

# OPTIMAL TOLERANCE ALLOCATION IN A COMPLEX ASSEMBLY USING EVOLUTIONARY ALGORITHMS

Ramesh Kumar, L.\*; Padmanaban, K. P.\*\* & Balamurugan, C.\*\*\*

\* Department of Mechanical Engineering, Christian College of Engineering and Technology, Tamil Nadu, India

\*\* Department of Mechanical Engineering, SBM College of Engineering and Technology, Dinidigul, Tamil Nadu, India

\*\*\* Department of Mechanical Engineering, College of Engineering, Guindy, Anna University, Chennai, Tamil Nadu, India

E-Mail: plrameshkumar80@gmail.com, padmarubhan@yahoo.co.in, balamurugan.c@annauniv.edu

## Abstract

Tolerance design is a vital factor which influences product and process development. Further, it determines the manufacturing cost, the functionality and quality of a product. It is evident that optimal tolerance normally leads to produce ample parts, better operation of mechanical systems and excellent assembling. In contrast, tight tolerance leads to increase in manufacturing cost for an assembly. An ideal relationship exists among production cost and operation, while determining the optimum tolerance. Based on this relation a new approach by implementing the Non-traditional techniques: Genetic Algorithm (GA), Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) and Differential Evolution (DE) for determining the optimum tolerance, zero percentage rejection and manufacturing cost considering the varying quality loss constants for an assembly namely overrunning clutch assembly, is discussed in this paper. From the result obtained, it is evident that, the proposed approach is best suitable for solving problems involving complex assemblies.

(Received in May 2015, accepted in September 2015. This paper was with the authors 1 month for 1 revision.)

**Key Words:** Tolerance Allocation, Manufacturing Cost, Quality Loss, Evolutionary Algorithms

## 1. INTRODUCTION

In an industry, designing of parts is a very crucial issue. It is more laborious to ascertain the required tolerance that is essential for the given dimension. Dissimilarity in component dimension and its aspects is sure to happen. However, the estimation of the required variation in assembly parameter is the goal of tolerance optimization. Nevertheless, depending upon either on the assembly or any of its part, the cost of the product is immediately influenced by tolerance and manufacturability. Stiff tolerance leads to excessive cost, the loose tolerance results in the manufacturing concern or rejection of the part. Therefore, the traditional designers assign tolerance depending on their knowledge and according to what the handbooks prescribe or depending upon the prevalent standard [1]. Earlier approaches gives neither guarantee its function nor minimize the cost. Only when the cost tolerance relations are well established, mathematical prototypes can be designed to obtain proper tolerance [2]. Consequently, in term of an amalgamation of data of the product design, the numerical model becomes the foundation for an optimization approach. Hence, the development of tolerance design strategies becomes complex, yet important. Consequently, the learning of mathematical optimization approaches for tolerance approach becomes an acute research area owing to hypnotically and concrete reasons. For the past umpteen years various tolerance techniques have evolved and have led to the production of extraordinary quality goods. Thus, in the process of tolerance development, traditional tolerance methods play a vital role. Such approaches and processes help in the formation and assessment of proficiency about product and design process and cost evaluation.

## 1.1 Literature review

Numerous scholars were keen on improving different kinds of tolerance allocation approaches for complex assemblies. Fathi [3] resolved a designing of tolerance by incorporating the quality loss. In this approach, the optimization approach is implemented to obtain the part tolerance and also to minimize the cost of production. Taguchi [4] to tolerance allocation has been extensively conferred in other works (Askin and Goldberg [5]; Zhang and Huq [6], Kapur [7]). Singh et al. [8] search extensive on Lagrange multiplier to find the optimal tolerance using the Genetic Algorithm. Sivakumar et al. [9] and Noorul Haq et al. [10, 11] introduced evolutionary algorithms to get the optimum tolerance for the mechanical assemblies. Prabhakaran et al.[12] introduced the non-traditional approach to find the optimum tolerance and to overcome the traditional tolerance approach. Prabhakaran et al. [13] imported a metaheuristic approach as an ant colony algorithm, to simultaneously allocate the tolerance and manufacturing cost. Krishna and Rao [14] incorporated an approach to reduce the cost of manufacturing. Huang and Shiau [15] evolved the tolerance synthesis of a sliding vane rotary compressor's parts to improve quality and reduce manufacturing cost. Huang and Zhong [16] used the optimization method to reduce the manufacturing cost, tolerance and also to advance the rate of acceptance of the parts. Singh et al. [17] suggested a methodology by allowing the continuous cost function to get the optimal solution using the GA. Siva Kumar et al. [18] introduced a hybrid methodology for optimal solution synthesis for models involving different manufacturing process. Muthu et al. [19] suggested non-traditional method to minimize the cost of manufacturing and improve quality and to obtain optimal solution by considering both least manufacturing cost and quality loss over the lifecycle of the invented assembly. Lu et al. [20] suggest an approach to attain least manufacturing cost and proper assimilability product to obtain an optimal solution. Cheng and Maghsoodloo [21], Wu et al. [22] and Huang et al. [23] used nonlinear programming method to design the tolerance. Cheng and Tsai [24, 25] solved allocation of statistical tolerances with exponential cost function and obtained closed-form solution of optimal tolerances. Gill [26] considered a manufacturing cost as important for the manufacturing processes using activity based cost model. Lestan et al. [27] have proposed the evolutionary optimization techniques, considered to be a Genetic Algorithm (GA) to measure the scheduling efficiency. Sasiadek [28] suggested theoretical approach for an assembly involving elements design and complex machine assembly. Govindarajalu et al. [29] suggested a methodology for a piston cylinder assembly to reduce the manufacturing cost. Cheng et al. [30] used a tolerance design method for hydrostatic rotary table involving a different design methodology by implementing PSO. Sivakumar et al. [31] presented the tolerance design procedure for complex assemblies implementing DE and NSGA-II. Ragu [32] suggested a methodology to reduce the uneven progression of die clearance involved in sheet metal products.

