Sequencing Mixed-Model Assembly lines In Just-in-time Production Systems

A thesis submitted for the degree of Doctor of Philosophy

By

Ghorbanali Mohammadi-Khashouie

Department of Systems Engineering

Brunel University

August 2003

Abstract

This thesis proposes a new simulated annealing approach to solve multiple objective sequencing problems in mixed-model assembly lines. Mixed-model assembly lines are a type of production line where a variety of product models similar in product characteristics are assembled. Such an assembly line is increasingly accepted in industry to cope with the recently observed trend of diversification of customer demands.

Sequencing problems are important for an efficient use of mixed-model assembly lines. There is a rich of criteria on which to judge sequences of product models in terms of line utilization. We consider three practically important objectives: the goal of minimizing the variation of the actual production from the desired production, which is minimizing usage variation, workload smoothing in order to reduce the chance of production delays and line stoppages and minimizing total set-ups cost. A considerate line manager would like to take into account all these factors. These are important for an efficient operation of mixed-model assembly lines. They work efficiently and find good solution in a very short time, even when the size of the problem is too large. The multiple objective sequencing problems is described and its mathematical formulation is provided. Simulated annealing algorithms are designed for near or optimal solutions and find an efficiency frontier of all efficient design configurations for the problem.

This approach combines the SA methodology with a specific neighborhood search, which in the case of this study is a " *swapping two sequence*". Two annealing methods are proposed based on this approach, which differ only in cooling and freezing schedules.

This research used correlation to describe the degree of relationship between results obtained by method B and other heuristics method and also for quality of our algorithm ANOVA's of output is constructed to analyse and evaluate the accuracy of the CPU time taken to determine near or optimal solution.

Acknowledgements

First and the most thanks to God, in whom remembrance do hearts, find satisfaction.

My best gratitude, immense respect and honour are for my supervisor professor Khalil Hindi for his excellent inspiration, advice guidance through this research. I am particularly thankful to Dr. Mostafa Ozbayrak for his counsel, encouragement and trust.

My thanks are due to my second supervisor Dr. Seyed Hessameddin Zegordi for his invaluable help and guidance.

A special thanks to members of staff, Mrs Linda Hedges and Mrs Margaret Hodgson of Systems Engineering Department.

The financial support of the Ministry of Culture and Higher Education of the Islamic Republic of Iran is gratefully. Their help has made this study possible.

Special thanks go to my colleagues at the Shahid Bahonar University of Kerman, Iran for their cooperation.

Publications relating to this work

1. Patrick R. McMullen and Ghorbanali Mohammadi "Simulated Annealing Scheme for Sequencing Mixed-Model Production lines", International Journal of production research. Forthcoming.
2. Ghorbanali Mohammadi "Minimizing Product rates Variation in JIT production systems". Fourth International Conference on Operations and Quantitative Management, Tainan, Taiwan, January 2003.
3. Ghorbanali Mohammadi "Sequencing Mixed-Model Final assembly line Productions", Institute for Operations Research and Management Sciences, 7th Conference on Information systems and Technology (CIST), San Jose, California, USA, November 2002.
4. Ghorbanali Mohammadi "Best Possible Solution For Mixed-Model Assembly Line Balancing Problems", Canadian Operational Research Society (CORS), Canada, 2000.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction and Preliminaries

This chapter begins with an introduction to the Mixed-Model Sequencing Problem (MMSP) and the contribution of this research to the body of knowledge as it pertains to the MMSP. Section 2 provides terminology to clarify the problem definition. Section 3 describes the main goals of sequencing mixed-model production lines. Section 4 discusses the objectives of the research and section 5 concludes the chapter with the outline of this dissertation.

1.1 Problem Definition and Research Contributions

Mixed-model assembly lines produce a variety of products. This concept is to combine the economics of the old mass production techniques and the customization of craft production ideas. With this new type of production comes the responsibility of companies to determine an efficient method for scheduling the variety, or mix, of products. *The problem is to develop efficient and appropriate techniques for sequencing Mixed-Model production lines taking account of set-up costs, and the Just-In-time philosophy.* This research focuses on providing sequencing algorithms and set-up costs, which can be ensured. Keeping a constant rate of usage and minimizing deviations of actual workload from ideal workload of products on the line, leveling the load and smoothing-workload on each station on the final assembly line and minimizing set-up costs. These are three conflicting goals, which will be explained, in next chapters.

The following is an outline of the contributions of this dissertation to the body of knowledge pertaining to the sequencing Mixed-Model production lines. The contributions are presented such a theoretical standpoint, and also from a numerical experiment perspective.

1. From a math programming, this dissertation provides a new Annealing scheme for simulated annealing algorithm. In this research the goal of interest was to levels and smooth the problems. The first attempt at levelling or balancing the schedule was based on an overall goal of minimizing the variation of the actual production from the desired production. This is called the" *Usage goal*". The second attempt was used at smoothing the workload on the final assembly line in order to reduce the chance of production delays and line stoppages. This is called

the "*smoothing-workload goal* ". The third objective was to minimize the required set-up costs.

2. This research also explores a new annealing procedure. The performance and computational time of this algorithm heavily depends on the *annealing schedule* and its parameter tuning. This algorithm produces optimal solutions for small problem instances and near-optimal solutions for large problem instances.
3. This research also explores heuristics for solving the Mixed-Model sequencing for Just-In-time (JIT) production systems, which require producing only the necessary products in the necessary quantities at the necessary times.
4. An evaluative methodology for comparison of the results with existing optimal solutions is established.
5. From a practical standpoint, this research provides an improvement to the well-known and widely used "*Goal-Chasing I* "and "*Goal-Chasing 2* " algorithm of Monden.

The following is a list of tasks used to accomplish the theoretical and practical contributions explained above.

1. Introduced a set of new methods for our objective Mixed-Model sequencing problem and tested with several problem sets, M1, M2, and M3 with respect to the sequencing goals: leveling the production line. The following is a list of the methods tested:

   . Method A and Method B- tested the two heuristics for comparison with each other.

   . Method B-tested- This method is tested with the EDD solution, and existing optimal solution, for comparison of the method B.

   . Pairwise exchange mechanism-Designed and tested this mechanism for determing a best manufacturing sequence.

2. Investigated the implications of algorithms A, B, and EDD solution, and optimal solution on three problem sets, and evaluated the resulting tradeoffs between these sequencing to meet the levelling Just-In-Time production line.
3. Evaluated the problem of minimizing deviations of actual workload from ideal workload. The following is a list of the methods tested:

- Set M1 and tables 5-10-5.12 assembly time required by stations and models (Simple, Moderate and Complex) are introduced to test with respect to the smoothing-workload.
- Method A and B-tested the two heuristics for comparison with each other to minimize the workload problem.

4. Evaluated the problem of minimizing required set-up costs of usage variation and smoothing the workload.
5. We classify all potential to the multiobjective optimisation problem into dominated and nondominted (efficiency *frontier*) solutions.
6. We used correlation to describe the degree of relationship between results obtained by Method B and optimal solution.
7. For quality of our algorithm ANOVA's of the output is constructed to analyse and evaluate the accuracy of the CPU time taken to determine near or optimal solution.

1.2 Terminology

The key terms that define the elements of the problem are: mixed-model sequencing, just in time, sequencing problems, optimization, Simulated Annealing, neighborhood search.

Mixed-Model Sequencing is used when a variety of products, similar in nature, are produced alternately on the same production line. Differences in the products may vary in small ways such as the colour of paint on a component, or they may vary greatly, such as a base model versus a luxury package, or even front wheel drive and rear wheel drive automobiles. Intentionally mixing the products, balancing the scheduling, smoothing the workload. This method of manufacturing replaces the well-known large batch production method.

JUST-IN-TIME (JIT) systems are very important in the world of manufacturing today. The "Just-In-Time" production philosophy is the foundation of the Toyota process. This concept refers to the manufacturing and conveyance of only what is

needed, when it is needed, and in the amount needed. In addition, a minimum amount of inventory is kept on hand. This enhances efficiency and allows quick response to change.

Pull System. Flow manufacturing is based on a pull orientation. What this means is that all materials are brought into the process (or into the plant from suppliers) only when demanded by production activities. In the highest-level view, a customer demand triggers all activities and material acquisition.

Sequencing problems. Finding an ordering, or permutation, of a finite collection of objects, like jobs, that satisfies certain conditions, such as precedence constraints.

Optimization can be said to be a variation of input parameters in order to minimize or maximize, or reach a target value for a cost function, within given constraints. The word optimization commonly refers to the mathematical handling of parameters in order to maximize or minimize a mathematical function. It is, however, possible to use the word to describe handling of more complex systems.

Combinatorial optimization is the task of designing a set of entities, and deciding how they must be configured, so as to give maximum results.

Simulated Annealing (SA) is an algorithmic approach to solving optimization problems.

1.3 Production Scheduling Goals

When a company is determining a production schedule in the Just-In-Time environment, there are several criteria that must be met and goals that are considered in order to efficiently and effectively meet the customer's demands. Three of the goals companies consider when scheduling in this environment follow:

1.3.1 Usage Variation

A basic and important goal is to level the final production. Smoothing the production line consists of keeping a constant rate of usage of work in final assembly

line in the system. This consistency helps to respond to customer's demands for a variety of products, without holding large inventories.

1.3.2 Balance the workload

Another important goal is to *balance the workload on the final assembly line.* The workload-smoothing problem minimizes deviation of actual workload from the ideal workload.

1.3.3 Set-ups

Minimizing set-ups is also important in a production line. The number of set – ups in sequencing Just-In- Time production line is quite straightforward. Set-ups in this research are sequence-dependent.

1.3.4 Research Objectives

The main objective of this research is to provide a method that can be used for sequencing a mixed-model Just-In –Time production line. This research will provide information that will increase the body of knowledge pertaining to mixed-model scheduling, which will aid in knowledgeable scheduling decisions.

1.3.5 Organization of Dissertation

In chapter 2, the relevant Mixed-Model assembly lines literature is reviewed. Chapter 3 reviews the Simulated Annealing literature. Chapter 4 presents the Mixed-Model sequencing problem and our proposed algorithms. Chapter 5 provides the results of computational testing of the usage variation and workload approaches with the required set-up costs and finally chapter 6 presents analysis and conclusions

2 Literature Survey

In this chapter we present the relevant literature on mixed-model sequencing problems (MMSP). Section 2 lists the specific references and provides a description of each. This is a broad survey, which covers several aspects of mixed-model sequencing found in reference journals. The two research efforts most important to this research are Monden (1983) and Miltenburg (1989).

2.1 Introduction

Assembly lines are special flow-line production systems that are typical in the industrial production of high quantity standardized commodities.

A mixed-model assembly line is a production line that produces a variety of different product models simultaneously and continuously. Each workstation specializes in a certain set of assembly work elements, but the stations are sufficiently flexible to perform their respective elements on different models. While one model is being assembled at one station, a different model is being assembled at the next station. Mass produced consumer products commonly assembled on mixed-model lines include automobiles (Weiner, 1985) and large and small home appliances. Variations in model styles, options, and sometimes brand names characterize these products.

2.2 Basic terms of assembly lines

Assembly-Assembly is the process of collecting and fitting together various parts in order to create a finished product (end item). Parts may be subdivided into (purchased) components and sub-assemblies.

Operation-An operation (task) is a portion of the total work content in an assembly process. The time necessary to perform an operation is called operation (task) time.

Station-A (work) station is a segment of an assembly line where a certain amount of work (a number of operations) is performed. Its dimension, the machinery and equipment as well as the kind of assigned work mainly characterize it. To this effect,

stations can be subdivided into manual or automated stations depending on the subjects performing the work.

Cycle Time-The cycle time represents the maximal amount of time a work-piece can be processed by a station.

Precedence Constraints- Due to technological restrictions, the ordering in which operations must be performed may be partially prespecified. This partial ordering of tasks can be illustrated by means of a precedence diagram (precedence graph; Prenting and Battaglin (1964)) which contains nodes for all operations and arcs (i,j) if an operation i must precede an operation j.

2.3 Classification of assembly line systems

Assembly line production systems are present in different industrial environments and are utilized to manufacture a large variety of products; in particular, they are used to produce consumer goods such as cars, engines, domestic appliances, television sets, computers, and other electrical appliances. The demands of products are rather different, and it is necessary to implement different production systems.

Assembly lines can be classified as single-model, mixed-model, and multi-model systems according to the number of models that are present on the line.

Single-Model assembly lines-Single model assembly lines have been used in single type or model production only. There are large quantities of the products, which have the same physical design (see Figure 2.1) on the line. Here, operators who work at a workstation execute the same amount of work when a sequence of products goes past them at a constant speed.

The main objective of this assembly line is to assign work-pieces and to choose the number of operators for stations, so that all stations have approximately the same amount of work, to find the minimum number of stations to satisfy a certain production output rate of the products. A cycle time must be chosen before work-pieces are assigned to stations in a single model assembly line.

Mixed-Model Assembly lines-Mixed-Model lines are usually used to assemble two or more different models of the same product simultaneously (see Figure 2.2). On the line, the produced items keep changing from model to model continuously.

Multi-Model Assembly lines-Several (similar) products are manufactured on one or several assembly lines. Because of significant differences in the production processes, rearrangements of the line equipment are required when product changes occur. Consequently, the products are assembled in separate batches in order to minimize set-up inefficiencies. While enlarging batch sizes reduces set-up costs, inventory costs are increased. (Scholl 1998). (See Figure 2.3).

Figure 2-1: Single-model line

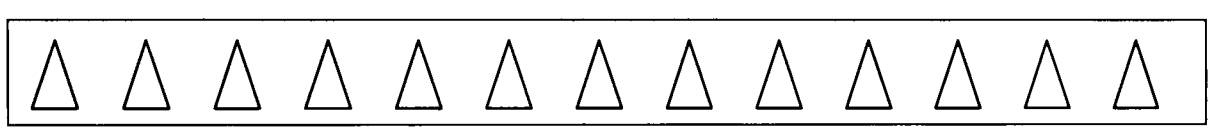


Figure 2-2: Mixed-model line

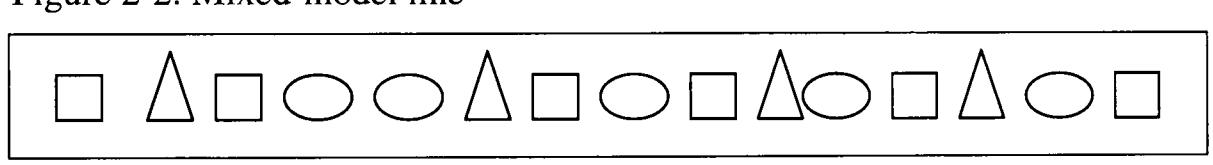
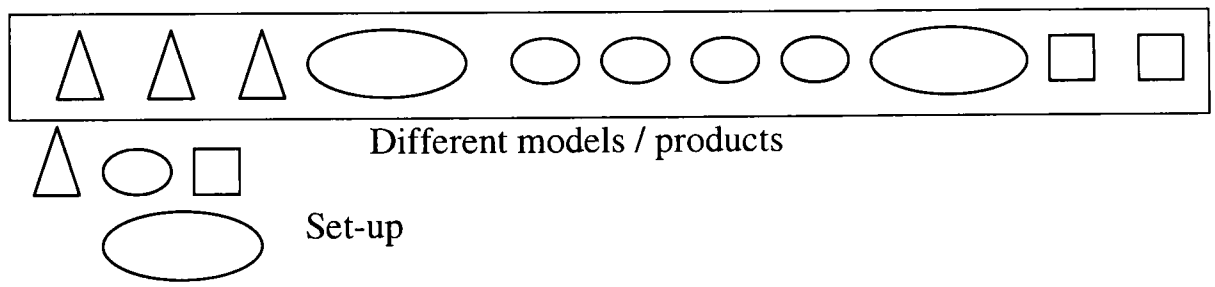


Figure 2-3: Multi-model line



2.4 Characteristics of Mixed-Model assembly lines

Many types of mixed-model assembly lines exist in industry (Wild 1972) and developing a classification system is best approached through identifying and distinguishing between several important design features. The characteristics of mixed-model assembly lines are as follow:

The conveyance system-This can either be a conveyor moving at a uniform speed, or else a stationary system where the product remains stationary at each station, and is only transferred to the next station after completion of the station's task.

The launching discipline-Launch interval is defined as a fixed-time interval at which successive work-pieces are fed into a station. There are two types of launching interval: (a) fixed rate launching in which the launching period is a weighted average of the total assembly time for all products to be assembled over all stations, in which case the production cycle time must be less than or equal to the model cycle time, and (b) Variable rate launching in which the launching period is the task time of the last product launched at the first station so the worker at the first station can start working on the next product immediately after completing work on the current product. When units are launched at a variable rate and in an arbitrary order, it is necessary for optimum utilization of the line that the worker cycle time be equal to the maximum model cycle time (Kilbridge and Wester, 1963).

The product's link to the conveying system-The product can be removed from the conveyor or else held stationary relative to the conveyor movement. Products that cannot be moved independently of the conveyor movement are referred to as being products fixed, whereas those with independent movement are called product moveable

Diversification in the customer's demand and the desire to produce to order, the required product in the required quantity, and to avoid large stocks of finished goods necessitate the use of one assembly line to handle mixed-model production schedules requiring the progressive assembly of several models of the same general product. The Just-in-time (JIT) production system has been well known world wide for achieving high efficiency. Just in time was originally developed by Toyota Motor Company in Japan to respond to various changes of circumstance such as fluctuating demand and diversified products (Ohno, 1986). To deal with frequent changes in demand and increasing variety in models, producing mixed-models on an assembly line has become widely adopted in the manufacturing industry around the world to

achieve flexibility and smooth part usage rates. The effective utilization of these lines requires that the following problems be solved (Okamaura and Yamashina 1979);

1- Determination of line cycle time,

2- Determination of the number and sequence of stations on the line,

3- Line balancing, a line balance (feasible task assignment) represents a feasible solution of a balancing problem. A feasible solution is characterized by the following properties: Because of its indivisibility each task is assigned to exactly one station. The precedence constraints are fulfilled, no task *j*, which must succeed a task, *i* is assigned to an earlier station than *i*. The station times of all stations (or the average station times) do not exceed the cycle time.

4- Determination of the sequence schedule for producing different products on the line.

2.5 Introduction to Just-in-time Systems

Notwithstanding the inherent controversies surrounding its definition, measurement and interpretation (Bailey & Hubert 1980; Caves et al. 1980), productivity analysis is generally recognized as an effective tool for evaluating the past performance and for assessing the effectiveness (both positive and negative) of actions taken to improve efficiency (Eilon et al. 1975, Eilon 1962). It is widely recognized that efforts to achieve the highest possible productivity will in themselves reap economic benefits (e.g. see Nelson 1981, Cosmetatos 1983). With the rapidly fluctuating economic landscape of the 1980's many North American manufacturers have been obliged to adjust their approach to production in order to retain and regain their foothold in an increasing more competitative global marketplace. The entrenched North American mindset of using the longest lead time for producing the largest lot size to obtain the lowest price would hold if one were basing price strictly in terms of manufacturing

costs (e.g. see Chyr et al. 1990). However, where is the benefit in achieving these price reductions if the manufacturer is not producing exactly what the customers want at exactly the time when they want it? One innovative approach used in repetitive (& other) manufacturing industries to answer this question is the just –in- time (JIT) strategy (Hannah 1987).

Ways of measuring and evaluating the effectiveness of JIT systems are as plural and divers (and also as controversial) as those for measuring the effectiveness in tradional manufacturing. In this study the focus will be on measuring the effectiveness of simulated annealing to scheduling in a JIT system.

The JIT approach was developed in post-second world war Japan at Toyota Motor Company by Taiichi Ohno, the former vice-president of manufacturing (Monden 1983). Just-in-time emphasises a continual process of removing waste and inefficiency from the production environment through high quality (Crosby 1984) and small lead times (Ohno 1988). The focus is one of solving production problems so that manufacturing operations become increasingly more efficient (Suri & Treville 1986). "*Just-In-Time*" refers to the actual production system whereby operations are activated just (and only) as they are needed. The JIT concept goes beyond the strict bounds of the production function and leads to more of a company-wide philosophy and way-of life. Companies using JIT treat production as an evolutionary operation (e.g. see Lamberecht & Decaluwe 1988). Bottlenecks to efficient production are identified, focused upon and eliminated until new bottlenecks appear, thereby regenerating the improvement cycle.

The successful implementation of a JIT system necessitates a high degree of teamwork and cooperation by all employees of a company (Crawford 1990; Johnson et al. 1989; Oliver 1990). Fukual (1983) describes how this team approach can be

implemented. Wastes such as scrap and rework must be reduced and eliminated through the high quality of the production itself (Ebrahimpour 1985). Total Quality Control (TQC) is the system approach to quality improvement within a company in which all employees are responsible for the monitoring of product quality. TQC is a part of the JIT process (see Hendrick 1987) and its use within Japanese companies is discussed in Ishikawa (1985). Hall (1983) describes in detail how the JIT process by producing only the necessary parts in the necessary quantities at the necessary times results in very low levels of all types of inventory (i.e. raw materials, work-in process and finished goods) which saves space (both in the warehouse and on the shop floor) and frees up resources which would normally be tied up in the idle inventory.

Just in time has been widely accepted and gained remarable attention among researcher as well as practitioners (Huang and Kusiak, 1998; Baykoc and Erol, 1998; Thesen, 1999; Kufteros, 1999; Ozbayrak et al 2002 and 2003).

Monden (1981a; 1981b; 1981c; 1981d) in a series of articles outlined the key features of Toyota's JIT system. Most of these features are common to any JIT system (e.g. see Stevenson 1990; Dilworth 1986; Gaither 1987) and will be summarized below:

2.5.1 Low Inventories

The most notable feature of JIT systems is the resultant low level of inventory. As indicated, all the types of inventory are reduced freeing up both space and resources. Production problems (i.e. poor quality, unreliable vendors etc), which might be hidden in the inventory of a traditional manufacturing environment are exposed in the JIT system and may be corrected in the evolutionary approach taken in problem solving.

2.5.2 Stable and Level Production Rates

A JIT production system requires a uniform rate of production within the system (Monden 1981,c). Gaither (1986) notes that Toyota in its monthly production tries to keep the same production schedule for every day of the month. Thus if only a few of a particular (automobile) model were needed in a month, some would be assembled in each day of the month. *If JIT is to work stable and level production schedules are a must.*

2.5.3 Reduction of Lot Sizes

JIT systems require small lot sizes in the production process (South 1986). In order to produce these small lots (often of size 1) it is necessary that the changeover cost from one product to another (measured in time and other resources) be negligible. These small sizes lead directly to reduce inventory levels throughout the factory. Fukuda (1983) describes the reduction of set-up times for machines to the desired one-touch (i.e. very rapid) set-ups (see also Spence & Porteus 1987).

