UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

MODELING OF THE MANUFACTURING AND ASSEMBLY COSTS FOR TOLERANCE DETERMINATION IN SELECTIVE ASSEMBLY

A Dissertation

SUBMITTED TO THE GRADUATE FACULTY

In partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

By

JIYOON LEE

Norman, Oklahoma
2002

UMI Number: 3073706

UMI®

UMI Microform 3073706

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

MODELING OF THE MANUFACTURING AND
ASSEMBLY COSTS FOR TOLERANCE DETERMINATION
IN SELECTIVE ASSEMBLY

A Dissertation APPROVED FOR THE
SCHOOL OF INDUSTRIAL ENGINEERING

BY

Dr. Shivakumar Raman, Chair

Dr. P. Simin Pulat

Dr. Theodore B. Trafalis

Dr. Thomas L. Landers

Dr. S. R. Gollahalli

ACKNOWLEDGEMENTS

First, I thank my God through Jesus Christ for completing my dissertation. I glorify Him who has been a leader and friend whenever I had hard times.

I am indebted to many people for the dissertation, who supported and guided me up to now. My sincere and deepest respect goes to my parents; father and mother, and father-in-law and mother-in-law. They believed me and showed the endless love during my study.

I express a deep appreciation from my heart to my advisor, Dr. Raman who supported me and taught good lessons that is useful through my life. In addition, my committee members, Dr. Pulat, Dr. Landers, Dr. Trafalis, and Dr. Gollahalli, deserve to accept my respects and appreciations.

My heart with love goes to my wife, Hyunsook, who encouraged me consistently, stood by me on obstacles, and endured long hard years. I love you with my whole heart! My children, Elliot Kokang and Ashley Khang-Eun, made me strong and happy with their love and endurance during those years. Without their love, I could not go through the tough times.

I would like to thank to all members of Norman Korean Baptist Church who prayed for me who were the most powerful supporting group. I also thanks to the colleagues who are in the School of Industrial Engineering. Especially, I thank to Dr. Chae-Bogk Kim who expressed a lot of concerns and love.

Now, it is the time to go forward with gripping the words, "Have I not commanded you? Be strong and of good courage; do not be afraid, nor be dismayed, for the Lord you God is with you wherever you go." (Joshua 1:9)

Table of Contents

List of Tables

List of Figures

Abstract

The manufacturing cost-tolerance optimization problem for tolerance determination is quite extensively published. However, the total manufacturing time to assembly optimization problem is not studied very extensively. Especially, the combined optimization of these apparently contradicting trends has not been studied very much in literature.

In this dissertation, the contradicting trends are modeled as a series of approximate bi-criteria optimization problems. Each of these problems is solved by developed heuristic methods that generate efficient solutions. These models are verified on several generated data sets.

The results serve as a first proof-of-concept for the combined tolerance-cost-time problem. This model can be used for process selection during process planning. Most importantly, this approach provides a mathematical basis to selective assembly.

CHAPTER 1

Introduction

1.1 Background

Important goals for the manufacturing industry include economic and timely delivery of products. Product tolerances are quite important in achieving these goals. Tolerances are critical in both product and process design. Tolerance assignment presents a significant impact on manufacturing cost, delivery time, and product quality [Zhang and Wang, 1993]. In other words, the goal for better, cheaper, and higher quality performance can be realized by analyzing and developing relationships between cost, tolerance, and time for manufacturing a part or an assembly.

Tolerance analysis and synthesis have been the focus of much research for the last three decades. The aim is to control production variation and to produce high quality parts which, in turn, increase competitiveness and market share. In tolerance analysis, assembly tolerances are usually modeled to determine critical assembly allowances. In tolerance synthesis, assembly tolerances are optimized with respect to manufacturing cost. The relationship between tolerance and cost is assumed or derived for use in subsequent automation [Chase et. al., 1990; Spotts, 1973; Peters, 1970; Sutherland and Roth, 1975; Wilde and Prentice, 1975; Wu et. al., 1988].

Systematic methodologies are needed for modeling manufacturing processes to evaluate processing alternatives for given tolerances. One recognized need central to the concept of design for manufacturability is how to provide this information to obtain more robust and cost-effective designs [ASME CRDT-15, 1990].

1.2 Context, Contribution, and Objective

Selective Assembly (SA) is a standard procedure for loosening the tolerances and conducting selective sorting. This way the original assembly tolerance can still be met. The major assumption is that the cost of sorting and assembly is not higher than the cost to manufacture tight tolerances. This procedure is largely qualitative and lacks a mathematical basis for comparing the losses in tolerance optimization. While manufacturing costs can be reduced by maintaining looser tolerances, the assembly cost could actually increase. The total manufacturing time to assembly comprises of manufacturing time and the time it takes to put together the assembly. The high cost of tighter tolerances could be tied to the high initial and running costs of expensive machinery and instrumentation. Looser tolerances can result in more time for assembly. Further, it could take longer time to manufacture using less expensive machines. The fundamental contribution of this research is in the development of a mathematical basis for considering the manufacturing cost as well as the total manufacturing time to assembly costs while selecting tolerances.

Accordingly, the objective of this research is to explain and analyze the mechanism of the relationship between cost, tolerance, and time, from a manufacturing perspective for selective assembly.

We use the widely reported, monotonically decreasing (exponential) relationship between cost and tolerances and an assumed increasing function between tolerances and time. These apparently different trends motivate us to study this problem as a bi-criteria optimization in principle, although approximated to solve the applied problem at hand. In other words, rather than solve a true multi-criteria model (response surface model), three individual approximate bi-criteria cases, most relevant to selective assembly and tolerance determination in manufacturing are solved. Numerical examples are developed to show the feasibility of the solutions. These tolerances are expected to help determine a suitable process sequence.

Chapter 2

Literature Review

2.1 Introduction

In mechanical design and manufacture, the tolerance assignment has an important effect in selecting manufacturing processes, in reducing manufacturing cost and in increasing quality and productivity. The introduction of the concept of interchangeability has made the design and manufacturing researchers keep in touch with the tolerance assignment since the seventies [Evans, 1974]. Until that time, many products were manufactured as individual entities with each part being fitted to its mated part.

The need for tolerance assignment arises from the '***interchangeability***' of parts that are '***sufficiently identical***'. But it would be impossible to make components identical by considering technological capabilities of manufacturing and measuring equipment. Therefore, an acceptable range about the nominal size called the tolerance is specified on parts.

Tolerance is defined as the maximum deviation from a nominal specification within which the component is still acceptable for its intended purpose [Wu et. al., 1988]. The assignment of tolerance makes us recognize the expected variations between components produced using manufacturing processes. In the traditional manufacturing fields, the design engineers or manufacturer have used their experiences, standard information, and handbooks to assign tolerances to the components and assemblies. Such tolerance assignments do not propose complete models. As a result, incorrect tolerance assignment caused lower quality of products at high costs of production.

Tolerance assignment should be sufficiently tight to guarantee that a component or an assembly will perform as intended from a statistical basis. Also, it should be noted that cost has to be minimized as low as possible within the operating range of the assembly. The design department tries to assign tight tolerances to the components and assemblies so as to guarantee the intended high quality and proper operation of the product. On the contrary, the manufacturing department is more concerned with loosening tolerances for lowering the cost. The conflict of concerns between these two departments must be resolved in a mutually agreeable fashion.

Various dimensional design tolerance models have been presented in the literature and each model has its own advantages and limitations. For a proper selection of a design tolerance method for intended manufacture, it is very useful to know each model's pros and cons and to compare the tolerance analysis and allocation methods. Wu et. al. [1988] proposed a good guideline for this, and hence the experimental procedure and results accomplished by them are presented to provide for the understanding of tolerance methodology.

2.2 Dimensional Tolerance Analysis Models

Tolerance analysis models are designed for calculation of the assembly tolerance from component tolerances. It should predict assembly tolerance close to actual assembly tolerance limits for minimizing predicted rejects or scrap. Also, different types of distributions of the component tolerances should be considered, because they play a significant role in the assembly dimensional tolerance. Four types of distributions of component tolerance are used in analyzing the tolerance analysis models that are shown in Fig. 2.1. These are: uniform, truncated normal, Weibull, and normal distributions. The uniform and truncated normal distributions are symmetric and the Weibull distribution is skewed. The characteristics and parameters of distributions used in the Wu et. al.'s [1988] paper are given as follows:

Uniform Distribution

A random variable *Y* has a uniform density, and is referred to as a continuous uniform random variable, if and only if its probability density function, *f(y)*, of a component dimensional tolerance is given by [Freund and Walpole, 1987]:

$$f(y) = \frac{1}{\beta - \alpha} = \frac{1}{\gamma} \quad for \quad \alpha < y < \beta \tag{1}$$

The parameters α and β are real constants with $\alpha < \beta$ and r means the tolerance range (β - α). In the uniform distribution, the mean μ and the variance σ^2 are expressed as:

$$\begin{aligned} \mu &= \frac{\alpha + \beta}{2} \\ \sigma^2 &= \frac{1}{12}(\beta - \alpha)^2 = \frac{\gamma^2}{12} \end{aligned} \tag{2}$$

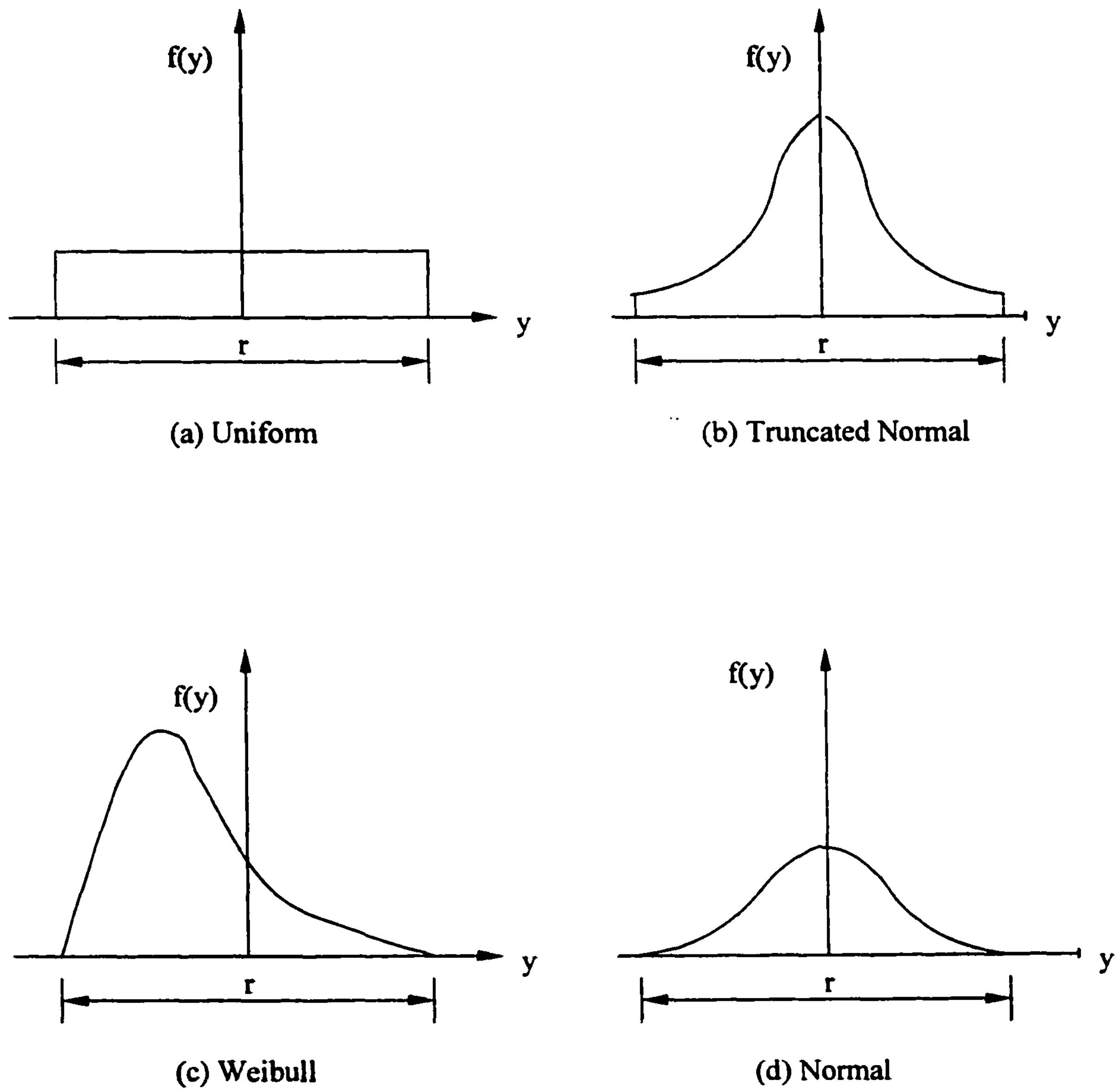


Fig. 2.1 Probability Density Function f(y) for Dimension y

They assumed that the mean value is on the x-axis, and is given by $\mu = 0$. The deviation multiplier z is the range of tolerance deviated by the standard deviation and is expressed as:

$$z = \frac{\gamma}{\sigma} = 2\sqrt{3} \tag{3}$$

Truncated Normal Distribution

The probability density function of a component dimensional tolerance with a truncated normal distribution [Wu et. al., 1988] is represented by:

$$f(y) = \frac{1.0478}{\sqrt{2\pi}\sigma_0}\exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_0}\right)^2\right] \quad for \quad \alpha < y < \beta \tag{4}$$

Also, the mean μ has the value of zero and the variance σ^2 has

$$\begin{aligned}\sigma^2 &= \int_{-2\sigma_0}^{2\sigma_0}\left\{\frac{1.0478}{\sqrt{2\pi}\sigma_0}\exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_0}\right)^2\right]y^2\right\}dy \\ &= \sigma_0^2\left[1-\left(\frac{4\sigma_0}{2\Phi(2\sigma_0)}-1\right)\phi(2\sigma_0)\right]\end{aligned} \tag{5}$$

where $\sigma_0 = r/4$ = standard deviation of the original normal distribution

$z = 4.547$.

Weibull Distribution

The Weibull distribution is defined as:

$$f(y) = \frac{\beta}{\delta}\left(\frac{y\gamma}{\delta}\right)^{\beta-1}\exp\left[-\left(\frac{y-\gamma}{\delta}\right)^{\beta}\right] \quad y \geq \gamma$$

where γ is the location parameter ($-\infty < \gamma < \infty$), $\delta>0$ is the scale parameter, and $\beta>0$ is the shape parameter. The mean and the variance of the Weibull distribution are:

$$\mu = \gamma + \delta\Gamma\left(1 + \frac{1}{\beta}\right)$$
$$\sigma^2 = \delta^2\left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma\left(1 + \frac{1}{\beta}\right)^2\right]$$

The values of each parameter are given with respect to $\gamma = -(r/2)$, $\delta = r/3$, and $\beta = 2$. By inserting those values of parameters, Weibull probability density function is given by:

$$f(y) = \frac{6}{r}\left(\frac{3y}{r} + \frac{3}{2}\right)\exp\left[-\left(\frac{3y}{r} + \frac{3}{2}\right)^2\right] \tag{6}$$

Also, the variance and the mean are represented by the following equations:

$$\begin{aligned} \sigma^2 &= \left(\frac{r}{3}\right)^2 - \left[\Gamma(2) - \Gamma(1.5)^2\right] = 0.02384r^2 \qquad (\sigma = 0.1544r) \\ \mu &= -\frac{r}{2} + \frac{r}{3}\Gamma\left(1 + \frac{1}{2}\right) = -0.020459r \end{aligned} \tag{7}$$

Then the deviation multiplier z becomes by the definition as:

$$z = \frac{r}{\sigma} = \frac{r}{0.1544r} = 6.4760 \tag{8}$$

Normal distribution

The probability density function of the normal dimensional tolerance distribution is given by:

$$f(y) = \frac{1}{2\pi}\exp\left[-\frac{1}{2}\left(\frac{y}{\sigma}\right)^2\right] \tag{9}$$

And the mean, standard deviation, and deviation multiplier are also given by:

$$\mu = 0 \quad \sigma = \frac{r}{6} \quad z = 6 \tag{10}$$

Eight tolerance analysis models have been discussed with comments as follows; Worst-case model, Statistical model, Spotts' modified model, Modified statistical model, Mean shift model, Monte-Carlo model, Moment model, Hybrid model.

(1) Worst-case Model

For the worst-case model [Peters, 1970, Chase and Greenwood, 1988], the assembly tolerance is the sum of each component's tolerance and each component dimension has its maximum or minimum limit so that the assembly can get the worst possible limits. The model is represented by:

$$T_A = \sum_{i=1}^{n} t_i \tag{11}$$

where T_A is an assembly tolerance, t_i is the i^{th} component tolerance, and n is the number of components.

The worst-case model is simple and leads to the 100% probability of satisfaction on the specified assembly tolerance, but it has the disadvantage of large value of resulting assembly tolerance. Also, the individual allocated component tolerances tend to be very tight for a given assembly tolerance.

(2) Statistical Model

The statistical model [Peters, 1970, Brooks, 1961, Chase and Greenwood, 1988] expresses the root sum squared (RSS) of the component tolerances as follows:

$$T_A = Z\left[\sum_{i=1}^{n}\left(\frac{t_i}{z_i}\right)^2\right]^{0.5} \tag{12}$$

where

Z = assembly deviation multiplier (6 for normal distribution)

z_i = deviation multiplier for the i^{th} component tolerance

= 3.4641 for uniform distribution

= 4.547 for truncated normal distribution

= 6.4858 for Weibull distribution

= 6 for normal distribution.

For the case of normal distribution with ±3σ range, the statistical model has a simple form with $z_i = 6$ and $Z = 6$ as follows:

$$T_A = \left[\sum_{i=1}^{n}(t_i)^2\right]^{0.5}$$

Brooks [1961] demonstrated that the benefits of this model were "higher quality design through better fits and clearances, lower manufacturing cost through wider part tolerances, and less scrap and rework through use of process controls". Especially when the number of components is large, the statistical model has the smallest assembly tolerance with respect to other models.

(3) Spotts' Modified Model

Spotts [1978] presented the model with the average between the arithmetic and the normal laws as follows:

$$T_A = 0.5\left[\left(\sum_{i=1}^{n} t_i\right) + \left(\sum_{i=1}^{n} t_i^2\right)^{0.5}\right] \tag{13}$$

This model uses both the worst-case model and statistical model.

(4) Modified Statistical Model

The statistical model is generalized as [Chase and Greenwood, 1988] :

$$T_A = CZ\left[\sum_{i=1}^{n}\left(\frac{t_i}{z_i}\right)^2\right]^{0.5} \tag{14}$$

where

C = correction factor to account for any non-ideal conditions

typical value 1.4 or 1.5 [Chase and Greenwood, 1988],

Z = deviation multiplier of the assembly tolerance, and

z_i = deviation multiplier of the i^{th} component tolerance.

"By using this model the assembly tolerance usually decreases, but it is sometimes larger than that of the worst-case model when a distribution with small z_i is used, unequalized tolerance chains are employed, or the number of component tolerances in the assembly is small" [Wu et. al., 1988].

(5) Mean Shift Model

Chase and Greenwood [1988] also proposed a different statistical model such as

$$T_A = \sum_{i=1}^{n} m_i t_i + \left\{ \sum_{i=1}^{n} \left[(1 - m_i) t_i \right]^2 \right\}^{0.5} \tag{15}$$

where

m_i = possible range of mean shift of the i^{th} component tolerance expressed as a fraction of its range.

The mean shift factor was defined as "a fraction of the specified tolerance range for the part dimension" [Chase and Greenwood, 1988]. For the normal, truncated normal, and uniform distributions, the value of m_i takes zero and then the mean shift model becomes the simple statistical model. The value of 0.4092 is assigned for the Weibull distribution. Factors ranging between 0 and 0.8 have been suggested. It is not appropriate to adapt this model unless detailed data for the distributions is fully known, since an accurate value of m_i is difficult to obtain at the early design stages.

(6) Monte-Carlo Model

By Monte-Carlo simulation, the upper and lower limits, X_{max} and X_{min}, respectively, of the assembly tolerance range are determined and expressed as:

$$T_A = \left(X_{max} - X_{min} \right) \tag{16}$$

For the Monte-Carlo simulation, 99.74% of the simulated assemblies fall in this tolerance range and it corresponds to the $\pm 3\sigma$ range of the normal distribution.

This model is appropriate for a skewed distribution, but not for the uniform distribution. The most significant advantage of this model is the reduction of the predicted rejection for all kinds of nonnormal distributions [Wu et. al., 1988]. Monte-Carlo simulation is a very useful method to model complex situations such as tolerance analysis in actual assembly operations where the product and process accuracy should be considered simultaneously [Redford et. al., 1981].

(7) The Moment Model

The moment model used by Chase and Greenwood [1988] is considered as a simple form which has only the first two moments, and assembly tolerance is assumed to be distributed normally. The moment model has an assembly tolerance expressed as:

$$T_A = \left(X_{\max} - X_{\min}\right) \tag{17}$$

where

$X_{max} = M_A + 3D_A,$

$X_{min} = M_A - 3D_A,$

$M_A = \Sigma m_i,$ and

$D_A = (\Sigma\sigma_i^2)^{1/2}.$

The type of component tolerance distribution takes an important role in the reduction of the assembly tolerance. For instance, while using the Weibull distribution, the assembly tolerance evaluated by this model has a smaller value with respect to the statistical model

(8) Hybrid Model

The hybrid model has the same assembly tolerance model as with the moment model whereby:

$$
\begin{aligned}
T_A &= X_{\max} - X_{\min}, \\
X_{\max} &= M_A + 3D_A, and \\
X_{\min} &= M_A - 3D_A.
\end{aligned} \tag{18}
$$

but the assembly mean tolerance and standard deviation are different with it because Monte-Carlo simulation is first used to create 1000 sample assembly tolerance values x_i. They will be used to calculate the moments of the assembly tolerance distribution directly. Thus,

$$
\begin{aligned}
&M_A = 0.001\sum_{i=1}^{n} x_i, \\
&and \quad D_A = \left(0.001\sum_{i=1}^{n} x_i^2 - M_A^{\ 2}\right)^{\frac{1}{2}}.
\end{aligned} \tag{19}
$$

The advantage of this model is a shorter run time than the Monte-Carlo model because smaller sample size (1000). But the hybrid model predicts larger rejects than the Monte-Carlo model. Published techniques are summarized and presented in Table 2.1 [Kumar and Raman, 1992].

Table 2.1. Summary of tolerance analysis techniques [Kumar and Raman, 1992]

No.	Model	Expression for Assembly tolerance	Notes
1	Worst-case model	$T_A = \sum_{i=1}^{n} t_i$	t_i = Component tolerance
2	Statistical model	$T_A = Z\left[\sum_{i=1}^{n}\left(\frac{t_i}{z_i}\right)^2\right]^{0.5}$	T_A = Assembly tolerance
3	Spotts' modified model	$T_A = 0.5\left[\left(\sum_{i=1}^{n} t_i\right) + \left(\sum_{i=1}^{n} t_i^2\right)^{0.5}\right]$	n = Number of components
4	Modified statistical model	$T_A = CZ\left[\sum_{i=1}^{n}\left(\frac{t_i}{z_i}\right)^2\right]^{0.5}$	m_i = Mean shift of the i^{th} component
5	Mean shift model	$T_A = \sum_{i=1}^{n} m_i t_i + \left\{\sum_{i=1}^{n}\left[(1-m_i)t_i\right]^2\right\}^{0.5}$	
6	Monte Carlo model	$T_A = (X_{max} - X_{min})$	$X_{max} = M_A + 3D_A$ $X_{min} = M_A - 3D_A$
7	Moment model	$T_A = (X_{max} - X_{min})$	M_A = Mean of assembly tolerance D_A = Standard Deviation of assembly tolerance
8	Hybrid model	Combination of 6 and 7 above	

2.3 Cost-Tolerance Functions

After the assembly tolerance model is chosen, and set within the required specification, an assembly cost function should be minimized to optimize component tolerances. Total assembly cost function is composed of two parts which depend on the tolerance values: a fixed part and a variable manufacturing part. Various mathematical functions have been presented to fit manufacturing cost-tolerance field data.

Wu et. al. [1988] examined the most frequently used Cost-Tolerance functions to compare their errors to the actual field data: Sutherland function, reciprocal square function, reciprocal function, exponential function, and Michael-Siddall function.

The relation between cost-tolerance function and sum of error square which utilizes the nonlinear least square method to fit each function to the field data and estimate its parameters is given by:

$$e = \sum \left[f_1 - f(a, t_i) \right]^2 , \tag{20}$$

where e = sum of the error square,

f = cost-tolerance function,

a = a vector of parameters to be estimated for each C-T function,

f_i = discrete value of cost of the i^{th} component, and

t_i = tolerance of the i^{th} component.

The value of e should be minimized as an index of good fit. The following are cost-tolerance functions proposed by various researchers.

First, Sutherland and Roth [1975] presented their cost-tolerance function at the n cost-tolerance pairs as follows:

$$f = bt^{-a} \tag{21}$$

where the parameters a and b are given by

$$a = \frac{\sum(\ln f_i)\sum(\ln t_i) - n\sum[(\ln f_i)(\ln t_i)]}{n\sum(\ln t_i)^2 - [\sum(\ln t_i)]^2}$$
$$b = \exp\left[\frac{\sum(\ln f_i) - a\sum(\ln t_i)}{n}\right]$$
$$n = \textit{number of } \cos t - \textit{tolerance pairs}$$

This function fits the field data well, but shows large fitting errors in the case of tight tolerances.

Spotts [1973] used the reciprocal square function as a cost-tolerance function such as:

$$f = \frac{a}{t^2} \tag{22}$$

where

$$a = \frac{\sum(f_i / t_i^2)}{\sum(1 / t_i^4)}$$

Chase and Greenwood [1988] used the reciprocal function to estimate the cost-tolerance function which has a simple form with only one parameter. This function provides a reasonable fit to the field data, but is not recommended for tolerance allocation with the geometric programming method. The reciprocal function is presented as:

$$f = \frac{a}{t} \tag{23}$$

where

$$a = \frac{\sum(f_i / t_i)}{\sum(1/t_i^2)}$$

As a more comprehensive cost-tolerance function, Speckhart [1972] applied the exponential function as a nonlinear expression which showed a good fit with the field data next to the Michael-Siddall C-T function.

$$f = a\exp(-bt) \tag{24}$$

where

$$a = \exp(z)$$
$$b = \frac{z\sum t_i - \sum t_i \ln f_i}{\sum t_i^2}$$
$$z = \frac{\sum(t_i \ln f_i)\sum t_i - \sum \ln f_i \sum t_i^2}{\left(\sum t_i^2\right) - n\sum t_i^2}$$

Also, Michael and Siddall [1981] adapted the following cost-tolerance function to minimize the total cost for assembly manufacturing.

$$f = at^{-b}\exp(-bt) \tag{25}$$

The constants in the function could be obtained by solving the following equations simultaneously:

$$zn - b\sum \ln t_i = \sum \ln f_i,$$
$$z\sum \ln t_i - b\sum(\ln t_i)^2 - d\sum t_i \ln t_i = \sum \ln t_i \ln f_i,$$
$$z\sum t_i - b\sum t_i \ln t_i - d\sum t_i^2 = \sum t_i \ln f_i, and$$
$$a = \exp(z).$$

2.4 Tolerance Allocation Methods

Many authors have proposed various kinds of tolerance allocation methods. Each method has its pros and cons, and has been applied to real manufacturing problems. To compare the tolerance allocation methods, Wu et. al. [1988] presented a case study where the total assembly tolerance was given as 0.05in., and which was to be allocated among five component tolerances while minimizing the total manufacturing cost.

The tolerance allocation methods considered were a proportional scaling method, constant precision method, Lagrange multiplier method, geometric programming method, discrete method, linear programming method, and nonlinear programming method. Among them, the proportional scaling method and the constant precision method are non-optimization methods and not cost-driven.

The exponential cost-tolerance function which is used for comparison of various tolerance allocation methods was be expressed as:

$$f = a_i \exp(-b_i t), \qquad i = 1,2,3,\ldots \tag{26}$$

And the worst-case model was used to calculate the assembly tolerance due to its simplicity. The constraint function to be considered for the study of tolerance allocation methods related the assembly tolerance to the individual components tolerance. The objective function is to minimize the manufacturing cost as:

$$\begin{array}{ll} \textbf{Min} & C \\ \textbf{Subject to} & \sum t_i \leq T \end{array}$$

where C = manufacturing cost,

T = assembly tolerance, and

t_i = tolerance of the i^{th} component.