However, Coelho [33] was deemed to: (1) optimal tolerance, cost related to loss in quality and manufacturing cost as the objectives; (2) use of self-organizing migrating algorithm (SOMA) and SOMA using Gaussian operator (GSOMA); (3) SOMA and GSOMA take more time to get the solution and they will not give the optimum solution globally for the different values of quality loss coefficient  $A$ ; and (4) in such a process, the computational efforts are at a high rate and other intelligent optimization techniques have their own limitations.

This paper takes significant steps to overcome the limitations of earlier techniques. Therefore this paper considers simultaneously, the optimal tolerance, quality loss and manufacturing cost as its sole objective. Briefly speaking, the followings are the benefits of the non-traditional method: 1) Search – method based on population. Therefore, world-wide optimum solution is viable. 2) It is simpler than the methods that are stated in the review. 3) It is easy to solve the problems to get global optimum solution for complex assemblies. 4) It is appropriate for resolving all kinds of complex assemblies. And (5) it is computationally

greater and quicker than the techniques stated. Thus, the solution of the tolerance design becomes easier and simpler to obtain by implementing this technique.

The major contributions of the tolerance region are cited below: 1. This paper suggests a common method to tolerance allocation. 2. The objectives that are cited in this paper, as given by Coelho [33], determine the optimal tolerance, reduce the total cost, zero percentage rejection and quality loss. 3. This paper uses the GA, NSGA-II and DE, which overcomes all the limitations found in the review.

The remaining portion of this paper is planned as follows: Section 2 deals with the problem description and optimization model. Section 3 briefly explains the optimization methods. Section 4 gives the simulation results and analysis. Conclusion and future work are given in Section 5.

## 2. PROBLEM DESCRIPTION AND OPTIMIZATION MODEL

Coelho [33] presented a one-way clutch assembly. This assembly clearly demonstrated the tolerance based on GA, NSGA-II and DE approaches. The three components overrunning clutch are the hub, the roller and the cage. Fig. 1 shows the overrunning clutch accumulated by establishing a cage within the hub and four rollers.

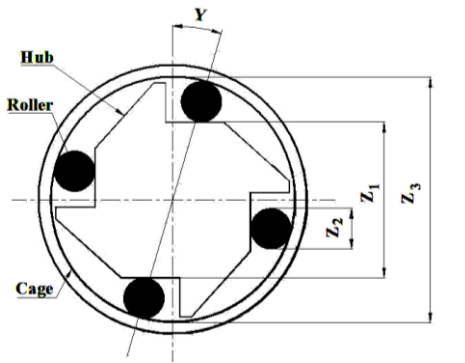


Figure 1: Overrunning clutch.

Table I gives the cost data for the clutch tolerance (tolerances in  $10^{-4}$  inches, cost in \$). The assembly response function is the contact angle  $Y$  between the roller and centre of the hub. Its value should be within the tolerance bound. It is shown as below.

$$Y = f(Z_1, Z_2, Z_3) = a \cos \left( \frac{Z_1 - Z_2}{Z_3 - Z_2} \right) \quad (1)$$

where,  $a$  is constant.

The values of  $Z_i$  ( $i = 1, 2, 3$ ) are 2.17706, 0.9, 4 inch, the nominal value and  $Y$  are  $0.122 \pm 0.035$  rad.

Table I: Cost data for the clutch tolerance (tolerances in  $10^{-4}$  inches, cost in \$).

Hub tolerance	Cost	Roll tolerance	Cost	Cage tolerance	Cost
2	19.380	1	3.513	1	18.637
4	13.220	2	2.480	2	12.025
8	5.990	4	1.240	4	5.732
16	4.505	8	1.240	8	2.686
30	2.065	16	1.200	16	1.980
60	1.240	30	0.413	30	1.447
120	0.825	60	0.413	60	1.200
-	-	120	0.372	120	1.033

Tolerance bounds are shown below:

$$0.0001 \leq t_1, t_3 \leq 0.0120 \text{ in}$$

$$0.0001 \leq t_2 \leq 0.0005 \text{ in}$$

The main aim of the proposed methodology is to minimize the manufacturing cost and quality loss.

