OPTIMAL TOLERANCE ALLOCATION FOR A MACHINE COMPONENT USING INTELLIGENT TECHNIQUE

A PROJECT REPORT

Submitted By

M. Sambathkumar - 71206402007

In partial fulfillment for the award of the degree of

MASTER OF ENGINEERING

in

CAD/CAM

DEPARTMENT OF MECHANICAL ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

ANNA UNIVERSITY :: CHENNAI 600 025

JUNE – 2008

Bonafide Certificate

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled "**Optimal Tolerance Allocation for a Machine Component Using Intelligent Technique**" is the bonafide work of

Mr. M. Sambathkumar - **Register No. 71206402007**

who carried out the project work under my supervision.

Signature of the HOD

Signature of the Supervisor

Internal Examiner

External Examiner

DEPARTMENT OF MECHANICAL ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE – 641 006

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE , TAMILNADU

DEPARTMENT OF MECHANICAL ENGINEERING & KCT-TIFAC CORE

ADVANCES IN MECHANICAL SCIENCES

CERTIFICATE

This is to certify that Mr/Ms/Mrs M. SAMBATHKUMAR has participated and presented a paper titled Optimal tolerance allocation for a ball screw assembly in the 2nd National Conference on "ADVANCES IN MECHANICAL SCIENCES" during 27- 28, March 2008.

Dr.P.PALANISAMY
CO-ORDINATOR

C Sivanandan
Dr.C.SIVANANDAN
CONVENOR & DEAN

Dr.JOSEPH V.THANIKAL
PRINCIPAL

BANNARI AMMAN INSTITUTE OF TECHNOLOGY
Stay Ahead

OPTEST 2008

TÜV CERT EN ISO 9001 Certificate : 0410021309
RWTÜV

Third National Conference on

Optimization Techniques in Engineering Sciences and Technologies

Certificate

This is to certify that Mr./Ms./Mrs. M. SAMBATH KUMAR

has participated / presented a paper entitled Optimal Tolerance Allocation for a Machine Component using Intelligent Techniques

in the Third National Conference on "Optimization Techniques in Engineering Sciences and Technologies" (OPTEST -2008), during 27-28 March 2008, organized by the Department of Mechanical Engineering, Bannari Amman Institute of Technology, Sathyamangalam.

G SASIKUMAR
Organizing Secretary

Dr A M K PODUVAL
Convener

Dr A SHANMUGAM
Chairman

Abstract

ABSTRACT

Tolerancing plays a major role in the performance and reliability of the assembly. Tolerance is used to control size, location, and geometry of dimensions of components to ensure that the components assemble and assembly meets the functional requirements. It is very important to know how to allocate tolerances economically for parts in a CAD/CAM system because this directly affects the machining cost of the parts. However tolerances are assigned out of intuition to satisfy design constraints and this approach is relying on the experience of the designers and the results may not be optimum. Based on the analysis results, modifications are made manually by a "trial and error" method. This method relies on the experience of the designers and the results may not be optimal.

This work explores an important aspect of tolerance charting; one of the main objectives in tolerance charting is to determine the working dimensions and tolerances at the lowest cost without violating the blueprint specifications. Here a new approach to tolerance charting is followed. Genetic algorithm techniques have been used in this work to find the optimal tolerance value and the work has been further extended by finding out the optimal tolerance value based on the cost output from PSO technique. Then the cost comparisons for the obtained optimal value based on genetic algorithm and PSO have also been presented in this work. It has been found out that the optimal tolerance associated for the selected machine component under specified conditions is achieved by PSO algorithm, than that of Genetic algorithm.

ஆய்வு சுருக்கம்

இயந்திர பாகங்களை ஒன்று கூட்டும் போதோ அல்லது அதன் செயல்பாட்டினை மேம்படுத்தும் போதோ பாகங்களின் அளவுகளில் ஏற்படும் மாறுபாடுகள் (பொறுதி) முக்கிய பங்கு வகிக்கிறது. ஒன்று கூட்டிய இயந்திர பாகத்தின் தோற்றத்தை கட்டுபடுத்தவோ, இடம் பெயர்க்கவோ, அளவினை குறைக்க அல்லது அதிகரிக்க செய்ய பொறுதி முக்கிய பங்காற்றுகிறது. இவற்றை செய்வதன் மூலம் தேவையான செய்லபாட்டு திறனை பெறமுடியும். ஒரு கணினி மூலம் வடிவமைத்தல் / கணினி மூலம் தயாரித்தல் அமைப்பில் உள்ள இயந்திர உதிரி பாகங்களுக்கு பொறுதி ஒதுக்கீடு செய்தல் முக்கிய செயல் ஆகும், ஏனென்றால் பொறுதியை சரியான முறையில் ஒதுக்கீடு செய்யாவிட்டால் அவை பாகங்களுக்கான தயாரிப்பு செலவுகளை நேரிடையாக பாதிக்கும், மேலும் ஒதுக்கீடு செய்யப்பட்ட பொறுதி வடிவமைப்பின் கட்டு திட்டங்களை பூர்த்தி செய்தாக வேண்டும். இந்த செயல்பாடுகள் அனைத்தும் வடிவமைப்போரின் அனுபவ அறிவை சார்ந்தது. இம்முறையின் முடிவுகளை பொறுத்து சில மாற்றங்களை "சரிபார்ப்பு மற்றும் பிழை" முறை மூலம் சுலபமாக மேற்கொள்ளலாம்.

இந்த திட்ட ஆய்வில், பொறுதி வரைபட முறை பயன்படுத்தப்பட்டுள்ளது. இந்த பொறுதி வரைபடத்தின் முக்கிய நோக்கம் யாதெனில் இயந்திர பாகத்தின் செயல்பாடின் அளவுகளையும் மற்றும் பொறுதிகளையும் மிக குறைந்த விலையில் எந்த ஒரு பாதிப்புமின்றி அதன் முக்கிய விவர வரையறைகளை பெறமுடியும்.

இந்த ஆய்வில் "மரபு நெறிமுறையை" பயன்படுத்தி உகப்பு பொறுதியின் அளவை கண்டறியபடுகிறது. மேலும் "துகள் பிழைத் தொகுதி உகப்பாக்கம்" முறை மூலம் இயந்திர பாகத்திற்கான விலை முடிவுகளை கண்டறியப்பட்டு அதனுடன் "மரபு நெறிமுறை"யின் விலை முடிவுகளுடன் ஒப்பிடப்பட்டுள்ளது. ஒப்பிடப்பட்ட விலை முடிவுகளை வைத்து "துகள் பிழைத் தொகுதி உகப்பாக்கம்" முறை மூலம் பெறபட்ட முடிவுகள் மற்ற முறைகளை காட்டிலும் நேர்த்தியான ஒன்று எனக் கண்டறியப்பட்டுள்ளது.

Acknowledgement

ACKNOWLEDGEMENT

The author grateful to his guide **Mr. S. Kalidass**, lecturer, Department of Mechanical Engineering, Kumaraguru College of Technology, Coimbatore for his excellent, utmost motivation, valuable advice, untiring support, timely suggestion, constant encouragement, enthusiasm, relentless patience, and inspiration throughout the study, holding in all the places.

I am grateful to **Dr. C. Sivanandan**, Dean and Head of the Department, Department of Mechanical Engineering for his valuable suggestions and timely help towards my project. I wish to express my deep sense of gratitude to **Dr. Joseph V. Thanikal**, Principal, Kumaraguru College of Technology for patronizing me, besides providing all assistance.

The author expresses his heartfelt thanks to **Dr. K. Sivakumar,** Professor, Department of Mechanical Engineering, Bannari Amman Institute of Technology, Sathyamangalam.

The author owes his immense gratitude and thanks to **Dr. P. Palanisamy,** project coordinator, Department of Mechanical Engineering, Kumaraguru College of Technology, Coimbatore, for his valuable guidance, encouragement and support during the entire course of this project.

The author owes his immense gratitude and thanks to **Dr. T. Kannan**, Associate Professor, Department of Mechanical Engineering, Kumaraguru College of

Technology, Coimbatore, for his valuable guidance, encouragement and support during the entire course of this project.

Words are inadequate in offering my thanks to the **faculty members** of the Department of Mechanical Engineering, Kumaraguru College of Technology for their encouragement and cooperation in carrying out my project work. My sincere thanks also to all **friends** for their help in all the matters during my project work.

Finally, I want to thank my dearest ones, my father **Mr. S. Mani**, my mother **Mrs. M. Manimegalai** and my dear sister **Miss. M. Revathi** for their support and encouragement throughout the different stages of my project work.

Content

CONTENTS

List of Tables

LIST OF TABLES

List of Figures

LIST OF FIGURES

Chapter 1

Introduction

CHAPTER 1

INTRODUCTION

1.1 DESIGN TOLERANCE

Tolerance can be defined, fundamentally, as limits or boundaries. In realm of mechanical engineering and product manufacturing, tolerance is defined as "The total amount by which a given dimension may vary, or the difference between the limits."

Tolerance technique has evolved over the years and has reached their current state through an ever increasing quest for efficiency in producing products that satisfy customer requirements. As society has progressed to the current standard of living, manufacturing technology has had a lot to do with the expectations that have developed; conversely the ever increasing demand for higher quality products and services has stimulated remarkable changes in manufacturing process and in the upstream product development and design processes.

Trying to fit a manufacturing process to the design tolerance just prior to production is a suboptimal approach in developing design and manufacturing process capability. Tolerances have a leading role in engineering, as it effects on the manufacturing process through deviations. Furthermore defined tolerances guarantee the interchangeability of parts. Technical systems are getting more complex in the development of engineering systems. Therefore one can find the tendency to decrease tolerance zones as a consequence, by keeping the correct function, which would mean making the design robust due to deviations of process chains. Coupling tolerances (clearance, deviation) and the calculation of stiffness (elastic deformation, subsuming thermal) of a product will give a correlation of both as a multi-criterion influence. Therefore, the tolerance contCouplingled constraints will give the designer a calculation model closer to reality.

1.2 NEED FOR OPTIMAL TOLERANCE

As design and functionality concern, tolerances should be as close to zero, which is not possible due to the manufacturing constraints. Thus the tolerance should respect the limited capability of required manufacturing processes as well as the functionality and assembly constraints as discussed by [1]. Mechanical tolerance specification is one of the most important of these potential meeting places.

Determining the optimal assignment of tolerances, especially for complex mechanisms, can be crucial to distinguish a product, that is cost-effective and on schedule from a product that is burdened by scrap, rework and delay. Designers often specify tight tolerances in order to maximize product performance. Manufacturers often prefer loose tolerances in order to maximize yield and lower production costs. Tolerance optimization provides development teams with a way to balance the competing requirements of product performance and total manufacturing costs.

Therefore tolerance can be highly desired to achieve customer's requirements and also produce the high quality product with optimal cost. Optimal tolerance can have the multiple objectives required by the customer requirements, including higher quality and low manufacturing cost. Optimal tolerance can be achieved by a procedure, which is executed iteratively by comparing various achieved solutions. The optimization algorithm begins with the initial design solutions and the iteratively check new design solutions in order to achieve the global optimal solutions.

Chapter 2

Literature Review

CHAPTER 2

LITERATURE REVIEW

The various tolerance synthesis methods are based on conventional optimization methods, quality engineering methods and methods based on genetic algorithm, simulated annealing are near fuzzy learning. The majority of the published articles on tolerance synthesis are based on optimization, most of which use the cost-tolerance models.

Ngoi et al. [1999] this paper describes a new approach for optimum tolerance allocation in assembly. The method allows all blueprint tolerances to be determined while ensuring that all the assembly requirements are satisfied. The algorithm is simple and hence it is suitable for all users. It reduces the amount of work and "guessing" required in the allocation of blueprint tolerances. Moreover, it is assured that the result obtained is an optimum and none of the assembly requirements are violated. Further work is being carried out to integrate tolerance charting with the above approach to determine the blueprint tolerances, working dimensions and tolerances concurrently.