To facilitate quick changeover, JIT systems tend to arrange equipment to handle streams of parts and products with similar processing requirements. Monden (1983) describes the U-turn format for the arrangement of machines, where small groups of workers attend several machines arranged in the "U" pattern corresponding to the flow of parts within a particular machine group. This grouping of machines and the reduction of in-process inventory allows for smaller factories to be developed if JIT is employed. Workers are expected to be proficient in the operation of several machines and must be able to assist their fellow workers in maintaining the schedule should someone fall behind. This entails that workers be multi-functional and capable of handling their own set-ups, making minor adjustments to the machines in their charge, and being able to perform minor repairs. This contrasts sharply to the strong

opposition to the multi-tasking requirements by the manufacturing union (Inman & Mehra 1989). Workers are responsible for checking the quality of their work and are expected to contribute to the problem solving process both for current problems and for those that may occur in the evolutionary process of the JIT system (Ishikawa 1985).

2.5.4 Pull Systems

A facility operating under JIT uses what is known as a "pull" system (for a more complete description of push and pull systems see Pyke & Cohen1990; Detoni et al. 1988). In this system work is moved from operation to operation only in response to demand from the next stage in the process. The control of this movement is the responsibility of the subsequent operation. Each workstation pulls the output from the proceeding station only it is needed. Output of the finished goods for the entire production facility is pulled by customer demand. Communication occurs backward through the system from station to station. Work moves "*Just-In-Time*" for the next operation and the flow of work is coordinated in such a way that the accumulation of excessive inventory between operations is avoided (Im & Schonberger 1988).

The communication of the demand can be achieved in a variety of ways. The most commonly used device is some variant of the kanban card system used at Toyota (the terms JIT production systems and kanban production systems are often interchangeable). When materials or work are required from the preceding station, a kanban card is sent authorizing the moving or work for parts. No part or lot can be moved or worked on without the use of these cards. Monden (1981b) describes the use of kanban cards at Toyota to control their JIT process.

The points above summarize the key features of any JIT system. To function successfully a facility using the JIT approach must integrate all of these factors. The major benefits arising from the use of JIT (Gaither 1987) are reduced inventory levels of raw materials, work-in-process and finished goods, increased product quality and a reduction of scrap and rework; a reduction in lead times and a greater flexibility in changing the production mix; a smoother flow of production with shorter set-up times, multi-skilled workers and fewer disruptions due to quality problems; reduced space requirements due to an efficient plant layout and lower inventory levels; and because the focus in JIT manufacturing is on solving production problems, the manufacturing operations become increasingly more streamlined and problem-free.

2.6 Just-In-time methods

Monden (1983) shows that the sequence of introducing models to the mixed-model assembly line is different due to the different goal or purposes of controlling the line. Monden defines that in sequencing mixed-model final assembly lines, two goals are important:

1- Levelling the work loads (total assembly time on each workstation on the line) among all stations within the line (goal chasing 1), and

2- Keeping a constant rate of usage for every part used by the line (goal chasing 2).

Goal chasing (1) recognizes that all models may not have the same operation time at any one workstation on the line. Most assembly lines may have enough flexibility to adjust to this without slowing down or stopping. However, if models with relatively longer operation times are successively scheduled, line stoppages or incomplete work may inevitably result. This goal seeks to smooth out the workload on the final assembly line to reduce the line inefficiencies such as idleness, work deficiency, utility work and work congestion.

Notation

α number of distinct product types, models

i index denoting a product type (i=1,2,..., α)

A_i number of product type i to be assembled

β number of parts whose usage is to be levelled

j index denoting a part (j=1,2,..., β)

b_{ij} number of j parts (j = 1,..,β) needed for producing one unit of a product type A_i (i = 1,..,α).

Q total number of products to be assembled in the production period, (i.e. the number of positions in the sequence)

$$= \sum_{i=1}^{\alpha} A_i$$

N_j total number of j parts required in the production period

$$= \sum_{i=1}^{\alpha} A_i \times b_{ij}$$

k index denoting the position in the sequence (k=1,2,...,k)

x_{jk} number of j parts required for producing products scheduled in positions from 1 to k.

$\frac{N_j}{Q}$ = Average necessary quantity of part j per unit of a product.

$\frac{k.N_j}{Q}$ = Average necessary quantity of part j for producing k units of Products.

Goal chasing 1 schedules a product one position at a time. At position k, It experimentally schedules each product and calculates the resulting distance from the cumulative part usage goal. The product that has the smallest distance is scheduled in position k.

Let S_k be the set of product types that can be scheduled in position k. when the algorithm begins, S is the set of all product types. When the A_i-th unit of product i is scheduled, then i is removed from the set S_k

Ideally, part j usage would be constant. Then, part j ideal cumulative usage would be linear over the production sequence-from zero at position zero to N_j at position Q. There is J of this ideal cumulative usage lines-one for each part. Each is a function of the position in the sequence. In equation (2.1), D_{ki} is the Euclidean distance from this ideal line to the actual cumulative part usage that would occur if product i were scheduled in position k.

The algorithm for goal chasing 1 is:

Step 1 Set k=1, $x_{j,k-1} = 0, (j = 1,..,\beta), S_1 = \{1,2,..,\alpha\}$.

Step 2 Set as kth order in the sequence schedule the product i^* such that

$$D_{ki^*} = \min_i \{D_{ki}\}, i \in S_{k-i},$$

$$\text{Where } D_{ki} = \sqrt{\sum_{j=1}^{\beta} \left(\frac{k.N_j}{Q} - x_{j,k-1} - b_{ij} \right)^2} \qquad (2.1)$$

Step 3 If all units of product i^* were ordered and included in the sequence schedule, then set

$$S_k = S_{k-1} - \{i^*\}.;$$

else set $S_k = S_{k-1}$.

Step 4 If $S_k = \emptyset$ (empty), the algorithm ends.

If $S_k \neq 0$, then compute $x_{jk} = x_{j,k-1} + b_{i^*j}, (j = 1,..,\beta)$ and go back to step 2 by setting k=k+1

According to goal chasing 2, in the Toyota production system, the variation in production quantities or conveyance times at preceding processes must be minimized. To dò so, the quantity used per hour (consumption speed) for each part in the mixed-model line must be kept as constant as possible. Toyota's sequencing method is designed to reach this second goal. Goal chasing 2's algorithm has the same as goal chasing method 1 except for step 2. Step 2 of the GC 2 algorithm is:

$$D_k = \sqrt{\sum_{j=1}^{\beta} \left(\frac{k.N_j}{Q} - x_{jk} \right)^2} \qquad (2.2)$$

This method chases the goal by scheduling the product whose parts will do the most "catching up"- less any getting ahead.

Unlike GC1, this method does not tentatively schedule each product and check the effect on cumulative part usages. Step 2 of GC1 sums the square of the distances of all parts from their ideal cumulative usage given the tentative scheduling of *i* in stage *k*. Step 2 of GC2, however, simply sums how far behind product *I* constituent parts

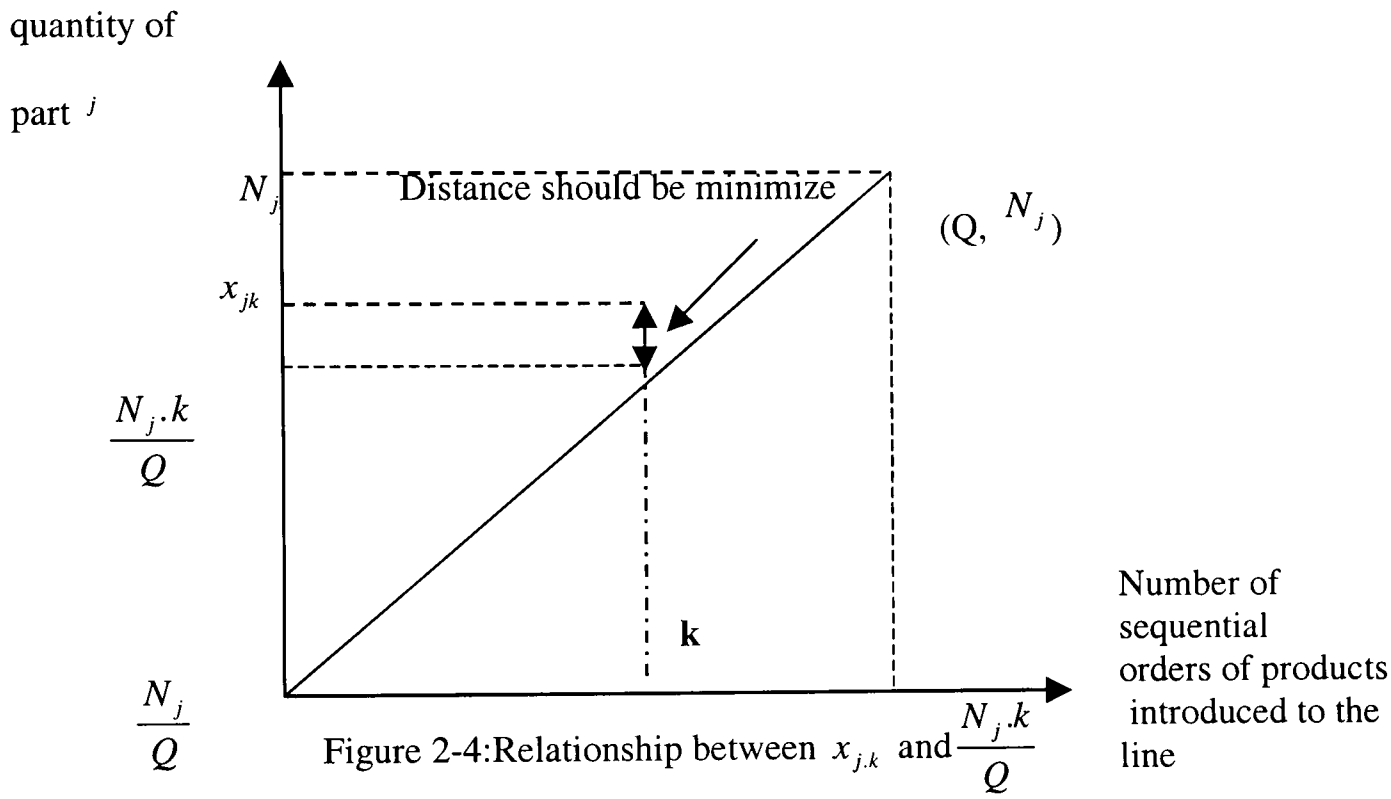


Figure 2-4:Relationship between $x_{j.k}$ and $\frac{N_j.k}{Q}$

are. In fact, b_{ij}, the number of part *j* units required by one unit of product *i*, does not even appear in method 2. This method breaks down when the b_{ij}, s are greater than 1.

In order to keep the consumption speed of a part *j*s constant, the amount of x_{jk} must be as close as possible to the value of $\frac{k.N_j}{Q}$. This is the basic concept underlying Toyota's sequencing algorithm and shown in Figure 2.4.

2.6.1 Goal chasing method 1

Distinctive computation schemes proposed for the solution of the goal chasing one are now discussed, emphasizing the ideas of these methods and their application rather than a detailed explanation of their theories.

2.6.1.1 Thomopoulos (1967)

Thomopoulos presents an approach that may be described as a greedy procedure to minimize the total cost of labour inefficiencies. For each position in the sequence, the inefficiency cost incurred by placing each model in that position is computed, and the model with the lowest cost is chosen. The procedure starts with the first position in the sequence and then sequentially considers each of the following positions. For one problem instance with six different models, the procedure performs well in comparison to five hundred randomly generated sequences.

2.6.1.2 Dar-El and Cother (1975)

Dar-El and Cother address the problem of sequencing so as to minimize the overall length of the assembly line while ensuring that there is no interference (i.e., utility work, idle time, congestion, and etc.). They propose a heuristic, which begins with a lower bound on the length of each station, which is equal to the maximum processing time of all models at station. Whenever infeasibility occurs, the length of a station will increase. The sequencing procedure is based upon the objective of spreading out the jobs within each model type as smoothly as possible. At any point in the sequence, the rank of a model is computed as the difference between the number of jobs of that model that should have been sequenced thus far and the number of jobs of that model actually sequenced. The model is considered in descending order of this ranking, and feasibility with respect to the constraint of no interference is checked. The first model that passes the feasibility check is selected for the given position in the sequence. If no model passes the feasibility check, the station length is increased by equal amounts and the process is repeated.

2.6.1.3 Okamura and Yamashina (1979)

Okamura and Yamashina propose a heuristics method to minimize the maximum distance that a worker must go from the origin of the station to complete work on all jobs. The heuristic involves moving jobs from one location to another in the sequence, or interchanging two jobs, to reduce the maximum distance. Candidate jobs to be moved or interchanged are selected from the regeneration cycle, which has the maximum distance from the origin. Here, regeneration is defined as an instant when the operator returns to the origin of his station and finds no work remaining to be done on jobs in the station. In the case of interchange, the other candidate job may be selected from a regeneration cycle with the smallest maximum distance from the origin.

Okamura and Yamashina suggest a procedure, which uses many different initial sequences. For each initial sequence, an improvement routine is applied in which jobs are moved until no improvement occurs, followed by an interchange of jobs until no improvement occurs. The best of the several sequences is the solution. They present empirical results for problems with up to 100 jobs, which suggest that the heuristic perform almost as well as a branch and bound procedure with a CPU time trap of 2 seconds.

2.6.1.4 Xiaobo and Ohno (1994)

Xiaobo and Ohno proposed an optimal sequence of units that minimizes total line stoppage.

Notation

t_m^k operation time of mode m, m=1,..., M by worker k;

M units of different models to be sequenced m=1,2,...,M

t^k passage time when a unit passes through workstation k, (the length of workstation k is $v_c t^k$);

p_n^k starting position of worker k for the nth unit in sequence;

it_n^k idle time of worker k after completing operation for the nth unit in sequence;

st_n^k line stoppage time caused by worker k for the nth unit in sequence;

IT^k total idle time of worker k;

ST^k total line stoppage time caused by worker k;

The idle time and line stoppage time before the line stops is:

$$it_n^k = \frac{1}{v_c}\max\left\{t_c v_c - p_n^k - v_c t_{\pi(n)}^k, 0\right\}$$

$$st_n^k = \frac{1}{v_c}\max\left\{p_n^k + v_c t_{\pi(n)}^k - v_c t^k, 0\right\}$$

When line stoppage occurs at some station, the whole assembly line will stop. Suppose that line stoppage occurs during the time interval of length S at some workstation. The idle time and the line stoppage will be:

1.If the remaining operation time is $\hat{t}_{\pi(n)}^k$

a- when $\hat{t}_{\pi(n)}^k \geq S$

$$it_n^k = \frac{1}{v_c}\max\left\{t_c v_c - p_n^k - v_c\left[t_{\pi(n)}^k - S\right], 0\right\}$$

$$st_n^k = \frac{1}{v_c}\max\left\{p_n^k + v_c\left[t_{\pi(n)}^k - S\right] - v_c t^k, 0\right\}.$$

b- when $\hat{t}_{\pi(n)}^k < S$. If $p_n^k + v_c\left(t^k{}_{\pi(n)} - \hat{t}_{\pi(n)}^k\right), t_c v_c$, then

$$it_n^k \frac{1}{v_c}\left[t_c v_c - p_n^k - v_c\left(t_{\pi(n)}^k - \hat{t}_{\pi(n)}^k\right)\right] + \left(S - \hat{t}_{\pi(n)}^k\right).$$

Otherwise

$$it_n^k = 0.$$

2-Worker k is idle (unit did not enter his station)

$$it_n^k = \frac{1}{v_c}\left[t_c v_c - p_n^k - v_c t_{\pi(n)}^k\right] + S.$$

The objective function of the sequencing problem is:

$$\text{minimize} \sum_{k=1}^{k} \sum_{n=1}^{M} st_n^k \ .$$

They found an optimal sequence to minimize the total line stoppage time for the mixed- model assembly line in the just-in-time production system. Lower and upper bounds of the total line stoppage time and the total idle time are derived and the branch-and-bound method is devised based on these bounds.

2.6.1.5 Xiaobo and Ohno (1997)

Xiaobo and Ohno proposed two algorithms to find an optimal sequence for mixed model assembly line to minimize the total conveyor stoppage time. A branch-and bound method is used to find an optimal solution for small size problems, and simulated annealing used to find sub-optimal solution for large-scale problems.

Notation

$\pi(n)$ the nth unit in the sequence;

$t_{\pi(n)}^k$ operation time of the nth unit in the sequence by worker k;

L^k length of work station k;

The objective function is to minimize the total conveyor stoppage time:

$$ST = \sum_{k=1}^{k} \sum_{n=1}^{M} st_n^k$$

The initial starting positions p_1^k, k=1,..,K, are equal to 0. Before a conveyor stoppage occurs, the line stoppage and the next starting position on n is:

$$st_n^k = \frac{1}{v_c} \max\left[p_n^k + v_c t_{\pi(n)}^k - L^k, 0 \right] = 0$$

$$p_{n+1}^k = \max\left\{ p_n^k + v_c \left[t_{\pi(n)}^k - t_c \right], 0 \right\},$$

Where k=1,...,K.

Suppose a line stoppage occurs at a station ℓ when the worker is operating the nth unit $(p_n^l + v_c t_{\pi(n)}^l - L^l, 0)$ and that the whole line is stopped. The starting position of worker k and a subsequent conveyor stoppage are influenced. If a conveyor stoppage lasts for a time interval of length S, for $k=\ell$, the starting position and the conveyor stoppage will be:

$$st_n^k = S = \frac{1}{v_c}\left[p_n^k + v_c t_{\pi(n)}^{\ell} - L^k\right],$$
$$p_{n+1}^k = L^k - v_c t_c.$$

At the moment of the conveyor stoppage occurs, for finding the starting position of worker k and the subsequent conveyor stoppage by the worker four cases may arise: for $k \neq \ell$,

Case 1: Worker k is operating for the n'th unit, and the remaining operation times $\hat{t}_{\pi(n)}^k \geq S$. the starting position and the line stoppage are:

$$st_{n'}^k = \frac{1}{v_c}\max\left\{p_{n'}^k + v_c\left[t_{\pi(n')}^k - S\right] - L^k, 0\right\},$$
$$p_{n'+1}^k = \max\left\{p_{n'}^k + v_c\left[t_{\pi(n')}^{\ell} - t_c - S\right], 0\right\};$$

Case 2: Worker k is operating for the n''th unit, and remaining operation time $\hat{t}_{\pi(n'')}^k, S$ and $p_{n''}^k + v_c\left[t_{\pi(n'')}^k - \hat{t}_{\pi(n'')}^k\right] \leq t_c v_c$. The conveyor stoppage time and the starting position of the worker k are:

$$st_{n''}^k = 0,$$
$$p_{n''+1}^k = 0;$$

Case 3: Worker k is operating for the n'''th unit, and the remaining operation time $\hat{t}_{\pi(n''')}^k, S$ and $p_{n'''}^k + \left[t_{\pi(n''')}^k - \hat{t}_{\pi(n''')}^k\right].t_c v_c$. The starting position of worker k and the conveyor stoppage time are:

$$p_{n'''+1}^k = p_{n'''}^k - v_c\left[t_{\pi(n''')}^k - \hat{t}_{\pi(n''')}^k\right] - t_c v_c,$$
$$st_{n'''+1}^k = \frac{1}{v_c}\max\left\{p_{n'''+1}^k + v_c\left[t_{\pi(n'''+1)}^k + \hat{t}_{\pi(n''')}^k - S\right] - L^k, 0\right\},$$
$$p_{n'''+2}^k = \max\left\{p_{n'''+1}^k + v_c\left[t_{\pi(n'''+1)}^k + \hat{t}_{\pi(n''')}^k - S\right] - v_c t_c, 0\right\};$$

Case 4: Worker k is idle after completing the operation for the n''''th unit.

$$st^{k}_{n''''} = 0,$$
$$p^{k}_{n''''+1} = 0.$$

2.6.1.6 Kim et al. (2000)

Kim et al. present a new method using a coevolutionary algorithm that can solve the line balancing and model sequencing problems at the same time. To enhance population diversity and search efficiency, a localized evolution strategy is adapted and methods of selecting symbiotic partners and evaluating fitness are developed.

2.6.1.7 Yano et al (2001)

Yano et al. propose a new line balancing approach for mixed-model assembly lines with an emphasis on how the assignment of tasks to stations affects the ability to construct daily sequences of customer orders that provides stable workloads on the assembly line, while also achieving reasonable workload balance among the stations.

2.6.2 Goal chasing method 2

Goal chasing 2 is to keep constant rates of part usage by the line. This goal is considered in the Toyota production system to be more important than goal 1 (Monden, 1983). This is because production would not be realized without achieving this goal. To keep constant rates of part usage implies an objective function that minimizes the total variation between the actual production rate and the ideal (constant) production rate for every model produced on the assembly line. This goal is to minimize the one-level variation. That is, the quantity of each part used by the mixed-model assembly line per unit time should be kept as constant as possible. This is called leveling, or balancing the schedule. Planning the final assembly schedules requires careful thought because they are the key points at which production is

levelled. If fabrication operations are set to supply final assembly system, then the final assembly schedule is the key that triggers the system. All planning is directed toward being able to develop and maintain level final assembly schedule. If these can be developed, production plans leading to them can be developed. There are several sequencing heuristics and an optimal procedure for minimizing the one-level variation at final stage of production lines. These heuristics are as follow:

2.6.2.1 Common Notations

M number of models or (product types)

k number of stages per time period, k=1,2,..., D_T

d_i product demand, d_1, d_2,.., d_n

D_T total product demand (units of products to be produced), $D_T = \sum_{i=1}^{n} d_i$

i index, product to be produced, i=1,2,...,n

r_i production ratio, the proportion of product i demand to the total product demand, $r_i = \frac{d_i}{D_T}$

$x_{i,k}$ total cumulative production of product i in periods 1 through k. Where k=1,2,..., D_T

j unit of product (j=1,2,..., d_i)

Stage the term will be used to indicate the order in which the products (models) are scheduled.

2.6.3 Miltenburg (1989)

Miltenburg provide a schedule for the final stage of multi-stage production system. He assumes that the products required approximately the same number and the mix of parts. Miltenburg achieves a constant rate of part usage by considering only the demand rates for the products, and ignoring the resulting part demand rates. The objective then is to schedule the assembly line so that the proportion of product produced (over a time period) to the total production is as close to the desired proportions of product as possible for all time periods.