The literature [33] explains the functionality of the manufacturing cost as given below.

$$C(t_1) = -0.731 + \frac{0.058}{t_1} \quad (2)$$

$$C(t_2) = -8.3884 + \frac{5.7807}{t_2} \quad (3)$$

$$C(t_3) = 0.978 + \frac{0.0018}{t_3} \quad (4)$$

$$\text{Manu. Cost } C(t_i) = C(t_1) + C(t_2) + C(t_3) \quad (5)$$

Loss in quality function is expressed as shown below:

$$\text{Quality loss : } Q(t_i) = \sum_{k=1}^k (A/t_k^2)(\sigma_k^2) \quad (6)$$

The total cost equation is expressed as:

$$\begin{aligned} \text{Minimize: Total cost } Z(t) &= \sum_{i=1}^3 [C(t_i) + Q(t_i)] \quad (7) \\ &= -33.3066 + \frac{0.058}{t_1} + \frac{23.1228}{t_2} + \frac{0.0018}{t_3} \\ &\quad + A(90.7029t_1^2 + 362.8110t_2^2 + 90.7029t_3^2) \end{aligned}$$

based on constraints shown below:

$$3.7499t_1 + 27.472t_2 + 3.722t_3 \geq 0.035 \quad (8)$$

where:

$t_1$  – tolerance of the hub,  $t_2$  – tolerance of the hub,  $t_3$  – tolerance of the roller,  
 $A$  – quality loss coefficient,  $t_k$  – tolerance stack up limit for  $k$ ,  $K$  – no. of dimensional chain,  
 $\sigma_k$  – standard deviation of  $k$ ,  $k$  – dimensional chain index.

### **3. OPTIMIZATION METHODS**

The proposed evolutionary techniques (GA, NSGA-II and DE) are stated as follows.

#### **3.1 Genetic algorithm**

Goldberg's GA [34] is a stochastic universal search. It is an optimization method that imitates the natural procedure in a biological origin. Darwin's evolutionary theory inspired the invention of GA. A GA activates the biological evolutionary method. First of all, the initial individual population is taken. And then, the mutation in every reproduction, the crossover and other genetic operators are applied. By doing such optimization, encoding of the population of each person is done on to a chromosome or a string. This denotes feasible solution to a concrete problem. About a known objective function, the aptness of an individual is estimated. By substituting the portions of their generated data, the extremely fit those are chances to duplicate. The result of such products is called new offspring, possessing

the best qualities of the parents. Then, later crossover by changing some generations in the strings, mutation is applied. As a result, among the entire population the new offspring can replace any unfit individual. Until the satisfactory solution is formed, this reproduction cycle or the evolution selection is repeated.

The basic procedure for GA can be explained as follows:

Step 1. Variable coding by proper representation.

Step 2. Random generation of chromosomes of the first population.

Step 3. Fitness task of every person in the population is evaluated.

Step 4. Halt when stopping situation arrives; otherwise, skip to 5<sup>th</sup> step.

Step 5. Mutation with crossover along with reproduction is implemented.

Step 6. A new creation form of the separate individuals is created to attain form step 5.

Step 7. Go to step 3.

### GA operators:

The parameter values of GA are: population size = 100, mutation probability = 0.15, total number of generations = 100, crossover probability = 0.8.

### 3.2 NSGA-II

Deb et al. [35] offered the NSGA-II. Basically, NSGA-II varies from non-dominated sorting genetic algorithm (NSGA) execution in several forms. Initially obtained optimal solutions are secured by NSGA-II as it is a robust securing system. Then, NSGA-II uses a faster non-dominated sorting technique. Since NSGA-II is self-governing it does not require a measurable limit. Fig.2 shows an iteration of NSGA-II, where,  $P_t$  – Parent population,  $Q_t$  – Offspring population,  $F_1, F_2, F_3$  – Non-dominated fronts,  $P_{t+1}$  – New parent population,  $R_t$  – Fast non-dominated fronts and  $N$  – Size of population.

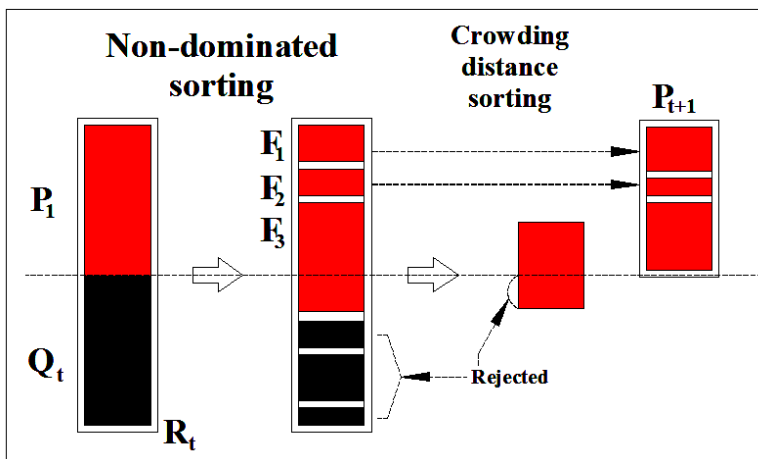


Figure 2: An iteration procedure of the NSGA-II algorithm.

### NSGA-II operators:

Parameters of NSGA-II technique are: variable type = real variable, size of population = 100, the probability of crossover = 0.7, real-parameter mutation probability = 0.25, real-parameter SBX parameter = 10, real-parameter mutation parameter = 100, total no. of generations = 100.

### 3.3 Differential evolution

Differential evolution (DE), suggested by Storn and Price [36] is a robust algorithm for complex assemblies, and utilizes the vector differences of entities for agitating the population segment. At the start, DE consist of a population of  $N$  and each entity is an  $n$ -dimensions

vector  $Y_i = \{Y_1, Y_2, \dots, Y_n\}$ . In the search space selected, the vectors are generated at random and during the process of evolution, entities will be handled by the process of crossover, mutation, and selection.