Bryan Ngoi Kok Ann et al. [1996] this paper presents a simple dimensional chains identification method. After establishing all necessary equations and constraints, a nonlinear objective function is formulated. Subsequently, all these relationships are submitted to an optimization software, OPTIVAR written in FORTRAN for the determination of the unknown variables.

Meifa Huang et al. [2006] this paper deals with a concurrent optimal tolerance design methodology for allocating assembly functional DGTs to the pertinent process DGTs has been presented. This method first converts the pertinent geometric tolerances into equivalent bilateral dimensional tolerances or additional tolerance Counterparts. The DGTs are then considered in the same integrated techniques for the tolerance transfer is the tolerance charting method, but it is basically restricted to the one dimensional case.

3.2.7 Tolerance Evaluation

It deals with how to assess the geometric deviations of a part using the data obtained from the coordinate measuring machines.

3.3 TOLERANCE AND COST RELATIONSHIP

The general characteristics of a manufacturing cost tolerance data curve, several general cost-tolerance relation models, including the exponential, reciprocal squared and the reciprocal powers models, were introduced. In addition, it fails to consider the valid range of a cost-tolerance curve to avoid infeasible solutions, and requires manual formulation.

Cost based optimal tolerance analysis techniques are very helpful in promoting economic design for functionality. They require a good deal of insight into developing a proper math model that relates cost and functional quality; once such a model is properly defined, the power of this optimal design tolerance become quite evident. In this work has concentrated on minimizing manufacturing cots and minimize the quality loss value.

Tolerance must be linked to more than the variability that originates in the manufacturing environment; it must have some costs that are incurred to make the product. These costs are primarily represented by the term unit manufacturing cost (UMC). Tolerances must further be developed in context of two more costs.
The life cycle cost (LCC) of the design will account for the broader that is repair and replacement cost associated with the use of the product. This metric is particularly important in industries that must repair and service the products they sell to satisfy customer expectations. Quality loss function (QLF) means when customers requirement could not reached by design parameter or it is deviated for the target point due to some manufacturing constrains, for this the manufacturer have to pay for it. The following Figure 3.1 gives the relationship between these three costs.

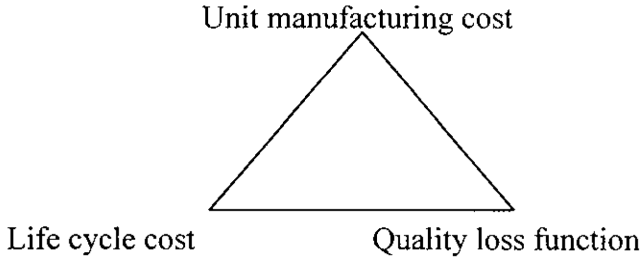

Figure 3.1 Relationship between costs

Tighter the tolerance, the more expensive it is to manufacture a part. This trend provides fundamental rule in selecting tolerances by the designers at the design phase; that is, tolerances should be chosen as large as possible as long as they meet the functional and assembly requirements of the part. It may be worth while to change designs to relax tolerance requirement for the cost purposes. Larger tolerances result in using less skilled machines, lower inspection costs and reduced scrapping of material.

3.4 QUALITY LOSS FUNCTION

Companies that are practicing on target engineering use an alternative approach to the limitations the step function exhibits as a measure of quality. The quality loss function was developed by Taguchi to provide a better estimate of the monetary loss incurred by manufacturers and consumers as product performance deviates from its target value. The quality loss function can be shown as equation (3.1).

$$L(y) = k(y-m)^2 \qquad \text{-------(3.1)}$$

Where, L(y) is the loss in dollars due to a deviation away from targeted performance as a function of measured response y of product; m is the target value of the product's response; and k is an economic constant called quality loss coefficient. Figure 3.2 illustrates the quality loss function. At y = m, the loss is zero, and it increases the further y deviates from m. the quality loss curve typically represents the quality loss for an average group of customers. The quality loss for a specific customer would vary depending on customer's tolerance and usage environment. However, it is not necessary to derive an exact loss function for all situations. That would be too difficult and not generally applicable. The quality loss function can be viewed on several levels:

- As a unifying concept of quality and cost that allows one to practice the underlying philosophy driving on target engineering

- As a function that allows one to relate economic and engineering terms in one model.

- As an equations that allows one to do detailed optimization of all cost, explicit and implicit, incurred by the firm, customers and society through the production and use of a product.

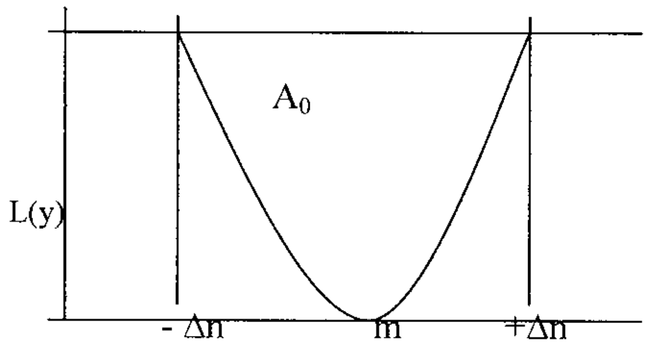

Figure 3.2 The Quality Loss Function

Chapter 4

Tolerance Allocation Methods

CHAPTER 4

TOLERANCE ALLOCATION METHODS

4.1 INTRODUCTION

Tolerance allocation is a design function. It is performed early in the product development cycle, before any parts have been produced or tooling ordered. It involves first, deciding what tolerance limits to place on the critical clearances and fits for an assembly, based on performance requirements; second, creating an assembly model to identify which dimensions contribute to the final assembly dimensions; third, deciding how much of the assembly tolerance to assign to each of the contributing components in the assembly.

A defective assembly is one for which the component variations accumulate and exceed the specified assembly tolerance limits. The yield of an assembly process is the percent of assemblies which are not defective. In tolerance analysis, component variations are analyzed to predict how many assemblies will be in spec. If the yield is too low, rework, shimming, or parts replacement may be required.

In tolerance allocation, an acceptable yield of the process is first specified and component tolerances are then selected to assure that the specified yield will be met. Often, tolerance design is performed by repeated application of tolerance analysis, using trial values of the component tolerances. However, a number of algorithms have been proposed for assigning tolerances on a rational basis, without resorting to trial and error.

4.2 PROPORTIONAL SCALING METHOD

The designer begins by assigning reasonable component tolerances based on process design guide lines. Then he sums the component tolerances by a constant proportionality factor. In this the relative magnitude of the component tolerances are preserved.

This method is demonstrated graphically in Figure. 4.1 for an assembly tolerance Tasm, which is the sum of two component tolerances, T1 and T2. The straight line labeled as the Worst Case Limit is the locus of all possible combinations of T1 and T2 which, added linearly, equal Tasm. The ellipse labeled Statistical Limit is the locus of root sum squares of T1 and T2 which equal Tasm. The following equations figure out these two cases.

Worst Case Limit

Tasm = T1 + T2 + T3 + ...+ Tn ----- (4.1)

Statistical Limit

$$Tasm = \sqrt{T_1^2 + T_2^2 + T_3^2 + \cdots\cdots + T_n^2} \quad \text{-----}(4.2)$$

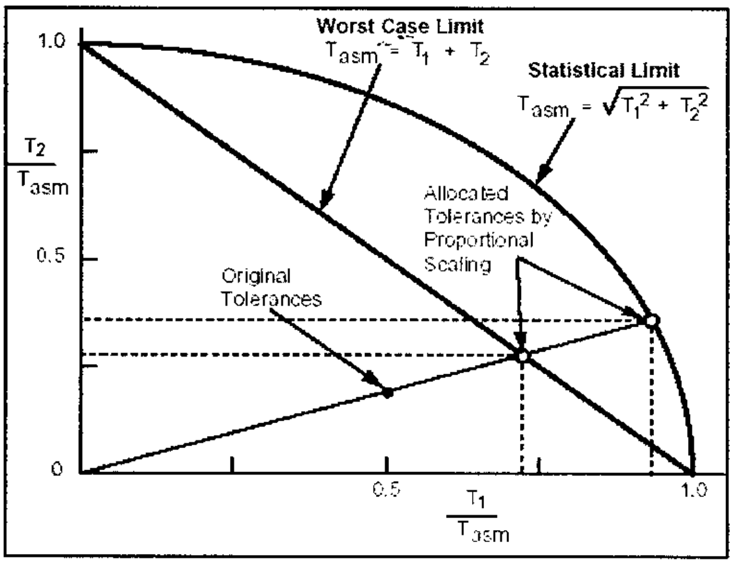

Figure 4.1 Graphical interpretation of tolerance allocation by proportional Scaling

4.3 ALLOCATION BY WEIGHT FACTORS

A more versatile method of assigning tolerances is by means of weight factors W. Using this algorithm, the designer assigns weight factors to each tolerance in the chain and the system distributes a corresponding fraction of the tolerance pool to each component. A larger weight factor W for a given component means a larger fraction of the tolerance pool will be allocated to it.

In this way, more tolerance can be given to those dimensions which are the more costly or difficult to hold, thus improving the producibility of the design. Figure 4.2 illustrates this algorithm graphically for a two component assembly. The original values for component tolerances T1 and T2 are selected from process considerations and are represented as a point in the figure, as before. The tolerances are scaled, similar to proportional scaling; only the scale factor is weighted for each component tolerance so the greater scale factors yield the least reduction in tolerance.

Worst Case Limit

Tasm = W1 T1 + W2 T2 +W3 T3 + ...+ Wn Tn -----(4.3)

Statistical Limit

$$\text{Tasm} = \sqrt{W_1T_1^2 + W_3T_2^2 + W_3T_3^2 \cdots\cdots\cdots + W_nT_n^2} \quad \text{----(4.4)}$$

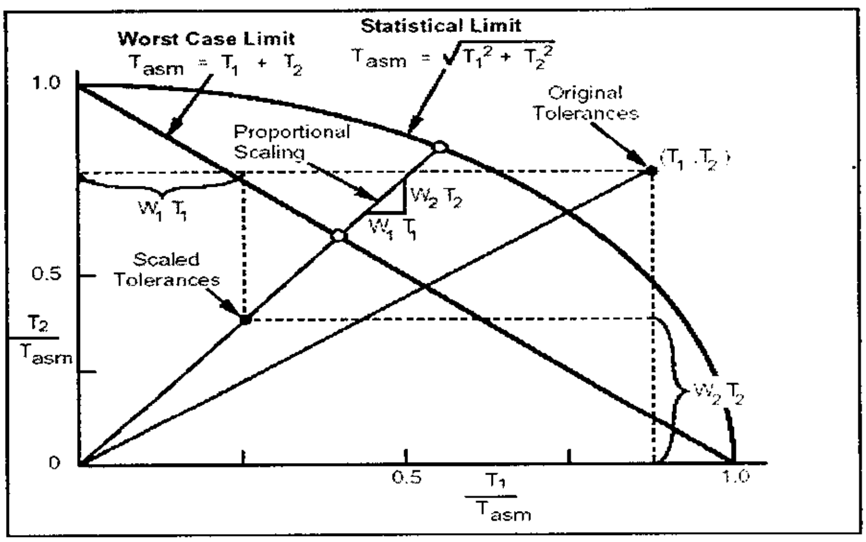

Figure 4.2 Graphical interpretation of tolerance allocation by weight factors.

4.4 CONSTANT PRECISION FACTOR METHOD

Parts machined to a similar precision will have equal tolerances only if they are of same size. As part size increases, tolerance generally increases approximately with the cube root of size.

$$\text{Tolerance } T_i = P\,(D_i)^{1/3} \qquad \text{----- (4.5)}$$

Where,

D_i = basic size of the part

P = Precision factor

Based on this rule, the tolerance can be distributed accordingly to the part size. Precision factor method is similar to the proportional scaling method, except there is no initial allocation required by the designer. Instead the tolerances are initially allocated according to the nominal size of each component dimension and the scaled to meet the specified assembly tolerance.