The Miltenburg's sequencing procedures starts with algorithm 1 to find the nearest integer point $(m_{1,k}, m_{2,k}, \ldots, m_{n,k})$, at each stage k, to $kr_1, kr_2, \ldots, kr_n$ for $\forall_i$. The optimal solution is found by algorithm 1, unless there is an infeasible production schedule, i.e., $m_{i,k} < m_{i,k-1}$; if the schedule is not feasible, then algorithm 3 goes on with the solution of algorithm 1 by finding the first stage ℓ where $m_{i,\ell} < m_{i,\ell-1}$. Setting δ to the number of different models for which $m_{i,k} < m_{i,k-1}$, and beginning at stage $\ell - \delta$, either heuristic 1 or heuristic 2 is used to schedule stages $\ell - \delta, \ell - \delta + 1, \ldots, \ell + \omega$. Where $\omega \geq 0$ and $\ell + \omega$ is the first stage in which the schedule determined by the heuristic matches the schedule determined by algorithm 1. Algorithm 3 with heuristic 2(M-A3H2) bases its scheduling decision on two successive stages by considering the sum-of-deviations for stages k and k+1.

2.6.4 Kubiak and Sethi (1991)

Kubiak and Sethi developed an optimal procedure to minimize the total one-level variation that is represented by a nonnegative convex function. This method requires the problem to be transformed into an assignment problem by a pair of indexes from the set I ={(i,j), i = 1,2,..n;j = 1,2,.., d_i } so the item (i, j) signifies the jth unit of the ith type product manufactured. Then all we need to complete the assignment problem is to obtain appropriate costs to specify the objective function. A regular assignment procedure then assigns all units of products to all possible positions in the sequence. If we let, C^i_{jk} be cost of assigning the j-th unit of product i to the k-th period, Z^{i*}_j Ideal position of j-th unit of product i to be produced, ℓ time period, ℓ = 1,2,.., D_T and ψ^i_{jl} represent the excess cost of having j units product of type i produced by period ℓ.

Assignment problem

Let $Z^{i*}_j = [(2j - 1)/2r_i]$ be the ideal period or position for the j-th unit of product i to be produced, i =1,2,..,n, j =1,2,.., d_i. If $k = Z^{i*}_j$, the j-th unit of product i has its ideal

position and $C^i_{jk} = 0$. If the j-th unit is produced too soon, k < Z^{i*}_j, then the excess inventory costs $\psi^i_{j\ell}$ are incurred in periods from ℓ = k to $\ell = Z^{i*}_j - 1$. On the other hand, if the j-th unit is produced too late, $k > Z^{i*}_j$, then the excess shortage costs

$$C^i_{jk} = \begin{cases} \sum_{\ell=k}^{z^{i*}_j - 1} \psi^i_{j\ell} & \text{if } k < z^{i*}_j \\ 0 & \text{if } k = z^{i*}_j \\ \sum_{\ell=z^{i*}_j}^{k-1} \psi^i_{jl} & \text{if } k > z^{i*}_j \end{cases}$$

ψ^i_{jl} are incurred in period from $\ell = Z^{i*}_j$ to $\ell = k$-1, which is

Where $\psi^i_{j\ell} = \left|(j - lr_i)^2 - (j - 1 - lr_i)^2\right|$,

$(i, j) \in I = \{(i, j)$: i = 1,...n, j=1,...,d_i }, l =1,..., D_T.

Let

$$x^i_{jk} = \begin{cases} 1 & \text{if } (i, j) \text{ is assigned to period } k; \\ 0, \text{otherwise} & \end{cases}.$$

* refers to values in an optimal solution.

The assignment problem is:

$$\text{Minimize} \quad \sum_{k=1}^{D_T} \sum_{(i,j)\in I} C^i_{jk} x^i_{jk}$$

Subject to

$$\sum_{(i,j)\in I} x^i_{jk} = 1, k = 1,2,..,D_T$$

$$\sum_{k=1}^{D_T} x^i_{jk} = 1, (i, j) \in I$$

$$x^i_{jk} = 0 \text{ or } 1,\ k = 1,2,..,D_T;\ (i, j) \in I$$

2.6.5 Inman and Bulfin (1991)

Inman and Bulfin provide earliest due date formulation and solution procedure to sequence a mixed-model just-in- time assembly system. They consider a facility, which manufactures n distinct products on a unit-by-unit basis with the objective to minimize the sum of both squared earliness and tardiness. It is first assumed that the processing times of all products are common and, for the convenience of discussion, it is also assumed that this common processing time is equal to 1. They first assign an "ideal due date" to every unit of each product to be sequenced on the assembly line. The ideal due date represents the times that all products are evenly spaced within the planning horizon (D_T). Inman and Bulfin algorithm is mathematically different, but intuitively similar to the objective function of Miltenburg (1989), and Kubiak and Sethi (1991).

And notation is:

X_{it} Cumulative production of product i at time t,

d_{it} The cumulative demand for product (i) at time t, $d_{it} = tD_T / (\sum d_i)$

t_{ik} The time at which the k-th unit of product (i) is needed

(ideal due date), $t_{ik} = [(k - 1/2)D_T]/d_i$, i=1,2, ,n and k=1,2,..., d_i

y_{ik} The time at which the k-th unit of product (i) is actually produced (completion time).

To minimize inventories and level parts availability; it appears appropriate to minimize the deviation between cumulative production and cumulative demand. To accomplish that they develop the following objectives:

$$\sum_{i=1}^{M} \sum_{t=1}^{T} (X_{it} - d_{it})^2 \qquad (O_1)$$

$$\sum_{i=1}^{M} \sum_{t=1}^{T} |X_{it} - d_{it}| \qquad (O_2)$$

By substituting (completion time) y_{ik} and t_{ik} (due-date) in above formula they get the following:

$$\sum_{i=1}^{M}\sum_{t=1}^{T}\left(y_{ik}-t_{ik}\right)^{2} \qquad (O_3)$$

$$\sum_{i=1}^{M}\sum_{t=1}^{T}\left|y_{ik}-t_{ik}\right| \qquad (O_4)$$

Define the time when the k th unit of product i needed-due date as;

$$t_{ik} = \left[(k-1/2)D_T\right]/d_i \qquad i=1,2,..,n \text{ and } k=1,2,..,d_i$$

Under (O_3) and (O_4) pair of objectives, they consider each unit of a product as a separate job. The sequencing problem is treated as a single-machine scheduling problem, where t_{ik} is the due date of job (i, k). Their objective (O_3) minimizes the sum over all jobs of the squared deviation between completion times and due dates. Similarly (O_4) minimizes the sum of the absolute value of deviations. They used Earliest Due Date (EDD) rule to find the sequence. The algorithm is as follow:

Step 1. Calculate the ideal due date for each unit (k) of each product (i). Set $k = 1$.

Step 2. At stage k, schedule the unit with the smallest t_{ik} (descending), $k = 1,2,d_i$,

i=1,2,..,M. Once a unit of product is scheduled, eliminate it from further consideration

Step 3. If $k = D_T$, stop; otherwise, let k=k+1; go to step 2

2.6.6 Sumichrast and Russell (1992)

Sumichrast and Russell develope a time spread method to smooth the workload at each assembly line station. The objective function is similar to the goal chasing 1's objective function, except that the individual terms representing quantities of parts and products in GC 1 have been replaced by their corresponding assembly times in time spread method. They compared time-speared method with goal chasing 1, goal chasing 2,and Miltenburg's algorithm 3 using heuristic 2.

Notation

$AT_{\ell(k-1)}$ = the actual time required at station ℓ to assemble the first k - 1 unit of the sequence

s = the number of assembly stations

$t_{i\ell}$ = the assembly time required by model i at station ℓ

T = the total time to assemble all items in sequence at all stations

T_ℓ = the total time to assemble all items in the sequence at station ℓ

Objective function-A model is assigned to position k in the sequence that will

$$\underset{i \in M`}{\text{minimize}} \sum_{\ell=1}^{s} \left(\frac{kT_\ell}{T} - AT_{\ell(k-1)} - t_{i\ell} \right)^2$$

2.6.7 Ding and Cheng (1993a)

Ding and Cheng developed a new algorithm for determination of the sequence to producing different products on the line. They used Miltenburg's mixed integer problem to minimizes the sum of square variations over the next two stages. At stage k of their algorithm, they minimize the sum of squared variations over the next two stages (stage k and k+1).

Algorithm - The algorithm minimizes the sum of squared variations over the next two stages (stage k and k+1).

Step 1. Set k=1 and let $x_{i,0} = 0$ for all i.

Step 2. Among the n products, determine the product s that has the

lowest $x_{i,k-1} - (k + \frac{1}{2})r_i$ (Break a tie arbitrarily.)

Step 3. Among the products, determine the product i that has the

$$\min\{\min_{i\neq s}\{x_{i,k-1} - (k+1)r_i\}, x_{s,k-1} + 1 - (k+1)r_s\}.$$

Step 4. If $s \neq t$ and $x_{s,k-1} - (k+\frac{1}{2})r_s - (x_{t,k-1} - (k+\frac{1}{2})r_t) > \frac{1}{2}(r_t - r_s)$,

Schedule product *i* in stage k. Otherwise, schedule product *s* in Stage *k*.

Step 5. Update $x_{i,k}$'s for all the products. If $k = D_T$, stop; otherwise,

Let *k*= *k*+1, go to step 2

2.6.8 Ding and Cheng (1993b)

Ding and Cheng their algorithm is similar to previous paper. They compared their new heuristic to Miltengburg's algorithm 3 heuristic 2 (M-A3H2). In this experiment they used 9 problem sets, which where conducted by Sumichrast and Russell (1990). The results show that: 1- the new heuristic procedure is similar to the M-A3H2 method in solution quality and much more efficient in computational time. 2-The procedure of the new heuristic approach is extremely simple.3- new heuristic procedure requires much less computer memory space than M-A3H2. 4-Simplicity of the new method can be viewed as a better alternative compared to M-A3H2 method for sequencing mixed-model assembly lines in just-in-time production systems.

2.6.9 Ding et al (2000)

Ding et al. consider the two-stage part-level variation as a heuristic solution approach, and in understanding the effect of minimizing the product-level variation. A simplification of the two-stage method is present. A transformed two-stage heuristic using product-level terms for reducing the par-level variation in sequencing mixed-model assembly line is provided.

2.6.10 Summary

The first term in the GC1 formula represents the quantity of component *j*, which would be required to assemble the first *k* items of sequence, if the component usage rate was constant. The other two terms represent actual usage for a particular partial sequence. Models, which may be assigned to position *k* are compared based on the sum of the squared differences over all fabricated components. Position *k* is filled by the required model, which minimizes this measure of no-uniform usage. At each step of building the sequence, the" distance" between the expected use of components and the actual use is minimized. However, no overall minimization is assured.

Goal chasing 2 selects the model for which the expected amount of each component used in the model under consideration most exceeds the actual amount used in models before the current position. In effect, it schedules the model composed of components, which most need to "catch up" to their average usage rates. Goal chasing 2 was created to approximate the solution produced by goal chasing 1 while reducing the computations.

Miltenburg consider the variation in production rates of the finished products in a mixed-model sequencing problem. Under the assumption that all models require the same number and mix of parts, Miltinburg pointed out that minimizing the variation in production rates of the finished products (the product-level problem) achieves minimizing the variation in part usage rates (the part-level problem).

Kubiak and Sethi transformed the product-level mixed-model sequencing problem into an assignment problem.

Inman and Bulfin propose the EDD approach for the product rate variation problem with the objective function of minimizing the sum of both squared earliness and tardiness. Inman and Bulfin objective function is similar to Miltenburg, and Kubiak and Sethi, but their algorithm mathematically different.

The sequencing method of Sumicharst and Russel is similar to the goal chasing 1, except that the individual terms representing quantities of parts and models in goal chasing 1 have been replaced by their corresponding assembly times in Sumicharst and Russell's objective function.

Ding and Cheng's algorithm (1993,a b) is another approach for the product rate variation problem. The procedure minimizes the next-two-stage (k and k+1) total squared deviation when a unit of a model is selected at stage k.

Summary of the all goal chasing 2 and their treatments are depicted in Table 2.1.

Table 2-1:Summary of the goal chasing method 2

Method [ref. No.]	Sequencing Criteria	Solution Method		Solution Quantity
		Description	Treatment	
Miltenburg [1989]	Minimizing the demand rates for the products.	Find the nearest integer point at each stage k. If there is an infeasible production schedule, then algorithm 3 goes on with the solution of algorithm 1 by using heuristic 1 or 2.	$\text{Min}\sum_{k=1}^{D_T}\sum_{i=1}^{n}\left(x_{i,k} - kr_i\right)^2$ $\text{s.t}\sum_{i=1}^{n} x_{i,k} = k$ $0 \le x_{i,k} - x_{i,k-1} \le 1$ $x_{i,k}$ *is* nonnegative int*eger* $\forall_k, \forall_i$	Optimal solution
Kubiak& Sethi [1991]	Minimizing the cost of assigning the *j*th unit of the product *i* to the *kth* period.	Product rate variation reduced to an assignment problem for all products produced on the line over a given time horizon.	$\text{Min}\sum_{k=1}^{D_T}\sum_{(i,j)\in I} C_{jk}^{i} x_{jk}^{i}$ $\text{s.t}\sum_{(i,j)\in I} x_{jk}^{i} = 1$ $\sum_{k=1}^{D_T} x_{jk}^{i} = 1,$ $x_{jk}^{i} = 0,1$	Optimal
Inman&Bulfin [1991]	Minimizing deviations or absolute deviations of the time at which the *k*th unit of product *i* is actually produced from the time at which the k unit of product	EDD approach for the product rate variation with the objective function of minimizing the sum of both squared earliness and tardiness. Defining the time when *j*th unit of product *i* is needed-due date- as $k_{ij} = (j - 1/2)r_i$ and considering each unit of a product as a separate job, the sequencing problem is treated as a single-machine scheduling	$\text{Min}\sum_{i=1}^{n}\sum_{k=1}^{D_T}\left\|y_{ik} - t_{ik}\right\|$	Optimal sequence

	(*i*) is needed (due-date).	problem with earliness/tardiness objective.		
Ding and Cheng [1993a,b]	Minimizing the next two stages of a unit of products.	Find the minimizes of the next-two-stage (k and k+1) total squared deviation when a unit of a product is selected at a stage k.	Minimize $\sum_{k=1}^{D_T}\sum_{i=1}^{n}\left(x_{i,k} - k.\frac{d_i}{D_T}\right)^2$	Optimal
Cakir and Inman[1992]	Finding the best sequencing product to level part usage in cases of non-zero/one product usage matrices.	Modify goal chasing method 2.The algorithm chases the goal of each part for each product, and compare the usage of the part for a particular product with its goal.	$F(i) = W_i(A_i \times \frac{k}{Q} - z_{i,k-1})$ for all $i \in S_k$ $W_i = \sum_{j \in B_i} b_{ij}$ for all i	Optimal
Sumichrast and Russel[1990]		The structure of the objective function in time spread is similar to the goal chasing method 1, except the individual terms representing quantities of parts and models in GC 1 have replaced by their corresponding assembly times in time TS.	Minimize $\sum_{j \in c_j}\left(\frac{kN_j}{Q} - A_{j(k-1)}\right)$	Optimal
Ding and Zhu [2000]	Minimizing transformed two-stage method.	Minimizing of the two-stage variation in the mixed-model sequencing problem of reducing the part-level variation is transformed to product-level forms.	Minimize $\sum_{k=1}^{D_T}\sum_{j=1}^{m}\left(\Phi_{j,k} - k\omega_j\right)^2$	Optimal

3 Simulated Annealing

3.1 Introduction

Simulated Annealing (SA), a method for obtaining good solutions to difficult optimization problems, has received much attention. The work began with Kirkpatrick et al. (1993), and Cerny (1985). They showed how a model for simulating the annealing of solids, as proposed by Metropolis et al. (1953), could be used for such optimization problems, where the objective function to be minimized corresponds to the energy of the states of the solid.

Since then, SA has been applied to many optimization problems occurring in areas such as computer (VLSI) design, image processing, molecular physics and chemistry, and job shop scheduling. There has also been progress on theoretical results from a mathematical analysis of the method, as well as computational experiments comparing the performance of SA with other methods for a range of problems. The aim of this chapter is to provide some understanding and guidance for those interested in using SA, particularly those involved in Operational Research.

Simulated Annealing is not the only heuristic search method to have received attention recently. Glover and Greenberg (1989) summarize the approaches offered by genetic algorithms, neural networks, tabu search, target analysis, as well as SA. Maffioli (1987) shows how simulated annealing can be considered as one type of randomized heuristic for combinatorial optimization problems. Vakharia and Chang (1990) compared the performance of SA to branch-and-bound and to two other scheduling heuristics for solving scheduling problems in cellular manufacturing systems. The problem involves scheduling the families of parts to be processed and scheduling the jobs within each family of parts. The objective is to minimize the makespan. Brusco and Jacobs (1993) applied SA to the 'cyclic staff scheduling' problems. The problem involves assigning staff to schedules where the demand for staff varies between periods in a workday and between days of the week. The objective is to minimize the number of employees to satisfy the forecast demand. Wang et al. (2001) formulates a model to solve the facility layout problem in cellular manufacturing systems. Their model assumes that the demand rate varies over the product life cycle. The objective is to minimize the total material handling cost and solve both inter and intra cell facility layout problems simultaneously.

Palshikar (2001) present a greedy nearest neighbhor approach to search the state-space to locate a lowest cost solution called" hill climbing", and then provide SA algorithm that mimics the natural process of the slow cooling of liquids, which leads to a solid that has the lowest energy. Applying the algorithm to the problem of producing an aesthetic drawing of a given graph.

3.2 Description of Simulated Annealing

Simulated annealing is a numerical optimization technique based on the principles of thermodynamics [23-30]. Annealing refers to the process in which a solid material is first melted and then allowed to cool by slowly reducing the temperature. The particles of the material attempt to arrange themselves in a low energy state during the cooling process. The collective energy states of the ensemble of particles can be considered the "configuration" of the material. The probability that a particle is at any energy level can be calculated by use of the Boltzmann distribution. As the temperature of the material decreases, the Boltzmann distribution tends toward the particle configuration that has the lowest energy. Metropolis et al. first realized that the thermal equilibrium process could be simulated for a fixed temperature by Monte Carlo methods to generate sequences of energy states (Metropolis, et al. 1953). The system is perturbed to yield a new configuration of the particles. The energy level before perturbation (Δf_s) and the energy level after perturbation (Δf_i) are compared. If Δf_s is greater than Δf_i $(\text{i.e.} \Delta f \rangle 0)$, the new perturbed system is accepted as the new configuration of particles. If $\Delta f \rangle 0$, the probability of accepting the perturbed system follows the Metropolis criterion shown in equation 3.1:

$$p = \exp(-\frac{Df}{kT}) \quad (3.1)$$

Where k is the Boltzmann constant and T is a fixed temperature. Using this criterion, the material will eventually reach its equilibrium configuration.

Shaffer (1996) applied this basic concept to numerical optimization problems. Simulated annealing (SA) applies the Metropolis criterion to a series of variable settings (configurations) for the system being optimized. The new variable settings are

obtained by perturbing the current configuration and can be thought of as steps or movements on the response surface. For numerical optimization, the response (objective) function score (R) replaces the energy terms. The concept of temperature is retained. However, it is now combined with k and used as an important control factor. The probability of accepting a detrimental step (i.e., $R > 0$, assuming minimization) is governed by equation 3.2.

$$p = \exp(-\frac{DR}{T}) \quad (3.2)$$

A random number, P, is drawn from a uniform random distribution on the interval [0,1]. If $P > p$, the detrimental step is rejected and a new step taken from the current position. If $p < P$, the detrimental step is accepted and the new configuration replaces the old one. A new step is then taken relative to this configuration. This criterion allows the possibility of a new variable being accepted as the new configuration even when it has a worse response function score than the current configuration. Moves that are very poor (i.e., large ΔR) are less likely to be accepted than moves that are not poor. This feature allows the algorithm to "walk" from local optima. New steps are taken until some termination criterion is reached.

Analogous to the physical process, temperature is slowly reduced causing the probability of accepting a poor move to decrease with time. The schedule by which temperature is reduced is called the cooling schedule and is critical to the success of SA. Two important parameters governing the cooling schedule are the step size for perturbation and temperature. The parameter settings suggested by most researchers are step sizes and temperature values that allow approximately 80% of the poor moves to be accepted (Kaliva, 1991).

The process is continued until an "*equilibrium*" state is achieved, then the temperature is lowered according to the annealing schedule. This procedure is repeated until the system *freezes*. At each temperature, the annealing schedule must allow the simulation to proceed sufficiently long for the system to reach the steady state condition (equilibrium point).

The primary advantage of SA is the ability to move from local optima. Thus, the ability to find the global optimum is not related to the initial conditions (i.e., the

starting point). SA is also very simple to implement. The primary disadvantages to SA are the subjective nature of choosing the SA configuration parameters (e.g. and step size) and that it typically requires more response or objective function evaluations than other optimization approaches. SA has been termed a "biased random walk". Unlike other optimization approaches that attempt to make intelligent moves on the response surface, the steps in SA are taken randomly. Thus, SA is classified as having a weak search heuristic. The fact that SA employs no knowledge of the response surface can be an advantage or a disadvantage depending on the application.

Simulated annealing has found wide application in analytical chemistry, as well as in many other fields of science and engineering [19,22]. Recently, several extensions to the original SA approach have been discussed [25-29]. One of these, generalized simulated annealing (GSA), has been found to simplify the configuration of the cooling schedule [25,26].

3.3 Methodology of simulated annealing

3.3.1 General algorithm

The simulated annealing procedure includes four basic components (Rutenbar, 1989): (*i*) configurations: all of the possible solutions for the combinatorial problem, (*ii*) move set: a set of allowable transitions. These transitions must be capable of reaching all of the configurations; (*iii*) cost function: a measure of how good any given configuration is; and (iv) cooling schedule: the annealing of the problem from a random to a good, frozen solution. Notably, the cooling schedule determines the initial temperature, the rule for decreasing the value of temperatures, the number of iterations for searching better configurations at each temperature, and the time at which annealing should be stopped. The general algorithm of SA derived from analogy with the physical annealing process described above is illustrated in pseudo-code in figure 3.1. In this figure, the outer loop represents the freezing process, in which the temperature is changed, whereas the inner loop described the equilibrium procedure that determines how many exchanges are to be attempted at each temperature.