(1). Proportional Scaling Method

The component tolerances are assigned by the process or product designer with a reasonable value at the first phase of design. Then, those tolerances are summed up and checked with respect to the specified assembly tolerance. If they do not meet the required tolerance, the designers scale the component tolerances by a constant proportionality factor to preserve the relative magnitudes of the tolerances.

Chase and Greenwood [1988] demonstrated this method to scale the component tolerances within a shaft and bearing assembly. The relations between the original component tolerances, t_i, and the scaled component tolerances, r_i, are given by

$$\frac{r_1}{t_1} = \frac{r_2}{t_2} = \frac{r_3}{t_3} = \frac{r_4}{t_4} = \ldots,$$

$$R = \sum r_i, \tag{27}$$

$$t_i = r_i \frac{T}{R}.$$

where R = scaled assembly tolerance, and

T = assembly tolerance.

The proportionality or ratio factor, p, is used to set the values of r_i and plays an important role in achieving favorable results. This method yields the highest average cost and maximum cost for production, but requires relatively short CPU time with the simplest formulation [Wu et. al., 1988].

(2). Constant Precision Method

Components machined to a similar precision have equal tolerances only if they are the same dimension. "As a components' dimension increases, tolerance usually increases approximately with the cube root of dimension" [Chase and Greenwood, 1988]. If there exists a relation between the original component dimensions, d_i, and the scaled component dimensions, r_i,

$$\frac{r_1}{d_1} = \frac{r_2}{d_2} = \frac{r_3}{d_3} = \frac{r_4}{d_4} = \ldots$$

then, the i^{th} component tolerance is obtained by the following equation [Wu et. al., 1988] as:

$$t_i = \frac{T r_i^{1/3}}{\sum_{j=1}^{n} r_j^{1/3}} \tag{28}$$

The Proportional Scaling method requires an initial allocation for the component tolerances, but this method allocates the tolerances according to the nominal size of each component dimension. The maximum and average cost are high but lower than that obtained with the proportional Scaling method. Also, the CPU time is short and the formulation is very simple.

(3). Lagrange Multiplier Method

If various dimensions in an assembly had different costs to produce, the Lagrange Multiplier method is very effective since it requires a short computer time to calculate the tolerances at the lowest manufacturing cost. Spotts [1973] presented the total cost C_v of an assembly as:

$$\begin{aligned} C_v &= c_1 + c_2 + c_3 + \dots \\ &= \frac{k_1}{t_1^2} + \frac{k_2}{t_2^2} + \frac{k_3}{t_3^2} + \dots + M_1' + M_2' + M_3' + \dots \end{aligned} \tag{29}$$

where c_i = total cost to produce the i^{th} part,

$k_i/t_i^2 = M_i$ = cost to produce the i^{th} part involved in the assembly, and

M_i = all the remaining cost for the i^{th} part.

The specified assembly tolerance t_v is supposed to be allocated arithmetically among the part tolerances, then it is represented by:

$$t_v = t_1 + t_2 + t_3 + \dots = \varphi(t_1, t_2, t_3) \tag{30}$$

Also, Eq. (29) could be represented by the function of t_i such as:

$$C_v = f(t_1, t_2, t_3, \dots) \tag{31}$$

To minimize the total cost C_v subject to the resulting tolerances to be satisfied, the following relationships should be satisfied.

$$\frac{\partial f}{\partial t_i}+\lambda\frac{\partial \varphi}{\partial t_i}=0 \qquad i=1,2,3,\ldots \tag{32}$$

where λ is a suitable multiplier that will cause all equations to be satisfied. Therefore, the total cost C_v will have a minimum with the part tolerances such as:

$$\begin{aligned} t_1 &= \frac{t_v}{1+\left(\frac{k_2}{k_1}\right)^{1/3}+\left(\frac{k_3}{k_1}\right)^{1/3}+\ldots} \\ t_2 &= \left(\frac{k_2}{k_1}\right)^{1/3} t_1 \\ t_3 &= \left(\frac{k_3}{k_1}\right)^{1/3} t_1 \end{aligned} \tag{33}$$

and so on.

The advantages of this method are its simplicity, shorter computer time with respect to other methods, and its capability to handle all the cost-tolerance functions except the Michael-Siddall relationship.

(4). Geometric Programming Method

The geometric programming method as designed to obtain least cost tolerances for an exponential cost model, by Wilde et. al. [1975]. The closed form derived was the same as that for the Lagrange Multiplier method, although the approach to derive the closed form was different.

The optimizing tolerances were acquired by the following form as:

$$t_i=\tau_1\left[\frac{T_a}{T}+\ln\left(\frac{r_i\tau_a}{R_a\tau_i}\right)\right] \qquad i=1,2,\ldots,n \tag{34}$$

where τ_i = characteristic tolerance

T_a = subassembly tolerance

t_i = semi tolerance of the i^{th} component

R_a = geometric means of the cost ranges

From the above equation, τ_i is a known constant called the characteristic tolerance, which plays an important role in the result. Normally, it is obtained either by nonlinear least squares curve fitting or as the negative reciprocal slope of the straight line obtained by plotting a variable cost versus the tolerance on semilogarithmic graph. The relation between the semitolerances and the subassembly tolerance were given as:

$$\sum_{i=1}^{n} t_i \le T_a \qquad (35)$$
$$t_i \ge 0 \qquad i = 1, 2, \ldots, n$$

Let R_a and τ_a be the geometric means of the cost ranges r_i and the characteristic tolerances, respectively. That is,

$$R_a = \prod_{i=1}^{n} (r_i)^{\tau_i/\bar{T}} \qquad (36)$$

$$\tau_a = \prod_{i=1}^{n} (\tau_i)^{\tau_i/\bar{T}} \qquad (37)$$

where

$$\bar{T} = \sum_{i=1}^{n} t_i$$

Like with the Lagrange Multiplier method, this method resulted in the lowest average manufacturing cost and a short CPU time. But only the exponential cost-tolerance function could be used.

(5). Discrete Method

Unlike the other methods which specified tolerances so that the assembly could function first and hopefully be the least expensive, this method guaranteed the true-position tolerancing with the least cost [Ostwald and Huang, 1977].

As discussed earlier in this chapter, the exponential cost-tolerance function was used as Eq. (25). In the Discrete method, the whole tolerance range on the function was segmented by discrete sections and the tolerance at each section was determined by the middle point in the section. There existed a corresponding discrete cost for a discrete tolerance determined this way.

The total cost, *C*, is represented by:

$$C = \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} COST(i, k_i) \tag{38}$$

where *COST* $(i,\ k_i)$ is a discrete cost of the i^{th} part of an assembly and k_i^{th} discrete value on the exponential cost-tolerance function. The constraint function for this objective function is

$$T - \sum t_i = T - \sum t(k_i) \geq 0 \tag{39}$$

where $t(k_i)$ is a discrete tolerance determined of ki^{th} discrete value.

As the number of discrete section increased, the computer time significantly increased, but reasonable tolerances could be acquired with small discrete sections. Though the average manufacturing cost and the computer time were high, this method had a simple formulation and could be applied to all kinds of cost-tolerance functions.

(6). Linear Programming

On each cost-tolerance curve of the assembly, the optimum component tolerance was obtained within the surrounding region of the optimum according to the linearized cost-tolerance curve. The objective function to minimize the total cost was [Patel,1980]:

$$C = \sum_{i=1}^{n} c_i(t_i) \tag{40}$$

where the t_i and c_i wee the tolerance size and manufacturing cost of the i^{th} linearized cost-tolerance curve. The constraint was given by the specified tolerance as:

$$\sum_{i=1}^{n} t_i \leq T \tag{41}$$

The most convenient algorithm to allocate the component tolerances was the simplex linear programming. As the segment size becomes smaller, the piecewise linearized function approached the continuous cost-tolerance function, and accordingly better optimum tolerances were acquired. Since this method used an iterative method, a longer computer time was needed.

(7). Nonlinear Programming

Because the cost-tolerance curve is actually a nonlinear function, more realistic tolerances can be obtained by nonlinear programming. But the formulations of the nonlinear programming are very seldom simple, and often longer computer time is required. The methods of the nonlinear programming which are widely used are the Hooke-Jeeves direct search method, random adaptive search method, David Fletcher-Powell method, Fletcher method, Jacobson-Oksman method, Powell's direct search method, simplex method, and reduced gradient method.

Lee et. al. [1993] provided a general framework for tolerance synthesis for nonlinear systems with multiple, dependent design constraints. Least-cost tolerance synthesis is mathematically formulated as a nonlinear programming (NLP) problem.

2.5 Tolerance Optimization for Process Selection

Besides methods to obtain the minimum manufacturing cost are methods that attempt an optimal selection of manufacturing processes during tolerance optimization.

Ostwald and Huang [1977] used the zero-one linear programming method for optimal selection of tolerances for a series of dimensions. For each component, a set of cost associated with the different specified tolerance values is obtained. An integer programming model is formulated by use of zero-one decision variables that selects only one tolerance value available. A zero-one algorithm developed by Balas [1965] is used to solve this problem and obtain the optimum set of tolerances for each component. The processes that correspond to these tolerances would then be used to manufacture the components for the assembly. However, if there are a total of N discrete points and suppose x_{ij} is the decision variable that selects the j^{th} point for the part i. Using binary tree enumeration, the number of combinations of the zero-one variable x_{ij} to be evaluated would be of the order of 2^N. Using Balas algorithm the authors claim that for moderate to large designs, up to 2^{30} in size, only 2% - 10% of the total number of combinations need be evaluated. However, this method is useful only in the case of worst case stack-up of tolerances.

Lee and Woo [1989] present a branch and bound algorithm to solve the above mentioned discrete tolerance selection problem. The computation time for this algorithm is much less, of the order of ps_i ($i = 1, 2, ..., N$), where s_i is the total number of discrete points on the tolerance-cost curves for the component i. This algorithm guarantees an optimum solution under linear as well as nonlinear (statistical) tolerance stack-up conditions.

Chase et. al. [1990] have presented three new methods - exhaustive search technique, univariate search procedure and sequential quadratic programming (SQP) technique - for automatically selecting the most economical manufacturing process for each part dimension from a set of alternative processes. In the exhaustive search method all the possible combinations of processes are evaluated and for each combination an optimization routine carried out to select the least value of cost. For example, consider an assembly made of N components, each of these components having n_i alternative manufacturing processes. Therefore, there are a total of $n_1 \times n_2 \times n_3 \times \ldots \times n_N$ combinations to be evaluated. Each of these combinations is solved as a separate optimization problem to get the optimal tolerances for that particular combination. After evaluating all the combinations, the solutions are compared and the one with the minimum objective cost value is taken as the optimal. Chase et. al. [1990] and Loosli [1987] have applied the

Lagrange multiplier technique to solve the non-linear optimization problem for each combination. With this type of procedure, the number of combinations to be evaluated increases geometrically with the number of parts and the number of processes.

Chase et. al. [1990] and Loosli [1987] also describe the univariate search problem. The number of combinations that had to be evaluated for a univariate search procedure is much less than those for the exhaustive search technique. Specifically, for an assembly of N components, with each part i having n_i of alternate processes, the number of combinations is $1 + n_1 + n_2 + n_3 + \ldots + n_N - N$. In this method, an initial arbitrary set of processes, one for each part, is selected and Lagrange multiplier method is applied to allocate the tolerances. The objective cost value is determined by this combination. The first univariate search is performed by evaluating each of the alternate processes of one of the parts, while holding the processes constant for the remaining $N-1$ parts. Of all the alternate processes evaluated, the process, which yields the minimum cost value, is fixed for that part. Next, alternate processes are evaluated on the next part and in a similar manner the process that gives the minimum cost is fixed for this part. This method is repeated until all the part levels are evaluated. Loosli and Chase et. al. demonstrate the univariate search method with an example in which there are three components that make up an assembly. Out of this, the first two parts have 2 alternate processes and the third has 3 alternate processes available.

To improve upon the above described univariate search technique and to obtain a better solution, Loosli [Loosli, 1990] proposed a continued univariate search method wherein starting from a current univariate optimum, the univariate search is repeated up to the point where there is no change in the process combination during the entire search. As each of the trial combination in the univariate search method is solved using the Lagrange multiplier technique, this imposes a limitation that bounded processes (processes defined only in a particular range of tolerance) cannot be handled. To overcome this difficulty, an "end-fixing" operation was performed by Loosli and Chase et. al. In this method, for each trial combination of the exhaustive search, tolerance allocation was performed without considering the process limits. Then for all parts whose tolerance values fell outside the limits, the tolerances were incremented (or decremented) in small steps. Keeping these tolerances fixed, tolerances on other parts were reallocated. This procedure was repeated till all the constraints were satisfied (i.e. all the tolerances allocated on each component fell within the specified tolerance limits).

In another method called the Sequential Quadratic Programming (SQP) technique, Chase et. al. use a non-linear optimization technique to solve a continuous optimization problem. This continuous optimization problem is different from the discrete optimization problem in that the binary coefficients x_{ij} are no longer restricted to take a value of 0 or 1 but are allowed to vary continuously between 0 and 1. In other words an equation expressed by $x_{ij} = 0$ *or* 1 is now replaced by 0 £ x_{ij} £ 1 $(\, i = 1, \ldots, N \quad j = 1, \ldots, M)$. The intent is that the optimization procedure will drive each of these binary coefficients to either a 0 or 1 automatically rather than selecting a combination of more than one process for any component. This is not true in all the cases and as such the SQP method does not always guarantee a feasible solution for all the problems. As pointed out by Zhang and Wang [1993], the SQP tended to select a combination of processes rather than selecting a single process to produce a single component for problems with non-overlapping curves or wide process ranges. With the analysis on different types of problems, they also conclude that the SQP algorithm often fails to find a global optimum for moderate to large problems.

Simulated Annealing (SA) is one of the techniques used to solve this combinatorial problem which is NP-hard. The running time of any algorithm that would guarantee an optimal solution to this problem (like the exhaustive search technique) is an exponential function of the size of the problem. SA is one of the heuristic approaches designed to give a good though not necessarily optimal solution, within a reasonable computing time. The SA algorithm was introduced by Kirkpatrick et. al. [1983] based on a model for simulating the annealing of solids [Metropolis, Rosenbluth, Rosenbluth and Teller, 1953]. Eglese [1990] describes the SA algorithm as applied to a combinatorial optimization problem. SA is a type of local search algorithm that starts with an initial solution. A neighbor of this solution is then generated (either randomly or by another mechanism) and the objective cost function value is evaluated. If the new solution is better than the current solution, the move (transition from current solution to the neighboring solution) is accepted and the current solution is replaced by the new one. However, to avoid getting trapped at a local minimum, the SA algorithm sometimes accepts a neighborhood move which increases the value of the objective cost function. The acceptance or rejection of an uphill move is controlled by a certain probability. The probability of accepting a move, which causes increment d in the objective function f, is normally set to *exp(-d / kT)* where *T* is a control parameter and *k* is a constant. This implies that small increases in *f* are more likely to be accepted than large increases and that when *T* is high most moves will be accepted, but as *T* approaches zero most uphill moves will be rejected. In SA the algorithm is started with a relatively large value of *T*, by attempting a certain number of neighborhood moves at each temperature, while the temperature parameter is gradually dropped. The different simulated annealing algorithms as applied to the discrete tolerance optimization problem differ in the move used to generate a neighborhood state, or in other words how a neighbor is defined.

Cagan and Kurfess [1992] describe two simulated annealing based approaches to determine the tolerance on each component, as well as its manufacturing process, in an assembly. The tolerance-cost relationship used is of the reciprocal type given by

$$C_i = \left\{ \frac{K_i}{(\Delta_i - A_i)} \right\} + B_i$$

where C_i is the cost of manufacturing component *I*,

D_i is the tolerance allocated on component *I*, and

K_i, A_i, B_i are component specific parameters.

In the first approach, a random combination of processes for each component is selected. If this combination is feasible (all the constraints are satisfied), an optimization technique is applied to obtain the optimal tolerances for each component for the selected set of manufacturing processes. Then another feasible state is generated by using another random combination of processes for each component and the optimal solution is obtained for this state. If this solution is better than the current one, it is accepted. If not, the solution might be accepted or rejected based on certain probability as described earlier. These steps are repeated till the temperature T is reduced to zero (i.e. the probability of accepting an uphill move becomes zero). The best solution obtained in all the trial evaluations is recorded, which may not necessarily be the global optimum. However, statistically if the algorithm is run for sufficient time it is likely to have found the optimal state. This algorithm can deal with nonoverlapping process limits.

In another approach, Cagan and Kurfess [1992] first start with an initial feasible solution, then randomly generate new tolerances in the neighborhood of the first n-1 components with the final component assigned the remaining available tolerance range. Based on the tolerance selected for each component the cost function for each component is evaluated and total cost of assembly is obtained. Based on the annealing technique this new solution is accepted or rejected. Termination is achieved when the temperature reaches zero and the best-recorded solution is taken as the final answer.

Cost-based analysis on process selections and determination of target process capability indices was conducted by Vasseur [1994] emphasizing the need to allocate tolerances and select manufacturing processes concurrently. He had shown that an optimum process selection was not compatible with the decision to impose uniform process capability indices for all processes. He cited as "process selection is best performed by taking into account the characteristics of the part manufactured as well as that of the final product considered." He counted the characteristics such as measurement uncertainty, inspection cost, and inspection delays as hindering conformance verification.

Zhang and Wang [1993] have coded and compared the simulated annealing technique to solve the discrete tolerance optimization model, with the SQP method on a range of problems. The SA algorithm which they used differs in that a move from the current configuration to the neighboring configuration. A move comprised of a random increment in the tolerance values on each component and a process selection combination after one move. The authors conclude that the SA algorithm is robust and effective for solving a wide variety of the discrete optimization problems whereas the SQP method is only applicable for a small range of problems. Although the SQP was found to be more efficient for smaller problems that have narrow process precision limits and overlapping cost curves, the SA algorithm often provided global or near-global optimum solutions for complex discrete optimization.

Process approximations have also been applied by Nagarwala et. al. [1995]. They used a slope-based (SB) method for tolerance optimization and process selection. After defining tolerance-cost curve using the process tolerance-cost curves for each component of the assembly, the optimal solution for the cost-effective process with the associated tolerance value was obtained using the SB method. They tested the SB method with test problems and obtained global solution or near-global solution. Kim et. al. [1999] developed the algorithm to obtain the least cost tolerance allocation for a bi-criteria optimization problem. Through the slope-defined investigation for the cost and tolerance, the optimal process selection was conducted by minimizing both the cost and tolerance. A similar approach to these is used in solving the bi-criteria problems in this dissertation.

Process selection in the above context is largely the selection of a simple process / component. Process planning is however more detailed in the sense that each component is manufactured through a combination of multiple processes / operations.

2.6 Process Planning

Process planning is defined as, "the function within a manufacturing facility that establishes the processes and process parameters to be used (as well as those machines capable of performing these processes) in order to convert a piece-part from its initial form to a final form which is predetermined (usually by a design engineer) on a detailed engineering drawing" [Chang and Wysk, 1985)].

Alternate process plans are defined as a set of process plans where each plan is capable of completing the machining activities required for a part, based on the availability of resources such as machines, cutting tools, and part fixtures, in a specific production environment. Alternate process plan may be used in several ways in a production system. Simulation and analytical models of production systems having alternate process plans permit flexibility in scheduling jobs in the system [Khoshnevis and Chen, 1990, 1993]. This flexibility allows the scheduler to perform functions such as balancing of workload on machines and routing of jobs to avoid bottleneck machines. Disruption in the production system due to machine failures, sudden changes in demand, and arrival of "hot" jobs, can be handled better by using alternate process plans for reactive rescheduling [Foote *et. al.*, 1993]. The selection of a specific process plan is usually based on multiple factors that include typical shop practices, processing times, processing costs, tolerance stack-ups, and work-piece deflections.

Process planning is an essential function within a discrete part manufacturing facility. Recent literature in process planning comes from the area of Computer Aided Process Planning (CAPP). CAPP approaches are classified into three categories: (i) variant process planning, (ii) generative process planning, and (iii) semi-generative process planning. Several knowledge-based and AI-based approaches have been used in CAPP for feature extraction, precedence representation, process selection, and sequencing.

2.6.1. Variant Process Planning

In variant process planning, the process plan for a new part is created by "recalling, identifying, and retrieving an existing process plan for a similar part, and then making the necessary modifications for the new part" [Alting and Zhang, 1989]. The principles of Group Technology (GT) are used for process planning, in this approach. Usually, a part family or a group of part families has a standard plan called a master plan. The process plan for a new part is obtained by retrieving and modifying the master plan of its part family.

A structured procedure for variant process planning is given by Chang and Wysk [1985] and Chang *et. al.* [1991]. The procedure for setting up a variant process planning system (preparatory stage) and for developing the process plan for an incoming part (production stage) is discussed. The preparatory stage involves coding of existing part drawings, formation of part families, and development of master process plans for each part family. The production stage involves coding of an incoming part, part family search, retrieval of master plan, and plan editing.

MIPLAN is a variant process planning system developed around 1976 [Alting and Zhang, 1989]. MIPLAN has been modified to a new process planning system called MultiCapp, based on the MULTI-IL GT scheme. A typical route sheet produced by the MultiCapp system is shown by Groover [1987]. CAPP is a very popular variant process planning system that was developed around the same period as MIPLAN [Link, 1976; Tulkoff, 1978]. It comprises of an extensive database from which process plans can be retrieved based on GT methods. A more recent variant process planning system is MICRO-CAPP [Wang and Wysk, 1986]. It is limited to rotational parts and uses the KK-3 coding scheme. Alting and Zhang [1989], and Chang and Wysk [1985] review some of the other variant process planning systems.

2.6.2. Generative Process Planning

Generative process planning works by creating the process plan based on logical procedures that a human process planner would use to convert the part from raw material to finished state [Groover, 1987]. In such an approach, "processing decisions are made based on decision logic, mathematical formulae, technology algorithms, and geometry based data" [Alting and Zhang, 1989]. In its ideal form, a generative process plan is obtained by extracting geometric data from a CAD file and then creating the process plan without any manual intervention.

The basic principles involved in generative process planning is described by Chang and Wysk [1985], and Chang *et. al.* [1991]. APPAS [Wysk, 1977] is one of the earliest generative process planning system, and is limited to prismatic parts. AUTAP [Eversheim and Holz, 1982] is a popular generative process planning system with a CAD interface. In conjunction with AUTAP-NC, it has the capability of generating part programs. Alting and Zhang [1989], and Chang and Wysk [1985] review several other generative process planning systems.

2.6.3. Semi-Generative Process Planning

The semi-generative approach is "a combination of the variant and the generative approaches for process planning" [Alting and Zhang, 1989]. An initial process plan is developed by variant or generative procedures. This initial plan is modified manually to obtain the final process plan for the part. Joshi *et. al.* [1994] give a detailed description of the design, development, and implementation of a semi-generative process planning system. Alting and Zhang [1989] review some of the semi-generative process planning systems.

2.6.4. Interpretation of Design Data

In many generative process planning systems, the first step is the interpretation of design data. A true generative process planning system should be capable of interpreting data from a CAD file. This is an area of ongoing research and there are several techniques given in the literature. Many of them are limited by the types of parts that are permitted or the nature of design representation in the CAD file.

Shah [1991] reviews three distinct methodologies that are prevalent for creating feature models for parts: (i) interactive definition, (ii) automatic recognition, and (iii) design by features. Of these, automatic recognition is the most popular method for CAPP. Wang and Wysk [1988] present an algorithm for automatic identification of machined surfaces on symmetric rotational parts. This algorithm is implemented on a wireframe model generated by AutoCAD. Yeh and Fischer [1991] use automatic recognition of features from Initial Graphics Exchange Specification (IGES) in the AMOPPS generative process planning system. Cho *et. al.* [1994] illustrate a procedure for identifying geometric entities and for forming primitive features from a Drawing Exchange Format (DXF) file. Two very important challenges in past feature recognition and interpretation exist in tolerance representation & transfer; and manufacturing precedence representation.

2.6.5. Precedence Information Representation

Precedence constraints between machining activities must be satisfied for generating feasible operation sequences. Several methods for obtaining and representing precedence information are seen in the literature. Prabhu *et. al.* [1990] use a datum table for representing precedence information between machining operations.

Yeh and Fischer [1991] use a fixed precedence relationship which does facing, then turning, and finally grooving operations. In a similar approach, Smith *et. al.* [1992] use a fixed hierarchical sequence for obtaining feature precedence for sheet metal parts in the FCAPP/SM process planning system. In expert system and AI-based approaches, the precedence information is predefined in the rule base [Alting *et. al.*, 1988]. The limitation of such procedures is that the precedence information is fixed and cannot be made part-specific. There is no flexibility for adding or removing precedence relationships that are often required to represent the true nature of the machining activity. Moreover, an "optimal" process plan for a specific part need not follow the fixed precedence relationship.

Irani *et. al.* [1995] restrict specific predecessor-successor relationships among part features by means of high penalty costs. This procedure can prevent a feature from immediately following another feature i.e., F1-F2 can be prevented by making it a high penalty transition. However, it cannot always prevent the latter feature from following the former feature at a later stage as in F1-F3-F2, and F1-F3 and F3-F2 may be low penalty transitions. Also, the procedure for finding such predecessor-successor relationships have not been investigated in their work.

In graph theory, a precedence graph is a very popular method for representing precedence relationships. Lee *et. al.* [1994] illustrate a simple precedence graph. Such a graph defines the set of activities (nodes) that must be completed before the start of other activities. The information represented by a precedence graph is similar to that represented by the datum table suggested by Prabhu *et. al.* [1990]. This is illustrated in Table 2.2, which is the datum table for the precedence graph shown in Figure 2.2. Precedence graphs are limited by the fact that mutually exclusive precedence relationships of the form "A precedes B OR C precedes D" cannot be represented.

Table 2.2. Datum Table

Surface	Datum
1	2
2	-
3	2,4
4	2,1
5	2

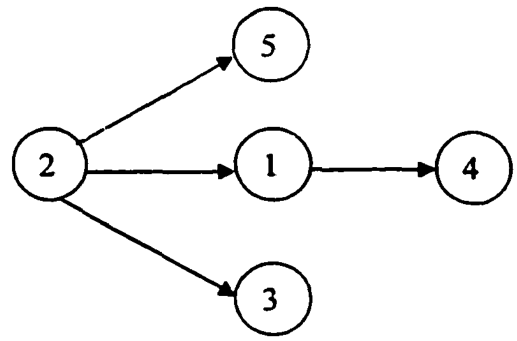


Fig. 2.2. Precedence Graph

The limitations of precedence graphs can be overcome by the use of AND/OR graphs. De Mello and Sanderson [1986] use AND/OR graphs for assembly sequence planning. This procedure is adopted by Mettala [1989], and Mettala and Joshi [1993] for representing precedence information for process planning. However, the procedure for obtaining precedence information from a design drawing for representing as an AND/OR graph is not discussed in most works. Also, unlike a datum table, it is not possible to represent any arbitrary set of precedence relationships in the form of an AND/OR graph.

Cho *et. al.*[1994] is one of the new works that describe an orderly procedure for obtaining precedence information from a CAD file. The design information in standard DXF file is converted to an AND/OR graph of primitive features based on feature precedence. This is later converted to an AND/OR graph of machining tasks. However, the representation of precedence information is purely based on the location of a particular volume of material with respect to other volumes of material within the part. It has no means of incorporating precedence relationships that occur due to factors such part dimensioning, work-piece location, shock loading of tools, thickness limitations, and standard practices that come naturally to a experienced machinist. Further, the precedence relationship is developed for a single machine case with single tool approach direction.

Korde *et. al.* [1992] use a feature-precedence graph (FPG) to represent precedence information. The precedence information is obtained by identifying resource independent constraints such as accessibility constraint, non-destruction constraint, and required-holding constraint on the part features.

More information on precedence representation is available in Mani [1996]. Some tolerance representation details are worked in Kumar and Raman [1992].

2.6.6. Preferred Sequences and Alternate Sequences

The use of alternate process plans has been illustrated in various research works. Foote *et. al.* [1993] have proposed alternate process plans for rescheduling in the face of disruptions in the production system such as machine breakdowns, changes in demand, and arrival of "hot" jobs. They have also proposed using alternate process plans for proactive scheduling. Khoshnevis and Chen [1990] and Chen and Khoshnevis [1993] have integrated alternate process plans with scheduling functions and have noticed improvements over traditional methods of process planning and scheduling. Lenderink and Kas [1993] have used alternate process plans in developing an integrated planning system called PART.