4.5 TAGUCHI METHOD

The Taguchi method not only determines tolerance but also determines the ideal nominal values for the dimensions. This is referred to as dimension. The method finds the nominal dimensions that allow the largest, lowest-cost tolerances to be assigned. It selects dimensions and tolerance with regards to their effect on a single design function. The method uses fractional factorial experiments to find the nominal dimensions and tolerance that maximize the so-called signal to noise ratio. The signal is a measure of how close the design function is to its desired nominal value. The noise is a measure of the variability of the design function caused by tolerances. The main disadvantage of the Taguchi method is its inability to handle more than one design function. Finding one design function for a product for a product may not be at all practical.

4.6 TOLERANCE ALLOCATION USING LEAST COST OPTIMIZATION

A promising method of tolerance allocation uses optimization techniques to assign component tolerances such that the cost of production of an assembly is minimized. This is accomplished by defining a cost-vs.-tolerance curve for each component part in the assembly. The optimization algorithm varies the tolerance for each component and searches systematically for the combination of tolerances which minimizes the cost.

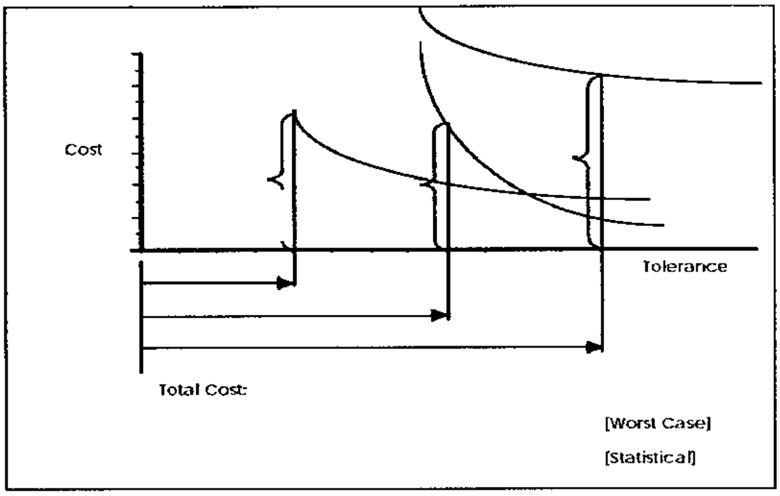

Figure 4.3 Optimal tolerance allocations for minimum cost

Figure 4.3 illustrates the concept simply for a three component assembly. Three cost-vs.-tolerance curves are shown. Three tolerances (T1, T2, T3) are initially selected. The corresponding cost of production is C1 + C2 + C3. The optimization algorithm tries to increase the tolerances to reduce cost; however, the specified assembly tolerance limits the tolerance size. If tolerance T1 is increased, then tolerance T2 or T3 must decrease to keep from violating the assembly tolerance constraint. It is difficult to tell by inspection which combination will be optimum, but you can see from the figure that a decrease in T2 results in a significant decrease in cost, while a corresponding decrease in T3 results in smaller increase in cost. In this manner, one could manually adjust tolerances until no further cost reduction is achieved. The optimization algorithm is designed to find it with a minimum of iteration. Note that the values of the set of optimum tolerances will be different when the tolerances are summed statistically than when they are summed by worst case.

4.7 TOLERANCE ANALYSIS Vs. TOLERANCE ALLOCATION

The analytical modeling of assemblies provides a quantitative basis for the evaluation of design variations and specification of tolerances. An important distinction in tolerance specification is that engineers are more commonly faced with the problem of tolerance allocation rather than tolerance analysis.

The difference between these two problems is illustrated in Figure 4.4. In tolerance analysis the component tolerances are all known or specified and the resulting assembly variation is calculated. In tolerance allocation, on the other hand, the assembly tolerance is known from design requirements, whereas the magnitudes of the component tolerances to meet these requirements are unknown. The available assembly tolerance must be distributed or allocated among the components in some rational way. The influence of the tolerance accumulation model and the allocation rule chosen by the designer on the resulting tolerance allocation will be demonstrated.

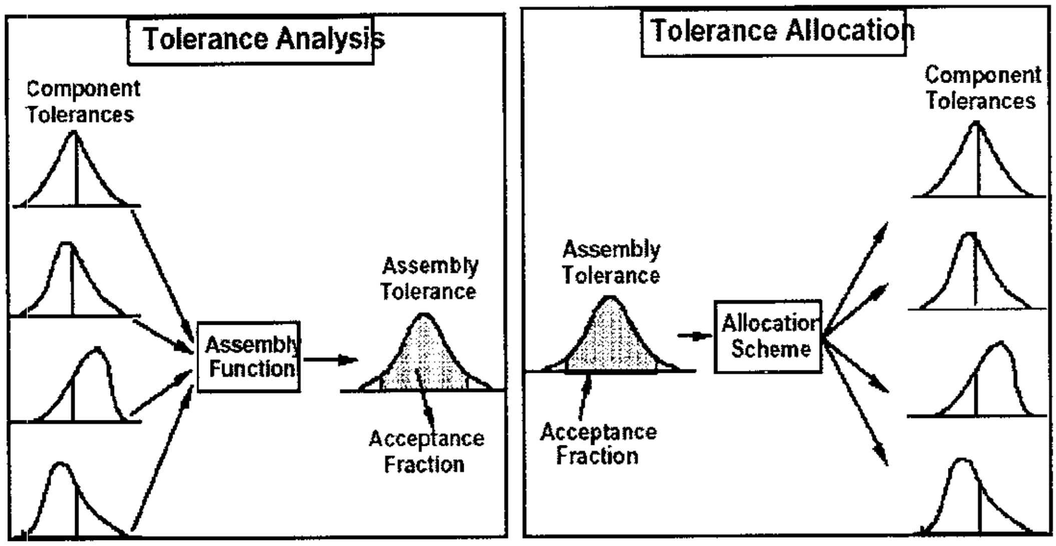

Figure 4.4 Tolerance Analysis Vs Tolerance Allocation

Another difference in the two problems is the yield or acceptance fraction of the assembly process. The assembly yield is the quality level. It is the percent of assemblies which meet the engineering tolerance requirements. It may be expressed as the percent of acceptable assemblies or the percent rejects. For high quality levels, the rejects may be expressed in parts-per-million (ppm), that is, the number of rejects per million assemblies.

In tolerance analysis the assembly yield is unknown. It is calculated by summing the component tolerances to determine the assembly variation, then applying the upper and lower spec limits to the calculated assembly distribution. In tolerance allocation, on the other hand, the assembly yield is specified as a design requirement. The component tolerances must then be set to assure that the resulting assembly yield meets the specifications. The rational allocation of component tolerances requires the establishment of a rule for distributing the assembly tolerance among the components.

Chapter 5

Optimization Techniques

CHAPTER 5

OPTIMIZATION TECHNIQUES

5.1 OPTIMIZATION

Optimization is the act of obtaining the best result under given circumstances. In design, constructing and maintenance of any engineering system, engineers/managers have to take many technological and managerial decisions at several stages, figure 5.1 shows the steps involved in the optimization process. The ultimate goal of all such decisions is to either minimize the effort required or maximize the desired benefit.

Mechanical engineers design mechanical equipments like pumps, turbines and heat transfer equipment for maximum efficiency and mechanical components like linkages, cams, and gears, machine tools for the purpose of achieving either a minimum manufacturing cost or a maximum component life. Production engineers are interested in designing optimum schedules of various machining operations to minimize the idle time of machines and the overall job completion time.

5.2 TYPES OF OPTIMIZATION TECHNIQUE

The following are the types of optimization. They are

1. Traditional optimization technique and
2. Non- Traditional optimization technique

5.2.1 Traditional and Non-traditional Techniques

Traditional techniques for optimization include linear programming, random search method, geometric programming, dynamic programming and integer programming.

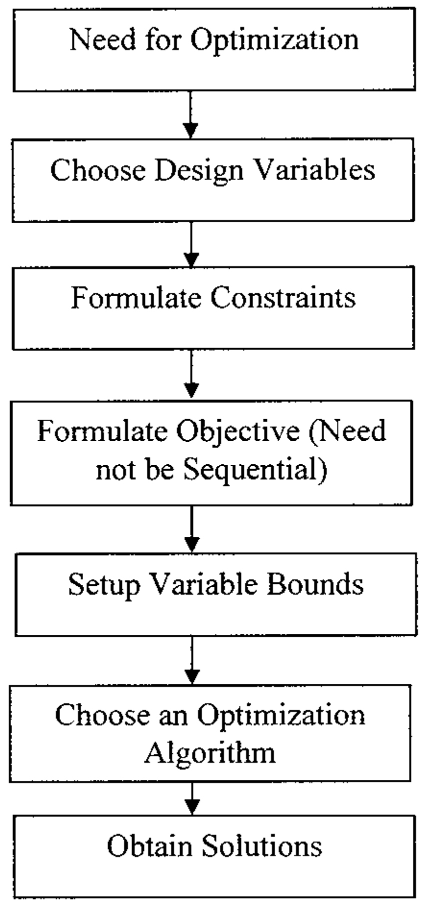

Figure 5.1 Optimization Process-Steps

5.2.1.1 Linear Programming

Linear programming is an optimization method applicable for the solution of problems in which the objective function and the constraints appear as linear functions of the decision variables. The constraint equations in a linear programming problem may be in the form of equalities or inequalities. The linear programming type of optimization problem was first recognized in the 1930's by economist while developing methods for the optimal allocation of resources.

Linear programming is considered a revolutionary development that permits us to make optimal decisions in complex situations. At least four Noble Prizes were awarded for contributions related to linear programming.

Although several other methods have been developed over the years for solving LP problems, the simplex method continues to be the most efficient and popular method for solving general LP problems.

5.2.1.2 Random Search Method

The random search method described for unconstrained minimization can be used, with minor modifications, to solve a constrained optimization problem. The basic procedure can be described by the following steps:

a. Generate a trial design vector using one random number for each design variable.
b. Verify whether the constraints are satisfied at the trial design vector. Usually, the equality constraints are considered satisfied whenever their magnitudes lie within a specified tolerance. If any constraint is violated, continue generating new trial vectors until a trial vector that satisfies all the constraints is found.
c. If all the constraints are satisfied, retain the current trial vector as the best design if it gives a reduced objective function value compared to the previous best available design. Otherwise, discard the current feasible trial vector and proceed to step 1 to generate a new trial design vector.
d. The best design available at the end of generating a specified maximum number of trial design vectors is taken as the solution of the constrained optimization problem.

5.2.1.3 Geometric Programming

Geometric Programming is a relatively new method of solving a class of non linear programming problems. It was developed by Duffin, Peterson and Zener. It is used to minimize functions that are in the form of polynomials subject to constraints of the same type. It differs from other optimization techniques in the emphasis it places on the relative magnitudes of the terms of the objective function rather than the variable. Instead of finding optimal values of the design variables first, geometric programming first finds the optimal value of the objective function. This feature is especially advantageous in situations where the optimal value of the objective function may be all that is of interest. In such cases, calculation of the optimum design vectors can be omitted. Another advantage of geometric programming is that it often reduces a complicated optimization problem to one involving a set of simultaneous linear algebraic equations. The major disadvantage of the method is that it requires the objective function and the constraints in the form of polynomials.

5.2.1..4 Dynamic Programming

Dynamic programming is a mathematical technique well suited for the optimization of multistage decision problems. This technique was developed by Richards Bellman in the early 1950s.

The dynamic programming technique, when applicable, represents of decomposes a multistage decision problem as a sequence of single-stage decision problems. Thus an N-variable problem is represented as a sequence of N single-variable problems that are solved successively. In most cases, these N sub-problems are easier to solve than the original problem. The decomposition to N sub-problems is done in such a manner that the optimal solution of the original N-variable problem can be obtained from the optimal solutions of the N one-dimensional problems. It is important to note that the particular optimization technique used for the optimization of the N single- variable problems is irrelevant. It may range from a simple enumeration process to a differential calculus or a nonlinear programming technique.