3.3.2 Annealing procedure

3.3.2.1 Initial temperature:

The number of iterations during the annealing process partly depends on the initial temperature. The procedures for setting the initial temperature may be broadly classified into two types. Most schemes determine the initial temperature as a fixed number prior to execution of the annealing process. Golden & Skiscim (1986) and Bukard & Rendl (1989) set high initial temperatures, while Wilhelm & Ward (1987) starts at a low temperature.

Figure 3-1:Simulated annealing algorithm in pseudo-code.

Select an initial solution $a = a^0 \in S$;
Select an initial temperature $T = T_0 > 0$;
Set temperature change counter t = 0;
Repeat Freezing process
  Set repetition counter $i = 0$;
  Repeat Equilibrium process
    Generate a solution b, a neighbour of a;
    Calculate $\Delta f = f(a) - f(b)$;
    If $\Delta f > 0$ then $a := b$;
      Else if random $(0,1) < \exp(-\Delta f / k_b T)$ then $a := b$;
    $i := i + 1$;
  Until $i = N(t)$;
  $t := t + 1$;
  T: =T (t);
Until stopping criterion true.

In Connolly's method (1990), the initial and final temperatures were determined by information obtained in trials prior to the annealing process. In these trials, a certain number of random moves were performed to record the resulting changes in the objective function. From the results, the minimum value Δf_{min} and the maximum

value Δf_{max} for the changes in the objective function were calculated for these exchanges. Using these values, the initial temperature, T_0 and the final temperature, T_f, were set according to equations (3.3) and (3.4), respectively:

$$T_0 = \Delta f_{min} + \frac{1}{10}\left(\Delta f_{max} - \Delta f_{min}\right), \qquad (3.3)$$

$$T_f = \Delta f_{min}. \qquad (3.4)$$

3.3.2.2 Temperature tuning (cooling):

One of the major issues that related to the annealing schedule is how to cool the temperature during the annealing process. In existing SA schemes, there are several types of procedures for cooling temperature. One is to employ a temperature lowering function based on the annealing schedule. Each annealing scheme has its own individual function. For example, Wilhelm & War (1987), Burkard & Rendl (1989) and McMullen & Frazier (2000) calculate the next temperature, T_{i+1}, using equation (3.5), where the parameter α is usually set close to one. Golden & Skiscim (1986) used Equation (3.6), which reduces the temperature by 1/25 of the initial temperature at each stage.

$$T_{i+1} = \alpha T_i \qquad 0 < \alpha < 1 \qquad (3.5)$$

$$T_{i+1} = T_i - \frac{T_0}{25} \qquad i = 0,\ldots,25 \qquad (3.6)$$

In another method information obtained from trails prior to the execution of annealing process is used. In Connolly (1990) during these trails the initial temperature T_0 and final temperature T_f are determined. The next temperature is calculated by Equation (3.7). In Equation (3.7), M refers to the number of pairwise exchanges examined, and calculated by Equation (3.8). In this equation parameter β usually has a small value, and therefore the temperature reduction proceeds slowly.

$$T_{i+1} = \frac{T_i}{1 + \beta T_i} \qquad \beta = \frac{T_0 - T_f}{M T_0 T_f} \qquad (3.7)$$

$$M = \frac{n(n-1)}{2} \qquad (3.8)$$

Lundy and Mees (1986) used equation (3.9) for cooling temperature.

$$T_{i+1} = \frac{T_i}{1+\beta T_i} \qquad \beta = \frac{T_0 - T_f}{T_0 T_f (M-1)} \qquad (3.9)$$

Where β is a Constant. This decreasing function is rewritten as:

$$T_{i+1} = \frac{T_i}{1+i\beta T_i}, i = 1,2,\ldots N-1 \qquad (3.10)$$

It is clear that when the temperature is too high, a lot of poor uphill moves are accepted. Conversely, when the temperature is too low the probability of falling into a local minimum is very high. Kirkpatrick, et. Al. (1983) mentioned that between these two extremes there is a critical band of the temperatures in the whole annealing process where very slow cooling is required. Equation (3.7) and the procedures for determing the initial temperature used in Connolly (1990) were designed based on this idea.

3.3.2.3 Equilibrium test:

Each SA scheme has its own means for testing Equilibrium State, testing whether the annealing process should proceeded to the next temperature. Most existing schemes test the Equilibrium State according to prescribed criteria independent of the problem characteristics.

For instance in Golden& Skiscim (1986) and Wilhelm & Ward (1987), a test as to whether the Equilibrium State has been reached is conducted after certain duration at each temperature. In both studies, this duration called an "*epoch*". In Golden & Skiscim (1986), it is represented as the *a priori* specified number of attempted exchanges including non-accepted exchanges, while in Wilhelm & Ward (1987) it is defined as *a priori* specified number of only accepted exchanges. In both methods, after the execution of each epoch, the value of the objective function is calculated, and the equilibrium test is conducted based on the current and previous values of the objective function. If the system reaches the Equilibrium State, then the temperature is

lowered. Otherwise, exchanges are repeated during the next epoch at the same temperature, and at the end of that epoch the equilibrium test is conducted again. Golden & Skiscim (1986) as shown in equation (3.11) fined different procedures for determining the equilibrium state. If the mean value of the objective function from the most recent epoch, j, at temperature T_i which is defined as l_j^i in this equation, is sufficiently close to any of the mean values at previous epochs, i.e., $l \in \left\{l_o^i,....,l_{j-1}^i\right\}$, then the systems is assumed to be at equilibrium.

Zegordi et al. (1995) proposed methods adopt "Move Desirability Table" as the equilibrium state of the system. This procedure allows user to test for equilibrium anytime during the annealing process.

On the other hand, Wilhelm & Ward (1987) use Equation (3.12) as the equilibrium test, where $\bar{f}_e$ is the mean value of the objective function during the most recent epoch at temperature T_i, and $\bar{f}'_e$ is the grand mean of the objective function for all preceding epochs at temperature T_i method A were designed based on this idea.

Kirkpatrick et al. (1983) uses the number of accepted and rejected pairwise exchanges as an equilibrium criterion.

$$\left[l_j^i - l\right] \leq \varepsilon \qquad (3.11)$$

$$\frac{\bar{f}_e - \bar{f}'_e}{\bar{f}'_e} \leq \varepsilon \qquad (3.12)$$

3.3.2.4 Termination criterion (Frozen test):

In general SA algorithms, the stopping criterion is specified in advance. These criterions usually depend upon the number of iterations for which an appropriate number of rejected or attempted transitions have taken place. It is shown that the stopping criterion has a great effect on the performance and CPU time of a SA algorithm (Kouvelis, et al, 1992).

3.4 Simulated Annealing and Neighborhood search

The neighbourhood NB(r) is defined as all assignments that can be reached from a given assignment when r elements are exchanged. For pairwise exchanged algorithms, the size of a neighbourhood is $|NB(2)| = \frac{1}{2}N(N-1)$.

At each iteration of the Simulated Annealing algorithm two elements of R are selected and the cost of exchanging their counterparts in objective function is computed. Most researchers choose the next potential assignment at random from the neighbour of the current solution. Connolly (1990) investigates a different selection mechanism, where the neighbourhood is searched sequentially. First, all possible pairwise exchange partners (u, v) are ordered lexicographically,

(1,2),(1,3),.....,(1, N),(2,3),.....,(N - 1, N), and then they are investigated sequentially. Hence, we first investigate the pairwise exchange (u, v) = (1,2), next (u, v) = (1,3),.... and finally (u,v) = (N - 1, N). Then, we start all over again with (u, v) = (1,2). Zegordi et. a. (1995) approach combines the simulated annealing methodology with a specific layout design rule called "Move Desirability Table". In this research we investigates a different selection mechanism, where swapping two members of sequence searches the neighbourhood. The following is an example of how a current solution is modified in the simulated annealing search.

Before Modification: ABABABCBABAAB

After Modification: CBABABABABAAB

3.5 Pairwise exchange algorithm

Ventura (2002) applies an adjacent pairwise interchange of assigned jobs until there was no improvement in the objective function.

Step 0. Set r_0 = r =1, a = a^0 = D_T -1, and q = 0.

Step 1. Consider the pair of sequences { $\sigma_0(r)$, $\sigma_0(r+1)$ }. If sequence $\sigma_0(r)$

And $\sigma_0(r+1)$ belong to the same item type go to step 4.

Step 2. Interchange product $\{\sigma_0(r), \sigma_0(r+1)\}$ to yield the new sequence σ.

Compare the new cost a.

Step 3. If $a < a^0$, replace σ_0 by σ and a^0 by a, and set $a^0 = \max\{r+1, D_T - 1\}$.

And if $q = 0$, set $r_0 = \min\{1, r-1\}$ and $q = 1$.

Step 4. If $q < a^0$, set $r = r+1$ and go to step 1.

Step 5. If $q = 1$, set $r = r_0$, $a = a^0$, go to step 1; else, stop.

4 Mixed-Model Sequencing Problem

4.1 Introduction

Products that have the same physical design and component requirements generally call for traditional assembly type of work in a mass production system. Sequencing the models is a most important issue that needs to be addressed for the setting up and the effective and profitable usage of a typical mixed-model assembly line.

A mixed-model assembly system is designed when the demand for a set of similar products is relatively low and the assembled models require a large warehousing or storage space and/or incur inventory cost. If the models are manufactured in large batches as in a mass production system, they will incur high inventory cost and/or storage or warehousing space. Therefore, a mixed product assembly system will obviously yield a lower output that closely meets the demands of the individual models, resulting in a lower inventory cost for low storage and/or smaller warehousing.

Mixed-model assembly lines can be found in industries such as consumer electronic manufacturing (TV, VCR, camera, telephone sets, scanners), automobile manufacturing (cars, jeeps), discrete manufacturing (household or file cabinets, refrigerators, washers, dryers), assembly of components on printed circuit boards (motherboard for different computer models or telephone sets), and fan manufacturing (table fans, ceiling fans, pedestrian fans).

Chapter 4 present a physical example of the Mixed-Model Sequencing problem and our proposed algorithms to defines the three main objectives involving minimizing the variation of the actual production from the desired production and smoothing the work load on the final assembly line with required set-ups.

4.2 The Sequencing Model

Let D_T = total number of products to be sequenced. A feasible solution I to the sequencing problem is a set of D_T products comprising a feasible schedule. The constraints of the problem are written as:

$$I \subseteq F, \qquad (4.1)$$

$$\|I\| = D_T, \qquad (4.2)$$

Where F is the set of products from which we can select. The set of allocations satisfying (4.1) and (4.2) will be denoted by X. There is O (D_T !) possible solutions. The problem description might be clarified with a physical example.

Suppose there are three different models of a product to be produced. If we are sequencing cars, we might consider one to be red car with a sunroof and manual transmission, the second a blue car with a plain roof and automatic transmission, and the third a white car with a plain roof and automatic transmission. The models can be represented as model A, model B, and model C or simply as A, B, C. Each model requires different demands.

Suppose we are to produce six units of model A's, six units of model B's and one unit of model C. A sequence based on batch techniques might be as follows:

Sequence 1: A, A, A, A, A, A, B, B, B, B, B, B, C.

If we decide to '*mix*' the models to take advantage of some of the results associated with mixed-model sequencing, we might determine two possible sequences as follows:

Sequence 2: A, B, A, B, A, B, A, B, A, B, A, B, C

Sequence 3: C, B, A, B, A, B, A, B, A, B, A, B, A

There are several ways a sequence can be derived. The numbers of possible sequences associated with this example are $\frac{D_T!}{\prod_{m=1}^{M}(d_m!)} = \frac{13!}{6!\,6!\,1!} = 12012$

This is *combinatorial nature of sequencing* problems.

4.3 Objective Function

This research deals with three objective functions:

minimize (UV) (4.3)

minimize (WL) (4.4)

minimize (St) (4.5)

Where (UV) is the variation of the actual production from the desired production objective, (WL) is the smoothing of the workload objective, and (St) the number of required set-ups.

4.3.1 First Objective

An important objective in mixed-model scheduling is to keep a constant rate of usage of work in final assembly line in the system. Consider M product model type with demands $d_1, d_2, ...d_m$ to be produced during a specified time horizon. Assume each product takes a *unit of time* to be produced so the number of product units to be sequenced in each period is given by $D_T = \sum_{m=1}^{M} d_m$ (total demand). If $r_m = \frac{d_m}{D_T}$, then the scheduling objective for the assembly line production is to keep the proportion of the cumulative production of product m to the total production as close to r_m as possible. A matrix x is used to represent the sequence. Let $x_{m,k} \in \{0,1\}$ $(k = 1,..., D_T, m = 1,..., M)$ be a production schedule. If $x_{m,k} = 1$, then product m will be produced during stage k, and $\sum_{m=1}^{M} x_{m,k} = k$, for all k, because only one product can be assembled during each stage. The index $k(k = 1,..., D_T)$

corresponds to the kth product unit to be assembled. We define position k in the sequence as stage k. Miltenburg (1990) developed a theoretical basis of this model. The objective function is as follow:

$$\text{minimize (UV)} = \sum_{k=1}^{D_T}\sum_{m=1}^{M}\left(x_{m,k} - k r_m\right)^2$$

Subject to:

$$\sum_{m=1}^{M} x_{m,k} = k \qquad k = 1,\ldots,D_T \tag{4.6}$$

$$0 \le x_{m,k} - x_{m,k-1} \le 1, \quad m = 1,..,M, \quad k = 1,..,D_T \tag{4.7}$$

$$x_{m,k} \text{ is a non-negative integer } \forall m, \quad \forall k, \tag{4.8}$$

The objective function penalizes the difference between the cumulative production and the ideal production amount of model m over stages 1 through k. Constraint (4.6) ensures that exactly k units are scheduled in periods 1 through k. Constraints (4.7) and (4.8) guarantee that for each model either one unit is scheduled in a given period or else it is not schedule at all.

When products are systematically placed in a sequence once at a time, they are placed in '*stage*'. Determining which product to sequence first is the determination of the stage 1 product. At stage two there are two products in the sequence, etc. At each stage of the sequence, we can calculate the variation of actual production from the desired production.

Example: Consider the production sequence where there are six units of product A, six units of product B and one unit of product C. One possible sequence would be BAABBACABBAAB. This production sequence results in nine set-up costs and usage variation equal to 4.62.

Table 4.1 shows the best possible solution for the particular example.

Table 4-1:Best possible solutions

Numbers	Best Sequence	Usage variation	Number of Set-ups
1	BAABBACABBAAB	4.62	9

4.3.2 Second Objective

Controlling a mixed-model assembly manufacturing facility, operating under a just in time (JIT) production control system, is by setting a production schedule for the last process in the facility, which is usually a mixed-model final assembly line. The workload-smoothing problem is to smooth the workload on the final assembly line to reduce the chance of production delays and stoppages.

M different products are assembled in the mixed-model final assembly line. The line consists of (S) number of stations between which products move until production is completed. The available production time at each station on the assembly line is fixed. Let $p_{m,s}$ be the processing time of model (m) on workstation s, where $s = 1,2,..,S$ stations, and $(t_{m,s})$ be the total processing time required by all requirements of model (m) on workstation s, $t_{m,s} = d_m p_{m,s}$. The workload-smoothing problem is minimizing the deviation of actual workload from the ideal workload. The following model is modified from smooth production loads model presented by Miltenburg et al. (1991).

$$\text{Minimize} \quad \text{WL} = \sum_{k=1}^{D_T}\sum_{m=1}^{M}\sum_{s=1}^{S}\left(p_{m,s}x_{m,k} - \gamma k\left(\frac{t_{m,s}}{T_p}\right)\right)^2$$

$$S.T.$$

$$\sum_{m=1}^{M} x_{m,k} = 1, \quad k = 1,\ldots,D_T$$

$$0 \le x_{m,k} - x_{m,k-1} \le 1, \quad m = 1,..,M,\ k = 1,..,D_T$$

$$x_{k,m} \text{ is a non} - \text{negative integer} \qquad \forall m, \quad \forall k$$

m model type; $m = 1,\ldots, M$

k position of a unit in the sequence; $k = 1,\ldots, D_T$

$p_{m,s}$ processing time of model m *on* workstation S

S number of stations on the assembly line

$t_{m,s}$ total processing time required by all requirements of model (m) on workstation s $(d_m p_{m,s})$.

T_p total processing time required by all models on workstation s over the planning horizon $\sum_{s=1}^{S}\sum_{m=1}^{M} p_{m,s}\ d_m$

γ cycle time, $\gamma = \dfrac{T_p}{(D_T)}$ (models are launched in fixed γ time interval to the assembly line).

A numerical example is now presented to clarify the workload production systems. This example sequences 3 different models on a final mixed-model assembly line. The assembly times by station and demand for each model are shown below.

Product i	1	2	3
Requirements,	100	100	100
Production time	4	5	4

Production time $T_p = \sum d_m t_{m,s}$, (100)(4) + (100)(5) + (100)(4) = 1300

Production demand $D_T = \sum d_m$, (100) +(100) +(100) =300

$$\left(\frac{t_{m1,s}}{T_p}\right) = \left(\frac{400}{1300}\right) = 0.31$$

$$\gamma = \frac{T_p}{(D_T)} = \frac{1300}{300} = 4.33$$

$$\text{Minimize} \quad WL = \sum_{k=1}^{D_T} \sum_{m=1}^{M} \sum_{s=1}^{S} \left(p_{m,s} x_{m,k} - \gamma k \left(\frac{t_{m,s}}{T_p} \right) \right)^2$$

Total smoothing workload for example is 182 and CPU time is 1.5833 minutes the sequence is as follow:

BCBACBABAACBBAAABBBCAACBBACCABBBCCBBBCCBCCACACACAA
CACCCCBCAABBBABABBCABBBBCBCAABBCACBABCCAABACBBBABC
BBBCAABCACBBAAAACBACBBAAACCAACCAABCCCCABAABBCCCCAB
CACBCCAAAACABBCAAABACAACABACBBCCAAABABAACCCBCCACBC
BBCCBBCCCBBBAAABCBBBAABAABAAABAABAABBAACAACCAAACBB
BBBABCCCBBAACBCACCBCCCBCCCABCBBAAABCCBBBCCCCACCBCC.

Following Miltenberg and Goldstein's (1991) model for the joint problem, we can state the joint problem as:

$$\text{Minimize} \quad \mathrm{P} = \sum_{\mathrm{k}=1}^{\mathrm{D_T}} \sum_{\mathrm{m}=1}^{\mathrm{M}} \left(\mathrm{x}_{\mathrm{m,k}} - kr_m \right)^2 + \sum_{s=1}^{S} \left(p_{m,s} + \gamma k \left(\frac{t_{m,s}}{T_p} \right) \right)^2$$

$$S.T.$$

$$\sum_{m=1}^{M} x_{m,k} = 1 \qquad \forall k$$

$$0 \le x_{m,k} - x_{m,k-1} \le 1, \quad m = 1,..,M; \quad k = 1,...,D_T$$

$$x_{k,m} \text{ is a non} - \text{negative integer} \quad \forall m, \forall k$$

Property of the objective function P to ensures good solution- At each stage k the heuristics make decisions by comparing alternatives against some desired levels of production which reflect, in a global sense, the optimal solution of the problem. That is, the desired production to stage k, so far as the usage variation goal is concerned, is kr_m for each output m, while the desired production to stage k, so far as workloading

goal is concerned, is $k\left(\frac{t_{m,s}}{T_p}\right)$ for each station s. But kr_m and $k\left(\frac{t_{m,s}}{T_p}\right)$ simply describe the relative location of stage k among all stages. The optimal solution to P is the production sequence, which in a global sense stays closest to these relative locations. Hence the heuristically solution is "pulled" toward the "optimal" solution at each stage and cannot drift too far away from it. Because of this property the heuristic give good solutions.

4.3.3 Third Objective

Costs of operations performed on every product but in different choices are affected by sequencing Mixed-Model assembly lines. For these types of operations, set-up costs may be incurred each time the operation switches from one choice to another. Therefore, an incentive exists to sequentially assembly products having the same choice of an operation.

The Third objective function is to minimize the number of required set-up costs(St) in a production sequence. Many assembly operations often require sequence-dependent set-ups. For instance, an automotive body station needs set-ups when the door types are changed. Similarly, an engine mounting station requires a set-up when the engine types are changed. Burns and Daganzo [5] addressed sequence-dependent set-ups cost and production capacity in determining an assembly line job sequence. In this research, a mathematical formulation considering sequence-dependent set-ups is developed and provided below:

$$\text{Min} \sum_{m=1}^{M}\sum_{k=1}^{D_T}\sum_{s=1}^{S}\sum_{r=1}^{M} X_{s,m,r}C_{s,m,r}$$

$$S.T.$$

$$\sum_{m=1}^{M}\sum_{r=1}^{M} x_{m,k,r} = 1 \qquad \forall k \quad (4.9)$$

$$\sum_{k=1}^{D_T}\sum_{r=1}^{M} x_{m,k,r} = d_m \qquad \forall m \quad (4.10)$$

$$x_{m,k,r} = 0 \quad or \quad 1 \quad \forall m,k,r$$

Where $C_{s,m,r}$ is the set-up costs required when model type is changed from m to r and $x_{k,m,r}$ is 1 if model type k and r are assigned, respectively, at the kth and (k+1) st position in a sequence; otherwise, $x_{k,m,r}$ is 0. In this work, it is assumed the set-ups time is sequence-dependent. Equation (4.9) is a set of position constraints indicating that every position in a sequence is occupied by exactly one product. Equation (4.10) imposes the restriction that all the demands should be satisfied.

Example: Consider the production sequence where there are five units of product A, three units of product B and two units of product C. Table 4.2 shows assembly times of these models. The results display in Table 4.3.

Table 4-2:Assembly time by station

	workstation				
Model	1	2	3	4	Demand
A	4	6	8	4	5
B	8	9	6	7	3
C	7	4	6	5	2

Table 4-3: Optimal solutions

Objective 1 (Usage variation)	Objective 2 (Workload)	Objective 3 (Set-ups cost)	Sequence
2.90	67.66	9.00	ABCAABACBA

In this study, the mixed-model assembly line is assumed to have been balanced in the design phase to accommodate the anticipated mix of models. Hence, each workstation can process at least the average amount of work without requiring additional, unassigned workers or without causing any stoppages or delays. In the

operational phase, however, if the models that require more work at a station than that allowed by the cycle time are consecutively schedule, the assembly line is forced to slow down or even stop in order to complete the required work. Workload-smoothing goal seeks to prevent this type of line inefficiencies where models with long production times followed models with shorter production times.