Mutation operation: in this process, with the distinct mutant methods, the developed path of a mutant vector  $\delta_i$  is different and there is a variation among the weighted vector difference. The approved mutant procedures are listed by Das and Suganthan [37] as shown below.

$$\begin{aligned}
- \text{DE/rad/1} : \delta_i &= Y_{R_1} + A(Y_{R_2} - Y_{R_3}) \\
- \text{DE/rad/2} : \delta_i &= Y_{R_1} + A(Y_{R_2} - Y_{R_3}) + A(Y_{R_4} - Y_{R_5}) \\
- \text{DE/best/1} : \delta_i &= Y_{best} + A(Y_{R_1} - Y_{R_2}) \\
- \text{DE/best/2} : \delta_i &= Y_{best} + A(Y_{R_1} - Y_{R_2}) + A(Y_{R_3} - Y_{R_4}) \\
- \text{DE/current to best/1} : \delta_i &= Y_i + A(Y_{best} - Y_i) + A(Y_{R_1} - Y_{R_2})
\end{aligned} \tag{9}$$

where the  $R_1, R_2, R_3, R_4, R_5$ , which are all different from the index  $i$ , are different among themselves individually and chosen at random within the range  $[1, N]$ .  $Y_{best}$  is the best one of the existent population. The factor that controls the difference of vectors is termed as scaling factor, denoted by  $A$ .

Crossover operation: From mutant vector  $\delta_i$  and the target vector  $Y_i$ ; the trial vector  $U_i$  is developed by binomial crossover as shown below.

$$U_{i,j} = \begin{cases} \delta_{i,j}, & \text{if } \text{ran}_j \leq CP \text{ or } j = j_{ran} \\ \delta_{i,j}, & \text{otherwise} \end{cases} \tag{10}$$

where  $i = 1, 2, \dots, N$ ;  $j = 1, 2, \dots, n$ ,  $j_{ran}$  is a chosen integer at random from  $[1, n]$  which secures  $U_i$ , acquired at the minimum of single individual from the mutant vector  $\delta_i$ ,  $\text{ran}_j$  is a uniform random number within 0 and 1.  $CP$  is the crossover probability and its value is between  $[0, 1]$ .

Selection operation: to preserve final ones for the next iteration, the developed trial vector  $U_i$  is correlated with the target vector  $Y_i$ . The procedure is defined as shown below.

$$Y_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } f(U_{i,g}) \leq f(Y_{i,g}) \\ Y_{i,g}, & \text{otherwise} \end{cases} \tag{11}$$

### DE implementation:

The basic procedure of DE is summarized as follows.

Step 1. Randomly initialize the population of entity for DE.

Step 2. The aim values of each individual are checked, and the optimum individual with the optimized objective value is selected.

Step 3. Based on Eq. (9) in order to attain all entities respective mutant vector, mutation operation is executed individually.

Step 4. Evaluate crossover operation among every individual and the related mutant vector based on Eq. (10) so as to attain individual's trial vector.

Step 5. The aim values of the trial vectors are checked.

Step 6. Perform selection operation among all entities and its related trial vector based on Eq. (11) in order to form new entities for the next generation.

Step 7. Optimized value among the best entity of the local new population is selected. The newly obtained entity is compared with the existing one and among them the optimal fit entity is selected and updated.

Step 8. From the step 3 entire procedure is repeated, if the criteria is not satisfied or else the output obtained is the optimal solution.

**DE operators:**

The parameters of DE technique are: strategy = DE/rand/1/bin, crossover constant  $CR = 0.9$ , number of population  $NP = 100$ ,  $F = 0.5$  and total number of generations = 100.

**4. SIMULATION RESULTS AND ANALYSIS**

This problem has been solved on a Pentium IV computer using VC++. The simulation results of GA, NSGA-II and DE techniques are listed in Table II, III and IV. The results obtained from Coelho [33] are listed in Table V and compared with GA, NSGA-II and DE. Better results are obtained with GA, NSGA-II and DE techniques compared to SOMA and GSOMA when the quality loss coefficient  $A = 0, 1, 52, 100, 300$  and  $520$  is considered. Convergent graph of GA, NSGA-II and DE for various values of  $A$  are exposed in Figs. 3, 4 and 5, respectively. From the Fig. 6, it is evident that when the value of  $A$  increases, the total cost also increases. From the Figs. 3, 4 and 5, it can be inferred that the closeness of results for the overrunning clutch assembly, indicates the consistency of performance of GA, NSGA-II and DE algorithm. GA converges only after 20 iterations for values of  $A = 0, 1, 52, 300$  and  $520$  whereas NSGA-II and DE algorithm converges within 20 iterations much earlier than SOMA and GSOMA. This shows that the NSGA-II and DE algorithms perform better than SOMA and GSOMA in terms of convergence, optimal results and reduced computational time.

Table II: Optimal tolerances, total cost, constraints and CPU time using GA.

$A$	$t_1$ ( $10^{-4}$ inch)	$t_2$ ( $10^{-4}$ inch)	$t_3$ ( $10^{-4}$ inch)	$Z(t)$	Constraints	CPU time(s)
0	0.01191	0.00049	0.01174	10.082	0.101793	5
1	0.00118	0.00050	0.01101	10.084	0.059140	5
52	0.01036	0.00050	0.00533	10.982	0.072423	5
100	0.00941	0.00046	0.00428	11.752	0.063853	5
300	0.00533	0.00050	0.00297	12.438	0.044777	5
520	0.00461	0.00050	0.00297	13.102	0.042077	5

Table III: Optimal tolerances, total cost, constraints and CPU time using NSGA-II.