There is very little literature available on formal procedures for developing alternate process plans. Prabhu *et. al.* [1990] present an efficient method for generating all feasible operation sequences for a part without enumerating all *n!* sequences and checking each sequence for feasibility. They use a recursive relationship for developing an out-trees structure where each branch in the out-tree is a feasible operation sequence. No procedure is suggested for evaluation of the operation sequences and finding a preferred sequence. Ben-Arieh and Kramer [1994] present a similar procedure for generating all feasible assembly sequences of a part.

The liberalization of a feature-precedence graph (FPG) to a state-transition (ST) graph by Korde *et. al.* [1992] generates all feasible process plans for a part. The ST graph is pruned by identifying certain resource independent constraints. It is suggested by Korde *et. al.* [1992] that a cost function based on certain optimality criteria be used for identifying a preferred process plan.

Smith *et. al.* [1992] use a network approach for finding a preferred machine sequence for sheet metal parts. An acyclic diagraph is developed where each level in the graph corresponds to a feature and contains all machines capable of producing that feature. Each arc in the diagraph represents the cost of transportation between the machines. Dijkstra's algorithm [Ahuja *et .al.*, 1993] is then used for obtaining the least cost machine sequence between start and end nodes. This method is very good for sheet metal operations since the diagraph can be developed based on a fixed feature precedence relationship (holes→slots→bends.). In process planning for machining, this procedure can be adopted only by fixing the operation sequence and then determining a least cost machine for the operation sequence. This restriction may prevent the "optimal" machine sequence from being determined.

Irani *et. al.* [1995] perform a constrained enumeration of feasible Hamiltonian Paths using the Latin Multiplication Method [Gibbons, 1985]. These paths are then evaluated based on penalty costs for machine changes, set-up changes, tool changes, and parameter changes. This procedure has the shortcoming that some of the generated sequences could end up being infeasible because of limitations in precedence information representation.

Cho *et. al.* [1994] describe a formal procedure for converting a CAD model of a product to a hierarchical set of process plans. Wysk *et. al.* [1995] and Lee *et. al.* [1994] illustrate how these process plans can be used for shop floor control function. By maintaining alternate process plans as AND/OR graphs, they are useful for making online decisions at a shop level, workstation level, and equipment level [Joshi *et. al.*, 1990]. A good process plan is chosen from the alternatives in real-time, depending on the condition of the shop floor. Since AND/OR graphs are used for representing alternatives in these works, there are certain limitations.

2.7 Multi-Criteria Optimization Problem

In engineering problems, the need for formulating a design with several criteria or design objectives is quite common. If there are opposing objectives in the formulation, the problem should be solved to find out the best possible design with satisfying opposing objectives and the subjective constraints. This type of problem is known as either a multi-objective, multi-criteria, or a vector optimization problem and can be defined as follows (Osyczka, 1985): "A vector of decision variables that satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria that are usually in conflict with each other. Hence, the term "optimize" means finding such a solution that would give the values of all the objective functions acceptable to the decision maker." As an example, in the design of an automobile an engineer may wish to maximize crash resistance for safety and minimize weight for fuel economy. This is a multi-objective problem with two opposing objectives, and thus a step towards improving one of the objectives, increasing crash resistance, is a step away from improving the other, increasing weight.

2.7.1 History and Background

Lebnitz and Euler used infinitesimal calculus to find the extreme values of functions. This made it possible for pioneers such as Newton to study various new fields of physics and mechanics. A French-Italian economist named Pareto developed the principle of multi-criteria optimization for use in economics. His theories became collectively known as Pareto's optimality concept. While having several objective functions, the notion of "optimum" changes, since trade-offs are sought rather than a single solution as in global optimization.

2.7.2 Pareto Optimality [Coello, 1999]

We say that a vector of decision variables $\vec{x^*} \in F$ is Pareto optimal if there does not exist another $\vec{x} \in F$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1, \ldots, k$ and $f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j.

It is expressed as $\vec{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\vec{x} \in F$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. We do not always obtain a single solution, but rather a set of solutions. It is called the Pareto optimal set. The vectors $\vec{x}^*$ corresponding to the solutions included in the Pareto optimal set are called non-dominated. It is also called as an efficient solution, and defined as "A feasible solution is efficient if there is no other feasible solution which is better with respect to every criterion." [Coello, 1999]. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front.

A Multiple-Objective (MO) optimum design problem is solved in a manner similar to the Single- Objective (SO) problem. In a SO problem, the idea is to find a set of values for the design variables that, when subject to a number of constraints, yield an optimum value of the objective (or cost) function. In MO problems, the designer tries to find the values for the design variables that optimize the objective functions simultaneously. In this manner, the solution is chosen from a so-called Pareto optimal set. In general, for multi-objective problems the optimal solutions obtained by individual optimization of the objectives (i.e., SO optimization) is not a feasible solution to the multi-objective problem. Multi-criteria mathematical programming handles the deterministic criterion with infinite alternatives.

It is worth to trace the path of emergence and development of methods and techniques to solve the multiple-criteria optimization problem. There are several useful books that explain principles and tools for multi-criterion decisions such as Multiple Criteria Optimization [Steuer, 1966], Multicriteria Decision Making [Zeleny, 1982], and Multiple Objective Decision Making and Application [Hwang et. al., 1979]. The problem of multiple criteria in linear programming was also tackled by Kuhn and Tucker (1951), and later by specialists in operational research such as Hitch (1953), Klahr (1958) and several others.

Linear Programming method for Multi-dimensional Analysis of Preferences (LINMAP) was introduced by Srinivasan and Shocker [1973]. The method was based on the paired comparison between alternatives. After ranking and weighting procedures, the objective is obtained suitable to the decision maker. Goal programming method was introduced by Charnes and Cooper [1961]. They solved the problem of multicriterion choice in linear programming by a search for a solution at minimal distance from a multicriterion goal, generally non-achievable, set by the decision maker. The first applications quickly demonstrate the GP interest in a number of areas (Charnes et. al., 1963, 1968). Later, numerous variants and a number of impressive applications followed (Charnes and Cooper, 1977). Zanakis and Gupta (1985) and later Romero (1986, 1991) and Schneiderjans (1995) have itemized hundreds of papers dealing with a wide range of problems.

The most decisive element of the eighties was the introduction of computer methods in to multicriterion decision making. By 1970, interactive methods were proposed; the main novelty was the ease with which they could be installed and the fact that the tasks which computers and microcomputers were now capable of performing had a powerful influence on the design of methods [Sadagopan et. al, 1982, 1986; Wallenius, 1975; Zionts et. al, 1976, Shin et. al. 1991]. Steuer [1977] used the Interval Programming to generate a cluster of efficient solutions with the number of 2p+1 where p is the number of criteria. Also Geoffrion et. al. [1972] applied the interactive method to the operation of an academic department. More information about multi-criteria optimization problem could be obtained through the reference survey of Coello [2002].

2.8 Cost and Time Analysis and Recent Literature

With respect to the cost-tolerance analysis, there are few approaches which discuss cost-time analysis. Bukchin and Tzur [2000] proposed an optimal and a heuristic algorithm for the problem of designing a flexible assembly line when several equipment alternatives are available. Its objective was to minimize total equipment costs, given a predetermined cycle time (production rate). They did not relate the time factor to the tolerance analysis, but used the time factor for getting the equipment cost which resulted in obtaining the manufacturing cost.

A new method, Cost Tolerance Sensitivity Analysis (CTSA), of determining which features are critical and non-critical early in the design phase when cost information is uncertain, was presented by Gerth and Pfeifer [2000]. They used minimum cost tolerancing methods combined with designed experiments to determine which features are sensitive or insensitive to uncertainties in the cost-tolerance curve estimates.

They did not use ANOVA tool to analyze, and concluded that one cannot determine the critical dimension critically based on the stack up function alone, nor based on the processing costs. The most difficult aspect of the methodology is obtaining cost tolerance information. Therefore, the research area of how cost tolerance information should be structured, obtained, organized, and disseminated should be emphasized. With this point of view, cost tolerance and cost time analysis are critical to manufacturing industry.

Using analytical and mathematical approaches, it is not possible to quantify and transform these imprecise and subjective criteria and random factors into proper input variables in the cost estimation model. Therefore, Jahan-Shahi et. al. [1999] used fuzzy sets probability distribution approaches to tackle the problem of uncertainty in cost estimation in order to generate reliable cost estimates in flat plate processing industry. They suggest that application of fuzzy approach can overcome the limitations of traditional mathematical formulations of cost estimation relationships.

In general, literature does not provide any optimization models that attempt to study tolerances under both time and cost constraints. However, such an analysis will be particularly useful in cases such as selective assembly. In selective assembly, assembly allowances are maintained at the expense of component tolerances. This can cause time expenses in sorting and pairing.

Chapter 3

Mathematical Formulation

3.1 Introduction

The main interest for designers and manufacturers is to manufacture an assembly with the least cost and the most reliable tolerance within a given time period. Many researchers and scholars devote themselves to determine the relationship of cost-tolerance and cost-time to obtain the least manufacturing cost. To accomplish a company's expected profits in a fast changing market, the least cost-tolerance-time model and optimal manufacturing processes should be considered. Most of the models that have been developed and applied to the real manufacturing fields deal with only the cost-tolerance relation to achieve their goals.

In the continuously fast-changing modern market, time plays an important role in reducing the total manufacturing cost and in meeting deadlines. Sometimes customers want to hasten the normal manufacturing time for their special occasions, and/or manufacturing lines require more flexible tolerance range to meet their equipment conditions. In such cases, it is important to consider time in addition to cost and tolerance. It is the primary objective of this research to develop an optimization model based on the cost-time-tolerance relation. Further, this research seeks to evaluate, qualitatively the selection of manufacturing processes to meet time and cost objectives.

3.1.1 Primary Objective

The least cost-tolerance-time model for an optimal manufacturing process is considered. The objective of this study is to determine a mathematical model for integrating the influences of manufacturing cost and total manufacturing time to assembly in the determination of tolerances. There exists the tradeoffs between the cost-tolerance relationship and the cost-time relationship to obtain the best quality, the least cost, and delivery of the most competitive product on time to the customers or markets. By considering the tradeoffs between factors, the most optimal process should be presented to the decision-maker for choosing the final process in accordance to his or her priorities. The mathematical formulation is a multi-criteria optimization model to minimize the total manufacturing cost, the total manufacturing time to assemble a final product, and the total assembly tolerance.

3.1.2 Overview

There are three key factors to be considered which are time, cost and tolerance. Each model has a bi-criteria optimization problem with one factor as a constraint form. For instance, Model 1 has two objective functions. One is a minimization of time, the other is a minimization of cost. In this case, the other factor, tolerance, is given as a constraint with allowable ranges. A mathematical formulation with two objectives and a set of constraints provide a series of effective solutions to get the most optimal solution. By using a computer code, every effective solution could be obtained through a full enumeration. In this research, a new algorithm is proposed, and it would extract many solutions without a full enumeration within an efficient computer running time.

The algorithm developed gives not only the guarantee to get a set of effective solutions comparable in speed to published algorithms, but also, the general understanding of the manufacturing process fitted to the customer's special requirements.

3.2 Mathematical Models

3.2.1 Road to the general problem

Let's consider an assembly with N components. Let T be the desired assembly tolerance value. For each component i, it is assumed that n_i many processes are available. Let c_{ij} and t_{ij} be the tolerance-related manufacturing cost and the achievable tolerance value for component i with process j respectively. The decision variable x_{ij} is a zero-one variable taking the value of one if process j is selected for the i^{th} component and zero otherwise. The mathematical model for selecting the best process sequence can be defined as follows:

$$[M1] \quad Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij} x_{ij} \tag{3.1}$$

$$st. \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij} x_{ij} \le T \tag{3.2}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.3}$$

$$x_{ij} = \{0,1\} \qquad (3.4)$$

where i = Index of the component ($i = 1,2,\ldots\ldots,N$)

j = Index of the process ($j = 1,2,\ldots\ldots,n_i$)

$$X_{ij} = \begin{cases} 1 & \textit{if component i is produced in process j} \\ 0 & \textit{otherwise} \end{cases}$$

c_{ij} = Normal manufacturing cost

when component i is produced in process j

t_{ij} = Tolerance when component i is produced in process j

T = Total tolerance

C = Total budget.

The above model assumes the worst case tolerance stack-up which simply adds the tolerances selected to determine the final assembly tolerance. Chase et al. (1990) investigated zero-one search and concluded that it is impractical. Then, three new methods (exhaustive search, univariate search and sequential quadratic programming) were proposed. For the deterministic integer programming, Kusiak and Feng [1995] presented three examples - single loop dimensional chain, multiloops dimensional chain with large number of component dimensions, and integer programming formulations. In order to adapt to the nonlinear case, they used experimental design method and Taguchi method. Balakrishnan [1993] presented the shortcomings of three methods proposed by Chase et al. [1990] and modified the mathematical formulation expressed above to be suitable for a multiple-choice knapsack model. Then he employed the branch and bound algorithm in Sinha and Zoltners [1979] to solve the problem. However, these approaches are suitable only for the single objective case.

According to T value, a desired assembly tolerance which is pre-specified may be of interest for the decision maker to see how much additional cost will be incurred in order to select the best process sequence with the next least final assembly tolerance. In fact, one may be interested in the following bi-criteria (cost and tolerance) integer programming problem.

[M2] $$Min \sum_{i=1}^{N} \sum_{j=1}^{n_i} c_{ij} x_{ij} \tag{3.5}$$

$$Min \sum_{i=1}^{N} \sum_{j=1}^{n_i} t_{ij} x_{ij} \tag{3.6}$$

$$st. \quad \sum_{i=1}^{N} \sum_{j=1}^{n_i} t_{ij} x_{ij} \leq T \tag{3.7}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.8}$$

$$x_{ij} = \{0,1\} \tag{3.9}$$

By using a simple algorithm called slope-based method with some conditions, all efficient solutions could be generated for M2 problem. Besides cost and tolerance to figure out the manufacturing scheme in real fields, there is one more criteria to consider which is the time component.

Let τ_{ij} be the completion time related to the assembly cost and corresponding tolerance value for component i with process j. The multi-criteria optimization problem for the cost, tolerance, and time could be expressed as follows:

$$[\mathrm{M3}] \quad Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}x_{ij} \tag{3.10}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij}x_{ij} \tag{3.11}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} \tau_{ij}x_{ij} \tag{3.12}$$

$$s.t. \quad \sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.13}$$

$$x_{ij} = \{0,1\} \tag{3.14}$$

The formulation [M3] would be an ***optimization problem*** to obtain the most efficient process with decision factors such as cost, time, and tolerance. This is a three criteria integer programming problem. Eqs. (3.10), (3.11), and (3.12) represent minimization of total manufacturing cost, minimization of total assembly tolerance, and minimization of total manufacturing time to assemble of the final product, respectively. If the total assembly tolerance is known, the problem [M2] becomes the problem [M1] with a single objective function by using the constraint method. By the same reasoning, the problem [M3] becomes a bi-criteria integer programming problem which contains manufacturing cost and manufacturing time to assemble as objective functions and assembly tolerance as constraint.

3.2.2 Model 1: Cost-Time Model

As stated above, the problem [M3] becomes a bi-criteria integer programming problem with a fixed assembly tolerance value T. Minimization of total manufacturing cost and Minimization of total manufacturing time to assemble as objective functions, is expressed as follows:

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}.x_{ij} \tag{3.15}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} \tau_{ij}.x_{ij} \tag{3.16}$$

$$s.t \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij}.x_{ij} \le T \tag{3.17}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.18}$$

$$x_{ij} = \{0,1\} \tag{3.19}$$

With this model, a relationship between cost and time will be found within a predetermined tolerance and also will be discussed with respect to the three factors that affect process planning and selection. After obtaining the relationship between cost and time, if we set the manufacturing time to assemble as a fixed value, within allowable value of tolerance, a model that shows the relationship between cost and tolerance should be obtained.

3.2.3 Model 2: Cost-Tolerance Model

In the model 1, total assembly tolerance was fixed to obtain the efficient solutions, but in the model 2, we set a total manufacturing time to assemble Ω as a fixed value. From the formulation [M3], the minimization of total manufacturing cost and minimization of total assembly tolerance are set as the objective functions. Therefore, the mathematical formulation for the model 2 could be expressed as follows:

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}x_{ij} \tag{3.20}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij}x_{ij} \tag{3.21}$$

$$s.t \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} \tau_{ij}x_{ij} \le \Omega \tag{3.22}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.23}$$

$$x_{ij} = \{0,1\} \tag{3.24}$$

3.2.4 Model 3:Tolerance-Time Model

The relationships between cost and time, and between cost and tolerance are discussed by models 1 and 2. The one that should be discussed is a relationship between time and tolerance. In this case, a manufacturing cost is fixed as a constraint, and the minimization of total assembly tolerance and total manufacturing time to assemble are set as objective functions, and it is expressed as follows:

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij}x_{ij} \tag{3.25}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} \tau_{ij}x_{ij} \tag{3.26}$$

$$s.t \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}x_{ij} \le \Delta \tag{3.27}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\dots,N \tag{3.28}$$

$$x_{ij} = \{0,1\} \tag{3.29}$$

Between these three models, the nature of relationship between the tolerance, time, and cost is handled. For instance, while manufacturing cost increases while tight tolerances are specified, the manufacturing time reduces due to better accuracies: when the manufacturing cost reduces, the assembly cost (time) goes up. Thus, we have arrived at a first modeling scheme to describe the tolerance, time, and cost trade-offs in a practical setting.

Chapter 4

Solutions Approach

For each model developed in the preceding chapter, the algorithms leading to efficient solutions are developed and presented here. In each section, algorithms are developed and explained to obtain the efficient solutions with step-by-step procedures. The typical cost-versus-tolerance function and cost-versus-time function are presented in Figure 4.1

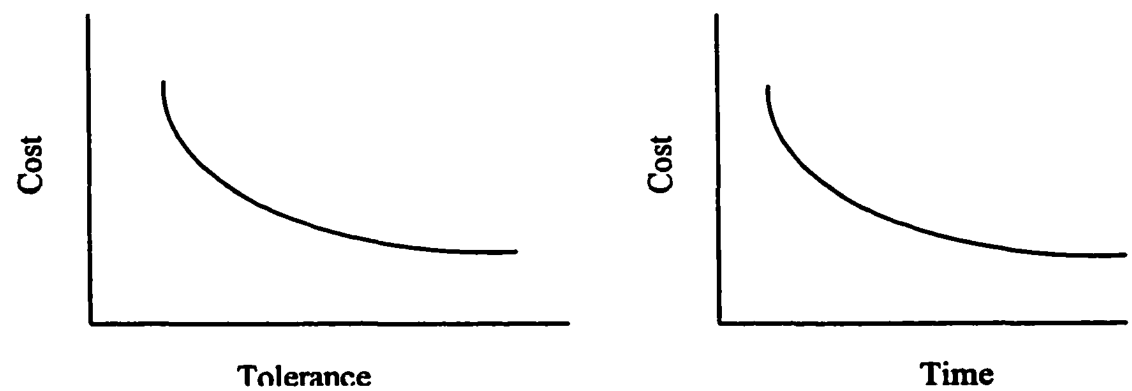


Figure 4.1 Typical Cost vs.Tolerance and Cost vs.Time Function

4.1. Cost-Time Model (Model 1)

The cost-tolerance function in the literature [Chase et. al. 1990] shows that cost is increasing as tolerance decreases. In project management applications, the relationship between cost and time is of similar nature as the relationship between cost and tolerance in the tolerance allocation problem (i.e., cost is increasing as manufacturing time decreases). In real situations, the more the manufacturing cost, usually the less assembly time to assemble is required and / or the more tightened assembly tolerance is obtained. There are trade-offs between manufacturing cost and total manufacturing (manufacturing + assembly) time to assemble with total assembly tolerance constraint.

Given the total assembly tolerance, T, sort n_i points to an increasing order of assembly tolerances (i.e., t_{i1}, t_{i2}, ..., t_{in_i}) as well as manufacturing time to assembles (i.e., τ_{i1}, τ_{i2}, ..., τ_{in_i}) and decreasing order of manufacturing costs (i.e., c_{i1}, c_{i2}, ..., c_{in_i}). Here, the cost-time ratio R_{ij} for component i is defined as $(c_{ij} - c_{i,j+1}) / (\tau_{i,j+1} - \tau_{ij})$, where $i = 1,2,\dots,N$ and $j = 1,2,\dots,n_i$. That is, the cost-time ratio denotes the cost saving when manufacturing time to assemble is increased from τ_{ij} to $\tau_{i,j+1}$.

Lemma 4.1.1. If total manufacturing time for each component is set to its lowest level (that is a base process sequence: $x_{11} = x_{21} = \dots = x_{n1} = 1$ and other x_{ij} variables are zero) and the total tolerance of dimensions does not violate the total assembly tolerance constraint, it is an efficient solution with respect to the total manufacturing time criterion.

Proof: Since we assign the manufacturing time of each dimension to its lowest level which is the shortest manufacturing time $x_{11}, x_{21}, \ldots x_{n1}$, it is impossible to obtain less manufacturing time than that of the base process sequence even though it requires maximum manufacturing cost. Since the total assembly tolerance is satisfied by the corresponding dimension tolerances, it is an efficient solution.

Consider a cost-tolerance-time model with discrete cost, tolerance and time points that is presented as model 1 in the preceding chapter. Since it would be desirable to report as many possible combinations of manufacturing cost and manufacturing time to assemble to a decision maker when total assembly tolerance is bounded, an algorithm generating all efficient solutions, if possible, would be more desirable. Therefore, a simple algorithm is presented to obtain efficient solutions. In order to generate efficient solutions for decision makers, when the assumption that $R_{ij} \geq R_{i,j+1}$ is relaxed, we will be needed to define R_{ij}^{m} to be $(c_{i,j+m} - c_{i,j}) / (\tau_{i,j} - \tau_{i,j+m})$, where $m = 1,2,\ldots,(n_i - j)$ and the following algorithm is constructed.

4.1.1 Algorithm 1

To obtain as many efficient solutions as possible, the following heuristic algorithm was applied that showed the efficient solutions with the tolerance condition.

Step 1: Read N, T, n_i, c_{ij}, $\tau_{i,j}$ and t_{ij} for $i = 1,2,\ldots N$ and for $j = 1,2,\ldots n_i$.

Step 2: Set the manufacturing time to assemble for each component to the lowest level (base process sequence). Store base process sequence and corresponding solution.

Step 3: Compute cost-time ratios R_{ij}^{m}, where $m = 1,2,\ldots,(n_i - j)$ and find the maximum cost-time ratio from the candidate set (if $x_{ij} = 1$, only R_{ij}^{m} are in the candidate set.). If there are ties, branch out from all solutions for further search of efficient solutions.

Step 4: Check the total assembly tolerance condition with all solutions in the tie solution set. If it is satisfied, store the solution (Note that if R_{pq}^{s} is chosen in Step 3, then the new solution is the same as the previous one except that $x_{pq} = 0$ and $x_{p,q+s} = 1$). Otherwise, delete R_{pq}^{s} from the candidate set.

Lemma 4.1.2. If the total assembly tolerance constraint is large enough (i.e. T is a large number) and $R_{ij} \geq R_{i,j+1}$ for all i and j, then algorithm 4.1.1. finds exactly $[1+(n_1-1)+(n_2-1)+\cdots+(n_N-1)]$ efficient solutions with respect to manufacturing cost and manufacturing time criteria.

Proof: Because of Lemma 4.1, we can obtain an efficient solution if the manufacturing time to assemble of each component is set to its lowest level. Then, since $R_{ij} \geq R_{i,j+1}$ (i.e., decreasing convex property), for component i, we can generate exact ($n_i - 1$) efficient solutions. Therefore, the sum of all the efficient solutions for all the dimensions provides the desired result.

The flow chart for algorithm 1 is shown in the Figure 4.2. Note that the number of evaluations by algorithm 1 is equal to those in the univariate search method by Fox (1973). To consider the validity of the algorithm developed, the problems which contain cost and tolerance values for each dimension were taken from Chase et. al. (1990) and the time factor was generated with the idea that as manufacturing cost decreases, the manufacturing time increases and as tolerance is tightened, assembly time decreases. For decision maker to determine the most suitable process for manufacturing a product/assembly, he or she should consider the time factor in accordance with the cost and tolerance for releasing a product / assembly on time and surviving in a competitive market.

An example with six different dimensions is taken to illustrate the algorithm 1 (Problem B in Appendix 1). It is a single line multi-level assembly system. Predefined total assembly tolerance is 23 units and it is assumed that the tolerances in this example are bi-lateral. Therefore, the total tolerance of the components 1 through 6 must not exceed the total assembly tolerance of 23 units. The procedure to obtain some of these efficient solutions is illustrated step by step as:

Step 1: $N = 6$. $n_3 = 3$ and all $n_i = 2$. $T = 23$.

Step 2: $x_{11} = x_{21} = \ldots = x_{61} = 1$ and all other $x_{ij} = 0$.

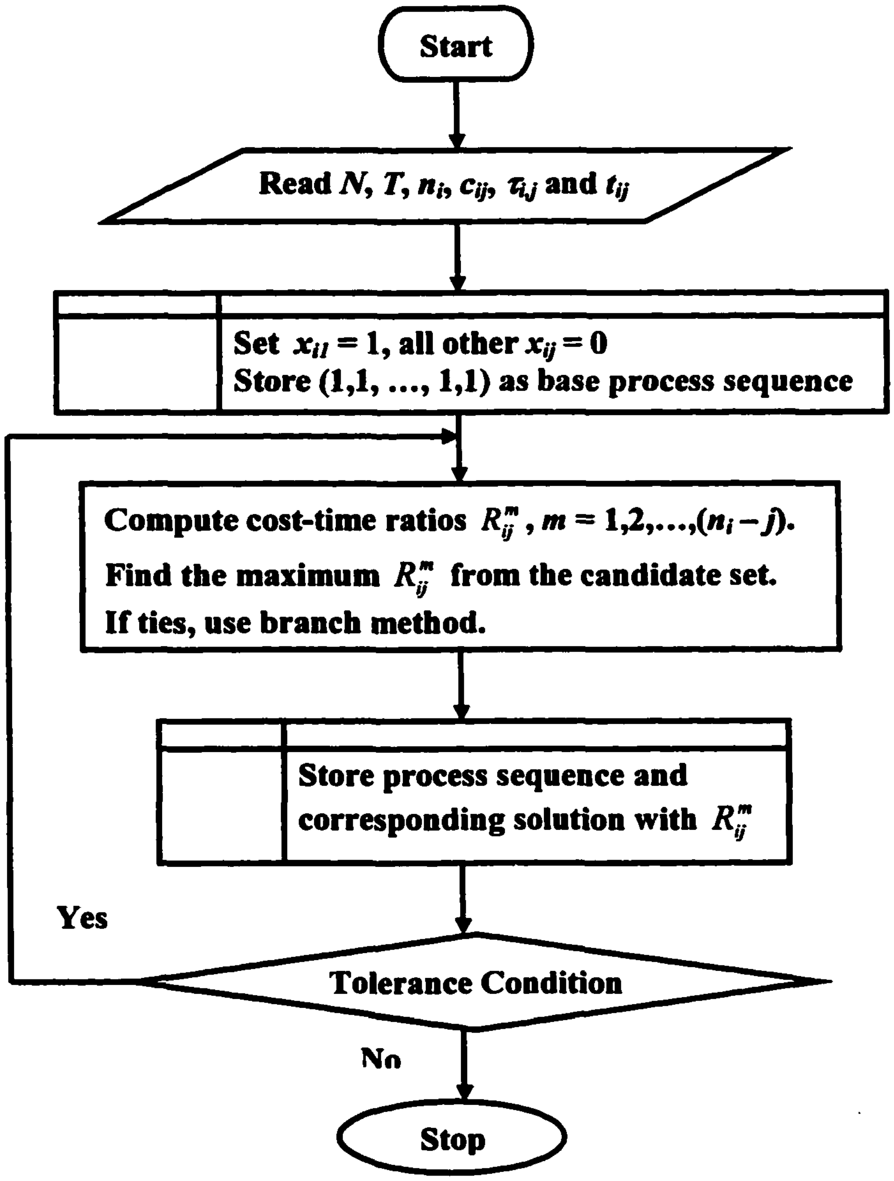


Figure 4.2 Flow Chart for Algorithm 1

It is a base process sequence (1,1,1,1,1,1).

$t_{11} + t_{21} + t_{31} + t_{41} + t_{51} + t_{61} = (3 + 2 + 2 + 3 + 3 + 2) = 15.$

$c_{11} + c_{21} + c_{31} + c_{41} + c_{51} + c_{61} = (9 + 6 + 8 + 7 + 8 + 9) = 47.$

$\tau_{11} + \tau_{21} + \tau_{31} + \tau_{41} + \tau_{51} + \tau_{61} = (5 + 3 + 3 + 1 + 2 + 3) = 17.$

Store solution (47, 17).