The dynamic programming technique suffers from a major drawback, known as the curse of dimensionality. However, despite this disadvantage, it is very suitable for the solution of a wide range of complex problems in several areas of decision making.

5.2.1.5 Integer Programming

When all the variables are constrained to take only integer values in an optimization problem, it is called an integer programming problem. When the variables are restricted to take only discrete values, the problem is called a discrete programming problem. When some variables only are restricted to take integer values, the optimization problem is called a mixed-integer programming problem. When all the design variables of an optimization problem are allowed to take on values of either zero or one, the problem is called zero-one programming problem.

Non-traditional techniques for optimization include fuzzy logic, search technique, genetic algorithm, Taguchi technique and response surface methodology.

5.2.1.6 Fuzzy Logic

Fuzzy logic has great capability to capture human commonsense reasoning, decision-making and other aspects of human cognition. It overcomes the limitation of classical logical systems, which impose inherent restrictions on representation of imprecise concepts. Vagueness in the coefficients and constraints may be naturally modeled by fuzzy logic. Modelling by fuzzy logic opens up a new way to optimize cutting conditions and also tool selection.

5.2.1.7 Genetic Algorithm

These are the algorithms based on mechanics of natural selection and natural genetics, which are more robust and more likely to locate global optimum. It is because of this feature that GA goes through solution space starting from a group of points and not from a single point. The cutting conditions are encoded as genes by binary encoding to apply GA in optimization of machining parameters. A set of genes is combined together to form chromosomes, used to perform the basic mechanisms in GA, such as crossover and mutation.

Crossover is the operation to exchange some part of two chromosomes to generate new offspring, which is important when exploring the whole search space rapidly. Mutation is applied after crossover to provide a small randomness to the new chromosomes. To evaluate each individual or chromosome, the encoded cutting conditions are decoded from the chromosomes and are used to predict machining performance measures. Fitness or objective function is a function needed in the optimization process and selection of next generation in genetic algorithm. Optimum results of cutting conditions are obtained by comparison of values of objective functions among all individuals after a number of iterations. Besides weighting factors and constraints, suitable parameters of GA are required to operate efficiently. GA optimization methodology is based on machining performance predictions models developed from a comprehensive system of theoretical analysis, experimental database and numerical methods. The GA parameters along with relevant objective functions and set of machining performance constraints are imposed on GA optimization methodology to provide optimum cutting conditions.

5.2.1.8 Scatter Search Technique (SS)

This technique originates from strategies for combining rules and surrogate constraints. SS is completely generalized and problem-independent since it has no restrictive assumptions about objective function, parameter set and constraints set. It can be easily modified to optimize machining operation under various economic criteria and numerous practical constraints. It can be extended as an on-line quality control strategy for optimizing machining parameters based on signals from sensors.

5.2.1.9 Taguchi Technique

Genichi Taguchi is a Japanese engineer who has been active in the improvement of Japan's industrial products and processes since the late 1940's. He has developed both the philosophy and methodology for process of products quality improvement that depends heavily on statistical concepts and tools, especially statistically designed experiments. Many Japanese firms have achieved great success by applying his methods. Wu (1982) has reported that thousands of engineers have performed tens of thousands of experiments based on his teachings. Sullivan (1987) reports that Taguchi has received some of Japan's most prestigious awards for quality achievement, including the Deming prize. In 1986, Taguchi received the most prestigious prize from the International Technology major contribution has involved combining engineering and statistical methods to achieve rapid improvements in cost and quality by optimizing product design and manufacturing processes.

5.2.1.10 Response Surface Methodology

Experimentation and making inferences are the twin features of general scientific methodology. Statistics as a scientific disciplines is mainly designed to achieve these objectives. Planning of experiments is particularly very useful in deriving clear and accurate conclusions from the experimental observations, on the basis of which inferences can be made in the best possible manner. The methodology for making inferences has three main aspects. First, it establishes methods for drawing inferences from observations when these are not exact but subject to variation, because inferences are not exact but probabilistic in nature. Second, it specifies methods for collection of data appropriately, so that assumptions for the application of appropriate statistical methods to them are satisfied. Lastly, techniques for proper interpretation of results are devised.

5.3 Advantages of Non-traditional Techniques

The advantages of Non-traditional techniques are

1. A population of points is used for starting the procedure instead of a single design point.
2. GAs use only the values of the objective function. The derivatives are not used in the search procedure.
3. Search method is naturally applicable for solving discrete and integer programming problems. For continuous design variables, the string length can be varied to achieve any desired resolution.
4. The objective function value corresponding to a design vector plays the role of fitness in natural genetics.
5. In every new generation, a new set of strings is produced by using randomized parents selection and crossover from the old generation.

Chapter 6

Problem Definition

CHAPTER 6

PROBLEM DEFINITION

6.1 OPTIMUM TOLERANCES FOR BALL SCREW ASSEMBLY

6.1.1 About Ball Screws

Ball screws are used in the conversion of rotary movement to linear movement, which translates torque into thrust. Ball screw assemblies consist of a screw and a nut. A steel ball is encased within the round nut in order to produce a coupling ling friction between the nut and screw. The nut itself can be made of either plastic or metal. The ball screw assembly is powered by a motor. As the motor generates torque, the rotating screw pushes the nut along the screw shaft, producing linear thrust.

There are a few variations of ball screws available for use in industrial settings. Ball screw manufacturers commonly produce ACME, Lead (pronounced leed) and Ground ball screws. They each differ in size and efficiency output on application. Lead screws do not actually use coupling lers to create movement but are placed in the same category as ball screws because of their similar function and capacities. ACME screws most widely used power screw and are a type of lead screw which creates friction between ball and nut.

Numerous industries, including aerospace, computer, electronic, automotive, and medical industries, utilize ball screws in product applications. Ball screw manufacturers can create ball screws that are used in medical equipment, material handling equipment, conveyors, and machine tools, among many other product applications. The most common use for ball screws is in aspects where linear motion is needed. They are often used alongside linear slides and linear actuators to create movement necessary to move parts and devices along a single axis.

Ball screws remain beneficial for a variety of reasons. Ball screw assemblies maintain high levels of efficiency, measuring approximately 90%, and maintain

low energy consumption levels. In addition, ball screws can be manufactured using a variety of techniques. Common techniques include conventional coupling 1ing, milling, and grinding. These advantages remain important considerations when choosing a screw assembly. Length of the screw is the most crucial component in choosing a ball screw for your application. They are classified however by diameter, either in English or metric and often manufacturers have both labels available. When replacing a ball screw assemblies, consider ball screw repair. Many ball screw manufacturers offer repair services as an alternative to the purchase of new ball screw systems.

6.1.2 Types of Ball Screws

- **ACME screws** are lead screws that create a sliding friction between screw and nut. ACME screws maintain a lower efficiency than ball screws, measuring about 30 %, but are often cost effective. Acme screws are often utilized in applications requiring high levels of accuracy at low speeds.

- **Ground ball screws** are produced using a grinding wheel instead of the conventional coupling 1ing technique. Ground ball screws offer close tolerances, but may be expensive to produce.

- **Jack screws** are frequently used in car jacks.

- **Lead screws** consist of a threaded shaft and nut, and create friction through sliding rather than through the coupling 1ing friction characteristic of ball screws. The efficiency of lead screws increases with increased lead. Lead screws are advantageous in managing high shock loads.

- **Metric ball screws** are designed according to metric system measurements, as opposed to the English system of measurements.

- **Miniature ball screws** , which consist of ball screws measuring as little as three millimeters in diameter, are used in applications in which minute products components are needed, such as industrial applications in the computer, electronic, fiber optics, and semiconductor industries. Miniature ball screws maintain high efficiency levels in spite of their size.

- **Screw jacks** are used in lifting jacks.

This model was proposed by Ngoi [1999], the ball screw assembly given in Figure 6.1 consist of four components; screw, nut, coupling, bearing.

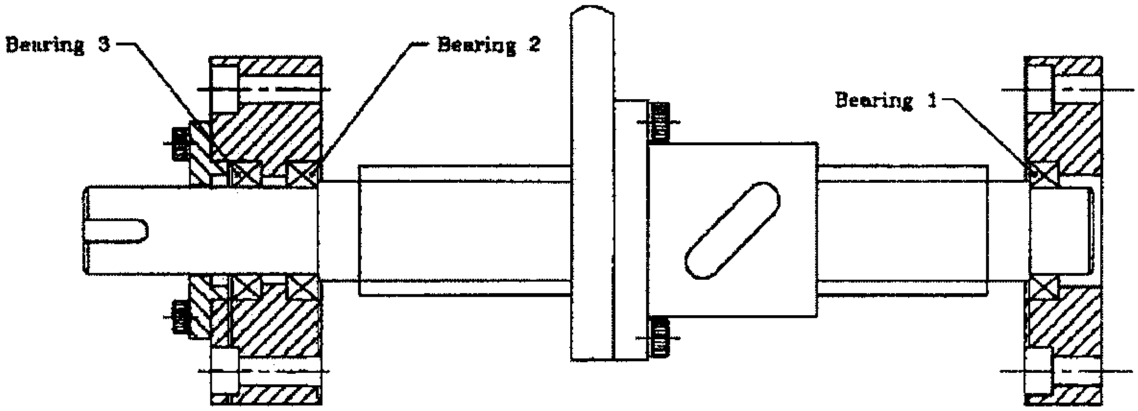

Figure 6.1 Ball Screw Assembly

6.2 INITIAL TOLERANCE ALLOCATION

In an assembly, some tolerances may not be constrained by the assembly requirements. Thus, it is necessary to set an upper limit to each tolerance. The limit should be chosen such that it can be easily attained and it should not violate any other requirements. The initial tolerance allocated based on Table 6.1

The purpose of this step is to first allocate the "loosest" tolerance possible to all dimensions and subsequently, it tightens all the necessary tolerances to meet the assembly requirements.

Table 6.1 Maximum Allowable Tolerance for Dimensions

Dimension	Maximum allowable tolerance
$L \leq 4$	±0.05
$4 < L \leq 16$	±0.1
$16 < L \leq 63$	±0.2
$63 < L \leq 250$	±0.3
$250 < L$	±0.4

6.2.1 Tolerance chart

The main tasks of tolerance charting include the identification of dimensional chains, the calculation of mean working dimensions, and the allocation of proper tolerances to the working dimensions for all machining cuts.

A tolerance chart is a visual graphic tool used by process engineers to determine the mean working dimensions and to assign the corresponding tolerances to the working dimensions for a new manufacturing process. The kernel of tolerance charting is the theory of tolerance chains. Once the tolerance chart is constructed, the first task is to identify tolerance chains among the working dimensions. Traditionally, a tracing method is used to identify tolerance chains in a tolerance chart. If the component consists of square-shouldered features only, the identification of the tolerance chains is not very difficult and a one-dimensional (1D) tolerance chart is good enough to deal with all problems that may be encountered. However, when an angular feature is involved, a two-dimensional (2D) tolerance chart must be constructed and the identification of the tolerance chains becomes much more difficult.