4.4 Efficiency frontier approach

For many real-world decisions making problems there is a need for simultaneous optimization of multiple objectives. As Hillier [1967] stated:

> "One possible approach is to use long-run profit maximation as the sole objective. At first glance, this approach appears to have considerable merit. In particular, the objective of long-run profit maximation is specific enough to be used conveniently, and yet it seems to be broad enough to encompass the basic goal of most organizations. In fact, some people tend to feel that all other legitimate objectives can be translated into this one. However, this is such an oversimplification that considerable caution is required! A numbers of studies have found that, instead of profit maximization, the goal of satisfactory profits combined with other objectives is characteristic of American corporations. In particular, typical objectives might be to maintain stable profits; increase (or maintain) one's share of the market, product diversification, maintain stable prices, improve worker moral, maintain family control of the business, and increase company prestige. These objectives might be compatible with long-run profit maximization, but the relationship is sufficiently obscure that it may not be convenient to incorporate them into this one objective."

Such multiobjective optimization problems require separate techniques, which are very different to the standard optimization techniques for single objective optimization. It is very clear that if there are two objectives to be optimized, it might be possible to find a solution, which is the best with respect to the first objective, and other solution, which is the best with respect to the second objective.

It is convenient to classify all potential solutions to the multiobjective optimization problem into *dominated* solutions and *nondominated* (*efficiency*)

solution. As solution x is dominated if there exits a feasible solution y not worse than x on all coordinates, i.e., for all objectives $f_i(i = 1,...,k)$:

$$f_i(x) \geq f_i(y) \qquad \text{for all} \quad 1 \geq \text{i} \geq \text{k. for minimization}$$

If a solution is not dominated by any other feasible solution, we call it non-dominated (or efficiency) solution. All efficiency solutions might be of some interest; ideally, the system should report back the set of all efficiency optimal points. In other disciplines, the term efficiency is also known as *Pareto optimality*, *admissibility*, or *noninferiority*. We call the set of all efficient points the efficient set.

One way to solve the multi-objective design problem is to set weights for the elements of the objective function, and then to develop an efficient method for solving the single objective problem. The major problem with this approach is finding the appropriate weight value: setting values should take into account many assumptions related to various costs. These cost elements are very difficult to evaluate and are affected by many factors, which differ from one environment to another, and from one assembly configuration to another. In addition, this additive model assumes a linear relation between objectives.

Usually it is much easier for a system designer to choose the preferable configuration from given (small number of) alternatives rather than to set weights for the objectives. In such a case an efficiency frontier approach is useful; the system designer should consider only efficient (not dominated) design configurations at the efficiency frontier. For any efficient configuration, which lies at the efficiency frontier there is no other configuration that is equal or superior in all performance measures; for any monotonic objective function $f(UV, WL, St)$, the optimal solution will be included in the set of configurations at the efficiency frontier.

The purpose of the efficiency frontier approach is to identify efficient configurations. If the group of efficient configurations turns out to be small enough, the designer can choose the preferred solution from among the efficient solutions, while taking into account local characteristics of the assembly system and the production environment. There is an axis for each of these efficient solutions, number of usage or workload on the y-axis, and number of required set-ups on the x-axis.

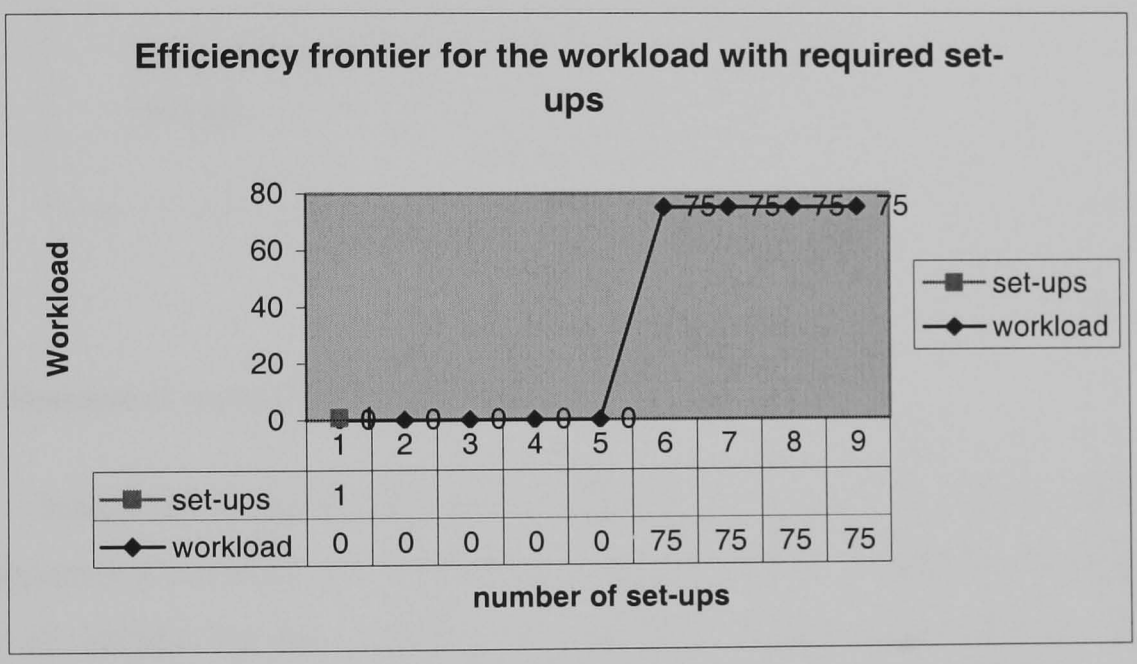
Finding an efficiency frontier is a difficult problem (Steuer, 1986). A hypothetical efficiency frontier is provided in Figures 4.1 and 4.2.

Example1. Consider the data given in Table 6.1 Simple structure with Low correlation and problem set M1 from Table 13 (appendix A) and apply algorithm B. The Gantt

charts of the schedules obtained in particular iterations are given in Figure 4.1 showing the *Efficiency frontier* for an example problem.

Each point on the efficiency frontier represents the minimum workload for each number of set-ups. The remaining points are "inefficient" or nondominated.

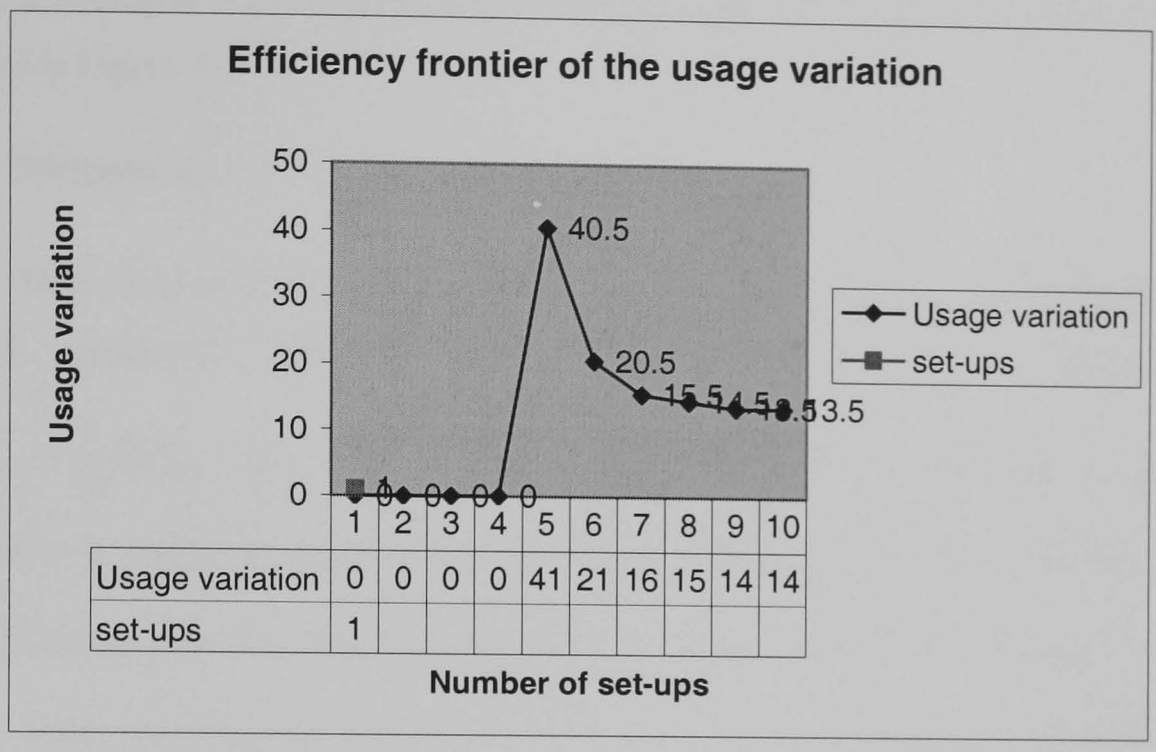
Figure 4-1: Efficiency frontiers for set-ups and workload.



Example2. Consider the data given in problem set M1 from Table 13 (appendix A) and apply algorithm B. The Gantt charts of the schedules obtained in particular iterations are given in Figure 4.2 shows the *Efficiency frontier* for an example problem.

Each point on the efficiency frontier represents the minimum usage variation for each number of set-ups. The remaining points are "inefficient" or nondominated.

Figure 4-2: Efficiency frontiers for usage variation with required set-ups.



	1	2	3	4	5	6	7	8	9	10
Usage variation	0	0	0	0	41	21	16	15	14	14
set-ups	1									

4.5 Simulated Annealing Algorithm

Simulated Annealing Algorithm is designed to produce an optimal solution to the sequencing mixed-model production problem with multi-objective functions in terms of minimizing the variation of the actual production from the desired production; workload-smoothing problem minimizes deviation of actual workload from the ideal workload and minimizes the set-ups cost.

The SA has great ability for deriving good solutions; this technique is known to be parameter sensitive and time consuming. The performance and computational time of this scheme heavily depend on the *annealing schedule* and its parameter tuning.

Based on analysis and comparison of existing simulated annealing methods, we proposed two methods with four algorithms, which differed only in cooling schedule and hence affected in computational time

4.5.1 Proposed Algorithms

The complete proposed algorithms for the annealing process procedures mentioned in chapter 3 are described here. The flowchart of the proposed method is illustrated in Figure 4.1.

4.5.1.1 Methods A

STEP 0: Determine an initial solution, which is selected from a population of 200000 randomly generated solution, and calculate T_0 and T_f by equations $T_0 = \Delta f_{\min} + \frac{1}{10}(\Delta f_{\max} - \Delta f_{\min})$, $T_f = \Delta f_{\min}$, respectively. The temperature T is initialised to T_0 and the iteration counter m, and equilibrium counter (r), are set to 0.

STEP 1: Compute the objective function, select efficiency frontier and randomly select starting point, set the temporary solution $a^* = a^0$, and the temporary function $E = f(a^0)$.

STEP 2: Generate a feasible neighborhood search by " pairwise exchange". For Pairwise exchanging, two unique products are randomly selected and exchanged. This new sequence, which obtained after exchange is referred to as the present solution and its objective value is determined (f_p).

STEP 3: Evaluate the value of the objective function after pairwise exchange: $\Delta f = f_p - f_c$. If Δf is less than or equal zero go to step 5;otherwise go to step 4.

STEP 4: Exchange acceptance process

(a) If $\Delta f \le 0$, then go to STEP 4b; otherwise go to STEP 4d.

(b) Accept the pairwise exchange and increment the iteration counter $m = m + 1$, and go to step 5.

(c) If $\Delta f \ge 0$, then go to STEP 4d, otherwise go to step 5.

(d) Compute (Metropolis) $P(\Delta f) = e^{\frac{\Delta f}{k_b T}}$ and select a uniform distribution with the range [0,1]. If the random number is less than $P(\Delta f)$, then go to STEP 4b; otherwise return to STEP 2.

STEP 5: Equilibrium test process

(a) If the value of objective function after exchange (E) is less than the best value found so far (f_p) go to step 5b;otherwise go to step 5c.

(b) Change the temporary solution, and if $m \langle e$ go to step 5c; otherwise go to Step 2.

(c) If $\frac{\left|\bar{f}_e - \bar{f}_g\right|}{\bar{f}_g} \leq \varepsilon$, go to step 6; otherwise go to step 2.

STEP 6: If *m,* the number of pairwise exchange examined, is greater than or equal *M*, then go to step 7; otherwise change the temperature according to equation (3.5), increment the equilibrium counter *r* =*r* + 1, and go to step 2

STEP 7: Stop.

4.5.1.2 Methods B

This method is similar to the previous one except for steps 0,1 and 6, which become as follows:

STEP 0: Determine an initial solution, which is selected from a population of 200000 randomly generated solution, and calculate T_0 and T_f by equations (3.3) and (3.4), respectively. The temperature T is initialised to T_0 and the iteration counter *m* set to 0.

STEP 1: Compute the objective function, which is the initial objective function $f(a^0)$, function $E = f(a^0)$. In executing SA, the temperature tuning β and *M* are calculated by:

$$T_{i+1} = \frac{T_i}{1+\beta T_i} \quad ,\beta = \frac{T_0 - T_f}{MT_0T_f} \quad ,M = 50 \times \frac{N(N-1)}{2}$$

set the temporary solution $a^* = a^0$, and the temporary function $E = f(a^0)$.

In executing SA, the cooling parameter on temperature β and M are calculated by equation (3.7) and (3.8), respectively.

STEP 6:

(a) If the new value of temperature is greater than or equal the final temperature T_f go to step 7; otherwise return to step 2.

STEP 7: STOP.

In the method A, uses a geometric decrement procedure for temperature tuning. It determines the next temperature according to a prescribed cooling function i.e. $T_{i+1} = \alpha T_i$ where α is a constant. Typical value of α used in practice lie between 0.8 and 0.99. Such function provides smaller decrements in T as zero temperature is approached. This function cools faster, and thereby contributes to saving computational time. The SA algorithm is stopped when the solution obtained at each temperature change is unaltered for a number of consecutive temperature changes. The final state is if *m,* the number of pairwise exchange examined, is greater than or equal *M*, said to correspond to the frozen state.

In the second method, information obtained during trails prior to the annealing process is utilized, and the system is cooled very slowly near an unknown critical temperature. Clearly if the system is kept too "hot" then too many bad uphill moves are accepted for any good solution to be reached while if it is too "cold" then the scheme will quickly drop into a local optimum and the remainder of the search will be a fruitless attempt to escape from it. Thus we would expect that somewhere between these two extremes ($\delta_{\min}, \delta_{\max}$) there must be an optimum fixed temperature.

By a suitable choice of M (see equation 3.9), the number of swaps examined, the user can easily control the running time of the algorithm. The final state is if the new value of temperature is greater than or equals the final temperature T_f. What is surprising is how effective the search becomes at this optimum temperature (see Figures 5.1, 5.2 and 5.3).

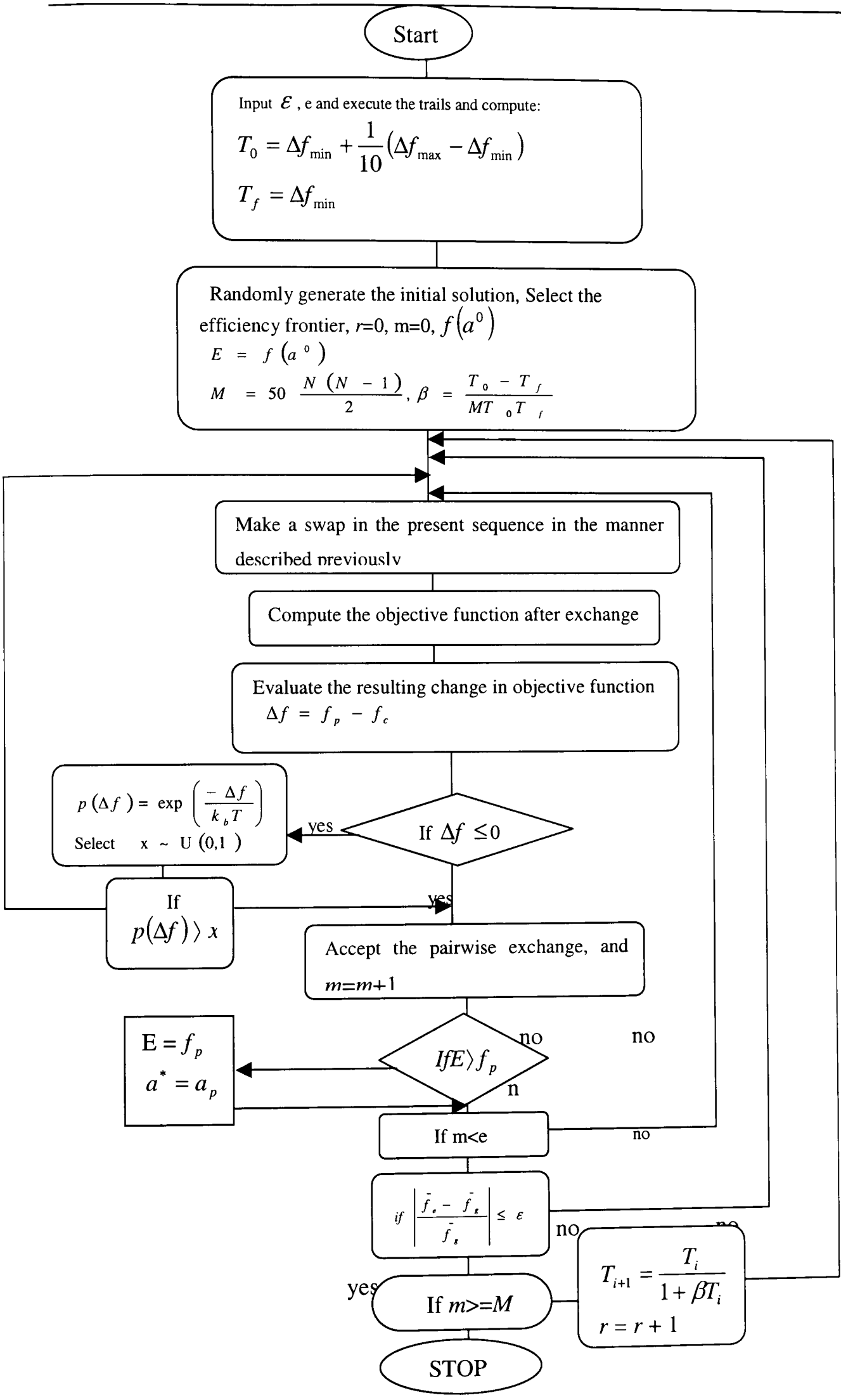


Figure 4-3: Flowchart of Methods A, B

5 Implementations and Results

In Chapter 5 we describe the method for evaluating the performance of the sequencing usage variation algorithms presented in section 1. In section 5.1, the small test problem instances are presented, and then their results are shown in sections 5.2 and 5.3. The medium-size test problem instances are presented with their respective results in sections 1.4 and 1.5. The large test problem instances are presented with their respective results in section 5.6. In section 2, we describe the method for evaluating the performance of the workload-smoothing method, and their computational results are presented in section 5.2. Chapter 6 provides analysis and insights into these results.

5.1 Evaluating The Sequencing Usage variations Method

Numerical experiments were conducted to evaluate the proposed methods A and B for objective function 1. These methods are used in view of the complexity of optimal solution methods, since the mixed-model sequencing problem with multiple objectives is NP-complete. Several test problems from the literature (Sumichrast & Russell, 1990) were used to evaluate the first objective function; see appendix A. These problem sets, M1 small-size problems with the total production $D_T = 20$ and $m = 5$, M2 medium-size problem with $D_T = 20$ and $m = 10$ and M3 large-size problem with $D_T = 100$ and $m = 15$, were solved by the proposed methods to compare the proposed methods with existing methods.We used the Earliest Due Date (EDD) (Inman1991) and existing optimal solution, which were presented by Meral et al. (2001).

The following average measures over nine problems are calculated for each solution approach:

- Mean squared deviation (MSD) = $\frac{1}{D_T}\sum_{k=1}^{D_T}\sum_{m=1}^{M}\left(x_{k,m} - kr_m\right)^2$,
- % Difference in the objective function = $\left(\frac{E_{algorithm} - E_{optimal}}{E_{optimal}}\right)\times 100$
- CPU time.

After evaluating the performance of the heuristic algorithms on small problems, we evaluate the solutions of medium and large problems. We also compare the solutions from the various algorithms with the "best" known solution. For the three sets of test problems, the methods of evaluation will be the same. The problems will be evaluated on the basis of the goal of minimizing the mean squared deviations of the actual production from the desired production. The algorithms are coded in Visual C++ and run on a Pentium 2 PC. During the experimentation, CPU time is taken in minutes.

Concerning parameter setting, Connolly's scheme was employed in our method B (see chapter 4 for detail). In method A, the tuning parameter for temperature α was set at 0.97 from the results of preliminary experiments.

To evaluate the quality of our algorithm, method B, we proposed another approach with different cooling temperature. In this approach final temperature is equal to 20 and initial temperature is equal to one. Several test problems from the literature (Sumichrast & Russell, 1990) were used to evaluate the quality of our method B; see appendix A. The results of the set M1 are shown in Table 5.1.

Table 5-1:Results of the problem with different cooling temperature

Problem name	Objective function	CPU
A	13.50	0.3000
B	11.80	0.2167
C	12.00	0.2167
D	10.25	0.2000
E	11.25	0.2000
F	11.45	0.2167
G	13.10	0.1833
H	13.35	0.2167
I	16.00	0.2000

Table 5.1 and Figures 5.1,5.2 shows the results of comparing our method B with different cooling temperature in set M1 problems. From the results in Table 5.1 and Figures 5.1, 5.2, it can be seen that method B gives near optimal solutions for small size problems with less computational time.

Figure 5-1:Comparisons of the Method B and constant temperature.