Step 3: R_{11} = 1.0, R_{21} = 1.0, R_{31}^1 = 2.0, R_{31}^2 = 1.2, R_{41} = 1.0, R_{51} = 1.67, R_{61} = 2.0. The maximum R_{ij} = R_{31}^1 = R_{61} = 2.0. Since we have two tie solutions, search both solutions (Tie solution set has R_{31}^1 and R_{61}).

Step 4: From the R_{31}^1 in the tie solution set,

$t_{11} + t_{21} + t_{32} + t_{41} + t_{51} + t_{61} = (3 + 2 + 4 + 3 + 3 + 2) = 17.$

$c_{11} + c_{21} + c_{32} + c_{41} + c_{51} + c_{61} = (9 + 6 + 4 + 7 + 8 + 9) = 43.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{51} + \tau_{61} = (5 + 3 + 5 + 1 + 2 + 3) = 19.$

Store process sequence (1,1,2,1,1,1) and corresponding solution (43, 19).

From the R_{61} in the tie solution set,

$t_{11} + t_{21} + t_{31} + t_{41} + t_{51} + t_{62} = (3 + 2 + 2 + 3 + 3 + 3) = 16.$

$c_{11} + c_{21} + c_{31} + c_{41} + c_{51} + c_{62} = (9 + 6 + 8 + 7 + 8 + 7) = 45.$

$\tau_{11} + \tau_{21} + \tau_{31} + \tau_{41} + \tau_{51} + \tau_{62} = (5 + 3 + 3 + 1 + 2 + 4) = 18.$

Store process sequence (1,1,1,1,1,2) and corresponding solution (45, 18).

Step 5: Since the solutions obtained in Step 3 do not violate the total assembly tolerance constraint, go to Step 3.

Step 3: From the process sequence (1,1,2,1,1,1), R_{11} = 1.0, R_{21} = 1.0, R_{32} = 0.67, R_{41} = 1.0, R_{51} = 1.67, R_{61} = 2.0. The maximum R_{ij} = R_{61} = 2.0. From the process sequence (1,1,1,1,1,2), R_{11} = 1.0, R_{21} = 1.0, R_{31}^1 = 2.0, R_{31}^2 = 1.2, R_{41} = 1.0, R_{51} = 1.67. The maximum R_{ij} = R_{31}^1 = 2.0.

Step 4: The only process sequence generated in Step 3 is the process sequence (1,1,2,1,1,2).

$t_{11} + t_{21} + t_{32} + t_{41} + t_{51} + t_{62} = (3 + 2 + 4 + 3 + 3 + 3) = 18.$

$c_{11} + c_{21} + c_{32} + c_{41} + c_{51} + c_{62} = (9 + 6 + 4 + 7 + 8 + 7) = 41.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{51} + \tau_{62} = (5 + 3 + 5 + 1 + 2 + 4) = 20.$

Store process sequence (1,1,2,1,1,2) and corresponding solution (41, 20).

Step 5: Since the solution does not violate the total assembly tolerance constraint, go to Step 3.

Step 3: From the process sequence (1,1,2,1,1,2), $R_{11} = 1.0$, $R_{21} = 1.0$, $R_{32} = 0.67$, $R_{41} = 1.0$, $R_{51} = 1.67$. The maximum $R_{ij} = R_{51} = 1.67$.

Step 4: The solution obtained in Step 3 is from a process sequence (1,1,2,1,2,2,).

$t_{11} + t_{21} + t_{32} + t_{41} + t_{52} + t_{62} = (3 + 2 + 4 + 3 + 8 + 3) = 23.$

$c_{11} + c_{21} + c_{32} + c_{41} + c_{52} + c_{62} = (9 + 6 + 4 + 7 + 3 + 7) = 36.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{52} + \tau_{62} = (5 + 3 + 5 + 1 + 5 + 4) = 23.$

Store process sequence (1,1,2,1,2,2) and corresponding solution (36, 23).

Step 5: Since the solution does not violate the total assembly tolerance constraint, go to Step 3. From the process sequence (1,1,2,1,2,2), $R_{11} = 1.0$, $R_{21} = 1.0$, $R_{32} = 0.67$, $R_{41} = 1.0$. The maximum $R_{ij} = R_{11} = R_{21} = R_{41} = 1.0$.

Step 4: The solution obtained in Step 3 is from process sequences (2,1,2,1,2,2,), and (1,1,2,2,2,2). However, both solutions violate the total assembly tolerance constraint.

Step 5: Since the solution search from tie solution set is completed and the solution with maximum cost-time ratio violates the total assembly tolerance constraint, stop.

The five efficient solutions obtained from the proposed algorithm are expressed in Fig. 4.5 a) and the values of cost and time are (47,17), (45,18), (43,19), (41,20), and (36,23) from process sequences (1,1,1,1,1,1), (1,1,1,1,1,2), (1,1,2,1,1,1), (1,1,2,1,1,2), and (1,1,2,1,2,2). By an explicit enumeration search, there are seven efficient solutions such as (47,17), (45,18), (43,19), (41,20), (40,21), (38,22), and (36,23) from process sequences (1,1,1,1,1,1), (1,1,1,1,1,2), (1,1,2,1,1,1), (1,1,2,1,1,2), (1,1,1,1,2,2), (1,1,2,1,2,1), and (1,1,2,1,2,2). Therefore, 71 percent of the efficient solutions are obtained by the algorithm 1. With a full exhaustive search, we need to generate 96 ($2^5 \times 3$) solutions but only five solutions are investigated with the proposed algorithm. Also, consider the two solutions not generated by the algorithm 1. These two solutions, (40, 21), and (38, 22), could be dominated by the linear combination of the solutions, (41, 20) and (36, 23).

Since there is an emphasis on obtaining the optimization of discrete cases, it is important to generate as many efficient solutions as possible for decision maker. Therefore, it is needed to develop another algorithm to satisfy the requirement.

4.1.2 Algorithm 2.

Now consider the solutions (40, 21) and (38, 22) obtained from process sequences (1,1,1,1,2,2) and (1,1,2,1,2,1). When running algorithm 1, processes in dimensions 3, 5, and 6 are investigated from the base process sequence (1,1,1,1,1,1). Note that the combination of processes in dimensions 3, 5, and 6 can generate solutions (40, 21) and (38, 22). By using the history when running algorithm 1, the following partial enumeration algorithm can be constructed.

Step 1: Run algorithm 1. Store the processes changed in each dimension from the base solution (1,1,...,1,1).

Step 2: Employ the partial enumeration search based on the processes selected for efficient solutions in running algorithm 1. Then, find the nondominated solutions. The flow chart is presented in the Figure 4.3.

Reconsider the same example applied in algorithm 1. Algorithm 2 can be illustrated step-by-step as follows:

Step 1: After running algorithm 1, the processes in dimensions 3, 5, and 6 are selected for efficient solutions.

Step 2: Processes in dimensions 1, 2, and 4 are fixed ($x_{11} = x_{21} = x_{41} = 1$). Processes in dimensions 3, 5, and 6 have 2 alternatives, respectively. The process sequences and corresponding solutions generated by partial enumeration search are as follows:

(1,1,1,1,1,1) = (47, 17)

(1,1,1,1,1,2) = (45, 18)

(1,1,1,1,2,1) = (42, 20)

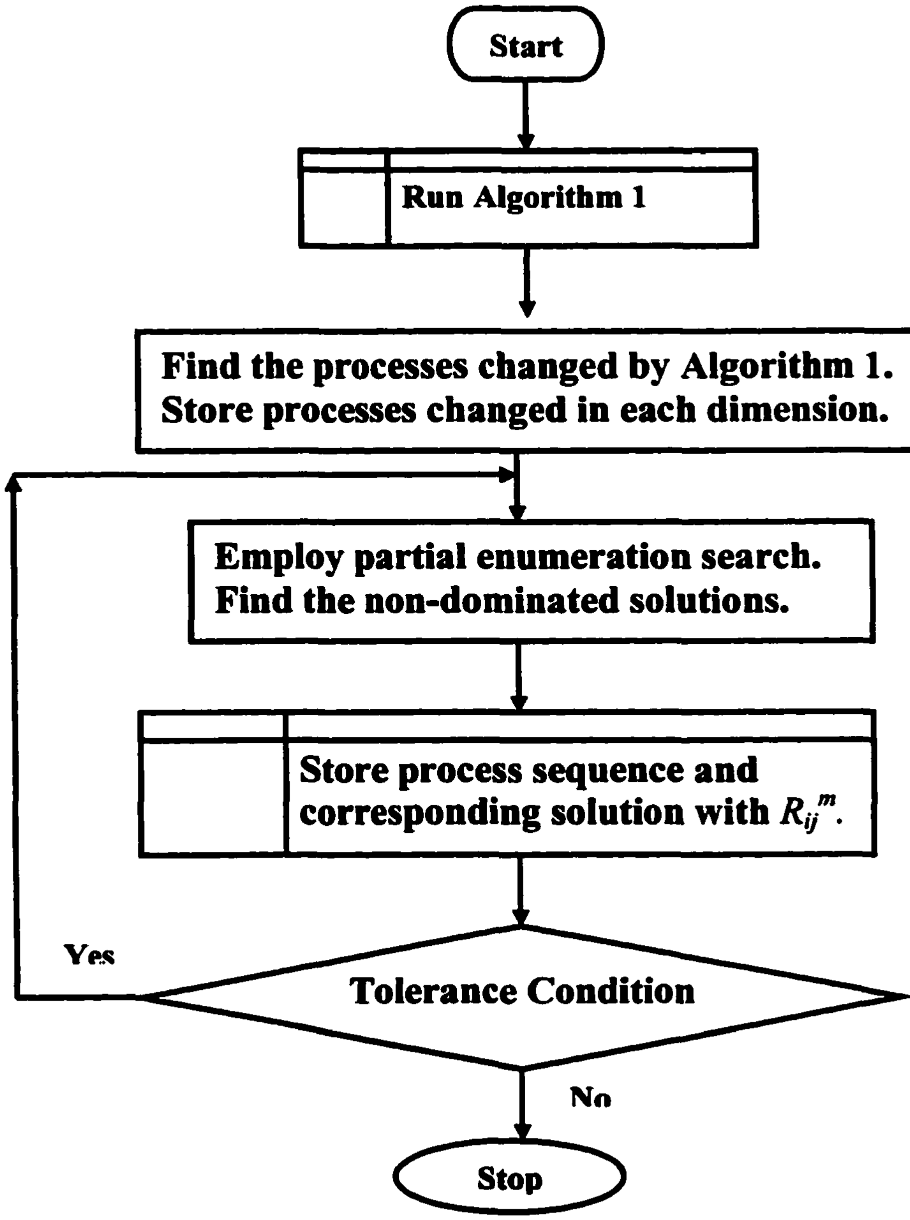


Figure 4.3 Flow Chart for Algorithm 2

$(1,1,1,1,2,2) = (40, 21)$

$(1,1,2,1,1,1) = (43, 19)$

$(1,1,2,1,1,2) = (41, 20)$

$(1,1,2,1,2,1) = (38, 22)$

$(1,1,2,1,2,2) = (36, 23)$

Since solution (42, 20) is dominated by (41, 20), there are seven nondominated solutions that are the efficient solutions generated. In algorithm 2., eight solutions (five solutions generated in step 1 are reproduced in step 2) are investigated. It found all seven efficient solutions. Compared with the explicit enumeration search, only seven percent of the solutions are searched. The efficient solutions for the same Problem B is presented in the Figure 4.5. b) as the relationship between cost and time.

If the total assembly tolerance constraint is large enough (i.e., T is a large value), then algorithm 1 searches all possible slopes from the candidate set. In this case, the partial enumeration in algorithm 2 presents too much computational burden. Now consider Problem C in the Appendix 1. The solution with the largest total assembly tolerance (T = 38) is from the process sequence (3,2,2,2,2,2,2) and it does not violate the total assembly tolerance constraint. Therefore, the partial enumeration in algorithm 2 is the same as the explicit enumeration.

4.1.3 Algorithm 3

In algorithm, the ratio based on the manufacturing cost difference and the manufacturing time difference is considered. From the base process sequence (1,1, ... ,1,1), the solution with the largest cost-time ratio is picked for the next solution. The idea of algorithm 3 is to search the solution with the minimum increase of manufacturing time or the minimum decrease of manufacturing cost. From the solutions in the solution set obtained in previous iterations, algorithm 3 tries to find the process sequence with the smallest increase of manufacturing time or the smallest decrease of manufacturing cost from the new solution just obtained. The sum of manufacturing time difference and manufacturing cost difference is the criterion to search the next solution since the less variation of differences provides more chance to generate efficient solutions. If there is a tie, it is broken based on the cost-time ratio. The algorithm 3 can be stated as in the next section.

Step 1: Sum up the manufacturing time differences and manufacturing cost differences. Pick the standard based on the minimum sum of differences (say manufacturing time is selected as a standard). Start the base process sequence (1,1, ... , 1,1) with the corresponding manufacturing time, manufacturing cost and tolerance as initial solution set.

Step 2: Search solution from the solution set, which generates the smallest increase of manufacturing time comparing the new solution just obtained. If there is a tie, break it based on the cost-time ratio. Compute manufacturing time, manufacturing cost and tolerance of the new solution.

Step 3: If the tolerance of new solution does not violate total assembly tolerance, store the new solution in the solution set and go to Step 2. Otherwise, find non-dominated solutions in the solution set and stop.

The flow chart for the algorithm 3 is presented in the Figure 4.4. It is better to consider the same example (problem B) again for a comparison and discussions. Algorithm 3 can be illustrated step by step as follows.

Step 1: The sum of manufacturing time difference (2+1+2+3+3+3+1=15) is less than the sum of manufacturing cost difference (2+1+4+2+3+5+2=19). Therefore, manufacturing time is the standard. The base process sequence (1,1,1,1,1,1) has (47,17,15) solution tuple.

Step2: The dimensions which provide the smallest increase from the solution set are the second processes in dimension 2 and in dimension 6. Both processes increase the manufacturing time only one unit (from 17 to 18). Because of the cost-time ratio, the process sequence (1,1,1,1,1,2) is selected. Corresponding solution tuple is (45, 18, 16).

Step 3: Since the new solution does not violate total assembly tolerance constraint, store the new solution in the solution set and go to step 2.

Step 2: There are three candidates to generate the smallest manufacturing time 19. From the process sequence (1,1,1,1,1,1) in the solution set, selecting the second processes in dimension 1 and in dimension 3 does increase manufacturing time from 17 to 19. Another candidate is the second process in dimension 2 from the

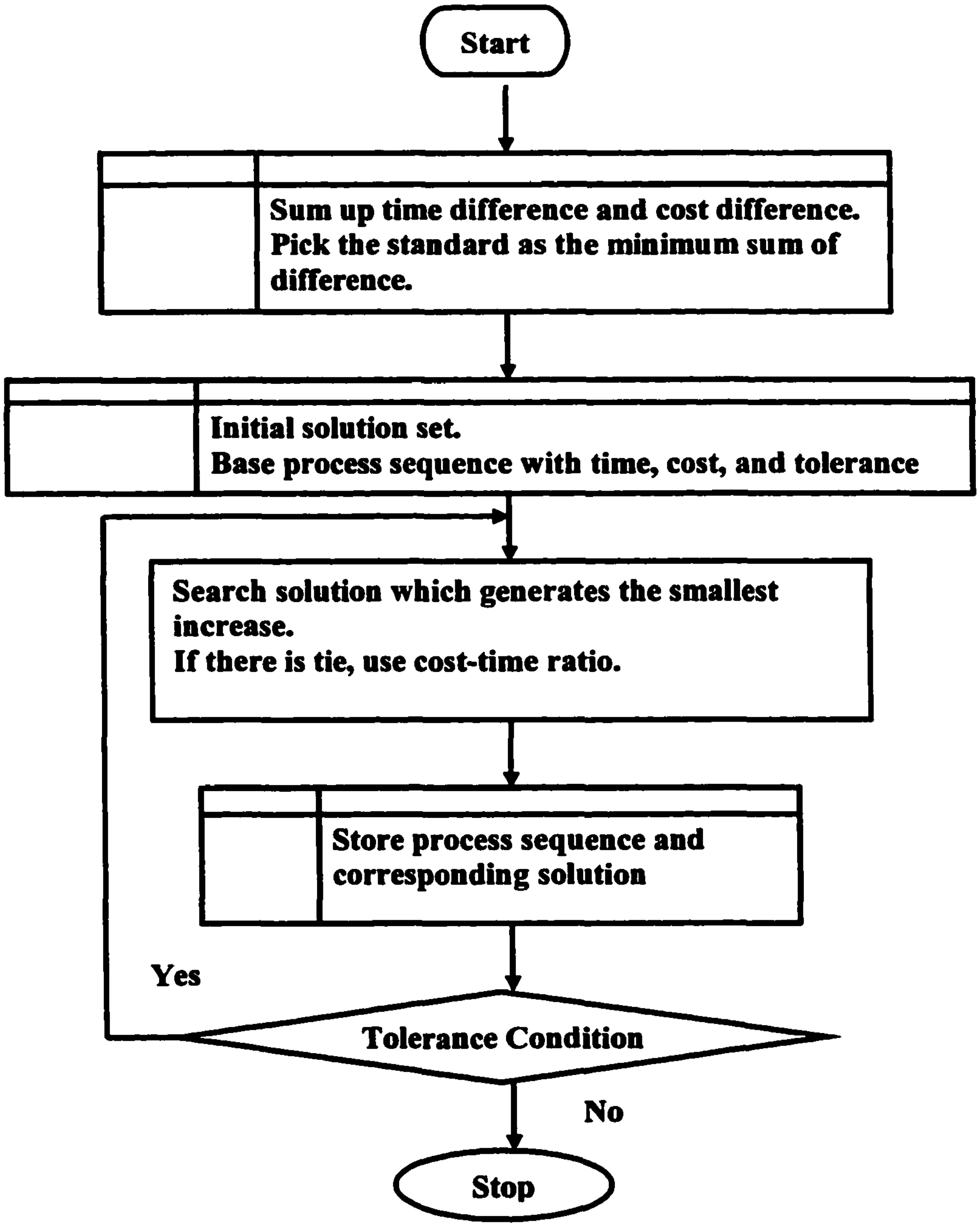


Figure 4.4. Flow Chart for Algorithm 3

process sequence (1,1,1,1,1,2). It increases manufacturing time from 18 to 19. The cost-time ratio enforces to choose the process sequence (1,1,2,1,1,1) whose solution tuple is (43, 19, 17).

Step 3: Because the new solution satisfies the total assembly constraint, store it.

Step 2: There are six possible candidates. Two candidates from process sequence (1,1,1,1,1,1), two from (1,1,1,1,1,2) and two from (1,1,2,1,1,1). The manufacturing times increases 3 units, 2 units and 1 unit from 17 to 20, from 18 to 20 and from 19 to 20, respectively. The process sequence (1,1,2,1,1,2) has the minimum cost-time ratio and it has (41, 20, 18) solution tuple.

Step 3: Store new solution in the solution set and go to Step 2.

Step 2: From the three process sequences in the solution set, four possible candidates can be considered. The selected process sequence with the minimum cost-time ratio, 1.67, is (1,1,1,1,2,2) from (1,1,1,1,1,2). The solution tuple is (40, 20, 21).

Step 3: Since it satisfies the total assembly tolerance constraint, keep it.

Step 2: There are five candidates with manufacturing time 22. Because of the cost-time ratio, the process sequence (1,1,2,1,2,1) is chosen from the (1,1,2,1,1,1). The new solution has (38, 22, 22) tuple.

Step 3: Store the solution in the solution set.

Step 2: There are five possible candidates; two candidates from process sequence (1,1,2,1,1,2), one from (1,1,1,1,2,2) and two from (1,1,2,1,2,1). The process sequence (1,1,2,1,2,2) whose solution tuple is (36, 23, 23), can be chosen.

Step 3: Store the solution. After then, the total assembly tolerance constraint can not be satisfied since the tolerance of new solution is the same as given total assembly tolerance. The nondominated process sequences in the solution set are as follows:

$(1,1,1,1,1,1) = (47, 17, 15)$

$(1,1,1,1,1,2) = (45, 18, 16)$

$(1,1,2,1,1,1) = (43, 19, 17)$

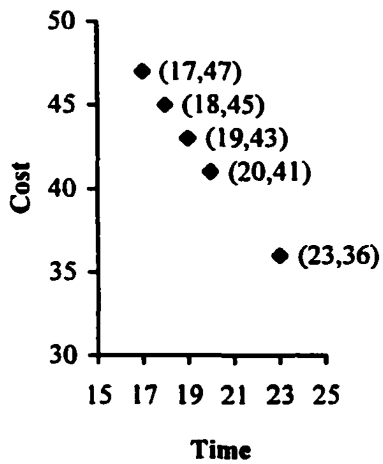
$(1,1,2,1,1,2) = (41, 20, 18)$

$(1,1,1,1,2,2) = (40, 21, 21)$

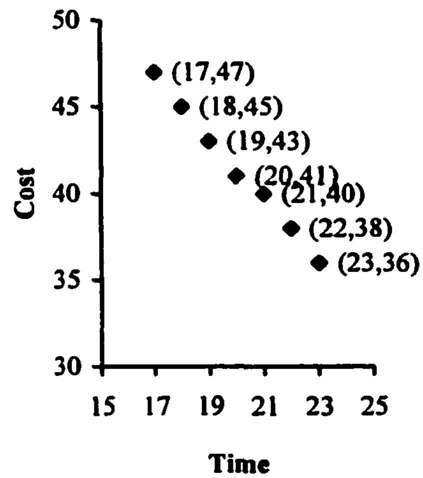
$(1,1,2,1,2,1) = (38, 22, 22)$

$(1,1,2,1,2,2) = (36, 23, 23)$

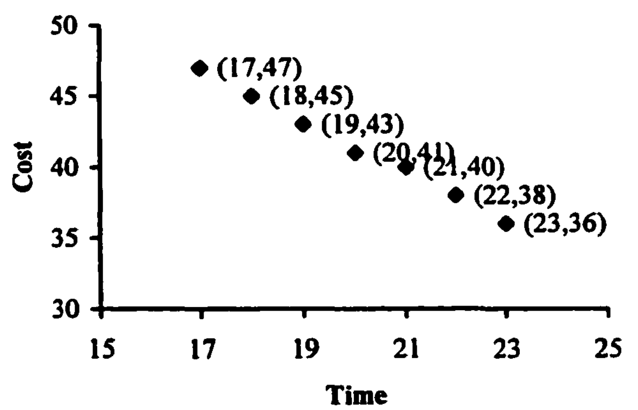
All seven efficient solutions are found after searching 7 full evaluations and 13 cost-time evaluations, and are presented in the Figure 4.5.c).



a) Algorithm 1



b) Algorithm 2



c) Algorithm 3

Figure 4.5 Cost-Time Relation for Problem B

4.2. Cost-Tolerance Model (Model 2)

With tighter tolerances of each process to satisfy the requirements of customers, total manufacturing cost will be increased. The function to explain the relationship between cost and tolerance shows an exponential decreasing nature in literature. As the assembly tolerance of dimensions increases, the total manufacturing cost decreases. Therefore, the decision maker should consider the trade-offs between manufacturing cost and assembly tolerance with total manufacturing time constraint.

In model 2, total manufacturing time to assembly Ω is fixed as a constraint, and the minimization of total manufacturing cost and minimization of total assembly tolerance are set as the objective functions. Mathematical formulation for the model 2 is expressed as follows from the preceding chapter:

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} c_{ij}x_{ij} \tag{3.20}$$

$$Min \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} t_{ij}x_{ij} \tag{3.21}$$

$$s.t \quad \sum_{i=1}^{N}\sum_{j=1}^{n_i} \tau_{ij}x_{ij} \le \Omega \tag{3.22}$$

$$\sum_{j=1}^{n_i} x_{ij} = 1 \quad for \quad i = 1,2,\ldots,N \tag{3.23}$$

$$x_{ij} = \{0,1\} \tag{3.24}$$

Given the total manufacturing time to assembly, Ω, sort n_i points to an increasing order of manufacturing time to assembles (i.e., τ_{i1}, τ_{i2}, ..., τ_{in_i}) as well as assembly tolerances (i.e., t_{i1}, t_{i2}, ..., t_{in_i}) and decreasing order of manufacturing costs (i.e., c_{i1}, c_{i2}, ..., c_{in_i}). Here, the cost-tolerance ratio R_{ij} for component i is defined as $(c_{ij} - c_{i,j+1}) / (t_{i,j+1} - t_{ij})$, where $i = 1,2,...,N$ and $j = 1,2,...,n_i$. That is, the cost-tolerance ratio denotes the cost saving when assembly tolerance is increased from t_{ij} to $t_{i,j+1}$.

Lemma 4.2.1. If the tolerance for each component is set to its lowest level (base process sequence: $x_{11} = x_{21} = ... = x_{n1} = 1$ and other x_{ij} variables are zero) and the assembly manufacturing time to assemble of dimensions does not violate the total assembly time constraint, it is an efficient solution with respect to the assembly tolerance criterion.

Proof: Since we assign the manufacturing tolerance of each dimension to its lowest level, it is impossible to obtain less manufacturing tolerance even though it requires maximum costs. Since the total manufacturing time to assembly is satisfied by the corresponding time, it is an efficient solution.

Similar to the development of algorithm 1 for Model 1, a simple algorithm could be presented to obtain efficient solutions in this case also. In order to generate efficient solutions for decision makers, when the assumption $R_{ij} \geq R_{i,j+1}$ is relaxed, we will need to define R_{ij}^{m} as $(c_{i,j+m} - c_{i,j}) / (t_{i,j} - t_{i,j+m})$, where $m = 1,2,\ldots,(n_i - j)$ and the following algorithm constructed.

4.2.1 Algorithm 1

To obtain as many efficient solutions as possible, the following algorithm was applied and it showed the efficient solutions with the time condition.

Step 1: Read N, Ω, n_i, c_{ij}, $\tau_{i,j}$ and t_{ij} for $i = 1,2,\ldots N$ and for $j = 1,2,\ldots n_i$.

Step 2: Set the assembly tolerance for each component to the lowest level (base process sequence). Store base process sequence and corresponding solution.

Step 3: Compute cost-tolerance ratios R_{ij}^{m}, where $m = 1,2,\ldots,(n_i - j)$ and find the maximum cost-tolerance ratio from the candidate set (if $x_{ij} = 1$, only R_{ij}^{m} are in the candidate set.). If there are ties, branch out from all solutions for further search of efficient solutions.

Step 4: Check the total manufacturing time to assembly condition with all solutions in the tie solution set. If it is satisfied, store the solution (Note that if R_{pq}^{s} is chosen in Step 3, then the new solution is the same as the previous one except that $x_{pq} = 0$ and $x_{p,q+s} = 1$). Otherwise, delete R_{pq}^{s} from the candidate set.

Lemma 4.2.2. If the total manufacturing time to assembly constraint is large enough (i.e. Ω is a large number) and $R_{ij} \geq R_{i,j+1}$ for all i and j, then algorithm 4.2.1. finds exactly $[1+(n_1-1)+(n_2-1)+\cdots+(n_N-1)]$ efficient solutions with respect to manufacturing cost and assembly tolerance criteria.

Proof: Because of Lemma 4.2.1, we can obtain an efficient solution if the assembly time of each component is set to its lowest level. Then, since $R_{ij} \geq R_{i,j+1}$ (i.e., decreasing convex property), for component i, we can generate exact $(n_i - 1)$ efficient solutions. Therefore, the sum of all the efficient solutions for all the dimensions provides the desired result.

To compare the results of Model 1 and Model 2, the same example (Problem B) is taken to illustrate the algorithm 1 of Model 2. It is a single line multi-level assembly system with six different dimensions. To ensure manufacturing within the assembly tolerance, total time to assemble is predefined as 23 units. Therefore, the total manufacturing time to assemble of the components 1 through 6 must not exceed the total time of 23 units. By the algorithm 1, the procedure is illustrated step by step.

Step 1: $N = 6$. $n_3 = 3$ and all $n_i = 2$. $\Omega = 23$.

Step 2: $x_{11} = x_{21} = \ldots = x_{61} = 1$ and all other $x_{ij} = 0$.

It is a base process sequence (1,1,1,1,1,1).