$A$	$t_1$ ( $10^{-4}$ inch)	$t_2$ ( $10^{-4}$ inch)	$t_3$ ( $10^{-4}$ inch)	$Z(t)$	Constraints	CPU time(s)
0	0.012000	0.000500	0.012000	10.0200	0.103398	3
1	0.012000	0.000500	0.012000	10.0462	0.103398	3
52	0.010019	0.000500	0.005767	10.9779	0.072745	3
100	0.007857	0.000500	0.004628	11.4335	0.06040	3
300	0.005222	0.000500	0.003207	12.4199	0.045254	3
520	0.004254	0.000500	0.002674	13.0471	0.053150	3

Table IV: Optimal tolerances, total cost, constraints and CPU time using DE.

$A$	$t_1$ ( $10^{-4}$ inch)	$t_2$ ( $10^{-4}$ inch)	$t_3$ ( $10^{-4}$ inch)	$Z(t)$	Constraints	CPU time(s)
0	0.012000	0.000499	0.012000	10.02000	0.103371	2.5
1	0.010020	0.000500	0.008361	10.04621	0.082429	2.5
52	0.010020	0.000500	0.008361	11.05389	0.082425	2.5
100	0.007856	0.000500	0.006104	11.48318	0.065914	2.5
300	0.005220	0.000500	0.003598	12.43126	0.046702	2.5
520	0.004254	0.000500	0.002674	13.04822	0.039640	2.5

Table V: Comparison of SOMA[33], GSOMA[33], GA, NSGA-II and DE.

A	SOMA [33]	GSOMA [33]	GA	NSGA-II	DE
0	11.63862	11.63786	10.082	10.0200	10.02000
1	11.64147	11.64223	10.084	10.0462	10.04621
52	11.78462	11.78578	10.982	10.9779	11.05389
100	11.91916	11.91896	11.752	11.4335	11.48318
300	12.46852	12.46874	12.438	12.4199	12.43126
520	13.04713	13.04712	13.102	13.0471	13.04822

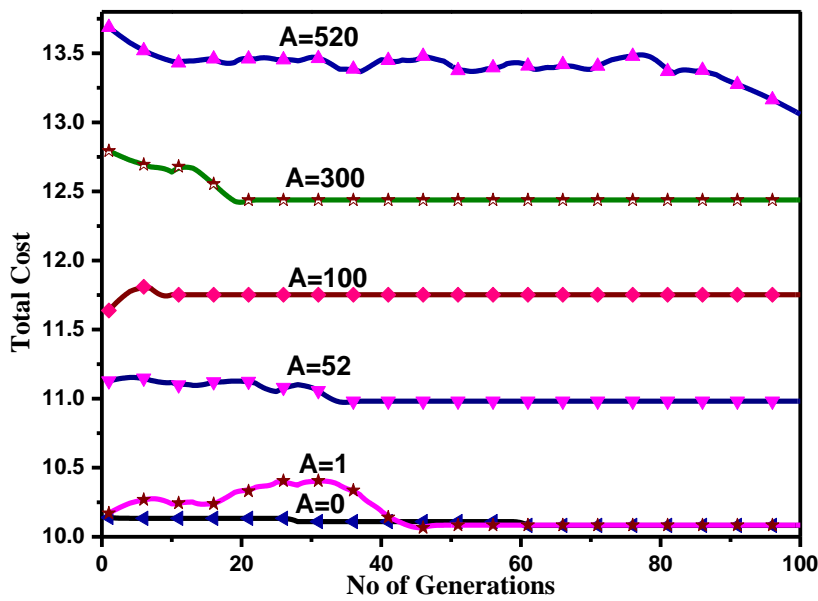


Figure 3: Convergent graph for different values of A using GA.

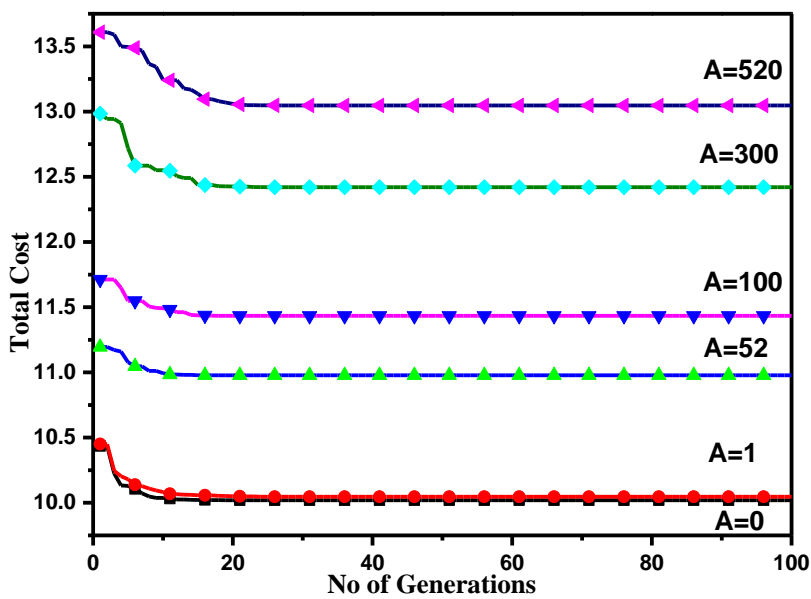


Figure 4: Convergent graph for different values of A using NSGA-II.