The tolerance chart diagrams are shown in the below:

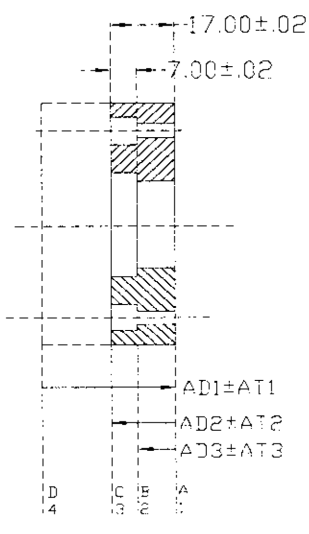

Figure 6.2 Tolerance Chart Diagram for part A

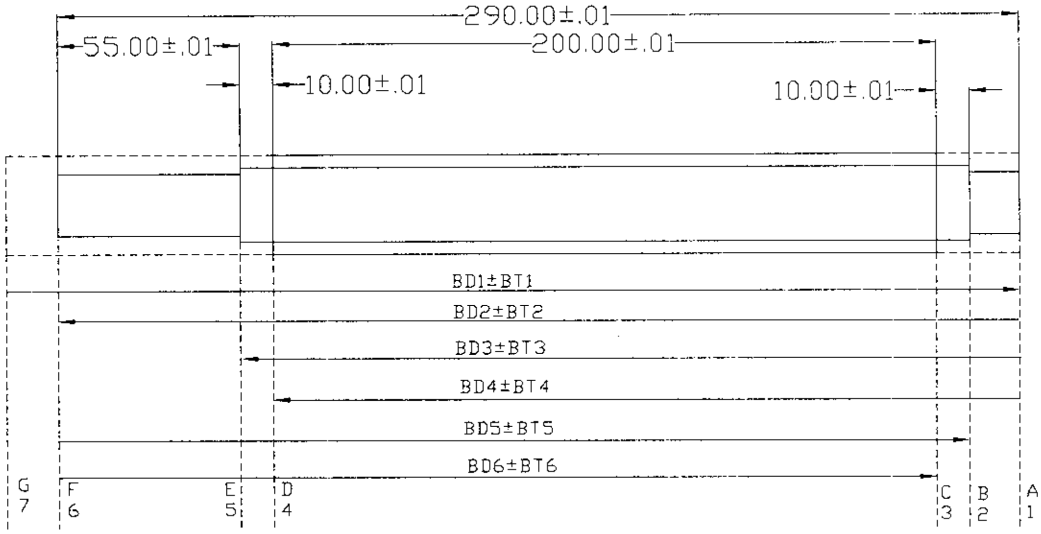

Figure 6.3 Tolerance Chart Diagram for part B

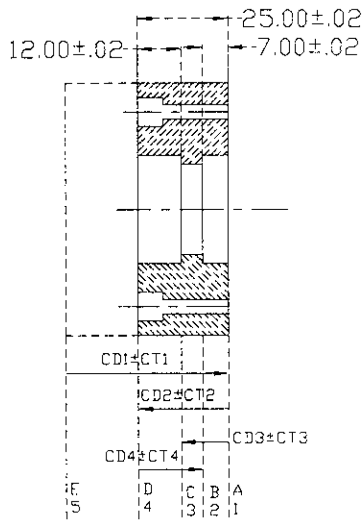

Figure 6.4 Tolerance Chart Diagram for part C

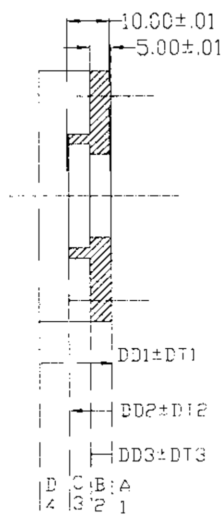

Figure 6.5 Tolerance Chart Diagram for part D

6.3 Formulating Objective Function and Constraint Equations:

The aim of the tolerance chart procedure is to loosen the process tolerance efficiently while ensuring that the blueprint specifications are not violated. The aim may be interpreted as maximizing the total additional tolerances to the existing process tolerance. It can, therefore, be formulated into an objective function and mathematically represented as:

Maximize $\sum = (*D1+*D2+*D3+ \ldots\ldots\ldots\ldots *Dn)$

Maximize $\sum = (*T1+*T2+*T3+ \ldots\ldots\ldots\ldots *Tn)$

Where

n = number of unknown working dimension or tolerance in the respective part.

Therefore the objective function of the working dimension and tolerances are

Maximize (AD1+AD2+AD3+BD1+BD2+BD3+BD4+BD5+BD6+
CD1+CD2+CD3+CD4+DD1+DD2+DD3) -----(6.1)

And

Maximize (AT1+AT2+AT3+BT1+BT2+BT3+BT4+BT5+BT6+
CT1+CT2+CT3+CT4+DT1+DT2+DT3) -----(6.2)

Having represented the objective function mathematically, the next step is to formulate the constraints. The constraints are governed by the relationship between the resultant tolerance and the tolerance specification the blueprint.

Equating the working dimension to the blueprint dimensions.

Example: $-AD3 + AD2 = 7$ ------(6.3)

Equating the working tolerance to the blueprint tolerance.

Example: $AT3+AT2 \leq 0.1$ -------(6.4)

Working dimension

Part 1

For BC:	-AD3 + AD2	= 7
For AC:	+AD2	= 17
For CD:	-AD2 + AD1	= 0.5

Part 2

For EF:	-BD3 + BD2	= 55
For DE:	-BD4 + BD3	= 10
For CD:	+BD6 - BD2 + BD4	= 200
For BC:	+BD5 - BD6	= 10
For AF:	+BD2	= 290
For FG:	-BD2 + BD1	= 0.5

Part 3

For AB:	+CD2 - CD4	= 7
For CD:	-CD3 + CD2	= 12
For AD:	+CD2	= 25
For DE:	-CD2 + CD1	= 0.5

Part 4

For BC:	-DD3 + DD2	= 5
For AC:	+ DD2	= 10
For CD:	-DD2 + DD1	= 0.5

Working tolerance

Part 1

For BC:	+AT3 + AT2	≤ 0.1
For AC:	+AT2	≤ 0.2

Part 2

For EF:	+BT3 + BT2	≤ 0.2
For DE:	+BT4 + BT3	≤ 0.1
For CD:	+BT6 + BT2 + BT4	≤ 0.3
For BC:	+BT5 + BT6	≤ 0.1
For AF:	+BT2	≤ 0.4

Part 3

For AB:	+CT2 + CT4	≤ 0.1
For CD:	+CT3 + CT2	≤ 0.1
For AD:	+ CT2 ≤ 0.2	

Part 4

For BC:	+DT3 + DT2	≤ 0.1
For AC:	+DT2	≤ 0.1

Chapter 7

Genetic Algorithm

CHAPTER 7

GENETIC ALGORITHM

7.1 GENETIC ALGORITHM

Genetic algorithms are a set of computer procedures of search and optimization based of the concept of the mechanics of natural selection and genetics. Holland made the first presentation of the GA techniques in the beginning of the 60's and further development can be credited to Goldberg.

Genetic algorithms implement Optimization strategies based on simulating the evolutionary law of natural selection, to obtain the "fittest individual", that is, the optimal solution. Genetic algorithms work as follows. First, the variable involved is coded into a suitable representation. Next, a population of chromosomes, usually randomly selected from the whole function domain, is created. This population is dynamic: at each iteration, some elements of it are chosen to reproduce, via suitable operators, according to their capability of adaptation to the environment. Key parameters of GA involve crossover probability, mutation probability, and population size.

7.2 GA PARAMETERS CONTROL

7.2.1 Crossover Probability

It denotes how often crossover will be performed. If crossover probability is 100%, then all offspring are made by crossover. If it is 0%, a next generation is made from exact copies of chromosomes from the old population. Typically, crossover probability should be high (larger than 50%).

7.2.2 Mutation Probability

It denotes how often parts of chromosomes will be mutated. If mutation probability is 100%, the whole chromosome is changed, if it is 0%, nothing is changed. Mutation generally prevents the GA from falling into local extremes. Mutation rate should be low generally (typically less than 5%). The Genetic Algorithm flowchart is shown in Figure 7.1.

7.2.3 Population Size

It denotes how many chromosomes are in population. If there are too few chromosomes, GAs has few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. In the current example the population size was set to eight members.

In an assembly, some tolerances may not be constrained by the assembly requirements. Thus, it is necessary to set an upper limit to each tolerance. The limit should be chosen such that it can be easily attained and it should not violate any other requirements. When the GA tool is opened, M-file is created and invoked in the GA tool of MATLAB 7.inside the simulation toolbox space, the GAs chose, randomly the initial population and scores parameters.

Then, based in the previous information the algorithm chose another setup, which was done and its data again fed into the algorithm. The process is continued until the optimum was found. The parameters of GA computations are shown in the Table 7.1.

In the GA, the population size, crossover rate and mutation rate are important factors in the performance of the algorithms. A large population size or a higher crossover rate allows exploration of the solution space and reduces the chances of settling for poor solution. However, if they are too large of high. it results in wasted computation time exploring unpromising regions of the solution space

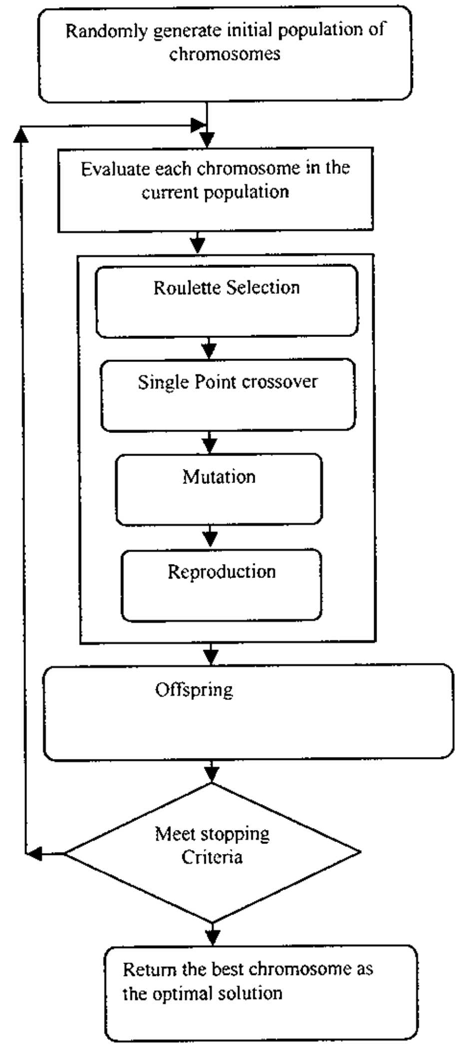

Figure 7.1 Genetic Algorithm flow chart

. Table 7.1 Parameters of GA computation

Population type	Double Vector
Population size	40
Fitness scaling function	Rank
Selection function	Roulette
Reproduction elite count	2
Crossover rate	100%
Crossover function	Single
Mutation function	Uniform
Mutation rate	1%
Number generations	63

7.3 OPTIMIZATION OF THE FUNCTION

Several numerical methods are available for optimization of non linear equation with constraints. A Genetic Algorithm method is efficient and quickest one, and this method was used to determine the optimum tolerance to match the assembly limit. The above mentioned objective function is fed into the M-file and run using the GA optimization tool box available in MATLAB 7 obtain the following results shown in the Table 7.2

Following plots and graphs are obtained for the optimized process parameters are shown in figure 7.2