$c_{11} + c_{21} + c_{31} + c_{41} + c_{51} + c_{61} = (9 + 6 + 8 + 7 + 8 + 9) = 47.$

$t_{11} + t_{21} + t_{31} + t_{41} + t_{51} + t_{61} = (3 + 2 + 2 + 3 + 3 + 2) = 15.$

$\tau_{11} + \tau_{21} + \tau_{31} + \tau_{41} + \tau_{51} + \tau_{61} = (5 + 3 + 3 + 1 + 2 + 3) = 17.$

Store solution (47, 15).

Step 3: $R_{11} = 1.0$, $R_{21} = 0.5$, $R_{31}^1 = 2.0$, $R_{31}^2 = 1.5$, $R_{41} = 0.75$, $R_{51} = 1.0$, $R_{61} = 2.0$. The maximum $R_{ij} = R_{31}^1 = R_{61} = 2.0$. Since we have two tie solutions, search both solutions (Tie solution set has R_{31}^1 and R_{61}).

Step 4: From the R_{31}^1 in the tie solution set,

$c_{11} + c_{21} + c_{32} + c_{41} + c_{51} + c_{61} = (9 + 6 + 4 + 7 + 8 + 9) = 43.$

$t_{11} + t_{21} + t_{32} + t_{41} + t_{51} + t_{61} = (3 + 2 + 4 + 3 + 3 + 2) = 17.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{51} + \tau_{61} = (5 + 3 + 5 + 1 + 2 + 3) = 19.$

Store process sequence (1,1,2,1,1,1) and corresponding solution (43, 17).

From the R_{61} in the tie solution set,

$c_{11} + c_{21} + c_{31} + c_{41} + c_{51} + c_{62} = (9 + 6 + 8 + 7 + 8 + 7) = 45.$

$t_{11} + t_{21} + t_{31} + t_{41} + t_{51} + t_{62} = (3 + 2 + 2 + 3 + 3 + 3) = 16.$

$\tau_{11} + \tau_{21} + \tau_{31} + \tau_{41} + \tau_{51} + \tau_{62} = (5 + 3 + 3 + 1 + 2 + 4) = 18.$

Store process sequence (1,1,1,1,1,2) and corresponding solution (45, 16).

Step 5: Since the solutions obtained in Step 3 do not violate the total manufacturing time to assemble constraint, go to Step 3.

Step 3: From the process sequence (1,1,2,1,1,1), $R_{11} = 1.0$, $R_{21} = 0.5$, $R_{32} = 1.0$, $R_{41} = 0.75$, $R_{51} = 1.0$, $R_{61} = 2.0$. The maximum $R_{ij} = R_{61} = 2.0$. From the process sequence (1,1,1,1,1,2), R_{11} = 1.0, R_{21} = 0.5, R_{31}^{1} = 2.0, R_{31}^{2} = 1.5, R_{41} = 0.75, R_{51} = 1.0. The maximum R_{ij} = R_{31}^{1} = 2.0.

Step 4: The only process sequence generated in Step 3 is the process sequence (1,1,2,1,1,2).

$c_{11} + c_{21} + c_{32} + c_{41} + c_{51} + c_{62} = (9 + 6 + 4 + 7 + 8 + 7) = 41.$

$t_{11} + t_{21} + t_{32} + t_{41} + t_{51} + t_{62} = (3 + 2 + 4 + 3 + 3 + 3) = 18.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{51} + \tau_{62} = (5 + 3 + 5 + 1 + 2 + 4) = 20.$

Store process sequence (1,1,2,1,1,2) and corresponding solution (41, 18).

Step 5: Since the solution does not violate the total manufacturing time to assemble constraint, go to Step 3.

Step 3: From the process sequence (1,1,2,1,1,2), R_{11} = 1.0, R_{21} = 0.5, R_{32} = 1.0, R_{41} = 0.75, R_{51} = 1.0. Since we have three tie solutions, search every solution (Tie solution set has R_{11}, R_{32}, and R_{51}).

Step 4: From the R_{11} in the tie solution set,

$c_{12} + c_{21} + c_{32} + c_{41} + c_{51} + c_{62} = (7 + 6 + 4 + 7 + 8 + 7) = 39.$

$t_{12} + t_{21} + t_{32} + t_{41} + t_{51} + t_{62} = (5 + 2 + 4 + 3 + 3 + 3) = 20.$

$\tau_{12} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{51} + \tau_{62} = (7 + 3 + 5 + 1 + 2 + 4) = 22.$

Store process sequence (2,1,2,1,1,2) and corresponding solution (39,20).

From the R_{32} in the tie solution set,

$c_{11} + c_{21} + c_{33} + c_{41} + c_{51} + c_{62} = (9 + 6 + 6 + 7 + 8 + 7) = 39.$

$t_{11} + t_{21} + t_{33} + t_{41} + t_{51} + t_{62} = (3 + 2 + 2 + 3 + 3 + 3) = 20.$

$\tau_{11} + \tau_{21} + \tau_{33} + \tau_{41} + \tau_{51} + \tau_{62} = (5 + 3 + 8 + 1 + 2 + 4) = 23.$

But this solution (39,20) with process sequence (1,1,3,1,1,2) is dominated by the solution (39,20) with process sequence (2,1,2,1,1,2), so this solution is not stored as an efficient solution.

As the final tie solution R_{51}, we have

$c_{11} + c_{21} + c_{32} + c_{41} + c_{52} + c_{62} = (9 + 6 + 4 + 7 + 3 + 7) = 36.$

$t_{11} + t_{21} + t_{32} + t_{41} + t_{52} + t_{62} = (3 + 2 + 4 + 3 + 8 + 3) = 23.$

$\tau_{11} + \tau_{21} + \tau_{32} + \tau_{41} + \tau_{52} + \tau_{62} = (5 + 3 + 5 + 1 + 5 + 4) = 23.$

Store process sequence (1,1,2,1,2,2) and corresponding solution (36, 23).

Step 5: Since the solution does not violate the total assembly tolerance constraint, go to Step 3. From the process sequence (2,1,2,1,1,2), $R_{21} = 0.5$, $R_{41} = 0.75$. The maximum $R_{ij} = R_{41} = 0.75$

Step 4: The solution obtained in Step 3 is from a process sequence (2,1,2,2,1,2,), and (1,1,2,2,2,2). However, both solutions violate the total assembly time constraint.

Step 5: Since the solution search from tie solution set is completed and the solution with maximum cost-tolerance ratio violates the total manufacturing time to assemble constraint, stop.

The six efficient solutions obtained from the algorithm 1 are expressed in Fig. 4.6. a) and the values of cost and tolerance are (47,15), (45,16), (43,17), (41,18), (39,20), and (36,23) from process sequences (1,1,1,1,1,1), (1,1,1,1,1,2), (1,1,2,1,1,1), (1,1,2,1,1,2), (2,1,2,1,1,2), and (1,1,2,1,2,2). By an explicit enumeration search, there are nine efficient solutions such as (47,15), (45,16), (44,19), (43,17), (41,18), (40, 21), (39,20), (38,22) and (36,23) from process sequences (1,1,1,1,1,1), (1,1,1,1,1,2), (1,1,1,2,1,1), (1,1,2,1,1,1), (1,1,2,1,1,2), (1,1,1,1,2,2), (2,1,2,1,1,2), (1,1,2,1,2,1), and (1,1,2,1,2,2). Therefore, 66.67 percent of the efficient solutions are obtained by the algorithm 1. To obtain the more efficient solutions for decision making, algorithm 2 is also applied to the same example as used with algorithm 1.

4.2.2. Algorithm 2

Now consider the solutions (40, 21) and (38, 22) obtained from process sequences (1,1,1,1,2,2) and (1,1,2,1,2,1). When running algorithm 1, processes in dimensions 1, 3, 5, and 6 are investigated from the base process sequence (1,1,1,1,1,1). Note that the combination of processes in dimensions 1, 3, 5, and 6 can generate solutions (40, 21) and (38, 22). By using the history when running algorithm 1, the following partial enumeration algorithm can be constructed.

Step 1: Run algorithm 1. Store the processes changed in each dimension from the base solution (1,1,...,1,1).

Step 2: Employ the partial enumeration search based on the processes selected for efficient solutions in running algorithm 1. Then, find the nondominated solutions.

Reconsider the same example applied in algorithm 1. Algorithm 2 can be illustrated step-by-step as follows:

Step 1: After running algorithm 1, the processes in dimension 1, 3, 5, and 6 are selected for efficient solutions.

Step 2: Processes in dimensions 2, and 4 are fixed ($x_{21} = x_{41} = 1$). Processes in dimensions 1, 3, 5, and 6 have 2 alternatives, respectively. The process sequences and corresponding solutions generated by partial enumeration search are as follows:

(1,1,1,1,1,1) = (47, 15)

(1,1,1,1,1,2) = (45, 16)

(1,1,2,1,1,1) = (43, 17)

(1,1,2,1,1,2) = (41, 28)

(1,1,1,1,2,2) = (40, 21)

(2,1,2,1,1,2) = (39, 20)

(1,1,2,1,2,1) = (38, 22)

(1,1,2,1,2,2) = (36, 23)

In algorithm 2, eight solutions (six solutions generated in step 1 are reproduced in step 2) are investigated. There are 9 efficient solutions by an explicit enumeration search, and algorithm 2 was applied to find 88.89 percent of efficient solutions. The efficient solutions for the same Problem B is presented in the Figure 4.6. b) with the relationship between cost and tolerance.

There is still one more efficient solution not obtained by algorithm 1 and 2 and the solution is (44,19) from process sequence of (1,1,1,2,1,1). Because algorithm 2 searched the efficient solution according to the partial enumeration of selected process that is identified by algorithm 1, the efficient solution including fixed process 4 in algorithm 1 is not considered. By applying the idea of algorithm 3, every possible efficient solution might be searched.

4.2.3 Algorithm 3

The idea of algorithm 3 in Model 2 is to search the solution with the minimum increase of assembly tolerance or the minimum decrease of manufacturing cost. From the solutions in the solution set obtained in previous iterations, algorithm 3 tries to find the process sequence with the smallest increase of assembly tolerance or the smallest decrease of manufacturing cost from the new solution just obtained. The sum of assembly tolerance difference and manufacturing cost difference is the criterion to search the next solution since the less variation of differences provides more chance to generate efficient solutions. If there is a tie, it is broken based on the cost-tolerance ratio. Once again the steps of algorithm 3 is set forth as follows:

Step 1: Sum up the assembly tolerance differences and manufacturing cost differences. Pick the standard based on the minimum sum of differences. Start the base process sequence (1,1, ... , 1,1) with the corresponding assembly tolerance, manufacturing cost and time as initial solution set.

Step 2: Search solution from the solution set, which generates the smallest increase of assembly tolerance comparing the new solution just obtained. If there is a tie, break it based on the cost-tolerance ratio. Compute manufacturing time, manufacturing cost and tolerance of the new solution.

Step 3: If the time of new solution does not violate total manufacturing time constraint, store the new solution in the solution set and go to Step 2. Otherwise, find non-dominated solutions in the solution set and stop.

For finding the efficient solution that is not obtained by algorithm 1 and 2, Problem B is chosen again. Algorithm 3 can be illustrated step by step as follows.

Step 1: The sum of assembly tolerance difference (2+2+2+2+4+5+1=81) is less than the sum of manufacturing cost difference (2+1+4+2+3+5+2=19). Therefore, assembly tolerance is the standard. The base process sequence (1,1,1,1,1,1) has (47,17,15) solution tuple.

Step2: The dimension that provides the smallest increase from the solution set is the second process in dimension 6. The process increases the assembly tolerance only one unit (from 15 to 16). Corresponding solution tuple is (45, 18, 16).

Step 3: Since the new solution does not violate total manufacturing time to assemble constraint, store the new solution in the solution set and go to step 2.

Step 2: There are three candidates to generate the smallest assembly tolerance 17. From the process sequence (1,1,1,1,1,1) in the solution set, selecting the second processes in dimension 1, 2, and 3 does increase manufacturing time from 15 to 17. The cost-tolerance ratio forces us to choose the process sequence (1,1,2,1,1,1) whose solution tuple is (43, 19, 17).

Step 3: Because the new solution satisfies the total manufacturing time to assemble constraint, store it.

Step 2: There are four possible candidates. Three candidates from process sequence (1,1,1,1,1,2), one from (1,1,2,1,1,1). The assembly tolerance increases 2 units and 1 unit from 16 to 18, and from 17 to 18, respectively. The process sequence (1,1,2,1,1,2) has the minimum cost-time ratio and it has (41, 20, 18) solution tuple.

Step 3: Store new solution in the solution set and go to Step 2.

Step 2: There is only one candidate with assembly tolerance 19. From the process sequences (1,1,1,1,1,1), the assembly tolerance increases 4 units from 15 to 19. The process sequence with the solution (44,19) is (1,1,1,2,1,1). The solution tuple is a (44,20,19). This solution is the one that could not be obtained by algorithm 1 and 2.

Step 3: Since it satisfies the total manufacturing time constraint, keep it.

Step 2: There are four candidates with assembly tolerance 20. Because of the cost-tolerance ratio, the process sequence (2,1,2,1,1,2) is chosen from (1,1,2,1,1,2). The new solution has (39,22,20) tuple.

Step 3: Store the solution in the solution set.

Step 2: There are five possible candidates; one candidate from process sequence (1,1,1,1,1,2), one from (1,1,2,1,1,1) and three from (1,1,1,2,1,1). The process sequence (1,1,1,1,2,2) whose solution tuple is (40,21,21), can be chosen.

Step 3: Since it satisfies the total manufacturing time to assemble constraint, store it.

Step 2: There is one candidate with assembly tolerance 22 from the process sequence (1,1,2,1,1,1). The second process of dimension 5 is chosen to have the new process sequence (1,1,2,1,2,1). The solution tuple is a (44,20,19).

Step 3: Since it satisfies the total manufacturing time to assemble constraint, store it.

Step 2: There is one more candidate with assembly tolerance 23. From the process sequence (1,1,2,1,1,2), the assembly tolerance increases 5 units from 18 to 23. The process sequence is (1,1,1,2,1,1) and the solution tuple is a (44,20,19).

Step 3: Store the solution. After then, the total manufacturing time to assemble constraint can not be satisfied. The nondominated process sequences in the solution set are as follows:

(1,1,1,1,1,1) = (47, 17, 15)

(1,1,1,1,1,2) = (45, 18, 16)

(1,1,1,2,1,1) = (44, 20, 19)

(1,1,2,1,1,1) = (43, 19, 17)

(1,1,2,1,1,2) = (41, 20, 18)

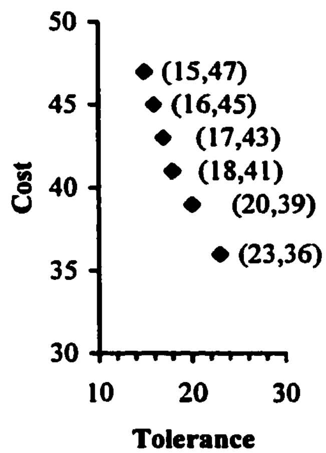
(1,1,1,1,2,2) = (40, 21, 21)

(2,1,2,1,1,2) = (39, 22, 20)

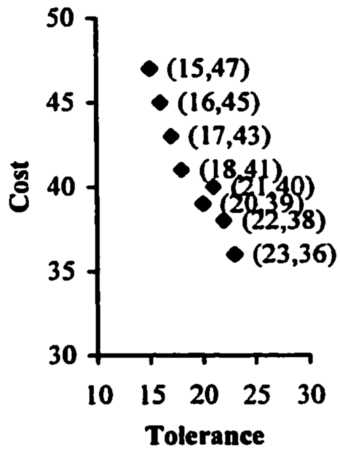
(1,1,2,1,2,1) = (38, 22, 22)

(1,1,2,1,2,2) = (36, 23, 23)

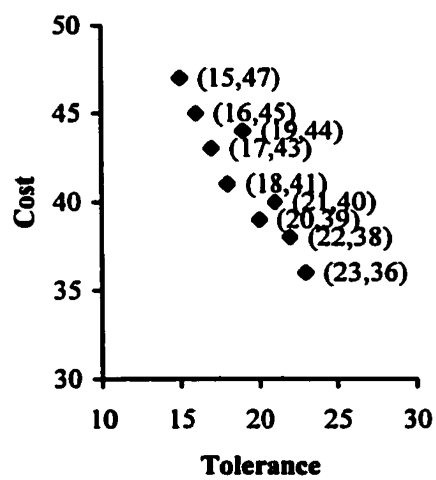
All nine efficient solutions are found after searching 7 full evaluations. The efficient solutions with the relationship between manufacturing cost and tolerance are presented in the Figure 4.6.c).



a) Algorithm 1



b) Algorithm 2



c) Algorithm 3

Figure 4.6 Cost-Tolerance Relation for Problem B

4.3. Tolerance-Time Model (Model 3)

The time-tolerance function indicates that manufacturing time is increasing as assembly tolerance increases. As each dimension has a loose tolerance, assembly tolerance is increased which results in requiring more time to complete. The basic time-tolerance function is indicated in figure 4.7.

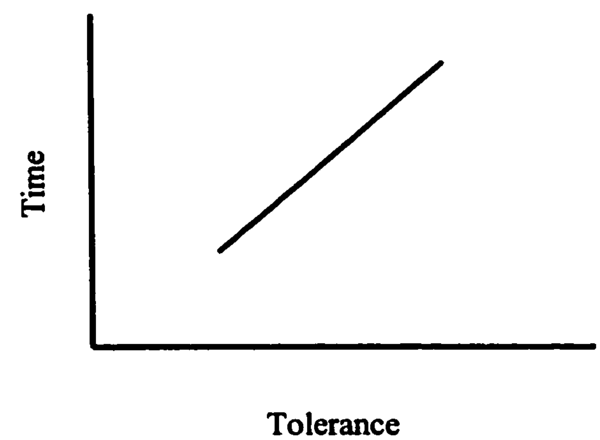
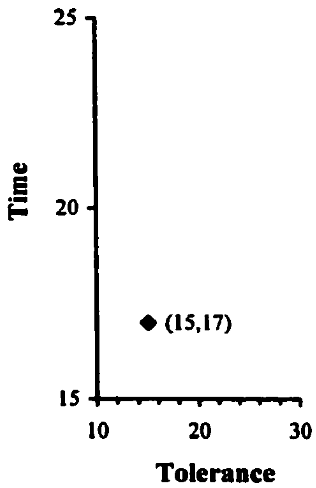
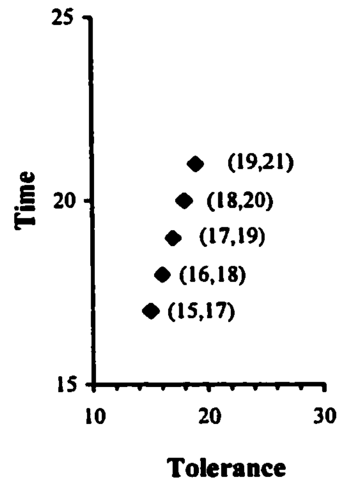


Figure 4.7 Time versus Tolerance Function

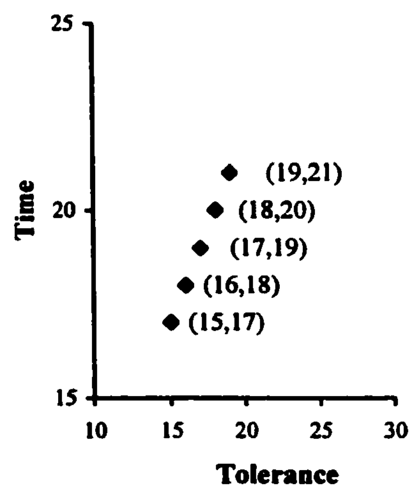
The mathematical formulations are already presented in the equations (3.26) through (3.29). The procedures to obtain three algorithms are similar to those of the cost-time model or cost-tolerance model and the algorithms of time-tolerance model are not detailed here. The same problem B is used to compare the results of each algorithm and the time-tolerance relation for the algorithm is presented in the Figure 4.8. As mentioned from the above figure, the relation between time and tolerance shows an increasing function that explains the



a) Algorithm 1



b) Algorithm 2



c) Algorithm 3

Figure 4.8 Time-Tolerance Relationship for Problem B characteristic of the manufacturing time to assemble according to the total assembly tolerance.

Considering the time and tolerance objectives, there are several same efficient solutions with different costs and processes after obtaining efficient solution set. If the manufacturing time is fixed, the decision maker can apply the tighter assembly tolerance because it guarantees the manufacturing of the higher quality production. However, if the least total manufacturing time with the tighter total assembly tolerance is the goal of the manufacturing company, and if the cost constraint provided by the client is set by a company, model 3 can be used to propose the better set of process selections to the decision maker.

Chapter 5

Analysis and Discussions

The primary intention of this chapter is to demonstrate the adaptability of the proposed model and solution methods to different problem sets. The problems vary in the number of candidate processes available for each dimension. For instance, drilling and punching are each candidate hole making operations with different costs and tolerances. Accordingly, a process selection may result in different levels of manufacturing assembly. Assembly with tighter tolerances is considered easier in general due to the closeness to perfect dimensions. This mirrors reality where interchangeability and standardization are both better affected by near-perfect dimensions. Also, there are cases where the high cost of tighter tolerances may be reflected purely through initial and operating expenses of high cost equipment and not through labor rates. In such cases, it is possible that a less expensive process may still take more time to run. Hence, total manufacturing time to assembly reflects both manufacturing and assembly times.

Eight problems taken from Chase et. al. (1990) are tested. The problem instances are shown in the Appendix 1. By using three algorithms, the solutions of process selection are obtained and efficiency of each algorithm is calculated. Without a full enumeration of solutions, the proposed heuristics enabled to obtain many efficient solutions within reasonable computer time. Complexity and other algorithmic and robustious quality issues are not tested or specified here as the scope of this research is limited to providing a feasible mathematical solution to selective assembly. It is to be noted that the OR aspect is not highlighted, but just used as a tool to prove the mathematical representation of practical problem.

5.1 Cost-Time Model (Model 1)

5.1.1 Algorithm 1

By applying the algorithm 1, the trade-off between cost and time has been obtained with a fixed tolerance for each problem. Figures 4.1.4 a) through c) show the optimal value of cost and time. Now the process for each dimension is needed for the process selection that, at the final stage, will be determined by the decision maker. The solutions of Problems A through H by algorithm 1 are presented in Figure 5.1. As noted from the figures, the process selection varies from the base process (that is, 1,1,1,1,.....,1,1) to others following the exponential function. Table 5.1 represents the computational results; the number of components, assembly tolerance, and the efficient solutions obtained. Also, Table 5.2 shows the number of efficient solutions, the number of solutions searched by each algorithm 1 and efficiency in finding the efficient solutions with corresponding manufacturing costs and times. Problem E has the lowest performance in obtaining the efficient solutions (58.33%) and Problem A has the highest efficiency (83.33%). Still, there are efficient solutions not found by algorithm 1.

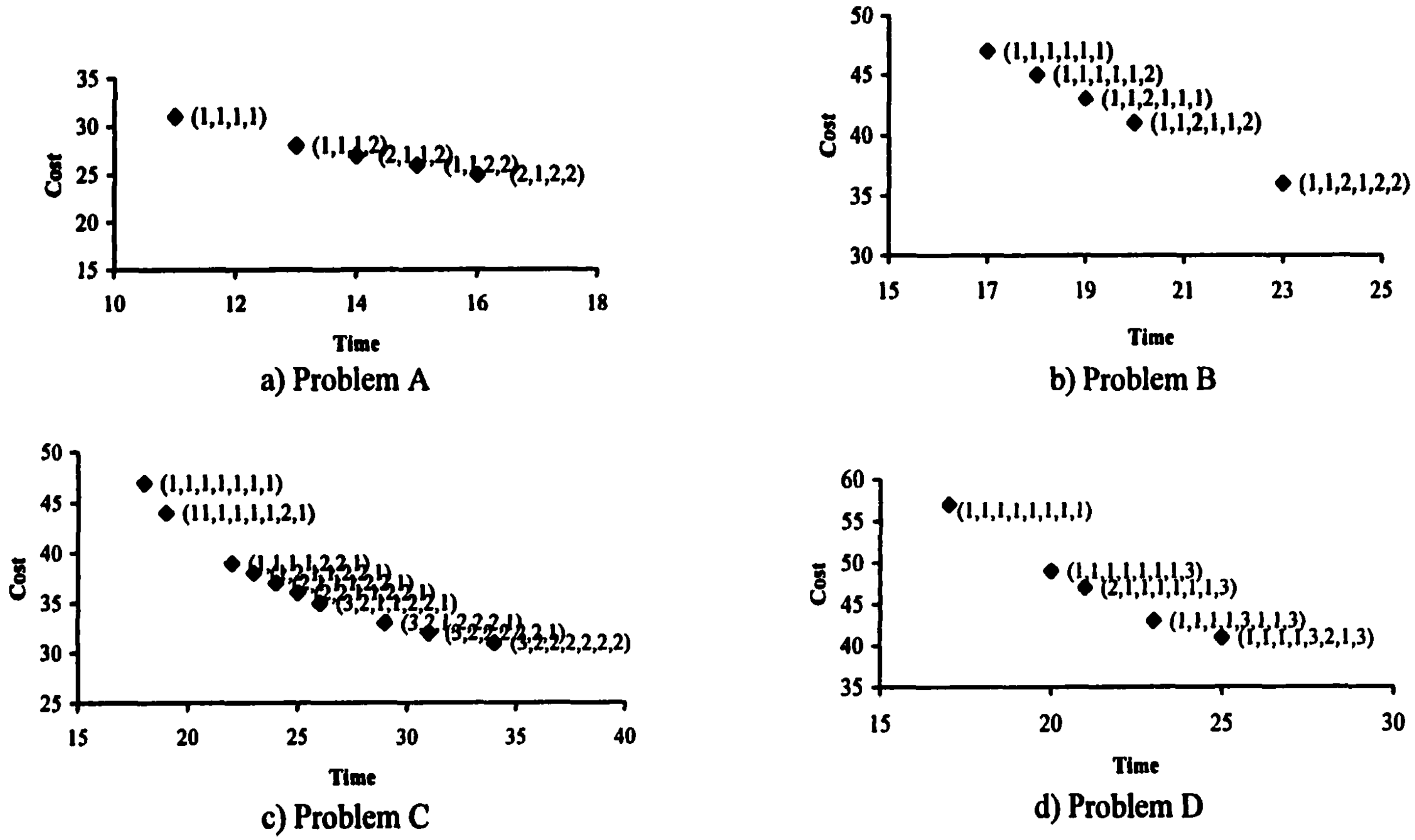


Figure 5.1 (a) Solutions for Problem A to D by Algorithm 1 (Model 1)

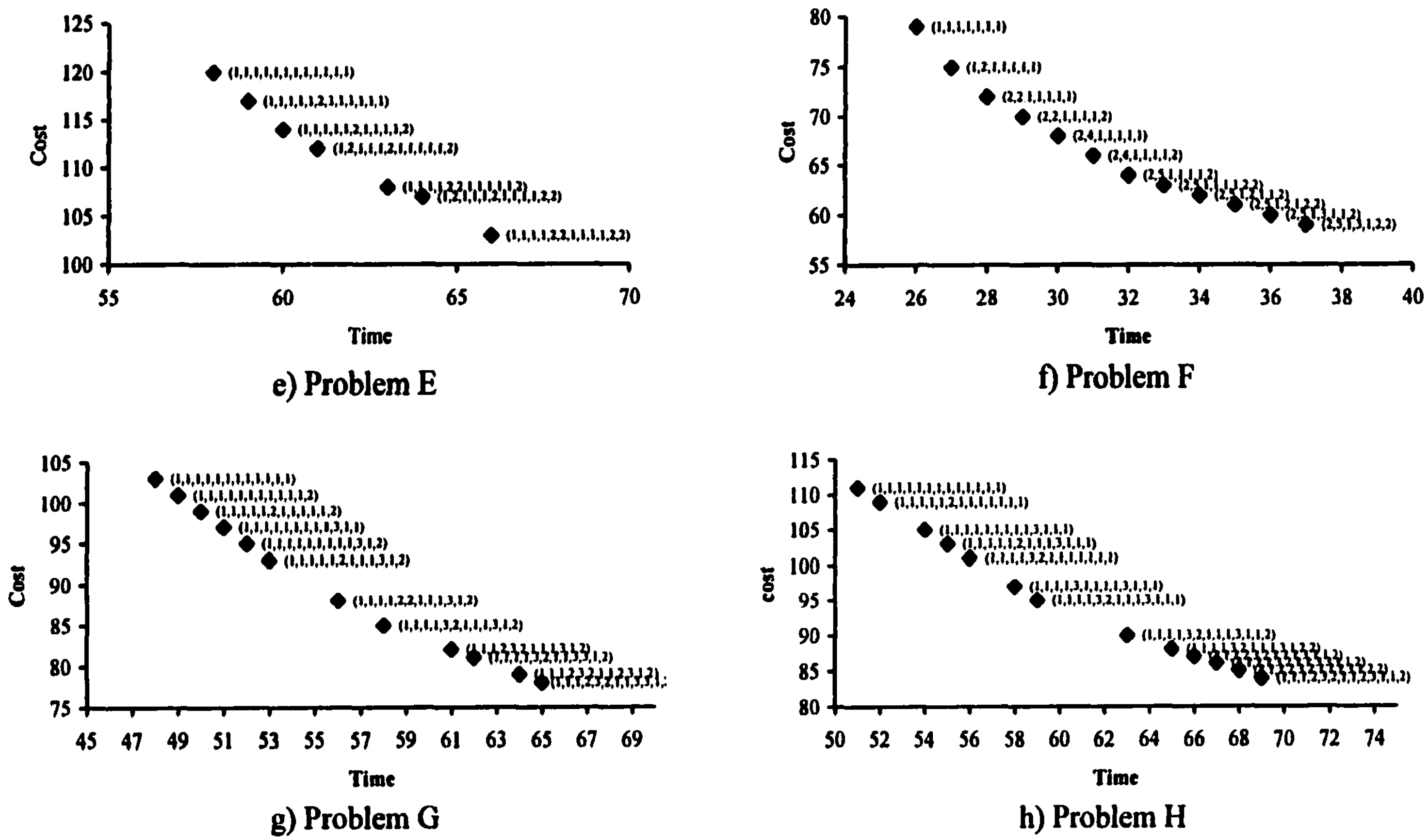


Figure 5.1 (b) Solutions for Problem E to H by Algorithm 1 (Model 1)