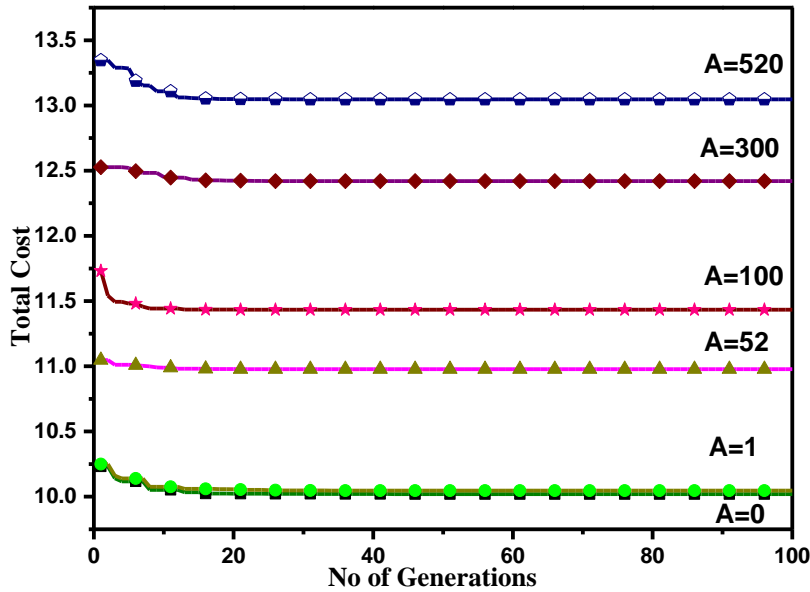


Figure 5: Convergent graph for different values of A using DE.

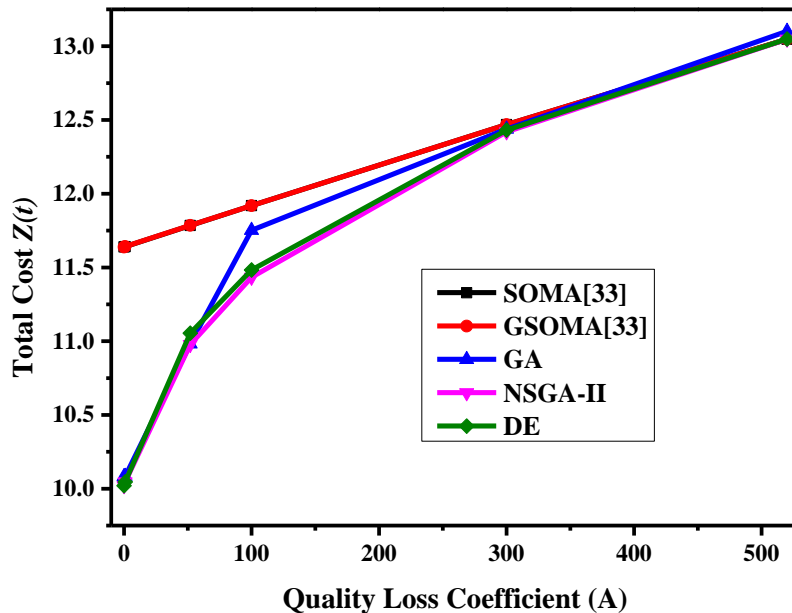


Figure 6: Comparison of total cost  $Z(t)$  Vs quality loss coefficient (A) using SOMA [33], GSOMA [33], GA, NSGA-II and DE.

## 5. CONCLUSION AND FUTURE WORK

This paper assures to achieve zero percentage rejection, quality loss and least cost by the implementation of GA, NSGA-II and DE. Moreover, there is a considerable reduction in computational effort, too. From the results it is evident that GA, NSGA-II and DE approaches are far more superior in their performance than SOMA and GSOMA when quality loss constants are considered. Thus, it is deemed that GA, NSGA-II and DE methods give better results than Cuelho [33]. As far as optimization issue is concerned, it is concluded that, GA, NSGA-II and DE are the apt solution. Obviously then, this paper keeps open the doors for more research on the mode of optimization methods to solve complex assemblies.

## Appendix A. Pseudo code for NSGA-II [35]

The pseudo code for NSGA-II algorithm is expressed as below:

1. Generate a casual parental population for size  $N$ .
2. Prioritize the population in the order of Non-domination.
3. All solutions are assigned with an equivalent fitness with the level of non-domination.
4. In order to generate a fresh  $N$ -size population offspring, the common selection for binary tournament, mutation operators and recombination are used.
5. The expanded  $2N$ -size population is formed by combining children population parental population.
6. The expanded population is classified depending on the non-domination.
7.  $N$ -size population is filed up with individuals, beginning from the best ones.
8. In order to make sure of the diversity, whether a front is able to somewhat fill the coming generation, the crowding comparison is invoked.
9. Till “stop” situation arrives, steps 2 to 8 are repeated. One “stop” is specified by a number of generations.