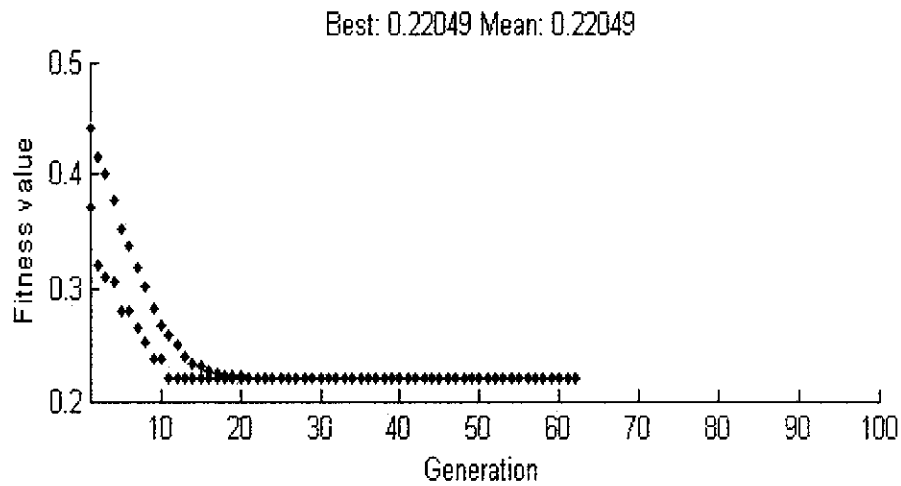

Figure 7.2 Best Fitness Graph

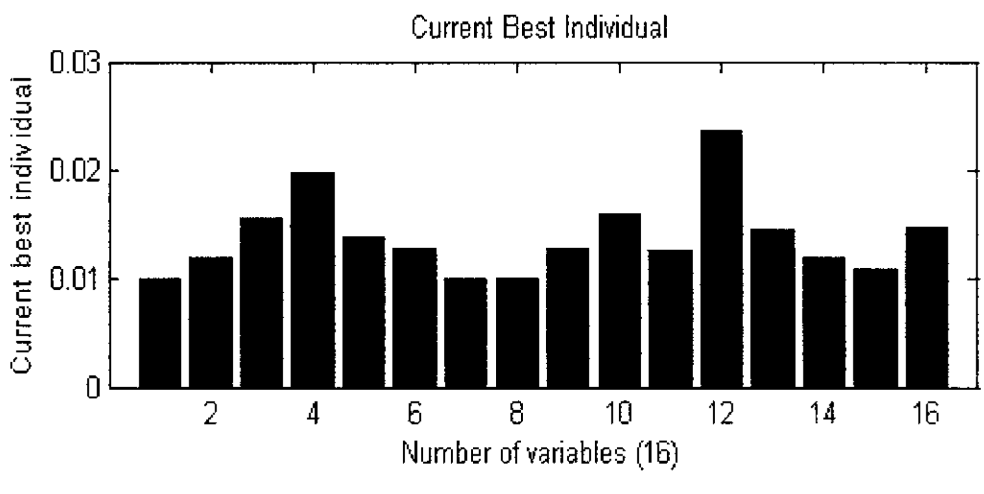

Figure 7.3 Best Individual Graph

Table 7.2 Best Individual Value from MATLAB

TOLERANCE	MATLAB
AT1	0.010
AT2	0.0119
AT3	0.0155
BT1	0.0197
BT2	0.0137
BT3	0.0127
BT4	0.010
BT5	0.010
BT6	0.0127
CT1	0.016
CT2	0.0125
CT3	0.0235
CT4	0.0145
DT1	0.0118
DT2	0.010
DT3	0.0146

Chapter 8

Particles Swarm Optimization

CHAPTER 8

PARTICLES SWARM OPTIMIZATION

8.1 BACKGROUND OF ARTIFICIAL LIFE

The term "Artificial Life" (Alive) is used to describe research into human-made systems that possess some of the essential properties of life. Alive includes two-folded research topic:

i. Alive studies how computational techniques can help when studying biological phenomena.
ii. ALife studies how biological techniques can help out with computational problems.

The focus of particles swarm optimization is on second Alife. Actually, there are already lots of computational techniques inspired by biological systems. For example, artificial neural network is a simplified model of human brain; genetic algorithm is inspired by the human evolution.

Here we discuss another type of biological system - social system, more specifically, the collective behaviors of simple individuals interacting with their environment and each other. Someone called it as swarm intelligence. There are two popular swarm inspired methods in computational intelligence areas: Ant colony optimization (ACO) and particle swarm optimization (PSO). ACO was inspired by the behaviors of ants and has many successful applications in discrete optimization problems.

8.2 PARTICLES SWARM OPTIMIZATION TECHNIQUE

The particle swarm concept originated as a simulation of simplified social system. The original intent was to graphically simulate the choreography of bird of a bird block or fish school. However, it was found that particle swarm model can be used as an optimizer.

As stated before, PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food.

PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions. An inertia weight factor that dynamically adjusted the velocity over time, gradually focusing the PSO into a local search, the particle updates its velocity and positions with following equation (8.1) and (8.2).

$$v[] = \omega \times v[] + c_1 rand(\ ) \times (pbest[] - present[]) + c_2 \times rand(\ ) \times (gbest[] - present[]) \quad \text{---- (8.1)}$$

$$present[] = present[] + v[] \quad \text{---- (8.2)}$$

Where,

v [ ] = Particle velocity

Present [ ] = Current particle (solution)

pbest [ ] = Best solution among the each particle

gbest [ ] = Best among defined as stated before.

rand ( ) = Random numbers between (0,1)

ω = Inertia Weights. Usually 0.8 or 0.9

C1, C2 are learning factors. Usually C1 = C2 = 2.

The nominal values of different component's Ball Screw Assembly.

Screw X1 = 11.836 inches

Coupling1 X2 = 1.0204 inches

Coupling2 X3 = 0.6938 inches

Tolerance boundary values of different component's Ball Screw Assembly

Screw X1 = 1 to 200 (in 10^{-4} inches)

Coupling1 X2 = 1 to 60 (in 10^{-4} inches)

Coupling2 X3 = 1 to 60 (in 10^{-4} inches)

8.3 OBJECTIVE FUNCTION

Combined objective function of minimizing the manufacturing cost and the cost associated with the quality loss function are considered in this work.

Manufacturing cost for single side tolerance values for

$$\text{Screw} \quad M(t_1) = 1 + \frac{0.1258}{t_1^{0.4653}} \qquad \text{-------(8.2)}$$

$$\text{Coupling1} \quad M(t_2) = 1 + \frac{0.1181}{t_2^{0.4383}} \qquad \text{-------(8.3)}$$

$$\text{Coupling2} \quad M(t_3) = 1 + \frac{0.1181}{t_3^{0.4383}} \qquad \text{-------(8.4)}$$

Total manufacturing cost $M(t_i) = M(t_1) + M(t_2) + M(t_3)$-------(8.5)

Cost associated with quality loss function

$$Q(t_i)=\sum_{k=1}^{k}\alpha_k^2\frac{A}{T_k^2}$$ -------(8.6)

where,

A is Quality loss coefficient.

T_k is the single side functional tolerance stackup limit for dimensional chain k

α_k is standard deviation of dimensional chain k

K is total no of dimensional chain

k is dimensional chain index

From the above equations the combined objective function can be formulated

Minimize $Y(t_1) = \sum_{i=0}^{3}\left[M(t_i)+Q(t_i)\right]$ ------(8.7)

Minimize $Y(t_i)=$

$$3+\frac{0.1258}{t_1^{0.4653}}+\frac{0.1181}{t_2^{0.4383}}+\frac{0.1181}{t_3^{0.4383}}+A*(90.77029*t_1^2+362.811*t_2^2+90.77029*t_3^2)$$

------(8.8)

8.4 ALGORITHM OF PARTICLES SWARM OPTIMIZATION

Most of evolutionary techniques have the following procedure:

i. Random generation of an initial population
ii. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
iii. Reproduction of the population based on fitness values.
iv. If requirements are met, then stop. Otherwise go back to 2.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee success.

However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm.

8.5 PSO PARAMETERS CONTROL

From the previous pages case, we learned that there are two key steps when applying PSO to optimization problems: the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. It is not like GA, which needs to change to binary encoding, or special genetic operators have to be used. Then we can use the standard procedure to find the optimum. The searching is a repeat process, and the stop criteria are that the maximum iteration number is reached or the minimum error condition is satisfied.

There are not many parameter need to be tuned in PSO. Here is a list of the parameters and their typical values. The number of particles: the typical range is 20 - 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

Dimension of particles: It is determined by the problem to be optimized,

Range of particles: It is also determined by the problem to be optimized, you can specify different ranges for different dimension of particles.

Vmax: it determines the maximum change one particle can take during one iteration. Usually we set the range of the particle as the Vmax.

Learning factors: c1 and c2 usually equal to 2. However, other settings were also used in different papers. But usually c1 equals to c2 and ranges from [0, 4]

The stop condition: the maximum number of iterations the PSO execute and the minimum error requirement. For example, for ANN training in previous section, we can set the minimum error requirement is one mis-classified pattern. The maximum number of iterations is set to 2000. This stop condition depends on the problem to be optimized.

Global version vs. local version: we introduced two versions of PSO. Global and local version. Global version is faster but might converge to local optimum for some problems. Local version is a little bit slower but not easy to be trapped into local optimum. One can use global version to get quick result and use local version to refine the search.

8.6 CODING SYSTEM

In order to solve the problem, the program has been written by using the VC++. First step, by using the random function, the tolerance values for the each components of the Ball Screw Assembly can be initialized. That can be called as the initial population. Before that we have to define the number of particles and number of iterations. These initial values are used to calculate the optimal cost by using the combined objective function. The ball screw assembly has three components such as Screw, Coupling 1 and Coupling 2; hence three variables in the objective function are t1, t2 and t3 respectively. These three variables are initialized then calculate the objective function. . This can be done for all particles used in the program.

After that we have to find the particles best (pBest) and global best (gBest), by using these values present particles velocities are updated for the next iterations. Like that each iteration will be updated the velocity of each particle. In final iteration we get the optimal cost and its optimal tolerance.

Figure 8.1 gives the flow of the program in order to find the optimal tolerance and optimal cost. Output of this program will be compared to the genetic algorithm results; its differences are tabled. The graphs are plotted. By absorbing that graphs, it gives the how the cost can be converging and meet optimal cost and optimal tolerance for each components.