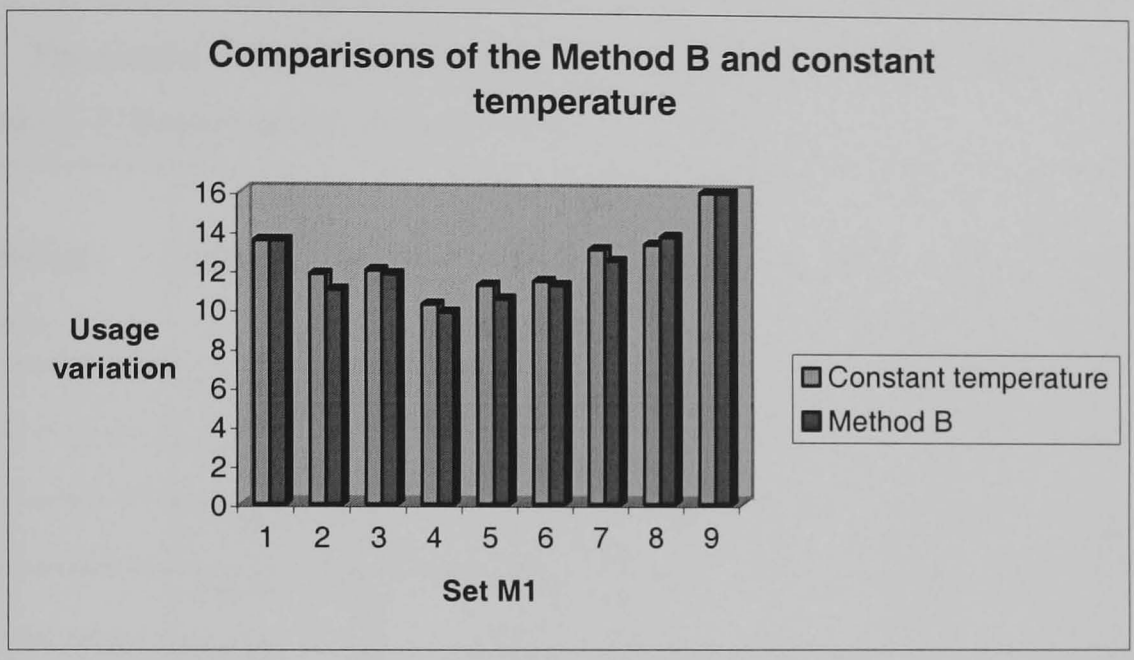
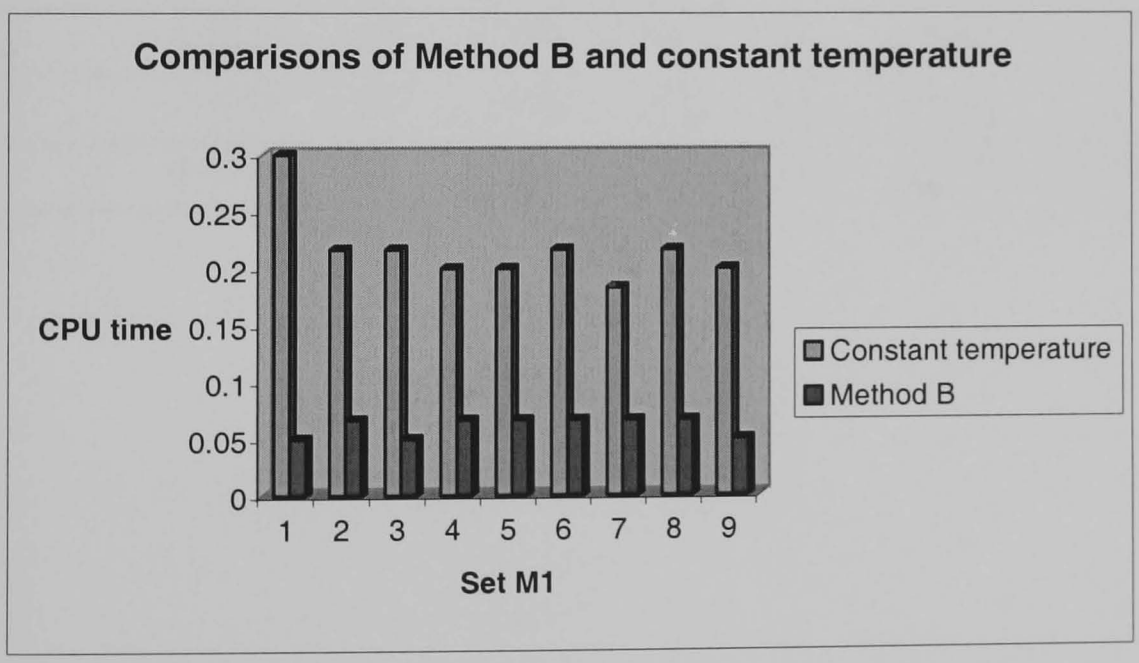


Figure 5-2:Comparisons of the Method B and constant temperature.



5.1.1 The Small Test Problem Results

To evaluate the algorithms using small test problems, a problem set M1 with total demand D_T =20 and m = 5 from appendix A is used. Nine different problems are used in set M1 to test each method. The problem set are A, B. C. D, E, F, G, H and I. Each problem has five products and the total demand is 20.

The results of these tests are broken by methods and shown in Table 5.1.

Table 5-2: Results of the problem Set M1

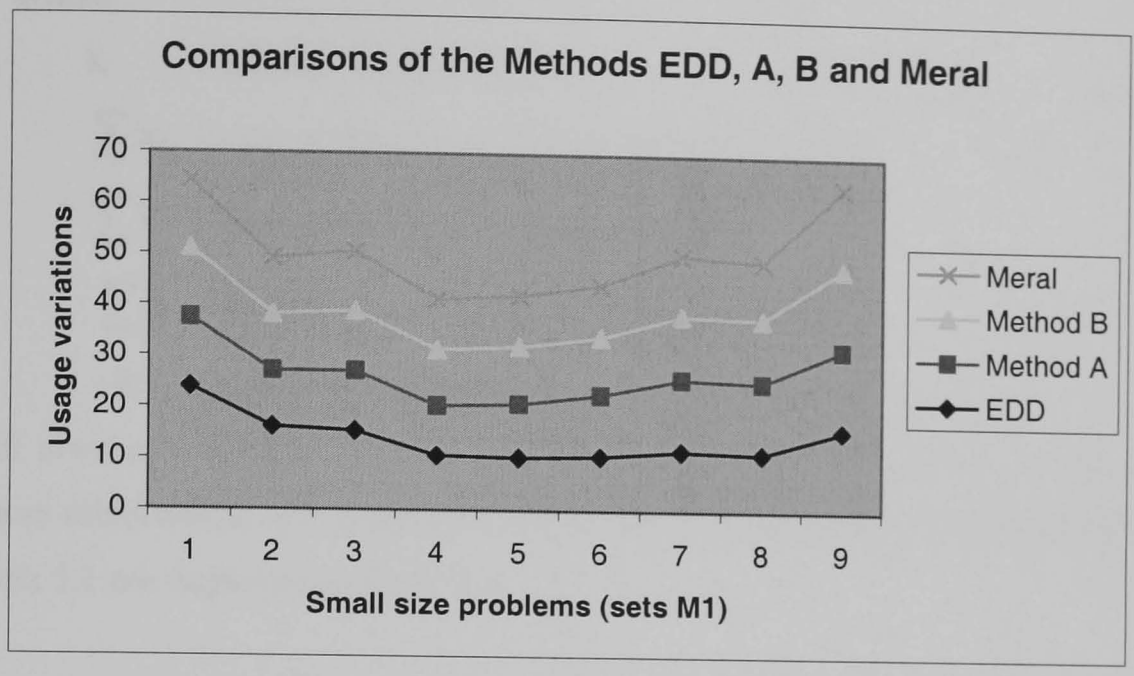
Problem name	Method A	Set-ups costs	CPU (minutes)	Method B	Set-ups costs	CPU (minutes)	% Difference (Methods)
A	13.50	9	0.0500	13.50	9	0.0500	0.0%
B	11.00	9	0.0500	11.00	9	0.0667	0.0%
C	11.80	11	0.0500	11.80	11	0.0500	0.0%
D	10.95	16	0.0500	9.85	15	0.0667	10.0%
E	11.25	19	0.0500	10.55	18	0.0667	0.06%
F	12.15	18	0.0667	11.25	18	0.0667	5.77%
G	14.20	19	0.0500	12.50	18	0.0667	11.97%
H	14.05	18	0.0500	13.75	18	0.0667	2.13%
I	16.00	18	0.0500	16.00	18	0.0500	0.0%

Table 5-3: Comparisons of the Methods A, B, EDD and Meral

Demand	Methods	Set M1
A	EDD	24.50
	M-A3H2	13.50
	Method A	13.50
	Method B	13.50
	Meral	13.50
B	EDD	16.20
	M-A3H2	11.00
	Method A	11.00
	Method B	11.00
	Meral	11.00
C	EDD	15.50
	M-A3H2	11.70
	Method A	11.70
	Method B	11.70
	Meral	11.70
D	EDD	10.65
	M-A3H2	9.85
	Method A	9.85
	Method B	10.95
	Meral	9.85
E	EDD	10.35
	M-A3H2	9.95
	Method A	10.55
	Method B	11.25
	Meral	9.95
F	EDD	10.65
	M-A3H2	10.25
	Method A	12.15
	Method B	11.25
	Meral	10.25
G	EDD	11.80
	M-A3H2	11.80
	Method A	14.20
	Method B	12.50
	Meral	11.80
H	EDD	11.35
	M-A3H2	11.35
	Method A	14.05
	Method B	12.50
	Meral	11.35
I	EDD	16.00
	M-A3H2	16.00
	Method A	16.00
	Method B	16.00
	Meral	16.00

Table 5.2 and Figure 5.1shows the results of comparing methods EDD, A, B and Meral in the small size problems.

Figure 5-3: Comparison of the Methods EDD, A, B, and Meral



From the results in Table 5.2 and Figure 5.1, it can be seen that method B gives near optimal solutions for small size problems.

5.1.2 Results Of The Small Test Problem

The results of our computational testing in Table 5.2 and Figure 5.1 shown that method B finds near optimal solutions for this problem set. The descriptive statistics of our methods are:

Methods	Mean	St.Dev	Variance	Sum	Minimum	Maximum	Range
B	12.31	1.61	2.6122	110.85	10.95	16.00	5.05
Optimal	11.71	1.95	3.83	105.40	9.85	16.00	6.15

We use correlation to describe the degree of relationship between results obtained by method B and the optimal solutions. The formula for the correlation used is:

$$R = \left[\frac{k\sum xy - \left(\sum x\right)*\left(\sum y\right)}{\sqrt{\left[k\sum x^2 - \left(\sum x\right)^2\right]*\left[k\sum y^2 - \left(\sum y\right)^2\right]}} \right] \qquad (5.1)$$

Where:

k = number of problems

$\sum xy$ = sum of the models B and optimal in set M1

$\sum x$ = Sum of model B in problem set M1

$\sum y$ = Sum of model optimal in problem set M1

R = stand for correlation

R will always between –1.0 and +1.0. If the correlation is negative, we have a negative relationship; if it is positive, the relationship is positive. Data needed for formula 5.1 are displayed in Table 5.3.

Table 5-4 : Descriptive statistics of small test problem

Problem (k)	Method B (x)	Meral (y)	$x*y$	x*x	y*y
1	13.50	13.50	182.25	182.25	182.25
2	11.00	11.00	121.00	121.00	121.00
3	11.70	11.70	136.89	136..89	136.89
4	10.95	9.85	107.86	119.90	97.02
5	11.25	9.95	111.94	126.56	99.00
6	11.45	10.25	117.36	131.10	105.06
7	12.50	11.80	147.50	156.25	139.24
8	12.50	11.35	141.88	156.25	128.80
9	16.00	16.00	256.00	256.00	256.00

Now, when we plug the values in Table 5.3 into formula 5.1. we find the correlation for our nine problems in set M1 to be 0.708, which shows a strong positive relationship between method B and optimal models in small size problem (set M1).

5.1.3 Testing the Significance of the correlation

Once we have computed the correlation, we can determine the probability that the observed correlation occurred by chance. That is, we can conduct a significance test. Most often we are interested in determining the probability that the correlation is a real one and not a chance occurrence. In this case, we are testing the mutually exclusive hypotheses:

Null Hypothesis	R = 0
Alternative Hypothesis	R 〈 〉 0

The way to test this hypothesis is to look at the statistical table to find the critical value of our correlation value (R). Our significance level of alpha = 0.05. This means that we are conducting a test where the odds that the correlation is a chance occurrence are no more than 5 out of 100. Our degree of freedom (df) is N-2, which is 9-2=7. We are doing a two-tailed test. After looking at the statistical table shown in appendix B, Table C.11 the critical value is 0.6664. This means that our correlation is greater than 0.6664 or less than -0.6664. Since our correlations is actually smaller than 0.6664, we conclude that our correlation is "statistically significant". We can accept the null hypothesis. Now we have confidence that our method B finds near optimal solutions.

5.1.4 The Medium size problem and the results

To evaluate the algorithms using medium test problems, a problem set M2 with total demand D_T =20 and m = 10 from appendix A is used. Nine different problems are used in set M2 to test each method. The problem sets are A, B, C, D, E, F, G, H and I. Each problem has ten models (products) and total demand is 20.

The results of these experiments are shown in Table 5.4.

Table 5-5: Results of the problem Set M2

Problem Name	Method A	Set-up costs	CPU (minutes)	Method B	Set-up costs	CPU (minutes)	% Difference (Methods)
A	30.75	18	0.1000	30.75	18	0.1333	0.0%
B	27.90	17	0.1000	26.80	17	0.1333	1.79%
C	28.55	20	0.0833	27.95	17	0.1333	2.10%
D	27.90	20	0.0833	27.90	20	0.1333	0.0%
E	28.75	19	0.1000	27.75	19	0.1167	0.0%
F	26.80	18	0.0833	25.80	18	0.1333	0.0%
G	28.15	19	0.1000	27.15	19	0.1000	0.0%
H	26.45	18	0.0833	26.45	18	0.1000	0.0%
I	33.00	20	0.1000	33.00	20	0.0833	0.0%

Table 5-6: Comparisons of the Methods EDD, B, and Meral

Demand	Methods	Set M2
A	EDD	60.75
	M-A3H2	30.75
	Method A	30.75
	Method B	30.75
	Meral	30.75
B	EDD	44.40
	M-A3H2	26.80
	Method A	27.90
	Method B	26.80
	Meral	26.80
C	EDD	45.95
	M-A3H2	27.15
	Method A	28.55
	Method B	27.95
	Meral	27.15
D	EDD	38.60
	M-A3H2	27.20
	Method A	27.90
	Method B	27.90
	Meral	27.20
E	EDD	39.95
	M-A3H2	27.55
	Method A	28.75
	Method B	27.75
	Meral	27.55
F	EDD	31.20
	M-A3H2	25.00
	Method A	26.80
	Method B	25.80
	Meral	25.00
G	EDD	33.35
	M-A3H2	25.75
	Method A	28.15
	Method B	27.15
	Meral	25.75
H	EDD	25.35
	M-A3H2	24.45
	Method A	26.45
	Method B	26.75
	Meral	24.15
I	EDD	33.00
	M-A3H2	33.00
	Method A	33.00
	Method B	33.00
	Meral	33.00

Table 5.5 and Figure 5.2 shows the results of comparing methods, EDD, A, B, and Meral in the medium size problems.

Figure 5-4: Comparisons of Methods EDD, A, B, and Meral

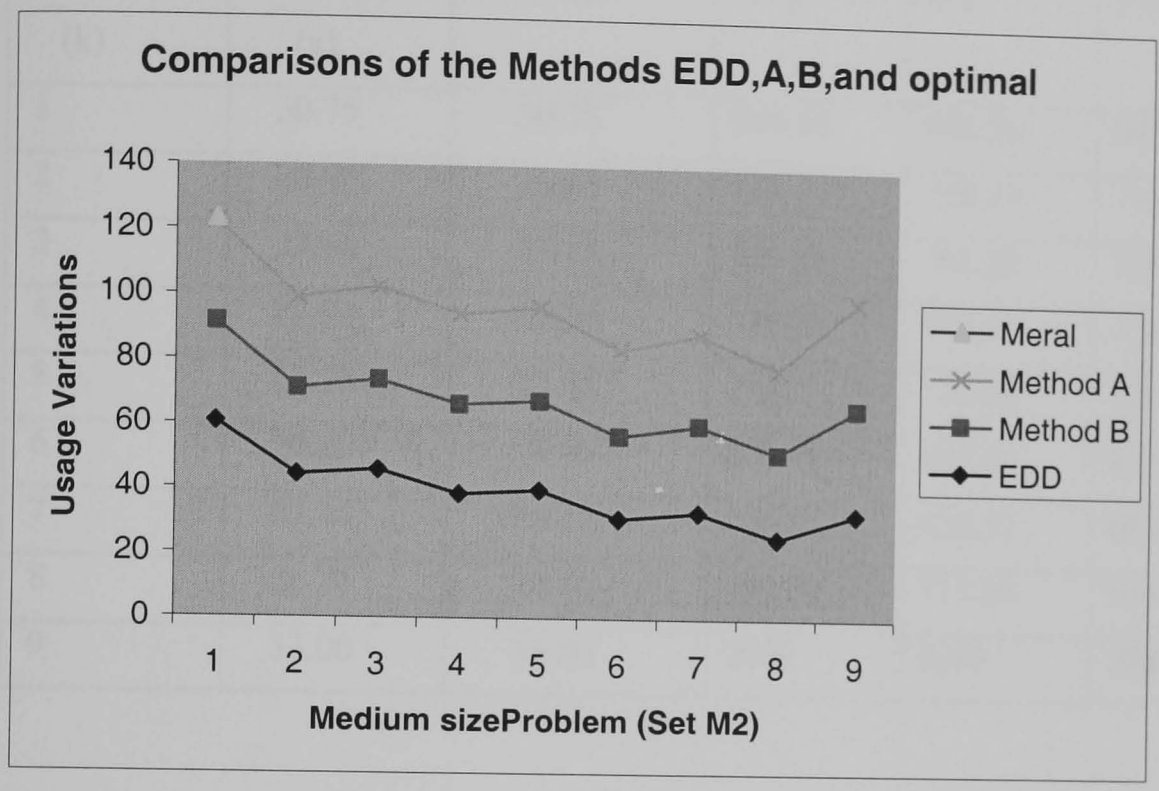


Table 5.5 and Figure 5.3 shows that method B gives a near optimal solution.

5.1.5 Results Of The Medium Test Problem

The results of our computational testing in Table 5.4 and Figure 5.2 shown that method B finds near optimal solutions for this problem set. The descriptive statistics of our methods are:

Methods	Mean	St.Dev	Variance	Sum	Minimum	Maximum	Range
B	28.20	2.25	5.10	253.85	25.80	33.00	7.20
Meral	27.48	1.98	3.945	247.35	24.15	33.00	8.85

Now, when we plug the value in Table 5.6to formula 5.1. We find the correlation for our nine problems in set M2 to be 0.708, which is a strong positive relationship.

Table 5-7: Descriptive statistics of Medium test problem

Problem (k)	Method B (x)	Meral (y)	x*y	x*x	y*y
1	30.75	30.75	945.56	945.56	945.56
2	26.80	26.80	718.24	718.24	718.24
3	27.95	27.15	758.84	781.20	737.12
4	27.90	27.20	758.88	778.41	739.84
5	27.75	27.55	764.51	770.06	759.00
6	25.80	25.00	645.00	665.64	625.00
7	25.15	25.75	647.61	632.52	663.06
8	26.75	24.15	646.01	715.56	583.22
9	33.00	33.00	1089	1089	1089

5.1.6 The Large size problem and the results

As with the small and medium problem sets, nine different demand patterns are used in set M3 to test each algorithm. The results of the large-size problems are displayed in Figure 5.3 and Tables 5.7 and 5.8.

Table 5-8: Results of the problem SetM

Problem name	Method A	Method B	% Difference (Methods)
A	255.22	254.16	0.415%
B	264.85	255.53	3.52%
C	254.72	264.28	-3.75%
D	269.45	261.77	3.01%
E	292.41	247.05	15.51%
F	293.87	254.85	13.28%
G	283.52	294.41	-0.36%
H	335.22	294.41	12.17%
I	367.95	154.06	58.13%

Table 5-9: Comparisons of the Methods in set M3

Demand	Methods	Set M3
A	EDD	522.70
	M-A3H2	213.94
	Method A	255.22
	Method B	223.14
	Meral	213.94
B	EDD	374.55
	M-A3H2	193.15
	Method A	264.85
	Method B	204.31
	Meral	189.95
C	EDD	321.00
	M-A3H2	254.72
	Method A	204.92
	Method B	186.72
	Meral	186.72
D	EDD	302.05
	M-A3H2	191.57
	Method A	269.45
	Method B	213.59
	Meral	187.49
E	EDD	237.71
	M-A3H2	186.01
	Method A	292.41
	Method B	178.81
	Meral	181.19
F	EDD	207.05
	M-A3H2	293.87
	Method A	203.53
	Method B	169.93
	Meral	169.93
G	EDD	182.91
	M-A3H2	176.53
	Method A	213.52
	Method B	205.41
	Meral	165.59
H	EDD	189.52
	M-A3H2	191.88
	Method A	335.22
	Method B	201.98
	Meral	177.60
I	EDD	193.05
	M-A3H2	193.05
	Method A	367.95
	Method B	227.70
	Meral	193.05

The results of our computational testing in Table 5.8 and Figure 5.3 shown that method B finds near optimal solutions for this problem set. The descriptive statistics of our large problems size are:

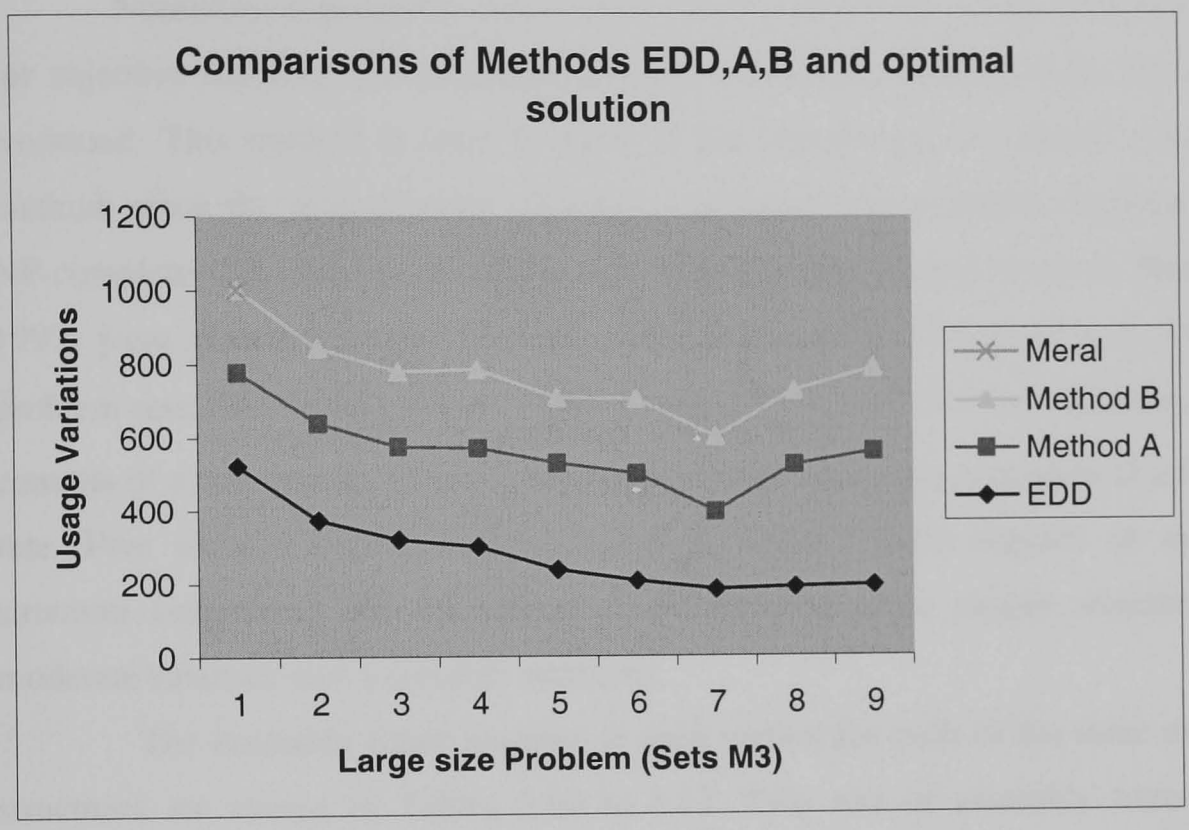
Methods	Mean	St.Dev	Variance	Sum	Minimum	Maximum	Range
B	207.04	197.29	14.04	1863.39	178.81	223.14	44.33
Optimal	185.05	100.40	10.01	1665.46	169.93	213.94	44.01

Table 5-10: Descriptive statistics of large test problem

Problem (k)	Method B (x)	Meral (y)	$x*y$	x*x	y*y
1	223.14	213.94	47738.5	49791.45	45770.32
2	204.31	189.95	38808.68	41742.57	36081.00
3	204.92	186.72	38262.66	41992.20	34864.35
4	213.59	187.49	40045.98	45620.68	35152.50
5	178.81	181.19	32398.58	31973.01	32829.81
6	203.53	169.93	34585.82	41424.46	28876.20
7	205.41	165.59	34013.84	42193.26	27420.04
8	201.98	177.60	35871.64	40795.92	31541.76
9	227.70	193.05	43957.48	51847.29	37268.30

Now, when we plug the value in Table 5.9 to formula 5.1. We find the correlation for our nine problems in set M3 to be 0.8621, which indicates a strong positive relationship. After looking at the statistical table shown in appendix B, Table C.11 the critical value is 0.6664. Since our correlations 0.8621 is actually smaller than 0.6664, we conclude that it is "statistically significant". We can accept the null hypothesis. Now we have confidence, that our method B finds near-optimal solution.