Table 5.1 Computational Results of Algorithm 1 for Cost-Time Optimization

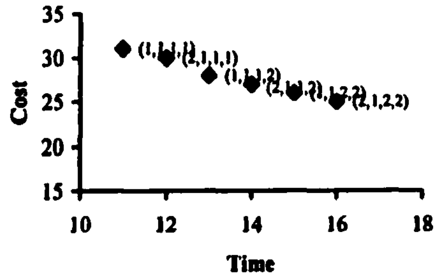
	Number of Components	Assembly Tolerance	Efficient Solutions (cost, time)
A	4	14	(31,11) (28,13) (27,14) (26,15) (25,16)
B	5	23	(47,17) (45,18) (43,19) (41,20) (36,23)
C	7	40	(47,18) (44,19) (39,22) (38,23) (37,24) (36,25) (35,26) (33,29) (32,31) (31,34)
D	8	33	(57,17) (49,20) (47,21) (43,23) (41,25)
E	12	40	(120,58) (117,59) (114,60) (112, 61) (108,63) (107,64) (103,66)
F	7	35	(79,26) (75,27) (72,28) (70,29) (68,30) (66,31) (64,32) (63,33) (62,34) (61,35) (60,36) (59,37)
G	12	36	(103,48) (101,49) (99,50) (97,51) (95,52) (93,53) (88,56) (85,58) (82,61) (81,62) (79,64) (78,65)
H	13	36	(111,51) (109,52) (105,54) (103,55) (101,56) (97,58) (95,59) (90,63) (88,65) (87,66) (86,67) (85,68) (84,69)

Table 5.2 Number of Efficient Solutions Searched by Algorithm 1 and Efficiency

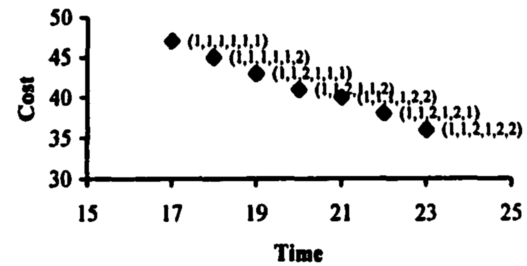
	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	6	5	83.33%
B	7	5	71.43%
C	13	10	76.92%
D	8	5	62.50%
E	12	7	58.33%
F	20	12	60.00%
G	18	13	72.22%
H	19	13	68.42%

5.1.2 Algorithm 2

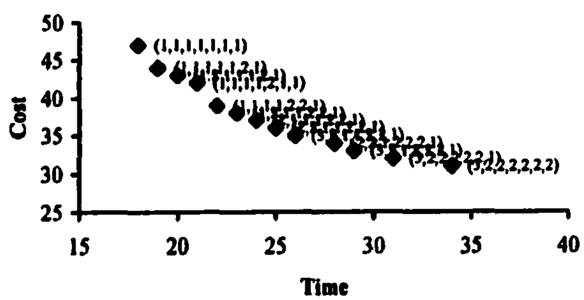
To find out the nondominated efficient solutions, algorithm 2 was used to obtain the solutions not found. The solutions of Problems A through H by algorithm 2 are presented in Figure 5.2. In most cases, efficient solutions were fully detected by algorithm 2 except problem D. Only Problem D has one more efficient solution that could not be obtained by algorithm 2. Table 5.3 represents a summary of analysis; the number of components, assembly tolerance, and the efficient solutions obtained. Also, Table 5.4 shows the number of efficient solutions, the number of solutions searched by algorithm 2 and efficiency in finding the efficient solutions with corresponding manufacturing costs and times.



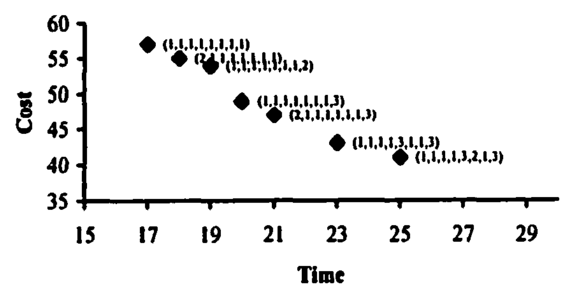
a) Problem A



b) Problem B

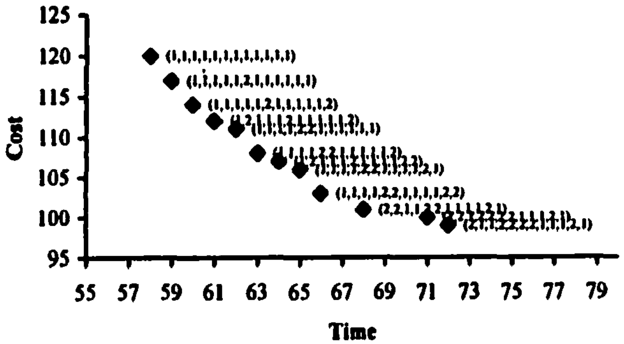


c) Problem C

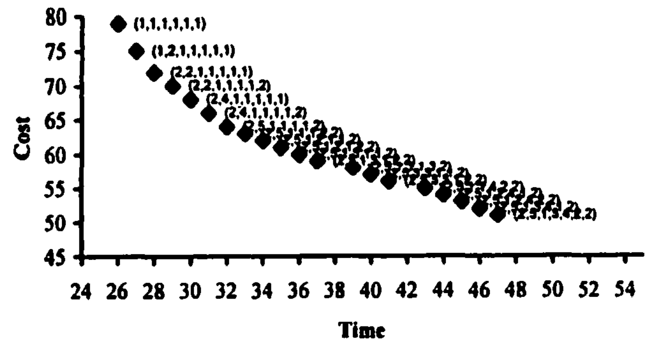


d) Problem D

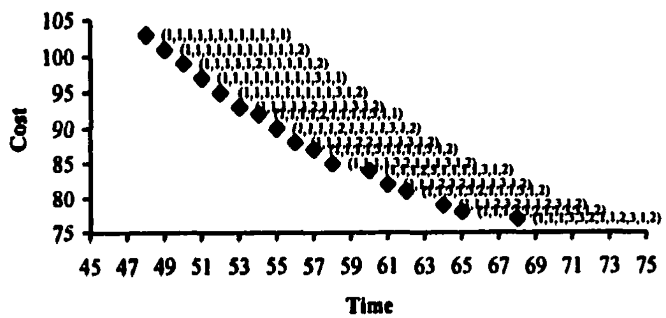
Figure 5.2 (a) Solutions for Problem A to D by Algorithm 2 (Model 2)



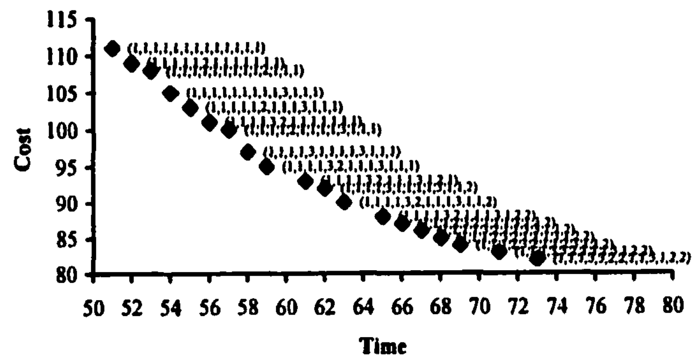
e) Problem E



f) Problem F



f) Problem G



h) Problem H

Figure 5.2 (b) Solutions for Problem E to H by Algorithm 2 (Model 2)

Table 5.3 Computational Results of Algorithm 2 for Cost-Time Optimization

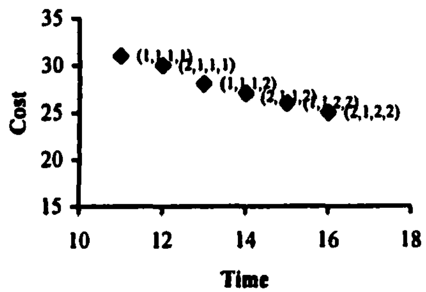
	Number of Components	Assembly Tolerance	Efficient Solutions (cost, time)
A	4	14	(31,11) (30,12)(28,13) (27,14) (26,15) (25,16)
B	5	23	(47,17) (45,18) (43,19) (40,21) (41,20) (38,22) (36,23)
C	7	40	(47,18) (44,19) (43,20) (42,21) (39,22) (38,23) (37,24) (36,25) (35,26) (34,28) (33,29) (32,31) (31,34)
D	8	33	(57,17) (55,18) (54,19) (49,20) (47,21) (43,23) (41,25)
E	12	40	(120,58) (117,59) (114,60) (112, 61) (111,62) (108,63) (107,64) (106,65) (103,66) (101,68) (100,71) (99,72)
F	7	35	(79,26) (75,27) (72,28) (70,29) (68,30) (66,31) (64,32) (63,33) (62,34) (61,35) (60,36) (59,37) (58,39) (57,40) (56,41) (55,43) (54,44) (53,45) (52,46) (51,47)
G	12	36	(103,48) (101,49) (99,50) (97,51) (95,52) (93,53) (92,54) (90,55) (88,56) (87,57) (85,58) (84,60) (82,61) (81,62) (79,64) (78,65) (77,68)
H	13	36	(111,51) (109,52) (108,53) (105,54) (103,55) (101,56) (100,57) (97,58) (95,59) (93,61) (92,62) (90,63) (88,65) (87,66) (86,67) (85,68) (84,69) (83,71) (82,73)

Table 5.4 Number of Efficient Solutions Searched by Algorithm2 and Efficiency

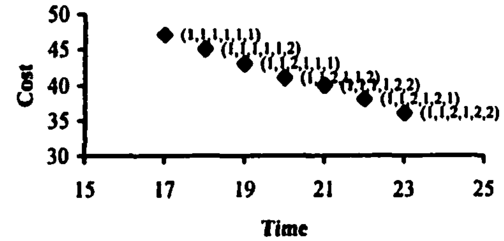
	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	6	6	100%
B	7	7	100%
C	13	13	100%
D	8	7	87.50%
E	12	12	100%
F	20	20	100%
G	18	18	100%
H	19	19	100%

5.1.3 Algorithm 3

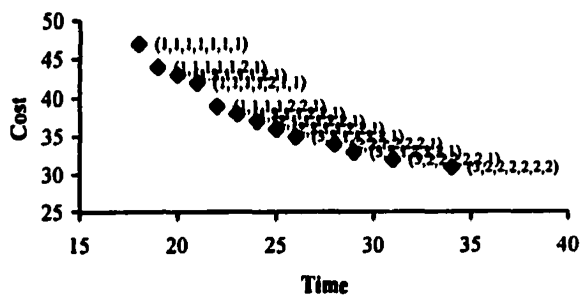
By applying algorithm 2, every efficient solution for Problem A through H was obtained except in the case of Problem D. It is the goal for system engineers to propose as many efficient solutions to the decision makers. Therefore, for the case of Problem D, algorithm 3 was used to determine the missed efficient solution (1,1,1,1,3,1,2,3) with the value of cost and time (40,26). Therefore, the table to show the computational results has only the difference in number of efficient solutions obtained with 8 in Table 5.4 and correspondingly a 100% efficiency. The endeavor to find out every efficient solution was fulfilled through



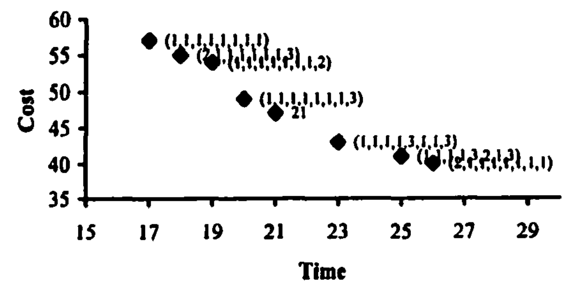
a) Problem A



b) Problem B

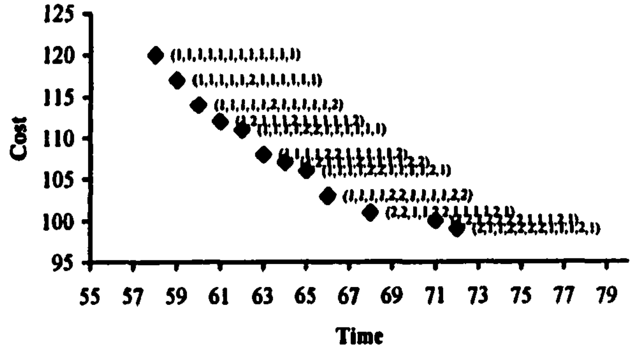


c) Problem C

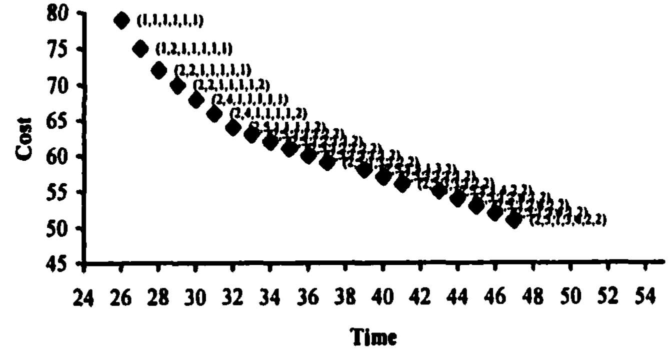


d) Problem D

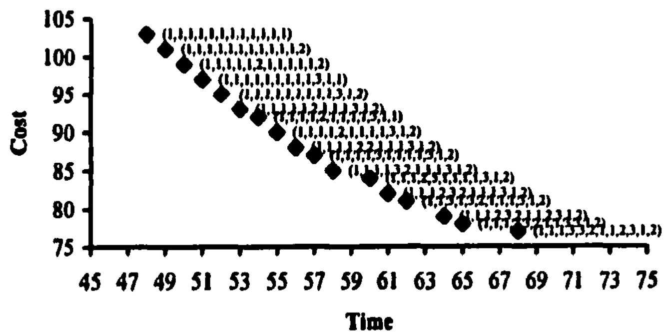
Figure 5.3 (a) Solutions for Problem A to D by Algorithm 3 (Model 1)



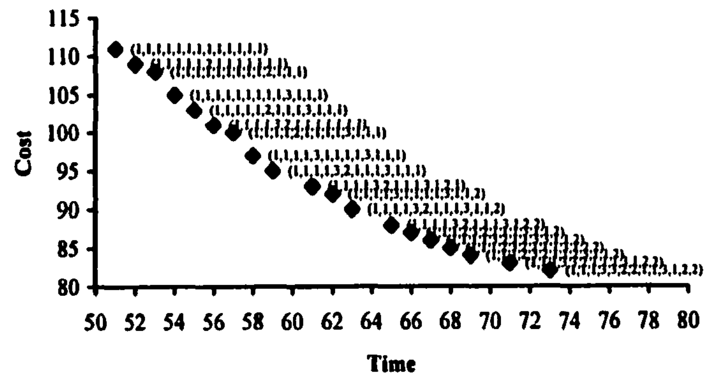
e) Problem A



f) Problem F



g) Problem G



h) Problem H

Figure 5.3 (b) Solutions for Problems E to H by Algorithm 3 (Model 1) the development of heuristic algorithms. The solutions by algorithm 3 is presented in the Figure 5.3.

5.2. Cost-Tolerance Model (Model 2)

5.2.1 Algorithm 1

By applying algorithm 1, the relationship between cost and tolerance are evaluated with a fixed manufacturing time to assemble for each problem. The solutions for Problem A through H by algorithm 1 are presented at Figure 5.4. As expected from literature (Figure 2 of Chase et. al. [1990]), the process selection varies from the base process sequence (that is, 1,1,1,1,.....,1,1) to the others following the exponential function. As assembly tolerance increases, the total manufacturing cost decreases.

Table 5.5 represents the computational results; the number of components, manufacturing time to assemble, and the efficient solutions obtained. Also, Table 5.6 shows the number of efficient solutions, the number of solutions searched by algorithm 1 for each problem and efficiency in finding the efficient solutions with corresponding manufacturing costs and assembly tolerance. Problem D has the lowest efficiency in obtaining the efficient solutions (42%) and Problem H has the highest efficiency (85.71%). Still, many efficient solutions are not to be found by algorithm 1.

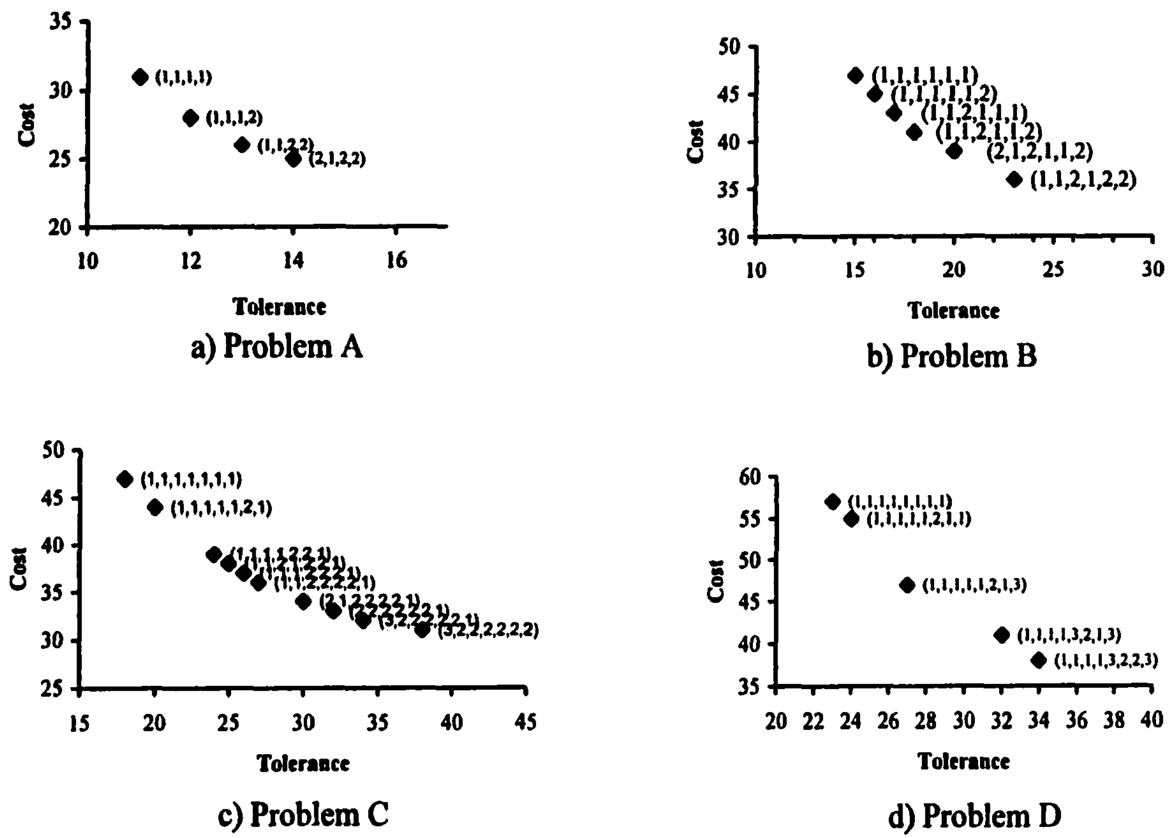


Figure 5.4 (a) Solutions for the Problem A to D by Algorithm 1 (Model 2)

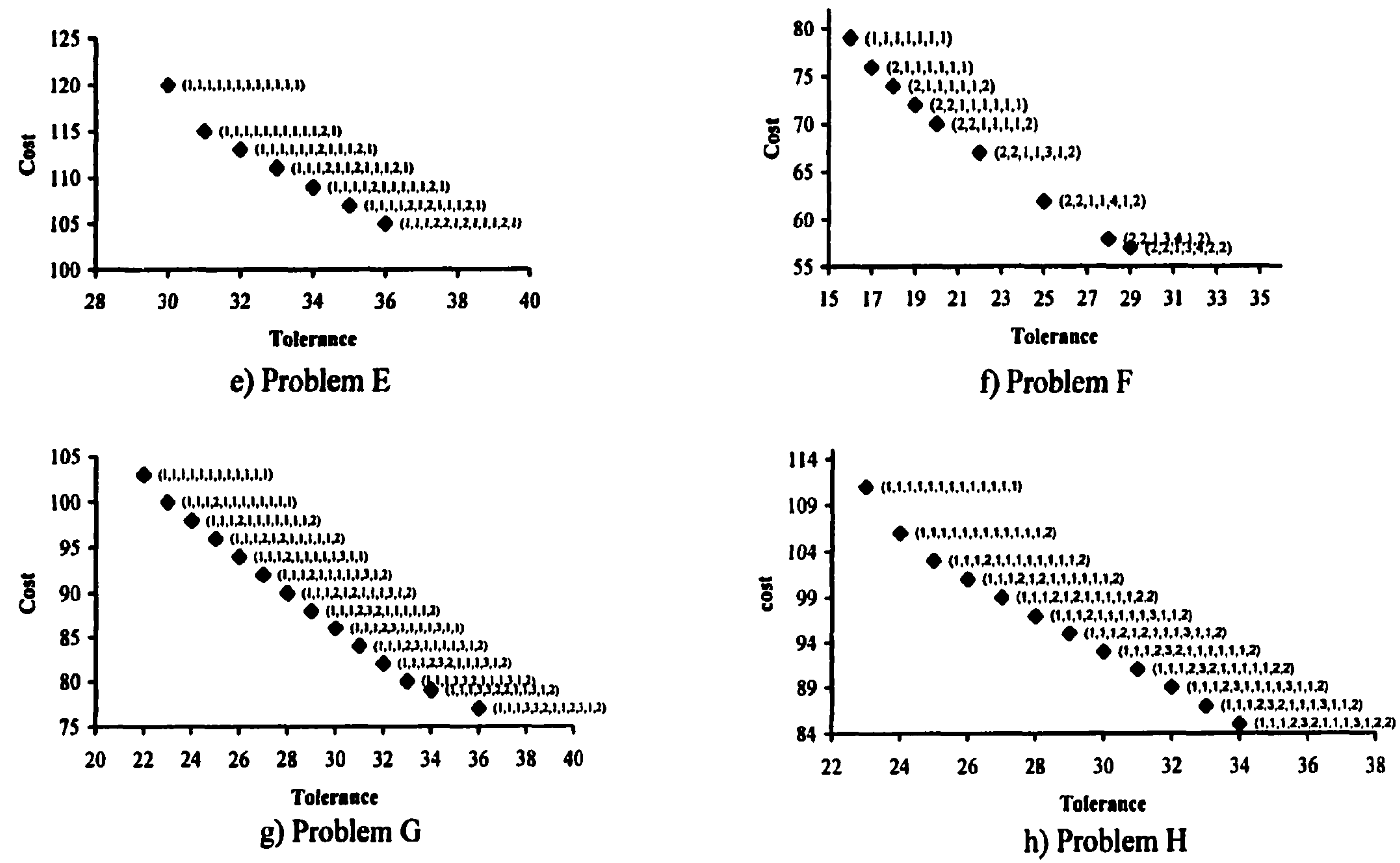


Figure 5.4 (b) Solutions of Problem E to H by Algorithm 1 (Model 2)

Table 5.5 Computational Results of Algorithm 1 for Cost-Tolerance Optimization

	Number of Components	Manufact. time	Efficient Solutions (cost, tolerance)
A	4	16	(31,11) (28,12) (26,13) (25,14)
B	5	23	(47,15) (45,16) (43,17) (41,18) (36,23) (39,20)
C	7	34	(47,18) (44,20) (39,24) (38,25) (37,26) (36,27) (34,30) (33,32) (32,34) (31,38)
D	8	30	(57,23) (55,24) (47,27) (41,32) (38,34)
E	12	70	(120,30) (115,31) (113,32) (111,33) (109, 34) (107,35) (105,36)
F	7	45	(79,16) (76,17) (74,18) (72,19) (70,20) (67,22) (62,25) (58,28) (57,29)
G	12	70	(103,22) (100,23) (98,24) (94,26) (92,27) (96,25) (90,28) (86,30) (84,31) (88,29) (82,32) (80,33) (79,34) (77,36)
H	13	70	(111,23) (106,24) (103,25) (101,26) (99,27) (97,28) (95,29) (93,30) (91,31) (89,32) (87,33) (85,34)

Table 5.6 Number of Efficient Solutions Searched by Algorithm 1 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	5	4	80.00%
B	9	6	66.67%
C	15	10	66.70%
D	12	5	42.00%
E	12	7	58.33%
F	17	9	52.94%
G	18	14	77.78%
H	14	12	85.71%

5.2.2 Algorithm 2

By the partial enumeration search, more efficient solutions were found by algorithm 2. The solutions obtained for Problems A through H by algorithm 2 are presented in the Figure 5.5. Unlike in Model 1 case, full solutions for only two problems (A and C) were fully detected by algorithm 2. Table 5.7 and 5.8 present the corresponding results.