8.7 PARAMETERS USED

The number of particles = 10 to 20

The number of iterations = 100 to 500

Dimension of particles = 3

Range of each particle

Tolerance of screw <=0.020

Tolerance of coupling 1 <= 0.0060

Tolerance of coupling 2 <= 0.0060

Velocity of each particle

Velocity of screw <=0.020

Velocity of coupling 1 <=0.0060

Velocity of coupling 2 <= 0.0060

Learning factors

c1= 1.5, c2 = 1.5

Inertia weight factor (ω) = 0.9

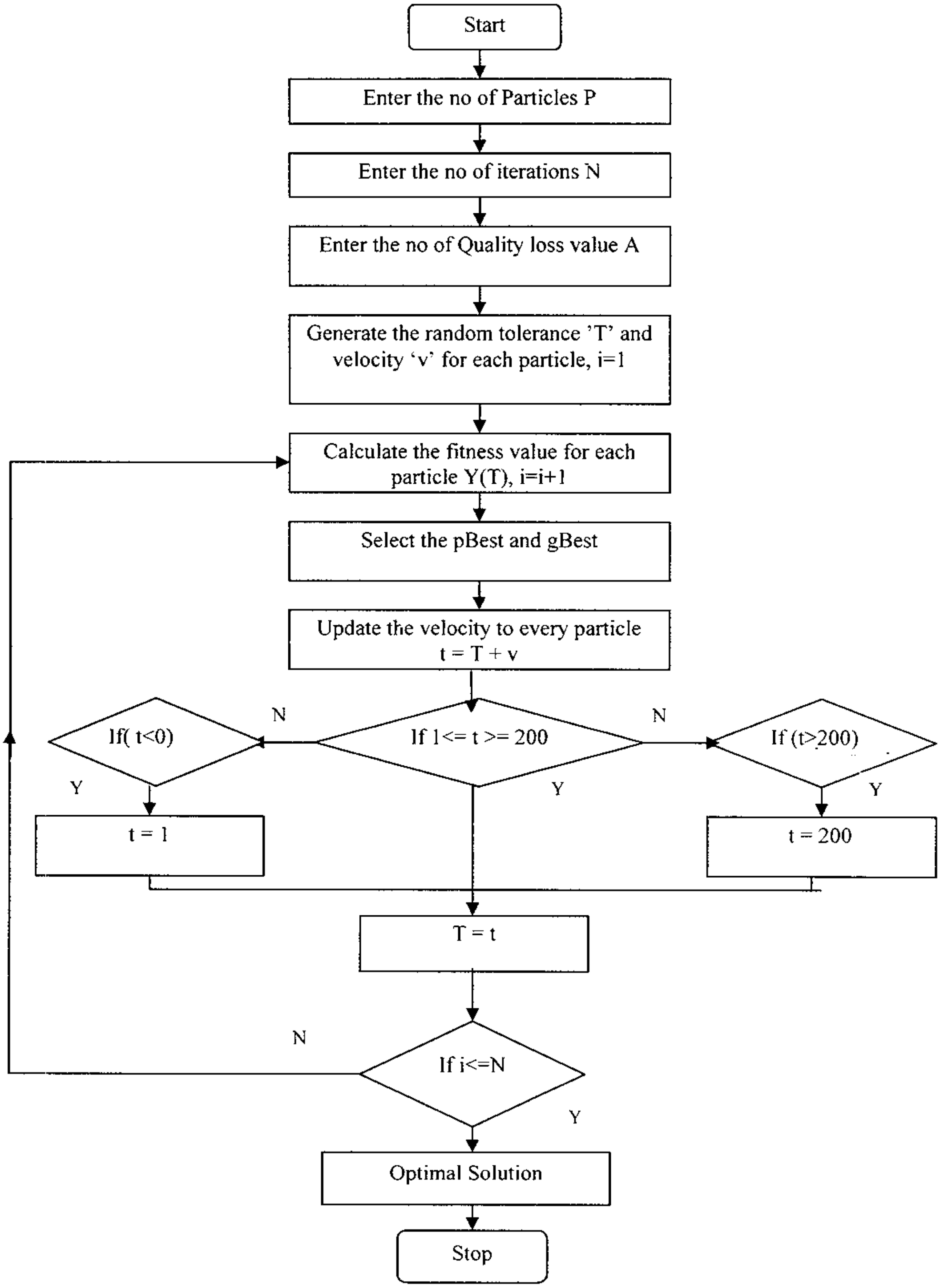

Figure 8.1 Flowcharts for Program

8.8 OPTIMUM SOLUTIONS FOR BALL SCREW ASSEMBLY FOR DIFFERENT QUALITY LOSS COEFFICIENT VALUES

The results clearly show that the PSO algorithm yields the optimal tolerance value for components of the ball screw assembly. The optimum tolerances are within the specified limits. The particle swarm optimization algorithm was run with a larger swarm size of 20, also used inertia weight factor of $\omega = 0.9$ and learning factors c1 = 1.5 and c2 = 1.5. The results of PSO are compared with those obtained with GA and are discussed in the succeeding section. Table10.1 display the optimal tolerance of individual components of the ball screw assembly and its cost for six different values of quality loss coefficient A (0, 1, 52, 100, 300 and 520). Also, the convergence of the solution is clearly portrayed in the graphs.

Table 8.1 Optimal tolerance and optimal Manufacturing Cost in Dollar

A	Tolerance by PSO in 10^{-4} inc			Y(T)
	T1	T2	T3	
0	200.0	60.00	60.0	9.501
1	200.0	60.00	60.0	9.553
52	77.2	39.4	60.0	11.789
100	59.2	30.1	53.2	11.923
300	37.9	19.2	33.9	12.567
520	30.3	15.3	27.1	13.189

The Figure 8.2 explains the cost convergence quality loss value of '0'. In this case, the optimal cost and optimal tolerance is obtained in the second iteration itself. When the quality loss values is '0'

The optimal tolerance for Screw = 200.00×10^{-4} inches

The optimal tolerance for Coupling 1= 60.00×10^{-4} inches

The optimal tolerance for Coupling 2 = 60.00×10^{-4} inches

The optimal manufacturing cost = 9.501 dollar

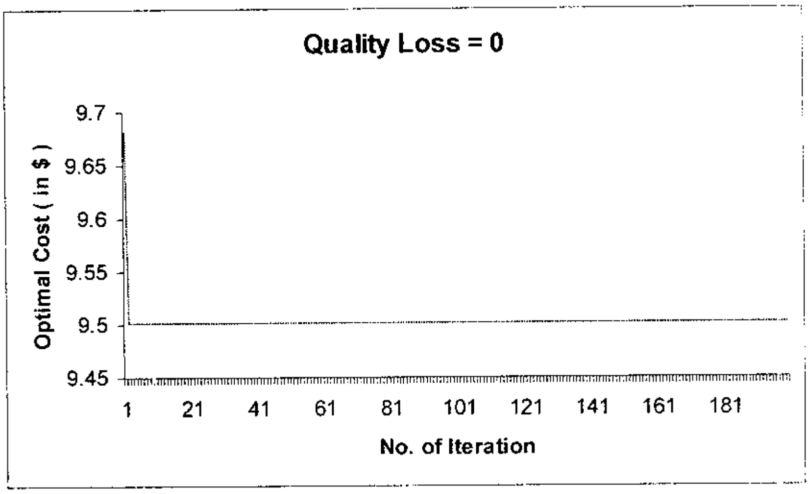

Figure 8.2 Graph for Quality Loss Coefficient '0'

The Figure 8.3 explains the cost convergence quality loss value of '1'. In this case, the optimal cost and optimal tolerance is obtained in the second iteration itself, with an increased cost than the previous one.

When the Quality loss value '1'

The optimal tolerance for Screw $= 200.00\times10^{-4}$ inches

The optimal tolerance for Coupling 1 = 60.00×10^{-4} inches

The optimal tolerance for Coupling 2 = 60.00×10^{-4} inches

The optimal manufacturing cost = 9.553 dollar

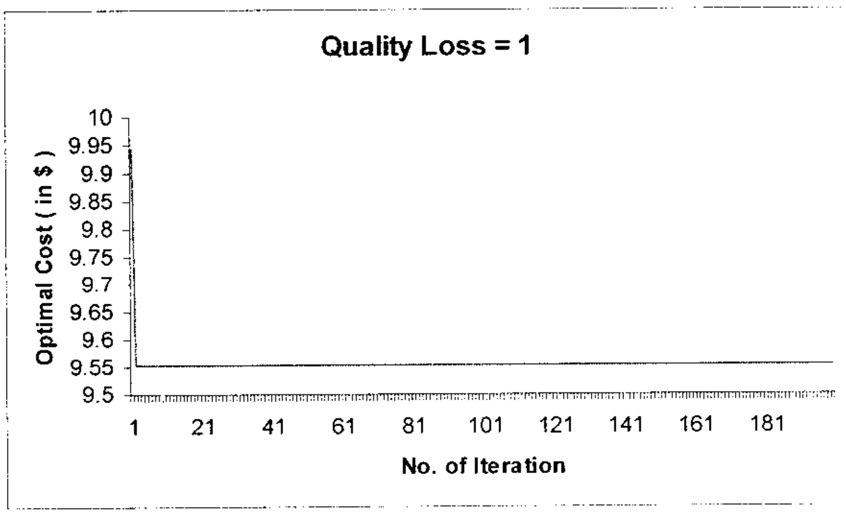

Figure 8.3 Graph for Quality Loss Coefficient '1 '

The Figure 8.4 explains the cost convergence quality loss value of '52'. In this case, the optimal cost and optimal tolerance is obtained in the fourth iteration. The optimal cost obtained in this case is more with reduced tolerance when compared with past two cases.

When the Quality loss value '52'

The optimal tolerance for Screw $= 77.2 \times 10^{-4}$ inches
The optimal tolerance for Coupling 1 $= 39.4 \times 10^{-4}$ inches
The optimal tolerance for Coupling 2 $= 60.0 \times 10^{-4}$ inches
The optimal manufacturing cost = 11.789 dollar

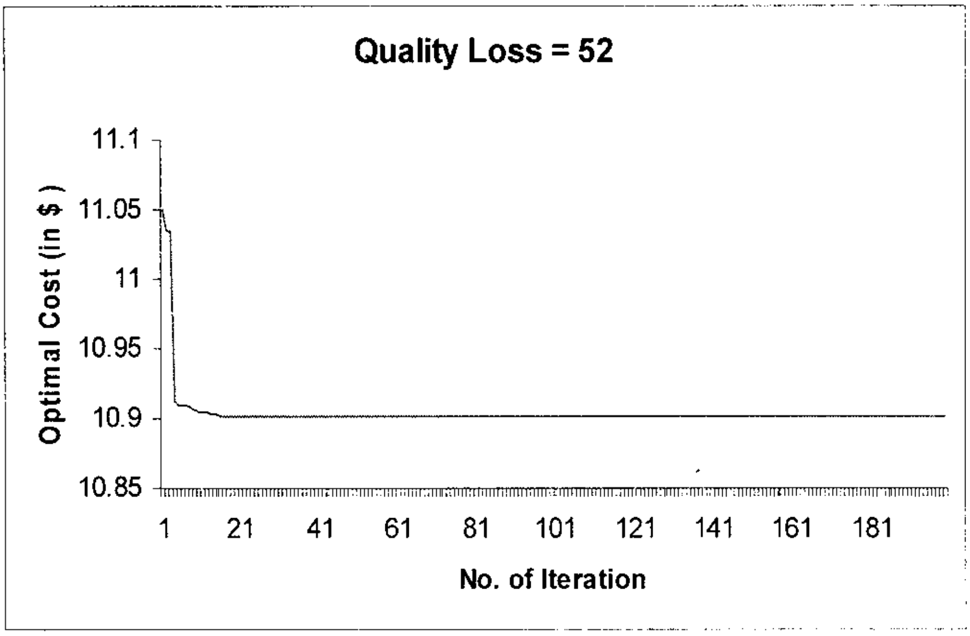

Figure 8.4 Graph for Quality Loss Coefficient '52'

The flowing Figure 8.5 shows optimal cost for the quality loss value of '100'. In this case the optimal cost and optimal tolerance is obtained at the seventh iteration. The tolerance is reduced and the cost is increased when compare the above three cases.

When the Quality loss value '100'

The optimal tolerance for Screw $= 59.2 \times 10^{-4}$ inches
The optimal tolerance for Coupling $1=30.10 \times 10^{-4}$ inches
The optimal tolerance for Coupling 2 $= 53.2 \times 10^{-4}$ inches
The optimal manufacturing cost = 11.923 dollar

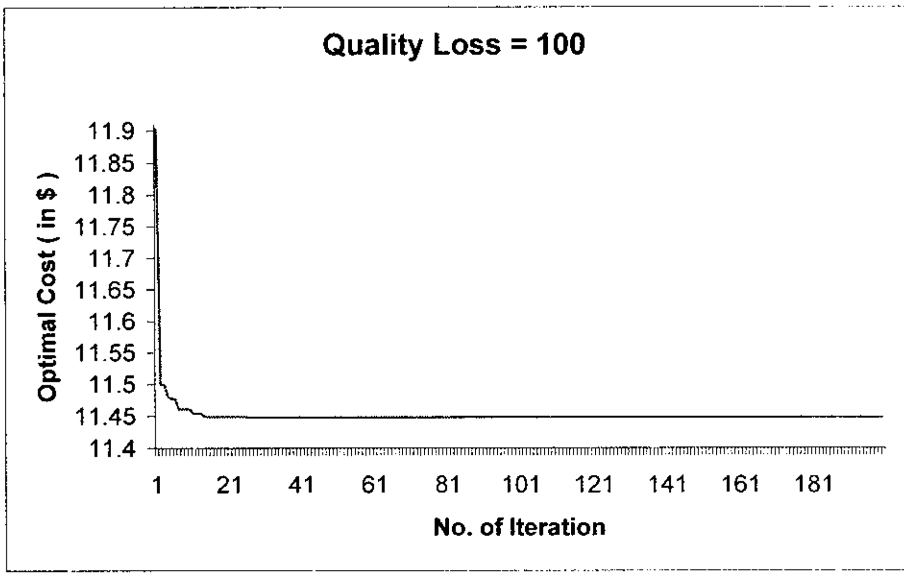

Figure 8.5 Graph for Quality Loss Coefficient '100 '

The flowing Figure 8.6 shows the case for quality loss value '300'. In this case also the optimal tolerance will be obtained at the 16^{th} iteration with an increased cost is increased when compared with the previous cases but tolerances for each component reduced.

