perform mutation, crossover, selection, and evaluation of the objective function for a specified number of generations:

while (gen<MAXGEN)

{ for i=1 to NP

{

– For each vector Xi (target vector), select three distinct vectors Xa, Xb, and Xc (select five if two vector differences are to be used) randomly from the current population (primary array) other than the vector Xi:

do

{ r1=random number*

r2=random number*NP

r3=random number*NP

} while

(r1=i) OR (r2=i) OR (r3=i) OR (r1=r2) OR (r2=r3) OR

(r1=r3)

– Perform crossover for each target vector X_i with its noisy vector $X_{n,i}$ and create a trial vector, $X_{t,i}$ Performing mutation creates the noisy vector.

– If CR=0, inherit all of the parameters from the target vector X_i, except one which should be from $X_{n,i}$

– For binomial crossover:

{p=random number

for n=1 to D

{if (p<CR)

$X_{n,i}=X_{a,i}+F(X_{b,i}-X_{c,i})$

$X_{t,i}=X_{n,i}$

} else $X_{t,i}=X_{i,i}$

}

– Again, the NP noisy random vectors that are generated should satisfy the constraint and the penalty function approach is followed as mentioned above.

– Perform selection for each target vector, Xi by comparing its objective value with that of the trial

vector, $X_{t,i}$, whichever has the minimum objective will survive for the next generation:

$C_{t,i}$=defunct.objective()

if ($C_{t,i}$<C_i)

new X_i=$X_{t,i}$

else new X_i=X_i

}

/* for i=1 to NP */

}

– Print the results (after the stopping criteria is met).

The stopping criterion is the maximum number of generations.

5. Results and Discussion:

Nowadays, the most efficient and the most sophisticated technique developed by industry to control tolerance buildups is represented by tolerance chart. Figure 2&3 shows the methods to arrive the suitable tolerance charting sequence in order to arrive the optimal tolerance allocation to the each machining operation. The proposed mixed exponential reciprocal cost tolerance model is arrived through the proposed tolerance charting technique. A DE based optimization procedure was developed to arrive the optimal machining tolerance of each operation in the given machine component. Table 1 shows the optimum working dimensions and the working tolerances with minimum machining cost using DE.

Table1. Optimum Working dimensions and Working tolerances

Minimum machining cost obtained = 25.14251066 $

Working dimensions	Working tolerances
D1=40.00	T1=0.02103798336
D2=65.00	T2=0.02252423043
D3=90.00	T3=0.02176450878

D4=120.02	T4=0.01536076968
D5=30.01	T5=0.01620928354
D6=30.00	T6=0.01786605805
D7=25.00	T7=0.01804482934
D8=25.00	T8=0.0226599748
D9=120.00	T9=0.01558089057

6. Conclusion:

This paper discusses the DE based optimal machining tolerances of machine elements by selecting the appropriate tolerance charting model. Tolerance charting was performed to determine the working dimension and machining tolerances with the blue print dimensions. A mixed exponential reciprocal nonlinear optimization model was developed with the machining cost as the objective function and the blueprint tolerances, process capabilities as the constraints. This proposed optimization procedure using DE achieve minimum machining cost and optimal tolerances. A VC++ program has been developed to solve the proposed optimization problem using DE.

REFERENCES

1. D.F Eary and G.E Johnson, "Process Engineering for manufacturing", prentice hall,pp 98-119, 1962.
2. O.R.Wade, "Tolerance control", Tool and Manufacturing Engineering, Handbook, Vol.I.pp2/60,1982.
3. D.Faingvelernt. D.wall and P.Bourdeb, "Computer Aided Tolerancing and dimensioning in process planning",Annals CIRP, 33(1) pp381-386, 1986.
4. J.R.He and P.R.Gibson, "Computer Aided Geometric dimensioning and tolerancing for process planning and quality control", International journal of advanced manufactureing technology, 7(1), pp 11-20, 1992.
5. B.K.A.Ngoi and C.K.Chua "A matrix approach to tolerance charting" International Journal of Advanced Manufacturing Technology 8(3), pp, 173-191, 1992.

6. B.K.A.Ngoi and S.Lfang,"Computer Aided Tolerance Charting" International Journal of predirection research, 32(8), pp, 1939-1957, 1994.
7. B.K.Angoi and Micael S,M.song, "Tolerance Synthesis adopting a nonlinear programming approach, " International Journal of Advanced Manufacturing Technology, 11: pp 387-393,1996.
8. Spectiharb, F.H., 1972, "Calculation of tolerance Based on a minimum cost approach," ASME Journal of Engineering for Industry, Vol.94, pp-997-
9. Spotts.M.F., "Allocation of Tolerances to minimize cost of Assembly", ASME Journal of Engineering for Industry, August pp.762-764.
10. Suferland.G.H andRoth.B. Mechanism design: Accounting for Manufacture Tolerances and cost on Function converting problems", ASME Journal of Engineering for Industry, Vol-98, pp-280-286, 1975.
11. Michael.W and Siddall.J.N., Optimization problems with Tolerance Assignment and Full Acceptance", ASME Journal of Mechanical Design, Vol.103, pp-842-848,1982.
12. Price K, Storn R (1997) Differential evolution—a simple evolution strategy for fast optimization. Dr Dobb's J 22(4):18–24, 78