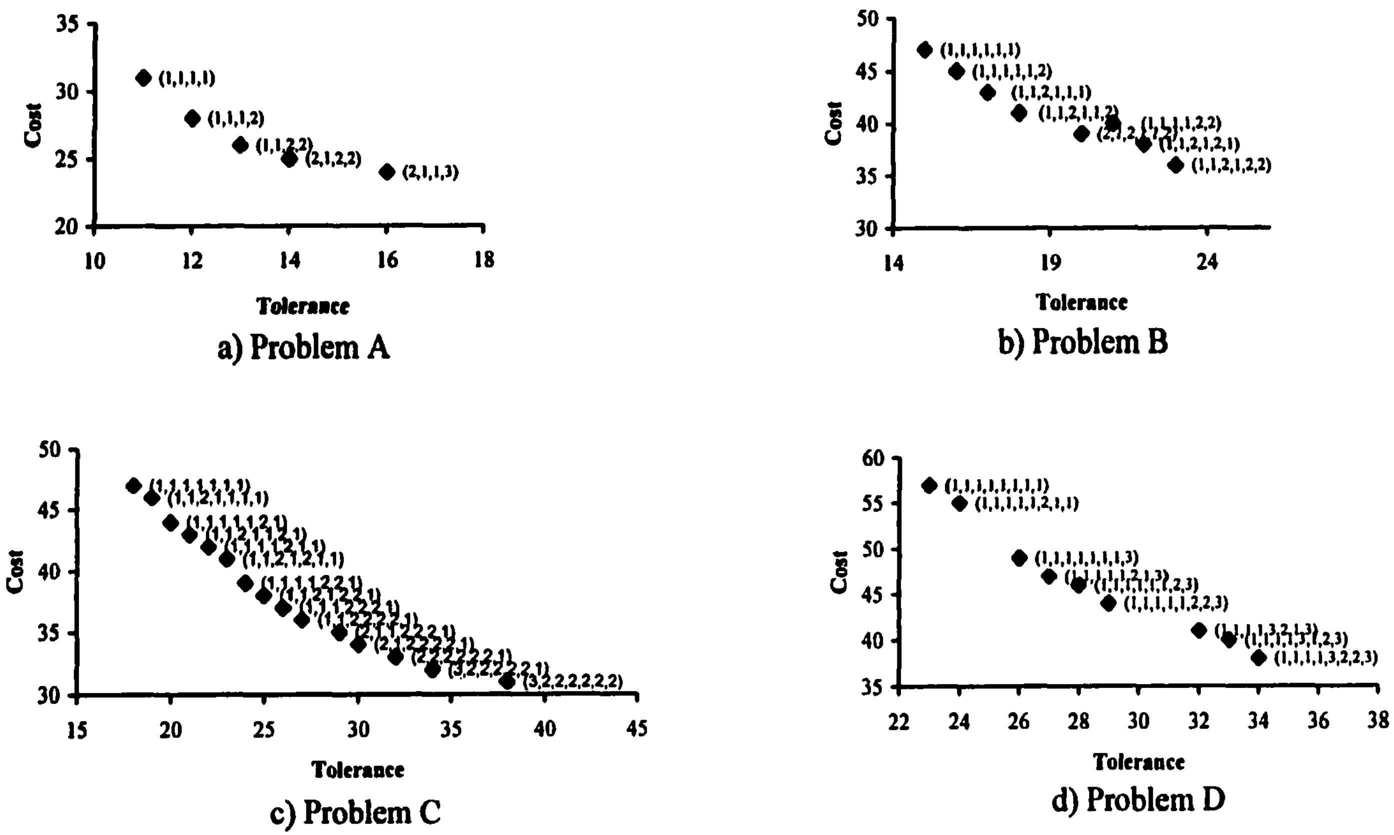


Figure 5.5 (a) Solutions of Problem A to D by Algorithm 2 (Model 2)

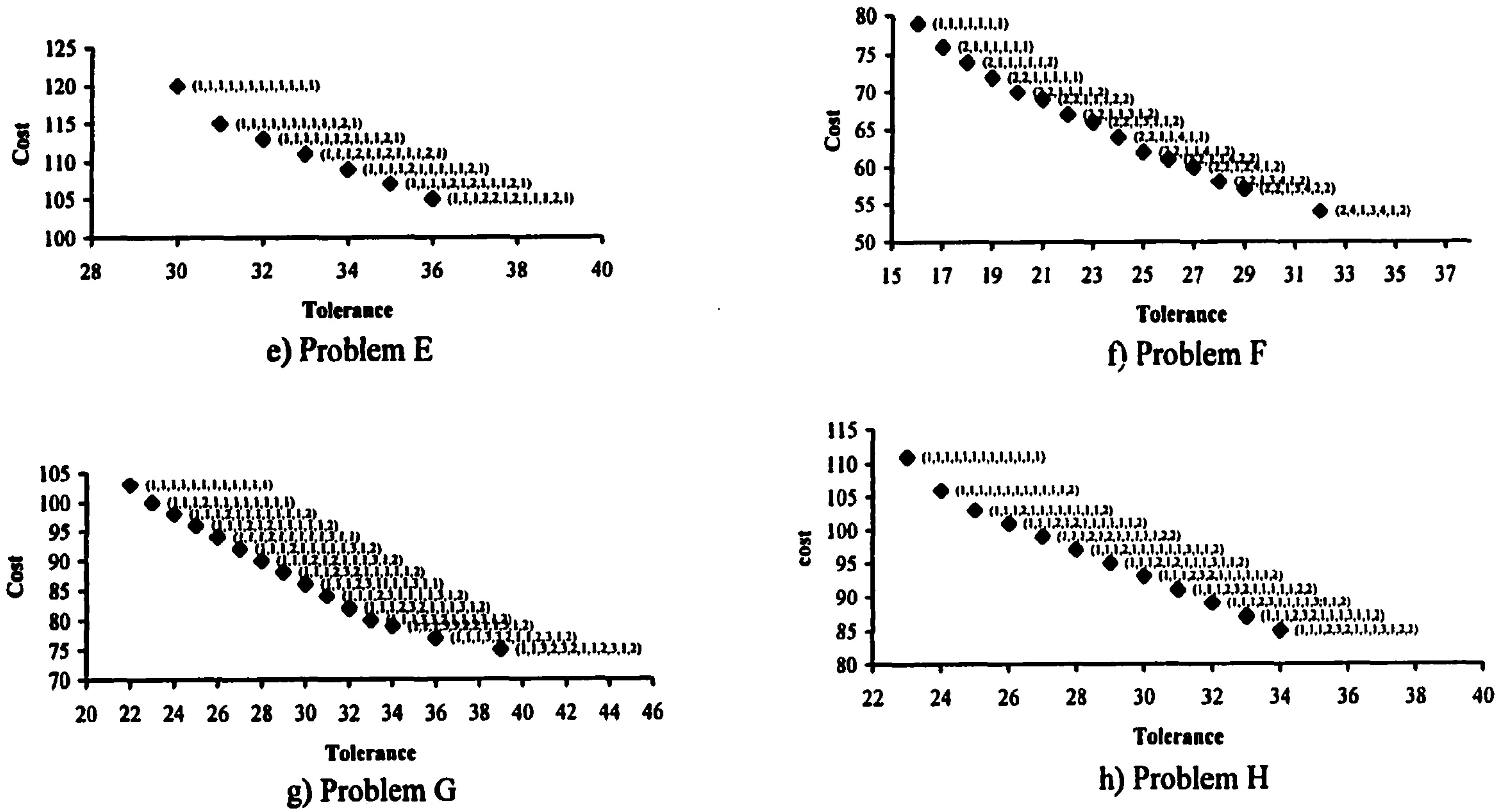


Figure 5.5 (b) Solutions of Problem E to H by Algorithm 2 (Model 2)

Table 5.7 Computational Results of Algorithm 2 for Cost-Tolerance Optimization

	Number of Components	Manufact. time	Efficient Solutions (cost, tolerance)
A	4	16	(31,11) (28,12) (26,13) (25,14) (24,16)
B	5	23	(47,15) (45,16) (43,17) (41,18) (40,21) (38,22) (36,23) (39,20)
C	7	34	(47,18) (46,19) (44,20) (43,21) (42,22) (41,23) (39,24) (38,25) (37,26) (36,27) (35,29) (34,30) (33,32) (32,34) (31,38)
D	8	30	(57,23) (55,24) (49,26) (47,27) (46,28) (44,29) (41,32) (40,33) (38,34)
E	12	70	(120,30) (115,31) (113,32) (111,33) (109, 34) (107,35) (105,36)
F	7	45	(79,16) (76,17) (74,18) (72,19) (70,20) (69,21) (67,22) (66,23) (64,24) (62,25) (61,26) (60,27) (58,28) (57,29) (54,32)
G	12	70	(103,22) (100,23) (98,24) (94,26) (92,27) (96,25) (90,28) (86,30) (84,31) (88,29) (82,32) (80,33) (79,34) (77,36) (75,39)
H	13	70	(111,23) (106,24) (103,25) (101,26) (99,27) (97,28) (95,29) (93,30) (91,31) (89,32) (87,33) (85,34)

Table 5.8 Number of Efficient Solutions Searched by Algorithm 2 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	5	5	100%
B	9	8	88.89%
C	15	15	100%
D	12	9	75.00%
E	12	7	58.33%
F	17	15	88.24%
G	18	15	83.33%
H	14	12	85.71%

5.2.3 Algorithm 3

By applying algorithm 3, every efficient solution for Problem A through H was obtained. From the Table 5.8, only problem A and C achieved 100% efficiency by using algorithm 2. To obtain as much as efficient solutions as possible to the decision makers, this algorithm plays a good role in arriving at the objectives proposed. The Figure 5.6 shows the solutions for the problems A to H. Tables 5.9 and 5.10 summarize the corresponding results.

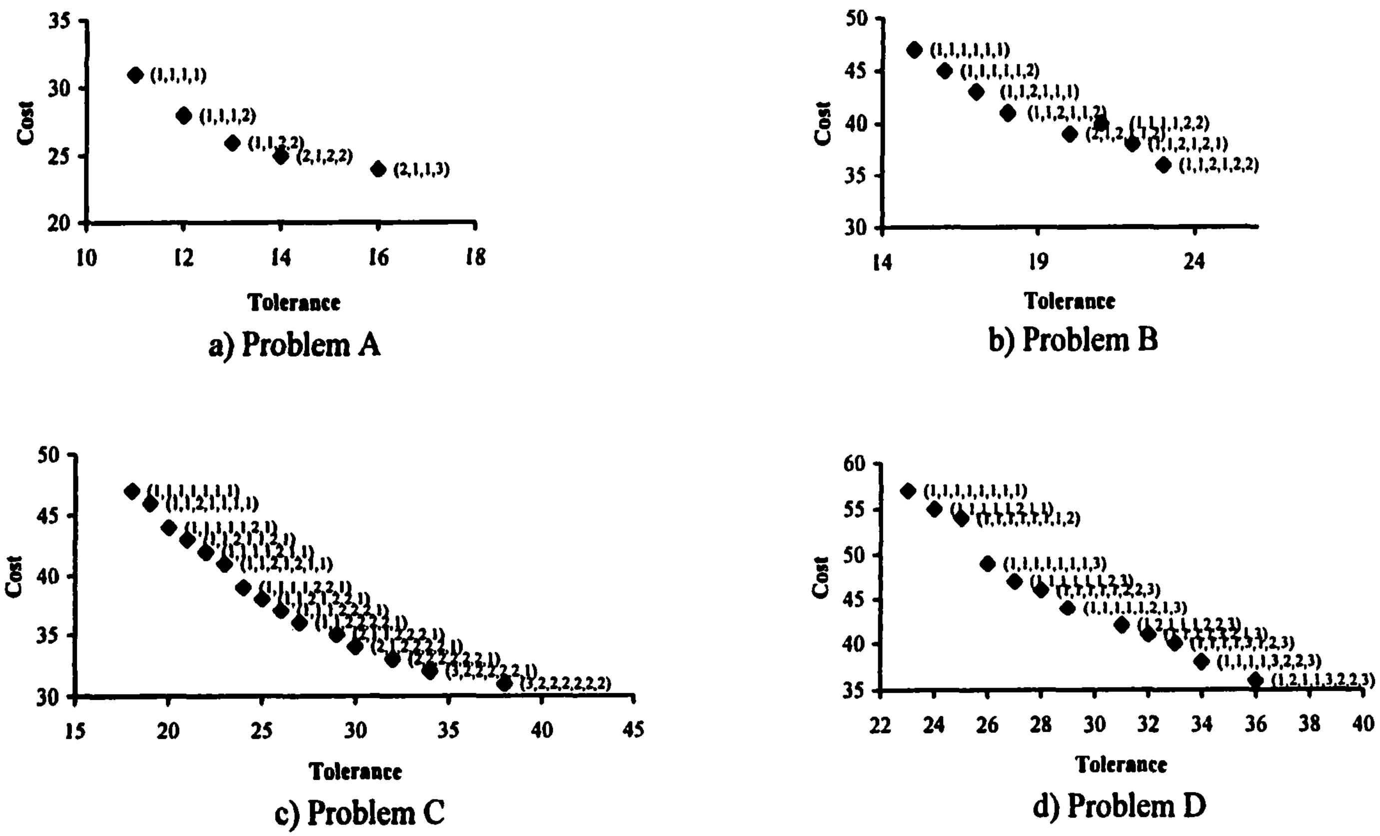


Figure 5.6 (a) Solutions of Problem A to D by Algorithm 3 (Model 2)

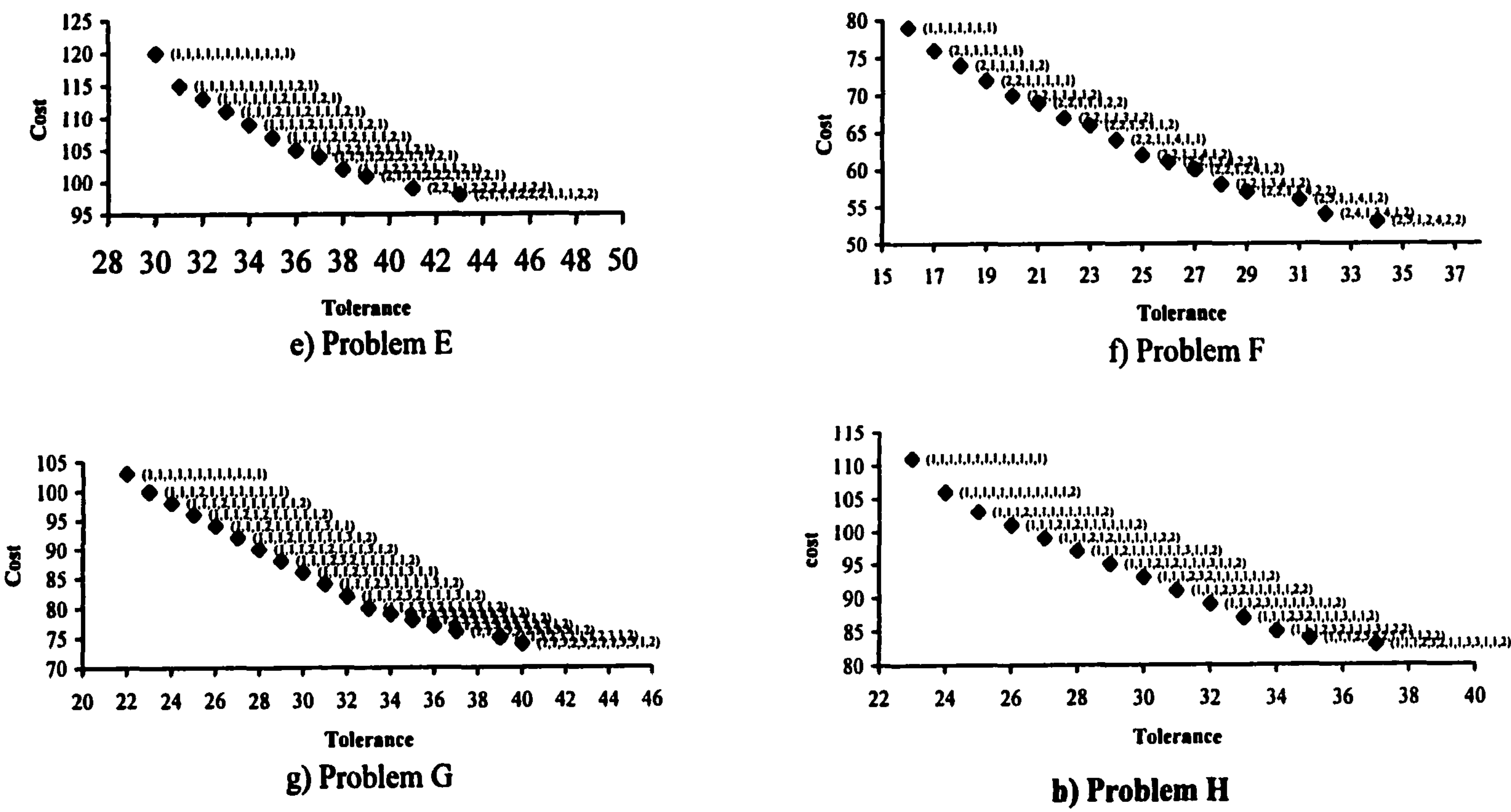


Figure 5.6 (b) Solutions of Problem E to H by Algorithm 3 (Model 2)

Table 5.9 Computational Results of Algorithm 3 for Cost-Tolerance Optimization

	Number of Components	Manufact. time	Efficient Solutions (cost, tolerance)
A	4	16	(31,11) (28,12) (26,13) (25,14) (24,16)
B	5	23	(47,15) (45,16) (44,19) (43,17) (41,18) (40,21) (38,22) (36,23) (39,20)
C	7	34	(47,18) (46,19) (44,20) (43,21) (42,22) (41,23) (39,24) (38,25) (37,26) (36,27) (35,29) (34,30) (33,32) (32,34) (31,38)
D	8	30	(57,23) (55,24) (54,25) (49,26) (47,27) (46,28) (44,29) (42,31) (41,32) (40,33) (38,34) (36,36)
E	12	70	(120,30) (115,31) (113,32) (111,33) (109, 34) (107,35) (105,36) (104,37) (102,38) (101,39) (99,41) (98,43)
F	7	45	(79,16) (76,17) (74,18) (72,19) (70,20) (69,21) (67,22) (66,23) (64,24) (62,25) (61,26) (60,27) (58,28) (57,29) (56,31)(54,32) (53,34)
G	12	70	(103,22) (100,23) (98,24) (94,26) (92,27) (96,25) (90,28) (86,30) (84,31) (88,29) (82,32) (80,33) (79,34) (78,35) (77,36)(76,37) (75,39) (74,40)
H	13	70	(111,23) (106,24) (103,25) (101,26) (99,27) (97,28) (95,29) (93,30) (91,31) (89,32) (87,33) (85,34) (84,35) (83,37)

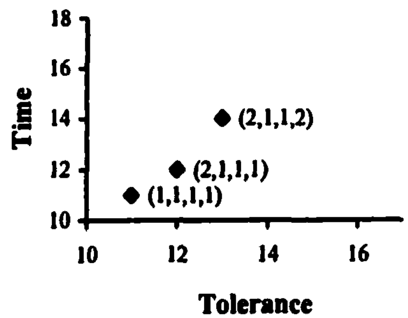
Table 5.10 Number of Efficient Solutions Searched by Algorithm 3 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	5	5	100%
B	9	9	100%
C	15	15	100%
D	12	12	100%
E	12	12	100%
F	17	17	100%
G	18	18	100%
H	14	14	100%

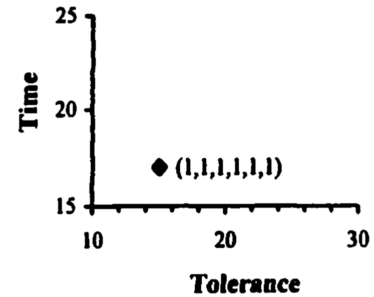
5.3 Tolerance -Time Model (Model 3)

5.3.1 Algorithm 1

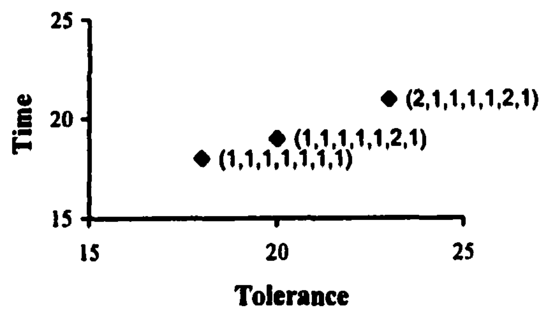
The procedures to obtain the efficient solutions for the three algorithms are the same and the computational results are presented in by the same manner. The solutions for Problems A to D are presented in the Figure 5.7, and the efficient solutions are noticeably fewer than the other models. Cost constraint provided many solutions by one unit change, but the efficient solutions are determined by the non-dominated solution's elimination from the solution set. By



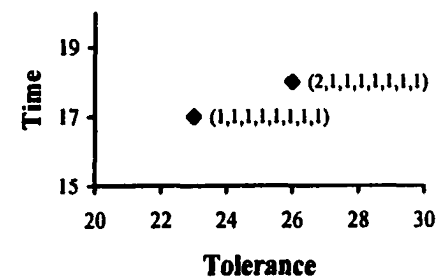
a) Problem A



b) Problem B



c) Problem C



d) Problem D

Figure 5.7 Solutions for Problem A to D by Algoritm 1 (Model 3) applying the algorithm 1, efficiency for the problem A is obtained at most 75 %. For the problem B, only the base process (1,1,1,1,1,1) is obtained. For the other problems, efficiency is calculated below 30 % by the algorithm 1. Table 5.11 and 5.12 show the computational result and the efficiency.

Table 5.11 Computational Results of Algorithm 1 for Time-Tolerance Optimization

	Number of Components	Assembly Cost	Efficient Solutions (time, tolerance)
A	4	27	(11,11) (12,12) (14,13)
B	5	40	(17,15)
C	7	40	(18,18) (19,20) (21,23)
D	8	50	(17,23) (18,26)

Table 5.12 Number of Efficient Solutions Searched by Algorithm 1 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	4	3	75%
B	6	1	16.67%
C	8	3	37.5%
D	7	2	28.57%

5.3.2 Algorithm 2

For obtaining more efficient solutions, algorithm 2 is applied to the problems, but the same number of efficient solutions obtained for the problems except problem B that has obtained four more efficient solutions. The solutions for Problems A to D are presented in the Figure 5.8. By applying the algorithm 2, efficiency for the problem B is obtained as 83.33% in Table 5.13 and 5.14.

Table 5.13 Computational Results of Algorithm 2 for Time-Tolerance Optimization

	Number of Components	Assembly Cost	Efficient Solutions (time, tolerance)
A	4	27	(11,11) (12,12) (14,13)
B	5	40	(17,15) (18,16) (19,17) (20,18) (21,19)
C	7	40	(18,18) (19,20) (21,23)
D	8	50	(17,23) (18,26)

5.3.3 Algorithm 3

By applying algorithm 3, problem A obtained all of the efficient solutions, but there were many efficient solutions not obtained for the problems. The solutions for problem A to D are presented in the Figure 5.9, and Tables 5.15 and 5.16 show the computational result and the efficiency.

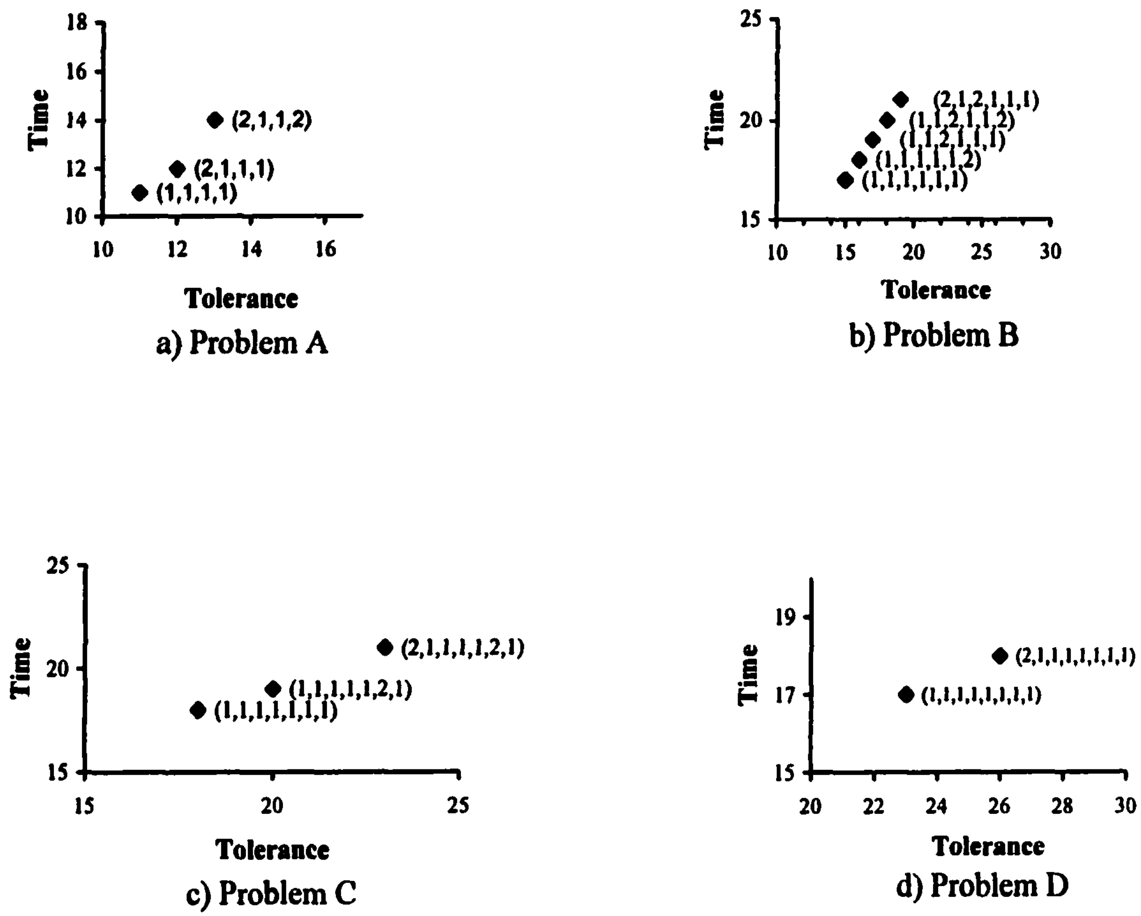
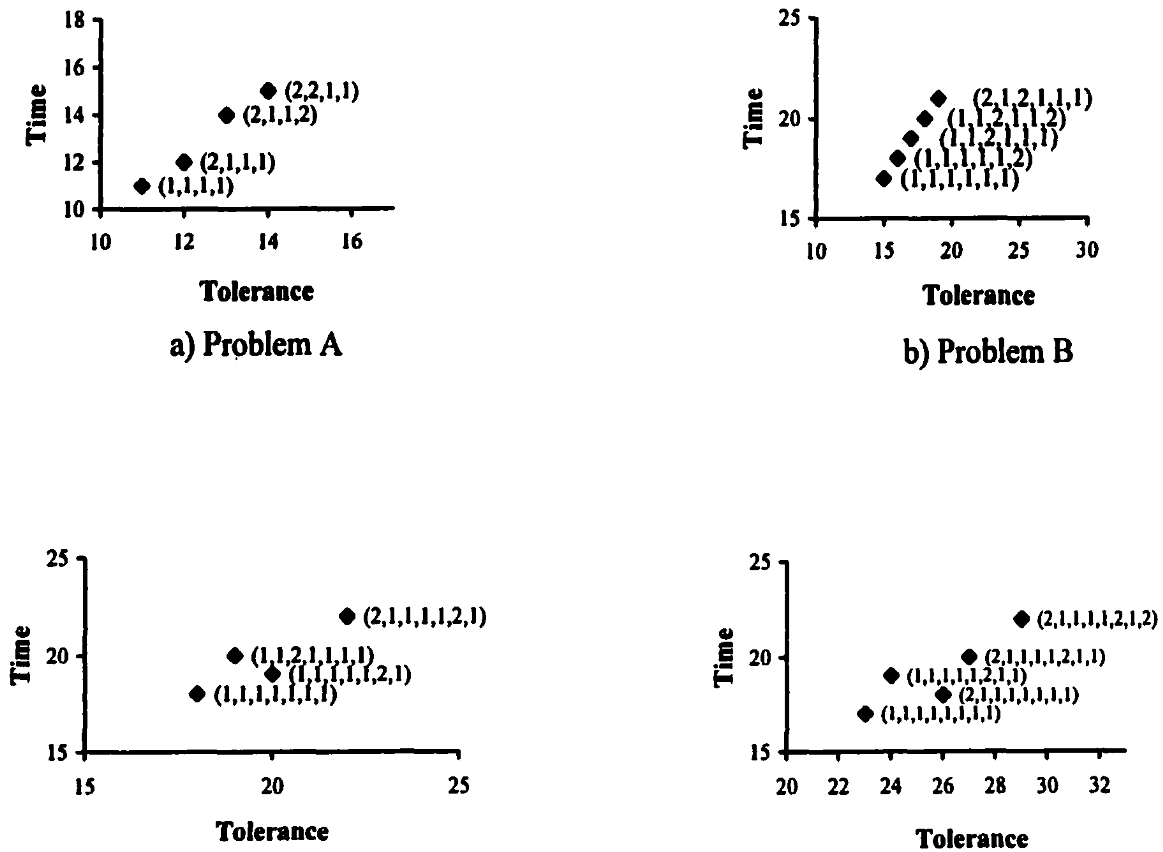


Figure 5.8 Solutions for Problem A to D by Algorithm 2 (Model 3)



a) Problem A

b) Problem B

c) Problem C

d) Prolem D

Figure 5. 9 Solutions for Problem A to D by Algorithm 3 (Model 3)

Table 5.14 Number of Efficient Solutions Searched by Algorithm 2 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	4	3	75%
B	6	5	83.33%
C	8	3	37.5%
D	7	2	28.57%

Table 5.15 Computational Results of Algorithm 3 for Tolerance-Time optimization

	Number of Components	Assembly Cost	Efficient Solutions (time, tolerance)
A	4	27	(11,11) (12,12) (14,13) (15,14)
B	5	40	(17,15) (18,16) (19,17) (20,18) (21,19)
C	7	40	(18,18) (19,20) (20,19) (21,23)
D	8	50	(17,23) (18,26) (19,24) (20,27) (22,29)

Table 5.16 Number of Efficient Solutions Searched by Algorithm 3 and Efficiency

	Number of Efficient Solutions	Number of Efficient Solutions Searched	Efficiency
A	4	4	100%
B	6	5	83.33%
C	8	4	50%
D	7	5	71.43%

In general, except in the case of model 3, the proposed approximate solution methods are quite effective. However, it must be noted that the algorithms used in this research are a first attempt to solve the formulated bi-criteria problems. More efficient OR-based approaches must be used to develop efficient, high-quality, and speedy solution methods.