When the Quality loss value '300'

The optimal tolerance for Screw = 37.9×10^{-4} inches

The optimal tolerance for Coupling 1 = 19.2×10^{-4} inches

The optimal tolerance for Coupling 2 = 33.9×10^{-4} inches

The optimal manufacturing cost = 12.567 dollar

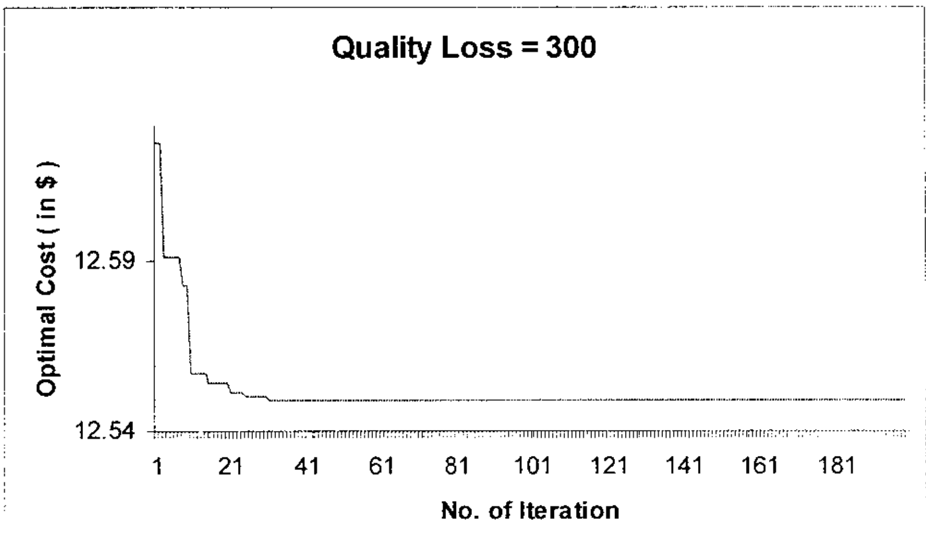

Figure 8.6 Graph for Quality Loss Coefficient '300'

The flowing Figure 8.7 shows the case for quality loss value '520'. In this case also the optimal tolerance is obtained at the 27th iteration with an increased cost and an reduced tolerance value.

When the Quality loss value '520'

The optimal tolerance for Screw = 30.3 $\times 10^{-4}$ inches

The optimal tolerance for Coupling 1 = 15.3 $\times 10^{-4}$ inches

The optimal tolerance for Coupling 2 = 27.1$\times 10^{-4}$ inches

The optimal manufacturing cost = 13.189 dollar

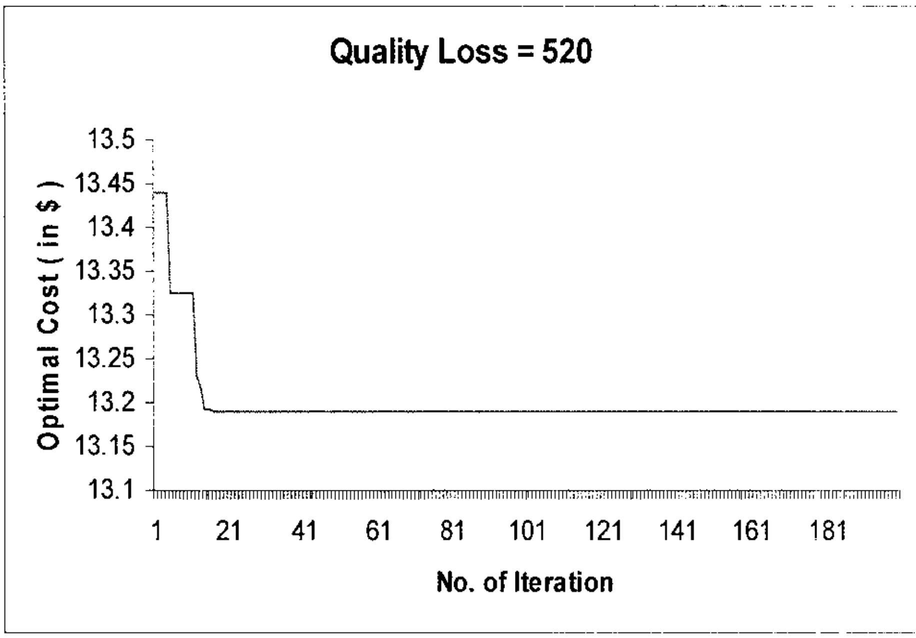

Figure 8.7 Graph for Quality Loss Coefficient '520 '

8.9 COMPARE PARTICLE SWARM OPTIMIZATION RESULTS WITH GENETIC ALGORITHM RESULTS

The results clearly show that the proposed PSO provides the optimal tolerance value, for various components of the ball screw assembly chosen, than that obtained with GA of previous work. The optimum tolerances obtained are within the specified limits.

Table 8.2 Comparison the PSO Results with GA Results

A	Tolerance by PSO in 10^{-4} inches			Cost in Dollars		Difference w.r.to PSO
	T1	T2	T3	Y_{ga}	Y_{pso}	GA
0	200.0	60.00	60.0	10.63	9.501	-1.129
1	200.0	60.00	60.0	10.731	9.553	-1.178
52	77.2	39.4	60.0	11.789	11.789	-0.887
100	59.2	30.1	53.2	11.923	11.923	-0.475
300	37.9	19.2	33.9	12.567	12.567	-0.018
520	30.3	15.3	27.1	13.189	13.189	0.000

The software is also run with a larger swarm size of 20, inertia weight factor of ω = 0.9 and learning factors c1 = 1.5 and c2 = 1.5 and the results are compared with those results of GA as shown in the Table 8.2 and also shows the comparisons of the results obtained by PSO (Ypso), Genetic algorithm results (Yga). This comparison clearly concludes that PSO technique yields the optimal tolerance value, for various components of the ball screw assembly, than that of Genetic algorithm (GA). But with an A = 520, PSO result exactly matches the results obtained with GA.

The Flowing Figure 8.8 illustrates the comparisons of the results obtained by PSO, Genetic algorithm (GA) results. This comparison clearly concludes that PSO technique yields the optimal tolerance value, for various Quality loss values 0, 1, 52, 100 and 300(A), than that of GA. But with an A = 520, PSO result exactly matches the results obtained with GA.

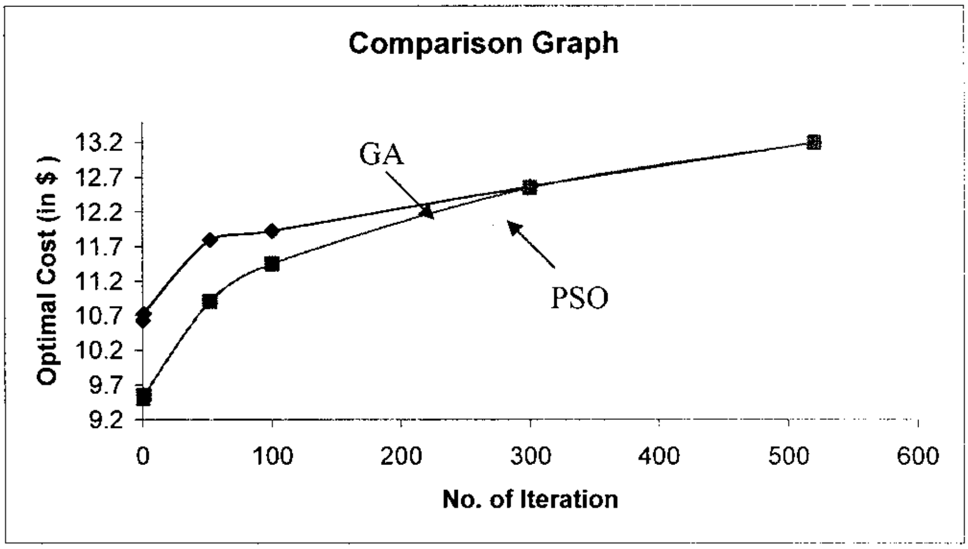

Figure 8.8 Comparison graphs

Chapter 9

Conclusion

CHAPTER 9

CONCLUSIONS

The following are some of the conclusions made regarding the work that have been carried out accordingly.

- Tolerance chart is a very useful tool for contCouplingl ing work piece dimension during manufacturing, and also allows all blueprint tolerance to be determined while ensuring that all the assembly requirements are satisfied.

- The optimal tolerance allocation using Genetic Algorithm makes it possible to achieve the global optimal tolerances, which matches the assembly limit and reduces the number of rejects and cost of production.

- Affirmed that the number of parent's combination is to be kept at the maximum in Genetic Algorithm.

- The Particle Swarm Optimization provides less manufacturing cost when compared to that of Genetic Algorithm.

- The optimal tolerance can be found out by means of manufacturing cost.

References

REFERENCES

1. Bryan Ngoi Kok Ann and Michael Scow Mgaw Seng (1996), "*Tolerance Synthesis Adopting a Nonlinear Programming Approach*", Int J Adv Manuf Technol, vol. 11: pp 387-393

2. Bryan R. Fischer, "*Mechanical Tolerance Stack up and Analysis*", Marcel Dekker Inc, New York.

3. Chou. C. Y and Chang. C. L (2001), "*Minimum-Loss Assembly Tolerance Allocation by Considering Product Degradation and Time Value of Money* ", Int J Adv Manuf Technol, vol. 17: pp 139–146.

4. Dimitrellou. S. Ch & Diplaris. S. C & Sfantsikopoulos. M. M (2006), "*Cost-competent Tolerancing in CAD*", Int J Adv Manuf Technol.

5. Huang. M, Gao. Y, Xu. Z and Li. Z (2002), "*Composite Planar Tolerance Allocation with Dimensional and Geometric Specifications*", Int J Adv Manuf Technol, vol. 20: pp 341–347.

6. Ibrahim Zeid (2003), "*CAD/CAM Theory and Practice*", Tata McGraw-Hill Edition, New Delhi.

7. Ji. P and Xue. J. B (2002), "*Process Tolerance Control in a 2D Angular Tolerance Chart*", Int J Adv Manuf Technol, vol. 20: pp 649–654.

8. Kenneth W. Chase, William H. Greenwood, Bruce G. Loosli, Loren F. Hauglund (1990), "*Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection*", Manufacturing Review, vol 3, no 1, pp 49-59.

9. Kenneth W.Chase (1999), "*Tolerance Allocation Methods for Designers*", ADCATS Report No. 99-6.

10. Kun-Lin Hsieh (2006), "*The study of cost-tolerance model by incorporating process capability index into product lifecycle cost*", Int J Adv Manuf Technol, vol 28: pp 638–642.

11. Meifa Huang & Yanru Zhong (2006), "*Dimensional and geometrical tolerance balancing in concurrent design*", Int J Adv Manuf Technol

12. Ngoi. B. K. A and Ong Jon Min (1999), "*Optimum Tolerance Allocation in Assembly*", Int J Adv Manuf Technol, vol. 15: pp 660–665

13. Prabhaharan. G, Asokan. P, Rajendran. S (2005), "*Sensitivity-based conceptual design and tolerance allocation using the continuous ants colony algorithm (CACO)* ", Int J Adv Manuf Technol, vol. 25: pp 516–526.

14. Prabhaharan. G, Asokan. P, Ramesh. P, Rajendran. S (2004), "*Genetic-algorithm-based optimal tolerance allocation using a least-cost model*", Int J Adv Manuf Technol, vol. 24: pp 647–660.

15. Shan and Roth. R. N (2003), "*Genetic Algorithms in Statistical Tolerancing*", Mathematical and Computer Modeling, vol. 38: pp 1427-1436

16. Siva Kumar. M, Kannan. SM (2007), "*Optimum manufacturing tolerance to selective assembly technique for different assembly specifications by using Genetic algorithm*", Int J Adv Manuf Technol, vol 32: pp 591-598.

17. Xiao-Feng Xie, Wen-Jun Zhang and Zhi-Lian Yang (2002), "*A Dissipative Particle Swarm Optimization*", Institute of Microelectronics, Tsinghua University, IEEE.

18. Ye. B, Salustri. F. A (2003), "*Simultaneous tolerance synthesis for manufacturing and quality*", Research in Engineering Design, vol. 14: pp 98–106.