Figure 5-5: Comparisons of Methods EDD, A, B, and Meral



5.2 Evaluating The Sequencing workload-smoothing Method

Numerical experiments were conducted to evaluate the proposed method A for objective function 2 to minimize deviation of actual workload from the ideal workload. This method is used in view of the complexity of optimal solution method, since the mixed-model sequencing problem with multiple objectives is NP-complete. Several test problems from the literature (Sumichrast & Russell, 1992) were used to evaluate the first objective function; see appendix A. These problem sets, five models with the total production (D_T) 20 and the assembly line consists of a ten-stations. Products are launched onto a moving conveyor at a fixed rate. Five models require different assembly times. Three degrees of model structure complexity are considered in the experiment: a simple structure; a moderate structure and a complex structure.

The assembly times required at each station for each of the three model structures are shown in Tables 5.10 to 5.12. Two sets of assembly times are provided for each model structure. One set of times reflects a high degree of correlation between the model structure and assembly time and the other set of times assumes little correlation between model structure and assembly time. (Sumicharst &Russell, 1992).

The following average measures over nine problems are calculated for each solution approach:

- Mean square deviation(MSD) $= \sum_{k=1}^{D_T} \sum_{m=1}^{M} \sum_{s=1}^{S} \left(p_{m,s} x_{m,k} - \gamma k \left(\frac{t_{m,s}}{T_p} \right) \right)^2$

- CPU time

We evaluate the performance of the heuristic algorithms similar to the first objective function (see section one).

5.2.1 The Computational Results

To evaluated the problem of minimizing deviations of actual workload from the ideal workload, a problem set M1 from appendix A and the assembly times required at each station for each of the three model structures are used to test each method.

The results of these tests are broken by the methods and are shown in Table 5.13. The results of our computational testing in Table 5.13 and Figure 5.4 have shown the Method B finds the better results in terms of CPU times for this problem set.

Table 5-11: Assembly time required by station and model: Simple structure.

Low Correlation

	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	6.41	0	0	0	1.28	3.85	7.69	1.28	3.85	0
2	5.13	6.41	2.56	2.56	1.28	7.69	8.97	2.56	3.85	6.41
3	2.56	1.28	6.41	2.56	1.28	0	7.69	6.41	2.56	7.69
4	1.28	2.56	1.28	4.91	1.28	0	5.13	7.69	7.69	7.69
5	7.69	1.28	7.69	0	3.85	1.28	7.69	6.14	3.85	8.97

High Correlation

	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	6.67	3.33	3.33	3.33	3.33	6.67	3.33	3.33	3.33	3.33
2	3.33	6.67	3.33	3.33	3.33	3.33	6.67	3.33	3.33	3.33
3	3.33	3.33	6.67	3.33	3.33	3.33	3.33	6.67	3.33	3.33
4	3.33	3.33	3.33	6.67	3.33	3.33	3.33	3.33	6.67	3.33
5	3.33	3.33	3.33	3.33	6.67	3.33	3.33	3.33	3.33	6.67

Table 5-12: Assembly time required by station and model: Moderate structure.

Low Correlation										
	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	2.47	5.56	1.23	0.62	2.47	4.78	0.93	2.78	6.17	2.78
2	1.39	2.78	3.55	4.46	4.63	2.16	5.25	2.16	0.77	1.23
3	6.95	2.16	3.40	2.78	5.56	3.86	4.48	3.24	1.54	2.93
4	4.94	2.47	2.16	2.93	3.24	4.63	2.31	3.86	4.94	6.95
5	3.24	3.40	6.77	3.09	1.85	6.48	4.17	3.86	5.09	3.86
High Correlation										
	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	2.35	4.06	2.35	1.50	1.50	4.49	2.35	4.06	4.92	1.50
2	1.93	2.35	2.78	4.06	3.21	3.21	4.06	1.50	1.93	1.50
3	5.34	2.35	2.35	1.50	4.92	3.37	5.34	1.93	1.50	2.78
4	4.06	1.50	3.21	1.93	4.49	3.21	3.37	2.78	4.92	5.34
5	3.37	4.06	1.93	2.35	2.35	4.92	3.37	2.78	3.37	2.78

Table 5-13: Assembly time required by station and model: Complex structure.

Low Correlation										
	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	4.49	0	1.23	1.23	1.23	2.45	7.36	1.23	3.68	0
2	4.49	26.14	2.45	2.45	1.23	8.59	7.36	2.45	3.68	6.14
3	2.45	2.45	4.91	2.45	2.45	1.23	8.59	4.91	2.45	7.36
4	3.68	2.45	2.45	4.91	2.45	1.23	6.14	7.36	7.36	7.36
5	7.36	1.23	7.36	1.23	2.45	2.45	7.36	6.14	3.68	7.36

High Correlation

	Station									
Model	1	2	3	4	5	6	7	8	9	10
1	2.35	4.06	2.35	1.50	1.50	4.49	2.35	4.06	4.92	1.50
2	1.93	2.35	2.78	4.06	3.21	3.21	4.06	1.50	1.93	1.50
3	5.34	2.35	2.35	1.50	4.92	3.37	5.34	1.93	1.50	2.78
4	4.06	1.50	3.21	1.93	4.49	3.21	3.37	2.78	4.92	5.34
5	3.37	4.06	1.93	2.35	2.35	4.92	3.37	2.78	3.37	2.78

Figure 5-6: Comparisons of Methods A and B

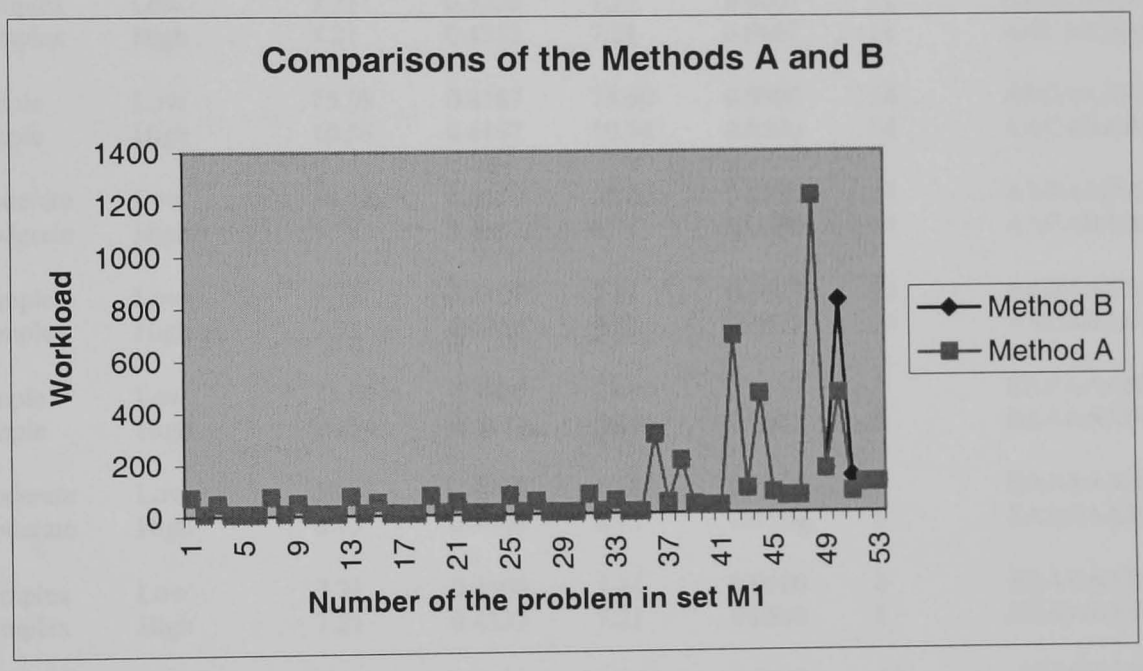


Table 5-14: Computational Results

Problem Name	Structure	Correlation	Method A	CPU (Minute)	Method B	CPU (Minute)	Set-ups	Sequence
	Simple	Low	74.99	0.4333	75.00	0.0500	9	AAAAADAAAAAACABAAEAA
	Simple	High	10.54	0.4500	10.54	0.0500	9	AAABAAADAAACAEAAAAAA
A	Moderate	Low	49.85	0.4333	49.58	0.0500	9	AAABAAADAAACAEAAAAAA
	Moderate	High	8.70	0.4500	8.70	0.0500	9	AAABAAADAAACAEAAAAAA
	Complex	Low	7.21	0.4661	7.21	0.0500	9	AAAAADAAAAAACABAAEAA
	Complex	High	7.21	0.4500	7.21	0.0500	9	AAABAAADAAACAAAAAAEA
	Simple	Low	75.20	0.4333	75.22	0.0500	11	AEACAAAAAAABADAABAAA
	Simple	High	10.54	0.4000	10.22	0.0500	11	AEACADAAAAABAAAABAAA
B	Moderate	Low	50.06	0.4333	50.06	0.0500	11	AACAADAAABAAAABAAEAA
	Moderate	High	8.70	0.4667	8.70	0.0500	11	AACAADAAABAAAABAAEAA
	Complex	Low	7.21	0.3333	7.21	0.0667	11	AEACAAAAAAABADAABAAA
	Complex	High	7.21	0.4333	7.21	0.0667	11	AACAADAAABAAAABAAEAA
	Simple	Low	75.58	0.4167	75.60	0.0500	14	ABCAAAEABAABAAADABAA
	Simple	High	10.54	0.4167	10.54	0.0500	14	AACABABAABAEADABAAAA
C	Moderate	Low	50.42	0.4500	50.42	0.0500	14	AABAADAABACBAABAAEAA
	Moderate	High	8.71	0.4667	8.71	0.0500	15	AACABABAABAEAAABADAA
	Complex	Low	7.21	0.3167	7.21	0.0677	15	AABAADAABACABABAAEAA
	Complex	High	7.21	0.4333	7.21	0.0677	15	AACABABAABAEAAABADAA
	Simple	Low	75.96	0.3500	75.99	0.0667	8	EAAAAABBBBAAAAADDCBC
	Simple	High	10.54	0.4167	10.54	0.0667	7	EAAAAABBBBAAAAADDBCC
D	Moderate	Low	50.83	0.4500	50.83	0.0500	8	EAAAAABBBBAAAAADDCBC
	Moderate	High	8.73	0.4000	8.73	0.0500	8	EAAAAABBBBAAAAADDCBC
	Complex	Low	7.21	0.4500	7.21	0.0500	8	EAAAAABBBBAAAAADDCBC
	Complex	High	7.21	0.4333	7.21	0.0500	8	EAAAAABBBBAAAAADDCBC
	Simple	Low	76.24	0.4333	72.58	0.0667	19	AEADBABADBABCBAACBAB
	Simple	High	10.54	0.4500	10.05	0.0667	19	AEADBABADABBCBAACBAB
E	Moderate	Low	51.08	0.4333	51.08	0.0667	19	AEADBABADBABCBAACBAB
	Moderate	High	8.73	0.4167	8.72	0.0667	7	AAAAECAADDBBBBBBBCAA
	Complex	Low	7.21	0.4333	7.21	0.0667	19	BADABAEABBACBABDABCA
	Complex	High	7.21	0.4333	7.21	0.0667	19	AEADBABADBABCBAACBAB
	Simple	Low	76.35	0.4333	76.38	0.0500	9	BBDAAAACCCCBBBBCEDAA
	Simple	High	10.54	0.4333	10.54	0.0500	9	BBDAAAACCCCBBBBCEDAA
F	Moderate	Low	51.22	0.4000	51.22	0.0667	9	BBDAAAACCCCBBBBCEDAA
	Moderate	High	8.75	0.3667	8.75	0.0667	9	BBDAAAACCCCBBBBCEDAA
	Complex	Low	7.21	0.4000	7.21	0.0667	9	BBDAAAACCCCBBBBCEDAA
	Complex	High	7.21	0.4000	7.21	0.0667	9	BBDAAAACCCCBBBBCEDAA

Table5.13	Continue							
	Simple	Low	305.83	0.4500	305.92	0.0500	9	AACCCCEEBBBDDDDBAAACB
	Simple	High	42.15	0.4500	42.15	0.0500	9	AACCCCEEBBBDDDDBAAACB
G	Moderate	Low	205.29	0.4333	205.29	0.0667	9	AACCCCEEBBBDDDDBAAACB
	Moderate	High	34.99	0.4500	34.99	0.0667	9	AACCCCEEBBBDDDDBAAACB
	Complex	Low	28.89	0.4167	28.89	0.0500	9	AACCCCEEBBBDDDDBAAACB
	Complex	High	28.89	0.4000	28.89	0.0500	9	AACCCCEEBBBDDDDBAAACB
	Simple	Low	688.34	0.4000	688.50	0.0500	11	BACAADCCCDDDEEEBBAAB
	Simple	High	94.81	0.4500	94.81	0.0500	11	BACAADCCCDDDEEEBBAAB
H	Moderate	Low	462.15	0.4000	462.15	0.0500	11	BACAADCCCDDDEEEBBAAB
	Moderate	High	78.72	0.4333	78.72	0.0500	11	BACAADCCCDDDEEEBBAAB
	Complex	Low	65.07	0.2000	65.07	0.0667	11	BACAADCCCDDDEEEBBAAB
	Complex	High	65.07	0.3167	67.07	0.0500	11	BACAADCCCDDDEEEBBAAB
	Simple	Low	1225.71	0.4333	1225.98	0.0500	10	BBBBEDDADEEECAAADCCC
	Simple	High	168.51	0.4000	168.51	0.0500	10	BBBBEDDADEEECAAADCCC
I	Moderate	Low	823.13	0.4167	462.15	0.0667	10	BBBBEDDADEEECAAADCCC
	Moderate	High	139.95	0.4333	78.72	0.0667	10	BBBBEDDADEEECAAADCCC
	Complex	Low	115.71	0.4167	115.71	0.0500	10	BBBBEDDADEEECAAADCCC
	Complex	High	115.71	0.4500	115.71	0.0500	10	BBBBEDDADEEECAAADCCC

6 Analysis and Conclusion

To better understand the results of the sets M1, M2 and M3, both heuristics were examined. Some of the characteristics and patterns of performance are observed and detained in this chapter. Section 6.1 describes ANOVA's associated with CPU time was taken during the experiment. Section 6.2 shows performance of algorithm on very large problems up to 1000 products to assure the quality of the algorithms. Section 6.3 gives a conclusion to the research.

6.1 ANOVA tests

For quality of our proposed methods A and B analysis of variance ANOVA's of the output can be constructed for analysing both the CPU time taken to determine near or optimal solution and for evaluating the accuracy of the heuristics used. Table 6.1 shows the CPU time was taken during the executing of the algorithms A and B. The ANOVA for evaluating the solution time in set M1, M2 and M3 are shown in Tables 6.2, 6.2 and 6.3.

Figures 6.1,6.2 and 6.3 shows the results of the ANOVA test for the CPU time was taking during the experiment of the Methods A and B compare to other heuristic.

The results of the ANOVA in sets M1, M2 and M3 indicate that the CPU time in proposed algorithms A and B to obtain the optimal or near optimal solution are significantly better than the problems with other heuristics. This conclusion is consistent with the cooling and freezing schedules used in our methods in the previous chapter. There are significant different in CPU time in large problems. In methods A and B to obtain the near optimal solution CPU time were 1.6 minutes, which is in the other solution the CPU time to obtain the optimal solution were 2.5 minutes.

Table 6-1: CPU times in sets M1, M2 and M3

Problem name	Methods	M1	M2	M3
		CPU	CPU	CPU
A	A	0.0500	0.08330	0.4830
	B	0.0500	0.08330	1.6000
	Other Heuristic	2.5000	2.49000	2.5000
B	A	0.0500	0.08330	0.4830
	B	.05000	0.10000	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
C	A	0.0500	0.08330	0.8430
	B	0.0500	0.10000	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
D	A	0.0500	0.08330	0.8430
	B	0.0667	0.11700	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
E	A	0.0500	0.10000	0.8430
	B	0.0667	0.13300	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
F	A	0.0500	0.10000	0.8430
	B	0.0667	0.13300	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
G	A	0.0500	0.10000	0.8430
	B	0.0667	0.13300	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
H	A	0.0500	0.10000	0.4930
	B	0.0667	0.13300	1.6000
	Other Heuristic	2.5000	2.50000	2.5000
I	A	0.0667	0.10000	0.4940
	B	0.0667	0.13300	1.6000
	Other Heuristic	2.5500	2.50000	2.5500

Table 6-2: ANOVA test for CPU time in set M1

Source of Variation	Sum of Squares	d. f.	Mean Squares	F
Interaction	35.99	2	17.99	1.4262E+05
Error	3.0279E-03	24	1.2616E-04	
Total	35.99	26		

Table 6-3: 3ANOVA test for CPU time in set M2

Source of Variation	Sum of Squares	d. f.	Mean Squares	F
Interaction	35.37	2	17.19	1.1056E+05
Error	3.7306E-03	24	1.5544E-04	
Total	34.38	26		

Table 6-4: ANOVA test for CPU time in set M3

Source of Variation	Sum of Squares	d. f.	Mean Squares	F
Interaction	18.43	2	9.213	9.2687E+04
Error	2.385E-03	24	9.940E-05	
Total	18.43	26		

Figures 6.1, 6.2 and 6.3 shows the differences in CPU time in the methods.