Algorithm 1 to 3 for models 1 & 2 each have different levels of reaching efficient solution. Algorithm 3 seems the most capable of generating efficient solutions in each case. The evolution of algorithms developed in this work followed the sequence of 1followed by 2, and then by 3. It is suggested that the decision maker choose the algorithm most suited for their purposes, from these.

Nevertheless, noting the differences in the three problems (Models 1, 2, and 3), it is concluded that the heuristics presented in these researches are better at resolving the conflicting trends (Modell 1 and 2) rather than proportional trend

(Model 3). The proportional trend tends to make the problem to deviate from a true bi-criteria optimization problem. The single criteria problem it tends to represent could be better solved with conventional OR tools.

Chapter 6

Conclusions

Manufacturing floors are often faced with prudent process selection decisions while designers have to contend with manufacturing costs and total manufacturing time to assembly. As can be appreciated by most concurrent engineers, the issues of cost, tolerance, and time must be evaluated in an integrated fashion during tolerance design.

This research is founded on tolerance determination and alternate process selection for effecting assembly. Tighter tolerances while quite expensive to manufacture are often easier in assembly completion. Since the penalties of trade-offs are difficult to evaluate without a former structure, a mathematical model with solutions is presented here.

The fundamental contribution of this research is to mathematically model the combined influence of total manufacturing time and manufacturing cost while determining tolerances and alternate machining processes. The nature of the problem has prevented any concrete mathematical study to this date. It is suggested through this thesis that the problem can be formulated as a multi-criteria optimization problem. Although, the practical zones of interest to most tolerance designers is in three specific problems within this integrative (multi-criteria) context: 1. minimizing manufacturing cost and manufacturing time to assemble for a given tolerance constraint. 2. minimizing cost and tolerances for a given manufacturing time to assemble constraint. 3 minimizing the tolerance and time for a given cost constraint. Heuristic methods are suggested based on optimization models to solve these three problems. Analysis is conducted with multiple problem sets to evaluate the feasibility and versatility of the model for standard scenarios. This research and the problem solutions are intended to serve as a first step towards designing decision support for tolerance designers in manufacturing shops. More sturdy mathematical solution methodologies must be employed in the future to solve the bi-criteria problems presented.

While evaluating the scenarios for problems, it is desired to propose as many efficient solutions as possible to the decision makers such that they can consider multiple alternates with respect to the factors considered. This is especially useful in shop environments during alternate process and production planning. Consequently, the models presented in this research have a manufacturing perspective that uses OR as a tool rather than otherwise.

More research on actual factory data sets will better verify the objectives and methodologies presented. It is proposed that this research be extended to multiple processes for single dimension problems, as well as to process scheduling problems. A total multi-criteria cost-tolerance-time formulation and solution with additional constraints on set-up times, measurement functions and process metrics will also serve to establish a comprehensive framework for process planning. An effective mathematical model for process planning will go a long way towards its computerization.

REFERENCES

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., (1993) *Network Flows – Theory, Algorithms, and Applications*, Prentice-Hall, New Jersey.

Alting, L., Zhang, H., and Lenau, T., (1988) "XPLAN – An Expert Process Planning System and its Further Development," Proceedings of the *27th International MATADOR Conference UMIST, UK*, pp. 20 –21.

Alting, L., and Zhang, H., (1989) "Computer Aided Process Planning: The State-of-the-art Survey," *International Journal of Production Research*, Vol. 27, No. 4, pp. 553-585.

Balakrishnan, N., (1993) "A Multiple-Choice Knapsack Model for Tolerance Allocation in Mechanical Assemblies," *IIE Transactions*, Vol. 25, No. 4, pp. 13-14.

Balas, E., (1965) "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Operations Research*, Vol. 13, pp. 517-546.

Bukchin, J. and Tzur, M., (2000) "Design of Flexible Assembly Line To Minimize Equipment Cost," *IIE Transactions*, Vol. 32, pp. 585-598.

Brooks, K. A., (1961) "Statistical Dimensioning Program," *Machine Design*, Vol. 33, pp. 140-145.

Cagan, Johathan and Kurfess, Thomas R., (1992), "Optimal Tolerance Allocation Over Multiple Manufacturing alternatives," *Advances in Design Automation*, ASME, Vol. 2, DE-Vol. 44-2, pp. 165-172.

Catron, B. A., and Ray, S. R., (1991) "ALPS: A Language for Process Specification," *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 2, pp. 105 – 113.

Chang, T. C., and Wysk, R. A., (1984) "Integrating CAD and CAM through Automated Process Planning Systems," *International Journal of Production Research*, Vol. 22, No. 5, pp. 877 – 894.

Chang, T. C., and Wysk, R. A., (1985) *An Introduction to Automated Process Planning Systems*, Prentice-Hall, New Jersey.

Chang, T. C., Wysk, R. A., and Wang, H. P., (1991) *Computer Aided Manufacturing*, Prentice-Hall, New Jersey.

Charnes, A., Cooper, W. W., and Ijiri, Y., (1963) "Breakeven budgeting and programming to goals", *Journal of Accounting Research*, Vol. 1, pp. 16-43.

Charnes, A., Cooper, W.W., Learner, D. B., and Snow, E. F., (1968) "Application of goal programming model for media planning", *Management Science*, Vol. 14, pp. 431-436.

Charnes, A., Cooper, W. W., (1977) "Goal Programming and multiple objective optimization", *European Journal of Operational Research*, Vol. 1, pp. 39-54.

Charnes, A., Gallegos, A., and Lu, H., (1996) "Robustly efficient parametric frontiers via multiplicative DEA for domestic and international operations of the Latin American airline industry", *European Journal of Operational Research*, Vol. 88, pp. 525-536.

Chase, K. W., and Greenwood, W. H., (1988) "Design Issues in Mechanical Tolerance Analysis", *Manufacturing Review*, Vol. 1, No. 1, pp. 50-59.

Chase, K. W., Greenwood, W. H., Loosli, B. G., and Hauglund, L. F., (1990) "Least Cost Tolerance Allocation for Mechanical Assemblies with Automated Process Selection", *Manufacturing Review*, Vol. 3, No. 1, pp. 49-59.

Chen, Q., and Khoshnevis, B., (1993) "Scheduling with Flexible Process Plans," *Production Planning and Control*, Vol. 4, No. 4, pp. 333-343.

Cho, H., Derebail, A., Hale, T., and Wysk, R. A., (1994) "A Formal Approach to Integrating Computer-Aided Process Planning and Shop Floor Control," *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 116, No. 2, pp. 108 – 116.

Coello, C. A. C. (1999) "A Comprehensive Survey of Evolutionary Survey of Evolutionary-Based Multiobjcetive Optimization Techniques," *Knowledge and Information Systems. An International Journal*, Vol. 1, No. 3, pp. 269-308, 1999.

Coello, C. A. C. (2002) List of references on evolutionary multiobjective optimization, http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

De Mello, H. L. S., and Sanderson, A. C., (1986) "AND/OR Graph representation of Assembly Plans," *Proceedings of the American Association of Artificial Intelligence*, Vol. 2, pp. 1113 – 1119.

Eglese, R. W., (1990) "Simulated Annealing: A Tool for Operational Research," *European Journal of Operational Research*, Vol. 46, pp. 271 - 281.

Evans, David H., (1974) "Statistical Tolerancing: The State of the Art. Part I. Background," *Journal of Quality Technology*, Vol. 6, No. 4, pp. 188 - 195.

Evans, David H., (1975) "Statistical Tolerancing: The State of the Art. Part II. Methods For Estimating Moments," *Journal of Quality Technology*, Vol. 7, No. 1, pp. 1 - 12.

Eversheim, W., and Holz, B., (1982) "Computer Aided Process Programming of NC-Machine Tools by Using the System AUTAP-NC," *Annals of the CIRP*, Vol. 31, No. 1.

Foote, B. L., Pulat, P. S., Badiru, A. B., Raman, S., and Kamath, M., (1993) "A Framework for Integrated Production Planning and Scheduling in a Hybrid Assembly Job Shop under Uncertainty," *Proceedings of the Second Industrial Engineering Research Conference*, Los Angeles.

Fox, R. L., (1973), *Optimization Methods for Engineering Design*, (Reading, MA: Addison-Wesley).

Freund, J. E., and Walpole R. E., (1987) *Mathematical Statistics*, 4th ed., Prentice-Hall Inc., New Jersey.

Geoffrion A. M., Dyer A. M., and Feinberg A., (1972) "An interactive approach for multicriterion optimization with an application to the operation of an academic department", *Management Science*, Vol. 19, pp. 357-368.

Gerth, R. J. and Pfeifer, T., (2000) "Minimum Cost Tolerancing under Uncertain Cost Estimates," *IIE Transactions*, Vol. 32, pp. 493-503.

Gibbons, H., (1985) *Algorithmic Graph Theory*, Cambridge University Press, New York.

Groover, M. P., (1987) *Automation, Production Systems, and Computer Integrated Manufacturing*, Prentice-Hall, New Jersey.

Hitch C. J., (1953) "Suboptimization in operations problems," *Operations Research*, Vol. 1, p.89.

Houtzeel, A., (1976) "The MICLASS system," *Proceedings, of the CAM-I's Executive Seminar – Coding, Classification, and Group Technology for Automated Planning*, Arlington, Texas.

Hwang, C. L. and Masud A. S., (1979) *Multiple Objective Decision Making Methods and Applications*, Springer-Verlag.

Irani, S. A., Koo, H. Y., and Raman, S., (1995) "Feature-Based Operation Sequence Generation in CAPP," *International Journal of Production Research*, Vol. 33, No. 1, pp. 17 – 39.

Jahan-Shahi, H., Shayan, E., and Masood, S., (1999) "Cost Estimation In Flat Plate Processing Using Fuzzy Sets," *Computers & Industrial Engineering*, Vol. 37, pp. 485-488.

Joshi, S. B., Wysk, R. A., and Jones, A., (1990) "A Scaleable Architecture for CIM Shop Floor Control," *Proceedings of the CIMCON 1990*, pp. 21 – 33.

Joshi, S. B., Hoberecht, W. C., Lee, J., Wysk, R. A., and Barrick, D. C., (1994) "Design, Development and Implementation of an Integrated Group Technology and Computer Aided Process Planning System," *IIE Transactions*, Vol. 26, No. 4, pp. 2-18.

Khahr C. N. (1958) "Multiples Objectives in Mathematical Programming", *Operations Research*, Vol. 6, pp.849-855.

Khoshnevis, B., and Chen, Q., (1990) "Integration of Process Planning and Scheduling Functions," *Journal of Intelligent Manufacturing*, Vol. 1, pp. 165-176.

Kim, C., Pulat, P. S., Foote B. L., and Lee, D., (1999) "Least Cost Tolerance Allocation and Bi-criteria Extension", *International Journal of Computer Integrated Manufacturing*, Vol. 12, No. 5, pp. 418-426.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., (1983) "Optimization by Simulated Annealing", *Science*, Vol. 220, pp. 671-680.

Korde, U. P., Bora, B. C., Stelson, K. A., and Riley, D. R., (1992) "Computer Aided Process Planning for Turned Parts Using Fundamental and Heuristic Principles," *Transactions of the ASME, Journal of Engineering for Industry*, Vol. 114, No. 2, pp. 31 – 40.

Kuhn H. W. and Tucker A. W., (1951) "Nonlinear Programming", *Proceedings of the second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley.

Kumar, S. and Raman, S., (1992) "Computer-Aided Tolerancing: The Past, the Present and the Future," *Journal of Design and Manufacturing*, Vol. 2, 29-41.

Kusiak, A. and Feng, C.-X., (1995) "Deterministic Tolerance Synthesis: A Comparative Case Study," *Computer-Aided Design*, Vol. 27, No. 10, pp. 759-768.

Lee, S., Wysk, R. A., and Smith, J. S., (1994) "Process Planning Interface for a Shop Floor Control Architecture for Computer-integrated Manufacturing," *International Journal of Production Research*, Vol. 33, No. 9, pp. 2415 – 2453.

Lee, W. -J., and Woo, T. C., (1989), "Optimum Selection of Discrete Tolerances," *Transaction of the ASME, Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 111, pp. 243-251.

Lenderink, A., and Kals, H. J. J., 1(993) "The Integration of Process Planning and Machine Loading in Small Batch Part Manufacture," *Robotics and Computer Integrated Manufacturing*, Vol. 10, No. ½, pp. 89 – 98.

Link, C. H., (1976) "CAPP – CAM-I's Automated Process Planning Systems," *Proceedings of the 13th Numerical Control Society Annual Meeting and Technical Conference*, Cincinnati, Ohio.

Loosli, B. G., (1987) "Manufacturing Tolerance Cost Minimization Using Discrete Optimization for Alternate Process Selection", *MS Thesis*, Brigham Young University, Provo, Utah.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. and Teller, A. H., (1953) "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, Vol. 26, No. 6, pp. 1087 - 1092.

Mettala, E. G., (1989) "Automated Generation of Computer Softwarein Computer Integrated Manufacturing," *Ph. D. Dissertation*, Penn State University, State College, Pennsylvania.

Mettala, E. G., and Joshi, S., (1993) "A Compact Representation of Alternate Process Plans/Routings for FMS control Activities," *Journal of Design and Manufacturing*, Vol. 3, pp. 91 – 104.

Michael W. and Siddall, J. N., (1981) "The Optimization Problem with Optimal Tolerance Assignment and Full Acceptance," *ASME, Journal of Mechanical Design*, Vol. 103, No. 4, pp. 842-849.

Nagarwala, M., Pulat, S. and Raman, S., (1995) "A Slope-based Method for Tolerance Allocation in Minimum Cost Assembly," *Concurrent Engineering: R & A*, Vol. 3, No. 4, pp. 319 – 328.

Ostwald, P. F. and Huang, J., (1977) "A Method for Optimal Tolerance Selection," *Journal of Engr. for Industry*, Trans. of the ASME, Serial B, Vol. 99, No. 3, pp. 558 - 565.

Osyczka, A., (1985) *Multicriteria Optimization for Engineering Design. In Design Optimization*, J. S. Gero, Ed. Academic Press, Inc., New York, NY, 193-227.

Pareto, V., (1896) *Cours D'Economie Politique*, Rouge, Lausanne, Switzerland.

Patel, A. M., (1980) "Computer-Aided Assignment of Manufacturing Tolerances," Proc. Of the 17th Design Automation Conference, Minneapolis, MN., pp. 129-133.

Peters, J., (1970) "Tolerancing the Components of an Assembly for Minimum Cost," *Journal of Engr. for Industry*, Trans. of the ASME, Series B, Vol. 92, No. 3, pp 677-682.

Pomerol, J.-C., and Barba-Romero, S., (2000) *Multicriterion Decision in Management: Principles and Practice*, Kluwer Academic Publishers, Boston/Dordrecht/London

Prabhu, P., Elhence, S., Wang, H. P., and Wysk, R., (1990) "An Operation Network Generator for Computer Aided Process Planning," *Journal of Manufacturing Systems*, Vol. 9, No. 4, pp. 283 – 291.

Redford, A. H., Swift, K. G., and Howie R., (1981) "Product Design for Automated Assembly," *Assembly Automation*, 2nd International Conference.

Romero, C., (1986) "A survey of generalized goal programming (1970-1982)", *European Journal of Operational Research*, Vol. 25, pp. 183-191.

Romero, C., (1991) "*Handbook of Critical Issues in Goal Programming*", Pergamon Press, Oxford.

Sadagopan S. and Ravindran A., (1982) "Interactive Solution of Bicriteria Mathematical Programs," *Naval Research Logistics Quarterly*, Vol. 29, No. 3.

Sadagopan S. and Ravindran A. (1986) "Interactive Algorithms for Multiple Criteria Nonlinear Programming Problems," *European Journal of Operational Research*, Vol. 25, pp. 247-257.

Shah, J. J., (1991) "Assessment of Features Technology," *Computer Aided Design*, Vol. 23, No. 5, pp. 331-343.

Shin, W. and Ravindran, A., "An Interactive Method for Multiple Objective Mathematical Programming Problems," *Journal of Optimization Theory and Application*, Vol. 68, No. 3.

Sinha, P., and Zoltners, A. A., (1979) "The Multiple-Choice Knapsack Problem," *Operations Research*, Vol. 27, No. 3, pp. 503-515.

Smith, J. S., Cohen, P. H., Davis, J. W., and Irani, S. A., (1992) "Process Plan Generation for Sheet Metal Parts Using An Integrated Feature-Based Expert System Approach," *International Journal of Production Research*, Vol. 30, No. 5, pp. 1175 – 1190.

Speckhart, F. H., (1972) "Calculation of Tolerance Based on a Minimum Cost Approach", *Trans. ASME, Journal of Engineering for Industry*, Vol. 94, pp. 447 – 453.

Spotts, M. F., (1973) "Allocation of Tolerances to Minimizes Cost of Assembly," *Journal of Engr. for Industry*, Trans. of the ASME, Serial B, Vol. 95, No. 3, pp. 762-764.

Spotts, M. F., (1978) "How to Use Wider Tolerances, Safely in Dimensioning Stacked Assemblies," *Machine Design*, Vol. 50, No. 9, pp. 60-63.

Srinivasan, V. and Shocker, A. D., (1973) "LP Techniques for Multi-Dimensional Analysis of Preferences," *Psychometrika*, Vol. 38, No. 3, pp. 337-369.

Steuer, R. E., (1977) "Multiple Objective Linear Programming with Interval Criterion Weights," *Management Science*, Vol. 23, pp. 305-317.

Steuer, R. E., (1986) *Multiple Criteria Optimization*, Wiley.

Sutherland G. H., and Roth B., (1975) "Mechanism Design: Accounting for Manufacturing Tolerances and Costs in Function Generating Problem," *Journal of Engr. for Industry*, Trans. of the ASME, Serial B, Vol. 99, No. 1, pp. 283-286.

Tulkoff, J., (1978) "CAM-I: Automated Process Planning (CAPP) System," *Proceedings of the 15th Numerical Control Society Annual Meeting and Technical Conference*, Chicago, Illinois.

Vasseur, H., (1994) "Manufacturing Quality And Process Precision: A Cost-Based Analysis," *Ph.D Dissertation*, Carnegie Mellon University.

Wang, H. P., and Wysk, R. A., (1986) "Applications of Microcomputers in Automated Process Planning Systems," *Journal of Manufacturing Systems*, Vol. 5, No. 2.

Wang, H. P., and Wysk, R. A., (1988) "AIMSI: A Prelude to a New Generation of Integrated CAD/CAM Systems," *International Journal of Production Research*, Vol. 26, No. 1, pp. 119 – 131.

Wilde, D., and Prentice, E., (1975) "Minimum Exponential Cost Allocation of Sure-Fit Tolerances," *Journal of Engr. for Industry*, Trans. of the ASME, Serial B, Vol. 97, No. 4, pp. 1395-1398.

Wu, Z., Eimaraghy, W. H., and Eimaraghy, H. A., (1988) "Evaluation of Cost - Tolerance Algorithms for Design Tolerance Analysis and Synthesis," *Manufacturing Review*, Vol. 1, No. 3, pp. 168 - 179.

Wysk, R. A., (1977) "An Automated Process Planning and Selection Program: APPAS," *Ph.D. Dissertation*, Purdue University, West Lafayette, Indiana.

Yeh, C. H., and Fischer, G. W., (1992) "A Structured Approached to the Automatic Planning of Machining Operations for Rotational Parts Based on Computer Integration of Standard Design and Process Data," *International Journal of Advanced Manufacturing Technology*, Vol. 6, pp. 285 – 298.

Zanakis, S. H., and Gupta, (1985) "A categorized bibliographic survey of goal programming", *Omega* 13 (3), pp. 211-222.

Zeleny, M, (1982) *Multicriteria Decision Making*, McGraw-Hill.

Zhang, C., and Wang, H. P., (1993) "The Discrete Tolerance Optimization Problem", *Manufacturing Review*, Vol. 6, No. 1, pp. 60-71.

Zionts S., and Wallenius J, (1976) "An Interactive Programming Method for Solving the Multiple Criteria Problem," Management Science, Vol. 22, No. 6, pp. 652-663.

Zionts S., and Wallenius J., (1983a) "Identifying efficients vectors: some theory and computational results", *Operations Research*, Vol. 28, No. 3, pp. 652-653.

Nomenclature

N = Number of components within an assembly

T = The desired assembly tolerance value

n_i = Number of available processes to manufacture the component i

i = Index of the component ($i = 1,2,......,N$)

j = Index of the process ($j = 1,2,.....,n_i$)

$$X_{ij} = \begin{cases} 1 & \textit{if component i is produced in process } j \\ 0 & \textit{otherwise} \end{cases}$$

c_{ij} = Normal manufacturing cost when component i is produced in process

t_{ij} = Tolerance when component i is produced in process j

τ_{ij} = Manufacturing time related to the manufacturing cost and corresponding tolerance value for component i with process j

T = Total tolerance

C = Total budget.

Ω = Total manufacturing time to assemble

Δ = Manufacturing cost

Appendix

Appendix 1.

Problem A

Dim.	Process	Toler.	Cost	Time
1	1	0.001	6.00	2
	2	0.002	5.00	3
	3	0.005	2.00	5
2	1	0.006	10.00	4
	2	0.008	8.00	7
3	1	0.003	7.00	3
	2	0.004	5.00	5
4	1	0.001	8.00	2
	2	0.002	5.00	4
	3	0.005	2.00	6

Assembly tolerance = 0.014

Assumption: Process, cost, and tolerance data is obtained from Chase et. al. [1990]. Time estimates are not what they provide. These estimates are generated based on the notion that looser tolerances tend to be more time expensive in assembly. This assembly costs and total assembly times are again assumed in the respective constraints, as necessary. In future studies, validation with actual data will be quite beneficial.

Problem B

Dim.	Process	Toler.	Cost	Time
1	1	0.003	9.00	5
	2	0.005	7.00	7
2	1	0.002	6.00	3
	2	0.004	5.00	4
3	1	0.002	8.00	3
	2	0.004	4.00	5
	3	0.006	2.00	8
4	1	0.003	7.00	1
	2	0.007	4.00	4
5	1	0.003	8.00	2
	2	0.008	3.00	5
6	1	0.002	9.00	3
	2	0.003	7.00	4

Assembly tolerance = 0.023

Problem C

Dim.	Process	Toler.	Cost	Time
1	1	0.001	9.00	5
	2	0.004	7.00	7
	3	0.006	6.00	8
2	1	0.002	3.00	3
	2	0.004	2.00	4
3	1	0.003	5.00	3
	2	0.004	4.00	5
4	1	0.006	6.00	1
	2	0.008	4.00	4
5	1	0.001	7.00	2
	2	0.005	2.00	5
6	1	0.002	8.00	3
	2	0.004	5.00	4
7	1	0.003	9.00	1
	2	0.007	8.00	4

Assembly tolerance = 0.040

Problem D

Dim.	Process	Toler.	Cost	Time
1	1	0.001	6.00	2
	2	0.004	4.00	3
2	1	0.002	8.00	3
	2	0.004	6.00	5
3	1	0.006	5.00	1
	2	0.008	4.00	4
	3	0.009	2.00	6
4	1	0.003	6.00	2
	2	0.005	4.00	5
5	1	0.002	8.00	3
	2	0.005	7.00	5
	3	0.007	2.00	6
6	1	0.003	6.00	2
	2	0.004	3.00	4
7	1	0.005	9.00	3
	2	0.007	7.00	6
8	1	0.001	9.00	1
	2	0.003	6.00	3
	3	0.004	1.00	4

Assembly tolerance = 0.033

Problem E

Dim.	Process	Toler.	Cost	Time
1	1	0.001	10.00	5
	2	0.003	7.00	7
2	1	0.003	7.00	3
	2	0.005	5.00	4
3	1	0.002	5.00	3
	2	0.004	3.00	5
4	1	0.006	3.00	1
	2	0.007	1.00	4
5	1	0.002	12.00	2
	2	0.005	6.00	5
6	1	0.001	10.00	3
	2	0.003	7.00	4
7	1	0.006	9.00	2
	2	0.007	7.00	4
8	1	0.001	9.00	3
	2	0.004	8.00	8
9	1	0.003	10.00	6
	2	0.005	9.00	9
10	1	0.001	15.00	9
	2	0.002	14.00	11
11	1	0.003	20.00	15
	2	0.004	15.00	18
12	1	0.001	10.00	6
	2	0.005	7.00	7

Assembly tolerance = 0.040

Problem F

Dim.	Process	Toler.	Cost	Time
1	1	0.002	10.00	5
	2	0.003	7.00	6
	3	0.004	6.00	8
	4	0.005	5.00	10
	5	0.007	3.00	14
2	1	0.002	12.00	3
	2	0.004	8.00	4
	3	0.006	7.00	5
	4	0.008	4.00	6
	5	0.010	2.00	7
3	1	0.003	9.00	3
	2	0.004	8.00	5
	3	0.006	6.00	7
	4	0.008	5.00	10
4	1	0.001	14.00	1
	2	0.003	12.00	3
	3	0.004	10.00	5
	4	0.006	9.00	8
5	1	0.002	20.00	8
	2	0.003	19.00	10
	3	0.004	17.00	13
	4	0.007	12.00	18
6	1	0.003	4.00	3
	2	0.004	3.00	4
	3	0.005	2.00	6
	4	0.006	1.00	8
7	1	0.003	10.00	3
	2	0.004	8.00	4
	3	0.008	7.00	9
	4	0.009	4.00	11

Assembly tolerance = 0.035

Problem G

Dim.	Process	Toler.	Cost	Time
1	1	0.001	10.00	5
	2	0.002	9.00	7
	3	0.003	8.00	9
2	1	0.001	9.00	3
	2	0.004	7.00	6
	3	0.005	6.00	8
3	1	0.002	8.00	3
	2	0.004	7.00	5
	3	0.006	4.00	7
4	1	0.003	7.00	1
	2	0.004	4.00	4
	3	0.005	2.00	8
5	1	0.001	10.00	2
	2	0.004	5.00	5
	3	0.005	2.00	7
6	1	0.001	6.00	3
	2	0.002	4.00	4
	3	0.003	3.00	6
7	1	0.002	7.00	2
	2	0.003	6.00	4
	3	0.006	5.00	8
8	1	0.004	10.00	3
	2	0.006	9.00	5
	3	0.008	8.00	7
9	1	0.001	10.00	6
	2	0.004	7.00	9
	3	0.005	6.00	10
10	1	0.002	9.00	9
	2	0.004	6.00	11
	3	0.005	3.00	12
11	1	0.003	8.00	5
	2	0.006	7.00	8
	3	0.008	6.00	10
12	1	0.001	9.00	6
	2	0.002	7.00	7
	3	0.004	6.00	9

Assembly tolerance = 0.036

Problem H

Dim.	Process	Toler.	Cost	Time
1	1	0.001	10.00	5
	2	0.002	9.00	7
	3	0.003	8.00	9
2	1	0.001	9.00	3
	2	0.004	7.00	5
	3	0.005	6.00	6
3	1	0.002	8.00	3
	2	0.004	7.00	5
	3	0.006	4.00	7
4	1	0.003	7.00	1
	2	0.004	4.00	4
	3	0.005	2.00	7
5	1	0.001	10.00	2
	2	0.004	5.00	5
	3	0.005	2.00	6
6	1	0.001	6.00	3
	2	0.002	4.00	4
	3	0.003	3.00	6
7	1	0.002	7.00	2
	2	0.003	6.00	4
	3	0.006	5.00	8
8	1	0.004	10.00	3
	2	0.006	9.00	5
	3	0.008	8.00	7
9	1	0.001	10.00	6
	2	0.004	7.00	9
	3	0.005	6.00	10
10	1	0.002	9.00	9
	2	0.004	6.00	11
	3	0.005	3.00	12
11	1	0.003	8.00	5
	2	0.006	7.00	8
	3	0.008	6.00	10

12	1	0.001	9.00	4
	2	0.002	7.00	6
	3	0.004	6.00	8
13	1	0.001	8.00	5
	2	0.002	3.00	9

Assembly tolerance = 0.036