## REFERENCES

- [1] Zhang, C.; Wang, H.-P. (1993). The discrete tolerance optimization problem, *Manufacturing Review*, Vol. 6, No. 1, 60-71
- [2] Chen, M.-C. (2001). Tolerance synthesis by neural learning and nonlinear programming, *International Journal of Production Economics*, Vol. 70, No. 1, 55-65, doi:[10.1016/S0925-5273\(00\)00044-X](https://doi.org/10.1016/S0925-5273(00)00044-X)
- [3] Fathi, Y. (1990). Producer-consumer tolerances, *Journal of Quality Technology*, Vol. 22, No. 2, 138-145
- [4] Taguchi, G. (1986). *Introduction to Quality Engineering*, Unipublications, White Plains, New York, 21-22
- [5] Askin, R. G.; Goldberg, J. B. (1988). Economic optimization in product design, *Engineering Optimization*, Vol. 14, No. 2, 139-152, doi:[10.1080/03052158808941207](https://doi.org/10.1080/03052158808941207)
- [6] Zhang, H. C.; Huq, M. E. (1992). Tolerancing techniques: the state-of-the-art, *International Journal of Production Research*, Vol. 30, No. 9, 2111-2135, doi:[10.1080/00207549208948140](https://doi.org/10.1080/00207549208948140)
- [7] Kapur, K. C. (1993). Quality engineering and tolerance design, Kusiak, A. (Ed.). *Concurrent Engineering: Automation, Tools, and Techniques*, Chapter 11, John Wiley & Sons, New York, 287-306
- [8] Singh, P. K.; Jain, S. C.; Jain, P. K. (2004). A genetic algorithm based solution to optimum tolerance synthesis of mechanical assemblies with alternate manufacturing processes – benchmarking with the exhaustive search method using the Lagrange multiplier, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 218, No. 7, 765-778. doi:[10.1177/095440540421800709](https://doi.org/10.1177/095440540421800709)
- [9] Sivakumar, K.; Saravanan, R.; Noorul Haq, A. (2008). Cost-tolerance modelling and optimisation of machining tolerance design through intelligent techniques, *International Journal of Machining and Machinability of Materials*, Vol. 3, No. 1-2, 162-189, doi:[10.1504/IJMMM.2008.017632](https://doi.org/10.1504/IJMMM.2008.017632)
- [10] Noorul Haq, A.; Karthikeyan, K.; Sivakumar, K.; Saravanan, R. (2006). Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly, *International Journal of Advanced Manufacturing Technology*, Vol. 27, No. 9-10, 865-869, doi:[10.1007/s00170-004-2274-5](https://doi.org/10.1007/s00170-004-2274-5)
- [11] Noorul Haq, A.; Sivakumar, K.; Saravanan, R.; Muthiah, V. (2005). Tolerance design optimization of machine elements using genetic algorithm, *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 3-4, 385-391, doi:[10.1007/s00170-003-1855-z](https://doi.org/10.1007/s00170-003-1855-z)
- [12] Prabhakaran, G.; Asokan, P.; Ramesh, P.; Rajendran, S. (2004). Genetic-algorithm-based optimal tolerance allocation using a least-cost model, *International Journal of Advanced Manufacturing Technology*, Vol. 24, No. 9-10, 647-660, doi:[10.1007/s00170-003-1606-1](https://doi.org/10.1007/s00170-003-1606-1)

- [13] Prabhakaran, G.; Asokan, P.; Rajendran, S. (2005). Sensitivity-based conceptual design and tolerance allocation using the continuous ants colony algorithm (CACO), *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 5-6, 516-526, doi:[10.1007/s00170-003-1846-0](https://doi.org/10.1007/s00170-003-1846-0)
- [14] Krishna, A. G.; Rao, K. M. (2006). Simultaneous optimal selection of design and manufacturing tolerances with different stack-up conditions using scatter search, *International Journal of Advanced Manufacturing Technology*, Vol. 30, No. 3-4, 328-333, doi:[10.1007/s00170-005-0059-0](https://doi.org/10.1007/s00170-005-0059-0)
- [15] Huang, Y. M.; Shiau, C.-S. (2006). Optimal tolerance allocation for a sliding vane compressor, *Journal of Mechanical Design*, Vol. 128, No. 1, 98-107, doi:[10.1115/1.2114893](https://doi.org/10.1115/1.2114893)
- [16] Huang, M.; Zhong, Y.-R. (2007). Optimized sequential design of two-dimensional tolerances, *International Journal of Advanced Manufacturing Technology*, Vol. 33, No. 5-6, 579-593, doi:[10.1007/s00170-006-0475-9](https://doi.org/10.1007/s00170-006-0475-9)
- [17] Singh, P. K.; Jain, P. K.; Jain, S. C. (2008). Optimal tolerance design of mechanical assemblies for economical manufacturing in the presence of alternative machines – a genetic algorithm-based hybrid methodology, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 222, No. 5, 591-604, doi:[10.1243/09544054JEM967](https://doi.org/10.1243/09544054JEM967)
- [18] Siva Kumar, M.; Kannan, S. M.; Jayabalan, V. (2009). A new algorithm for optimum tolerance allocation of complex assemblies with alternative processes selection, *International Journal of Advanced Manufacturing Technology*, Vol. 40, No. 7-8, 819-836, doi:[10.1007/s00170-008-1389-5](https://doi.org/10.1007/s00170-008-1389-5)
- [19] Muthu, P.; Dhanalakshmi, V.; Sankaranarayanan, K. (2009). Optimal tolerance design of assembly for minimum quality loss and manufacturing cost using metaheuristic algorithms, *International Journal of Advanced Manufacturing Technology*, Vol. 44, No. 11-12, 1154-1164, doi:[10.1007/s00170-009-1930-1](https://doi.org/10.1007/s00170-009-1930-1)
- [20] Lu, C.; Zhao, W.-H.; Yu, S.-J. (2012). Concurrent tolerance design for manufacture and assembly with a game theoretic approach, *International Journal of Advanced Manufacturing Technology*, Vol. 62, No. 1-4, 303-316, doi:[10.1007/s00170-011-3783-7](https://doi.org/10.1007/s00170-011-3783-7)
- [21] Cheng, B.-W.; Maghsoodloo, S. (2009). Optimization of mechanical assembly tolerances by incorporating Taguchi's quality loss function, *Journal of Manufacturing Systems*, Vol. 14, No. 4, 264-276, doi:[10.1016/0278-6125\(95\)98879-B](https://doi.org/10.1016/0278-6125(95)98879-B)
- [22] Wu, C.-C.; Chen, Z.; Tang, G.-R. (1998). Component tolerance design for minimum quality loss and manufacturing cost, *Computers in Industry*, Vol. 35, No. 3, 223-232, doi:[10.1016/S0166-3615\(97\)00087-0](https://doi.org/10.1016/S0166-3615(97)00087-0)
- [23] Huang, M. F.; Zhong, Y. R.; Xu, Z. G. (2005). Concurrent process tolerance design based on minimum product manufacturing cost and quality loss, *International Journal of Advanced Manufacturing Technology*, Vol. 25, No. 7-8, 714-722, doi:[10.1007/s00170-003-1911-8](https://doi.org/10.1007/s00170-003-1911-8)
- [24] Cheng, K.-M.; Tsai, J.-C. (2013). Optimal statistical tolerance allocation for reciprocal exponential cost-tolerance function, *Proceeding of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Vol. 227, No. 5, 650-656, doi:[10.1177/0954405412473720](https://doi.org/10.1177/0954405412473720)
- [25] Cheng, K. M.; Tsai, J. C. (2011). Optimal statistical tolerance allocation of assemblies for minimum manufacturing cost, *Applied Mechanics and Materials*, Vol. 52-54, 1818-1823, doi:[10.4028/www.scientific.net/AMM.52-54.1818](https://doi.org/10.4028/www.scientific.net/AMM.52-54.1818)
- [26] Gill, A. (2008). Identifying potential bottlenecks through activity under-utilization cost, *International Journal of Simulation Modelling*, Vol. 7, No. 4, 165-175, doi:[10.2507/IJSIMM07\(4\)1.104](https://doi.org/10.2507/IJSIMM07(4)1.104)
- [27] Lestan, Z.; Brezocnik, M.; Buchmeister, B.; Brezovnik, S.; Balic, J. (2009). Solving the job-shop scheduling problem with a simple genetic algorithm, *International Journal of Simulation Modelling*, Vol. 8, No. 4, 197-205, doi:[10.2507/IJSIMM08\(4\)2.138](https://doi.org/10.2507/IJSIMM08(4)2.138)
- [28] Sasiadek, M. (2015). Planning and analysis of mechanical assembly sequences in design engineering – Part I: the method, *Technical Gazette*, Vol. 22, No. 2, 337-342, doi:[10.17559/TV-20130428110530](https://doi.org/10.17559/TV-20130428110530)
- [29] Govindarajalu, J.; Karuppan, S.; Manoharan, T. (2012). Tolerance design of mechanical assembly using NSGA II and finite element analysis, *Journal of Mechanical Science and Technology*, Vol. 26, No. 10, 3261-3268, doi:[10.1007/s12206-012-0811-y](https://doi.org/10.1007/s12206-012-0811-y)

- [30] Cheng, Q.; Zhan, C.; Liu, Z.; Zhao, Y.; Gu, P. (2015). Sensitivity-based multidisciplinary optimal design of a hydrostatic rotary table with Particle Swarm Optimization, *Strojniski vestnik – Journal of Mechanical Engineering*, Vol. 61, No. 7-8, 432-447, doi:[10.5545/sv-jme.2015.2478](https://doi.org/10.5545/sv-jme.2015.2478)
- [31] Sivakumar, K.; Balamurugan, C.; Ramabalan, S. (2011). Evolutionary sensitivity-based conceptual design and tolerance allocation for mechanical assemblies, *International Journal of Advanced Manufacturing Technology*, Vol. 48, No. 1-4, 307-324, doi:[10.1007/s00170-009-2256-8](https://doi.org/10.1007/s00170-009-2256-8)
- [32] Ragu, K. (2014). Experimental analysis of die clearance distribution in a prestool assembly, *Transactions of FAMENA*, Vol. 38, No. 4, 55-64
- [33] Coelho dos Santos, L. (2009). Self-organizing migration algorithm applied to machining allocation of clutch assembly, *Mathematics and Computers in Simulation*, Vol. 80, No. 2, 427-435, doi:[10.1016/j.matcom.2009.08.003](https://doi.org/10.1016/j.matcom.2009.08.003)
- [34] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Publishing, Reading
- [35] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 182-197, doi:[10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
- [36] Storn, R.; Price, K. (1995). *Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces* (Technical Report TR-95-012), International Computer Science Institute, Berkeley
- [37] Das, S.; Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art, *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 1, 4-31, doi:[10.1109/TEVC.2010.2059031](https://doi.org/10.1109/TEVC.2010.2059031)