Figure 6-1: Comparisons of CPU in Methods A, B and Meral

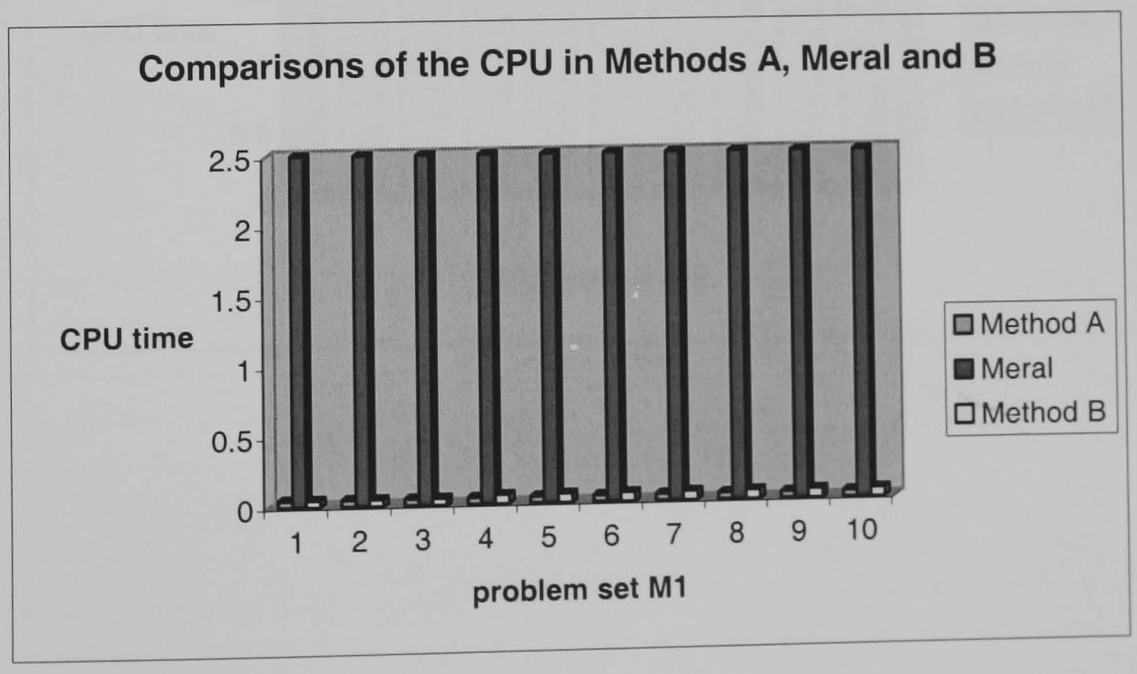


Figure 6-2: Comparisons of CPU in Methods A, B and Meral

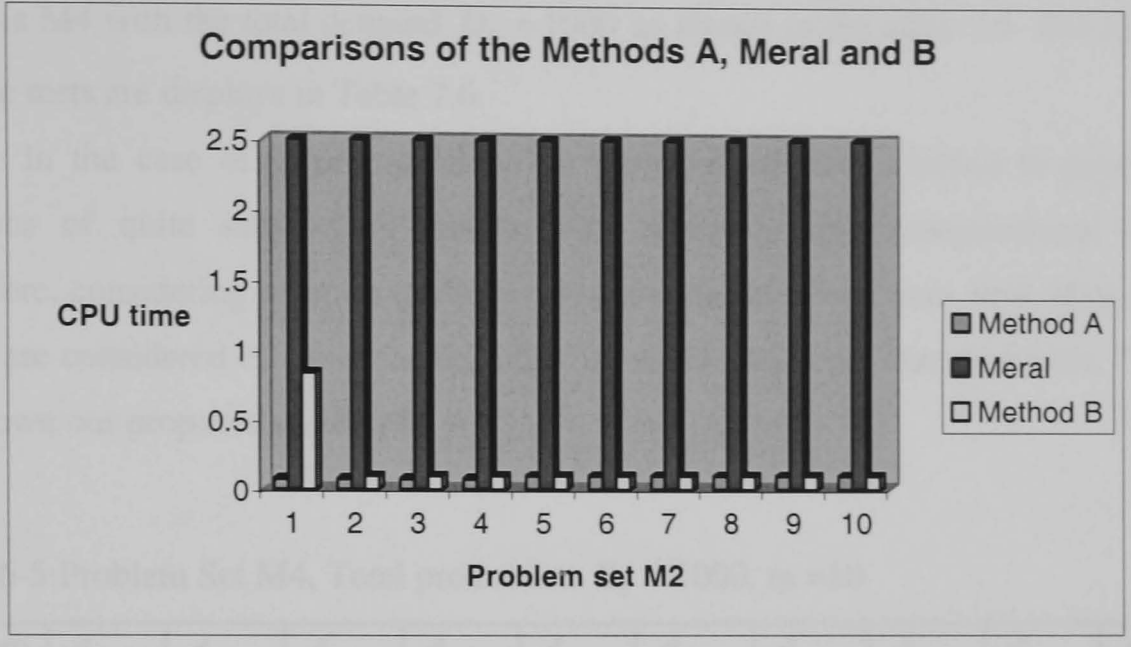
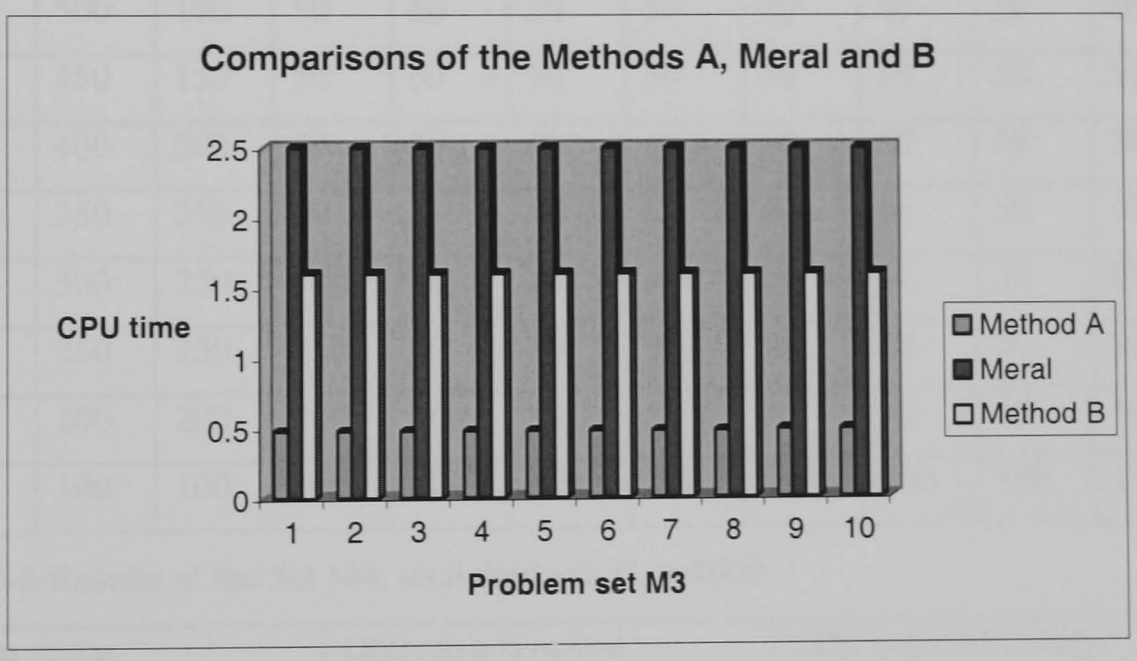


Figure 6-3: Comparisons of CPU in Methods A, B and Meral



6.2 Analysis with 1000 products

To evaluate the quality of the algorithms, we developed the new problem sets which is M4 with the total demand $D_T = 1000$ as shown in the table 7.5. The results of these tests are displays in Table 7.6.

In the case of large problems, the proposed method, Method B produced solutions of quite satisfactory quality with relatively less computational time. Therefore, considering solution quality as well as computational time, both Method A and B are considered efficient for dealing with small and larger size problems. Table 6.6 shown our proposed method B.

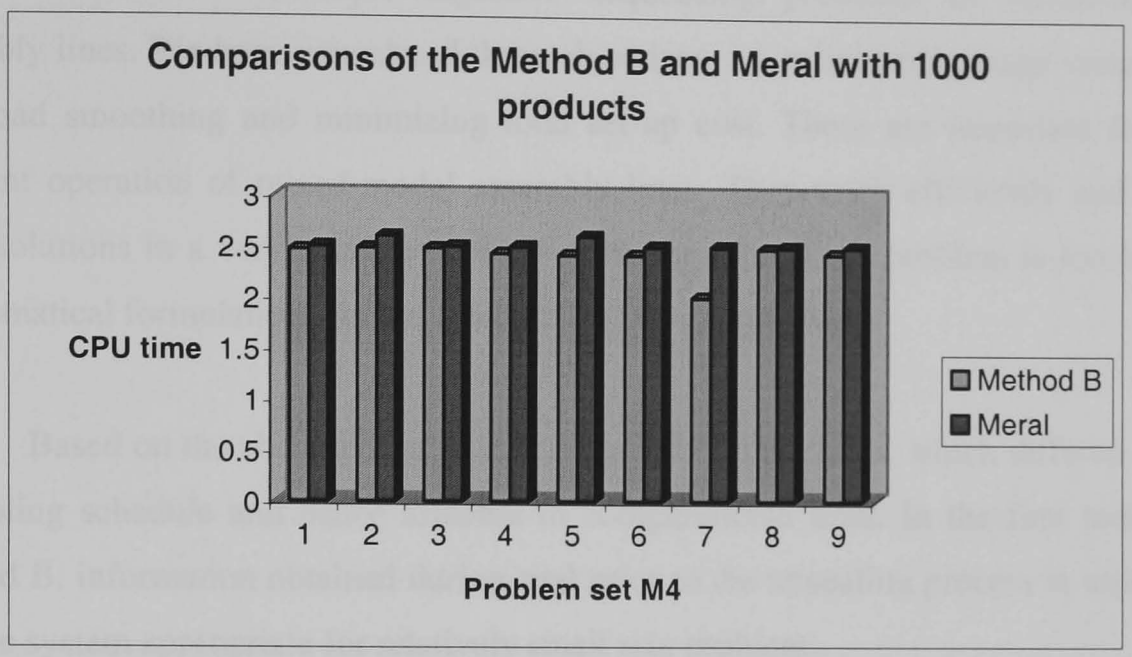
Table 6-5:Problem Set M4, Total production D_T =1000, m =10

Problem name	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
A	550	50	50	50	50	50	50	50	50	50
B	500	100	50	50	50	50	50	50	50	50
C	450	150	50	50	50	50	50	50	50	50
D	400	200	50	50	50	50	50	50	50	50
E	350	250	50	50	50	50	50	50	50	50
F	300	250	100	50	50	50	50	50	50	50
G	250	250	150	50	50	50	50	50	50	50
H	200	200	200	100	50	50	50	50	50	50
I	100	100	100	100	100	100	100	100	100	100

Table 6-6:Results of the Set M4, total demand D_T =1000

Problem name	Objective function	CPU time (Minutes)
A	15005.3999	2.4667
B	16204.1999	2.4833
C	17970.0999	2.4833
D	18006.2000	2.4333
E	19027.0999	2.4167
F	17536.4999	2.4167
G	20141.4999	2.0000
H	22015.0000	2.4833
I	24935.9999	2.4333

Figure 6-4: Comparisons of CPU for 1000 products



Based on the testing analysis of variance (ANOVA) and testing our proposed methods A and B with large number of variety of products, and experiments in chapter five, our method A uses a simpler procedure for temperature tuning, i.e., it determines the next temperature according to a prescribed cooling function. This function used in the proposed method A cools faster, and thereby contributes to saving computational time.

On the other hand, the second method, method B, information obtained during trials prior to the annealing process is utilized, and the system is cooled very slowly near an unknown critical temperature, thereby making this method appropriate for relatively small and medium size problems.

Conclusion

In this research, a new simulated annealing algorithm is developed to obtain optimal solutions to multiple objective sequencing problems in mixed-model assembly lines. We have considered three objectives, i.e. minimizing usage variation, workload smoothing and minimizing total set-up cost. These are important for an efficient operation of mixed-model assembly lines. They work efficiently and find good solutions in a very short time, even when the size of the problem is too large. Mathematical formulations for the three objectives are provided.

Based on the above concepts, we proposed two methods, which differed only in cooling schedule and hence affected in computational time. In the first method, method B, information obtained during trial prior to the annealing process is utilized, and the system appropriate for relatively small size problem.

On the other hand, the second method, method A uses a simpler procedure for temperature tuning, i.e., it determines the next temperature according to a prescribed cooling function. This function used in the present study cools faster, and thereby contributes to saving computational time.

In order to evaluate the proposed methods, we conducted a number of numerical computations using the standard problems of Sumichrast et.al. [52], as well as some very large problems with 1000 products developed in the mixed-model assembly lines problem for this research. Based on the computational results, Method B was shown to obtain higher quality solutions than the other methods for all size of problems, but the expense of relatively high computational effort in the case of large-scale problems. In order to obtain the optimal solution for large-scale problems, it may not be reasonable to look for a promising optimisation algorithm, which makes implicit enumeration with less memory requirements; the computational time may grow extremely high, unless a tight lower bound found, because the number of possible sequences is equal $D_T/(d_1!d_2!..d_m)$ which may become extremely large, depending on the length of the planning horizon and/or the demand figures.

Finally, although the simulated annealing algorithms proposed here are the mixed-model assembly lines problems, it is possible to apply these methods to other combinatorial optimizatin problems by finding rules or structure based on the characteristics of the problem. It may provide higher quality solutions with increasing efficiency in comparison with other heuristic methods available in the literature.

REFERENCES

References

Aarts, E. H. L., and Korst, J.H.M.," Simulated Annealing and Boltzmann Machines", 1989,Wiley, Chichester,

Aarts, E. H. L.and van Laarhoven, P.J.M.,"Simulated Annealing: Theory &Application", 1987, Kluwer Academic publisher, Dordrecht.

Baykoc, O. F., Erol, S.," Simulation modelling and analysis of a JIT production system," ***International Journal of Production Economics***, 55(1998), p.203-212.

Bailey, D. and Hubert, T., " Productivity Measurement", Gower, London, 1980.

Burkard, R. E., and Rendle, F., " A thermodynamically motivated Simulation procedure for combinatorial optimization problems", ***European Journal of operational research***, Vol.17, (1989), 169-174.

Burns, L. D. and Daganzo, C. F. " Assembly line job sequencing principle". ***INT. Iournal of production Research***, Vol.25, (1987), 71-99

Cheh, K.M., Goldberg, J.B., and Askin, R.G.,"A note on the effect of neighbourhood structure in simulated annealing", ***Computers operation research***, 14, (1991), 537-547.

Chyr, F. T. Lin and Ho, F-Y, " Compassion Between Just-in-time and EOQ Systems, ***Engineering Costs & Production Economics***, 18, (1990), 3, p.233-240.

Connolly, DT. " An improved Annealing scheme for the QAP", ***European Journal of Operational Research***, Vol.46, (1990), 93-100.

Crosby, L. " The Just-in-time Manufacturing Process Control of Quality and Quantity, ***Production and inventory Management***, 4, (1984).

DAR-EL, EM, and Cother, "Assembly line sequencing for model-mix". ***International journal of production research***, 13(5), (1975), p.463-477.

Ding, F. Y., and Cheng, L.,"A simple sequencing algorithm for mixed-model assembly lines in just-in-time production systems", ***Operations research***

letters, 13, (1993a) p.27-36.

Ding, F. Y. And Cheng, L., "An effective mixed-model assembly line sequencing heuristic for just-in-time production systems", ***Journal of operations Management***, 11, (1993 b), p.45-50.

Ding. Y. and Zhu, J, "A transformed two-stage method for reducing the par-usage variation and a comparison of the product-level and part-level solutions in sequencing mixed-model assembly lines", ***European Journal of Operational research***, 127, (2000), p.203-216.

Dilworth, J. "Production and Operations Management " 3rd ed., Random House Inc., New York, 1986.

Ebrahimpour, M. "An Examination of Quality Management in Japan Implications for Management in the United States", **Journal of Operations Management**, 5(4), 1985,p. 419-431.

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

Eilon, S. B. Gold and Soeson, J., "Applied Productivity Analysis for Industry" Perggamon Press, Oxford, 1976.

Golden, B. And Skiscim, C., "Using simulated annealing to solve routing and location problems, ***Nav.Res.Q.*** 33, (1986), p.261-279.

Gaither, N. "Production and Operations Management", 3rd ed., Stamford Connecticut, 1983.

Huang, C. C. and Kusiak, A., " Manufacturing control with a push-pull approach. ***International Journal of Production Research***, 36, (1998), p.251-275.

Hall, N.,"Zero inventories", 1983, Dow Jones-Irwin, Homewood, IL.

Hall, N., Kubiak, W., and Sethi, S. "Earliness-Tardiness Scheduling Problems, 11: Deviation of Completion Times about a Restrictive Common Due Date", ***Operations Research***, 39(5), (1991), p 847-856.

Hannah, K. "Just-in-time: Meeting the Competitive Challenge", ***Production and Inventory Management***, 28(3), 1987, p 1-3

Inman, R.R., and Bulfin, R.L.,"Sequencing just-in-time mixed-model assembly lines", ***Management science***, 37(7), (1991), 901-904.

Kim, Y., Kim, Y. K. and Kim, J. Y. 1"A coevolutionary algorithm for balancing and sequencing in mixed-model assembly lines", ***Applied intelligence*** 13 (2000), 247-258.

Kirkpatrick, s., Gelatt, JR., C. D., and Vecchi, M. P.,"Optimization by Simulated Annealing", ***IBM Computer Science/Engineering Technology Report***, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, (1982).

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

Koufteros, X. A.," Testing a model of pull production: A paradigm for manufacturing research using structureal equation modelling", ***Journal of Operations Management***, 17, (1999),p. 467-488.

Kouvelis, P., and Chiand, W.," A simulated annealing procedure for single row layout problems in flexible manufacturing systems", ***International Journal of Production research***, Vol. 30, (1992), 717-732.

Kubiak, W., and Sethi, S., "Level schedules for mixed-model assembly lines in just-in-time production systems", ***Management science***, 37(1), (1991), 121-122.

Kilbridge, MD, and Wester, L., "The assembly line model-mix sequencing problem". Proc. of the third int. conf. op. Res. English universities press (Paris: Dunod Editor), (1963).

Lundy, M., and Mees, A., "Convergence of annealing algorithm" ***Mathematical programming***. 34/1, (1986), 11-124.

McMullen, P. and Frazier, G. V. " A simulated annealing approach to mixed-model sequencing with multiple objectives on a just in time line", ***IIE***

Transactions, 32,(2000), p. 679-686.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E., "Equation of State Calculations by Fast Computing Machines", ***Journal of Chemical physics***, Vol. 21, (1953), p.1087-1092.

Meral, S. and Korkmazel, T. " Bicriteria sequencing methods for the mixed-mode assembly line in just in time production systems", ***European Journal Of operational Research***, 131, (2001), p.188-207.

McLaughlin, M. P., "Simulated annealing", ***Dr. Dobb's journal***, (1989), 26-37. 1

Miltenburg, G. J., "Level schedules for mixed-model assembly lines in just- in-time production system", ***Management Science***, 35(2),(1989),p.192-207.

Monden, Y., "Toyota production system" (Norcross, GA: Institute of Industrial Engineers Press, (1983).

Monden, Y., " What Makes the Toyota Production System Really Work" 13(1), ***Industrial Engineering***, 1981(a).

Monden, Y., " Adaptable Kanban System Helps Toyota Maintain Production" 13(5), ***Industrial Engineering***, 1981(b).

Monden, Y., " Smoothed Production Lets Toyota Adapt to Demand Charges and Reduced Inventory", 13(8), ***Industrial Engineering***, 1981(c).

Monden, Y., " How Toyota Shortened Supply Lot Production Time, Waiting Time and Conveyor Time", 13(9), ***Industrial Engineering***, 1981(d).

Miltenburg, G. J., "Level schedules for mixed-model assembly lines in Just-In-time production system", ***Management science***, 35(2). (1989), 192-207.

Meral, S., and Krokmazel, T., "Bicriteria sequencing methods for the mixed-mode assembly line in JIT production systems", ***European journal of operatioal research***, 131, (2000), 188-207.

Nelson, R. " Research on Productivity Growth and Productivity Differences Dead Ends and New Departures", ***Journal of Economic Literature***, 19, (1981).

Okamura, K., and Yamashina, H., "A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the Conveyor", ***International journal of production research***, 17, (1979), 233-247.

Ohno, T., "The origin of Toyota production system and Kanban system, In applying just in time": The American/Japanese Experience, Monden, Y. (ed.). 1986, Institute of Industrial Engineers Press. Nocross, GA.

Ohno, T., " Toyota Production System: Beyond Large Scale Production", Productivity Press, Cambridge, MA, 1988.

Oliver, N. " Human Factors in the Implementation of Just-in-time Production", ***International Journal of Operations and production Management***, 10(4), (1990), P 32-40.

Ozbayrak, M., Cagil, G., Kubat, C., " Performance modelling of an automated machining cell: Push/Pull control and comparisons", Research report Department of Systems Engineering, Brunel University,Uk, (2002).

Ozbayrak, M., Cagil, G., Kubat, C., " How successfully does JIT handle machine breakdowns in an automated manufacturing system? Integrated Manufacturing Systems", ***International Journal of Technology Management***, in press.

Prenting, T. O., and Battaglin, R. M., "The precedence diagram: A tool for analysis in assembly line balancing", ***Journal of industrial engineering***,15, (1964), 208-213.

Palshikar, G.K.,"Simulated Annealing: A Heuristic Optimization Algorithm", ***Dr. Dobb's journal***, (2001), 121-124.

Pyke, D. and Cohen, M. " Push and Pull in Manufacturing and Distribuation Systems", ***Journal of Operations Management***, 9(1), 1990, p 24-43.

South, J. " A Minimum Production Lot-Size Formula for Stockless Production",

Production and Inventory Management, 27(2), 1986.

Stevenson, W." Production/Operations Management", 3rd ed. Richard D. Irwin inc. Homewoowd 111,1990.

Suri, R. and DeTreville, S. " Getting from Just-in-time to Just-in-time, ***Journal of Operations Management***, 6(3), (1986), p 295-304.

Sumichrest, R. T., and et al., "A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system", ***int. J. prods. Res***., 30(1), (1992), p.199-214.

Sumichrast, R.T. and Russel, R.S.,"Evaluting mixed-model assembly line sequencing heuristics for just-in-time production systems", ***Journal of operations Management***, 9, (1990), 371-390.

Thomopoulos, N. T., "Line balancing sequencing for mixed-model assembly", ***Management science***, 14(2), (1994), p.59-75.

Thesen, A.," Some simple but Efficient Push and Pull Heuristics for production sequencing for certain flexible manufacturing systems, ***International Journal of Production Research***, 37,(1999),p.1525-1539.

Ventura, J. and Radhakrishnan, S. " Sequencing mixed-model assembly lines for a Just in time production system", ***Production Planning & Control***, 13 (2), (2002), p.199-210.

Xiaobo, Z. And Ohno, K., "A sequencing problem for a mixed-model assembly line in a just-in-time production system", ***computers and industrial engineering journal***, 27, (1994) p.71-74.

Xiaobo, Z. And Ohno, K., "Algorithms for sequencing mixed-models on an assembly lines in a just-in-time production system, ***computers and industrial engineering journal***, 32(1),(1997)p.47-56.

Wang, T. Y., Wu, K. B. and Liu, Y. W. " A simulated annealing algorithm for facility layout problems under variation demand in cellular manufacturing systems", ***Computers in Industry***, 46, (2001), p.181-188.

Wilhelm, M. R. And Ward, T. L., " Solving quadratic assignment problem by simulated annealing", ***IEE transactions***, Vol.1, No. 1, (1987),107-119

Weiner, S., "Perspective on automotive manufacturing, in the manufacturing of productivity and technology in manufacturing", edited by Paul R. Kleindorfer (New York: Plenum Press), (1985), p.57-71.

Wild, R., (1972), Mass production management, the design and operation of production flow line systems, (New York: John Wiley and sons, inc.).

Yano, C.A. and Matanachai, S., " Balancing mixed-model assembly lines to reduce work overload", ***IIE Transactions***, 33, (2001), p.29-42.

Zegordi, S. H., Itoh, K., Enkawa, T. and Chung, S. " Simulated annealing scheme incorporating move desirability table solution of facility layout problems". ***Journal of the Operations Research***, 38, (1995), p.1-20.

APPENDIX

Appendix A, tables 13-15

Table 13.

Problem Set M1, Total production $D_T = 20$, $m = 5$

Problem name	d_1	d_2	d_3	d_4	d_5
A	16	1	1	1	1
B	15	2	1	1	1
C	13	4	1	1	1
D	10	5	2	2	1
E	8	7	2	2	1
F	6	6	5	2	1
G	5	5	5	3	2
H	5	4	4	4	3
I	4	4	4	4	4

Table 14.

Problem Set M2, Total production $D_T = 20$, $m = 10$

Problem Name	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
A	11	1	1	1	1	1	1	1	1	1
B	10	2	1	1	1	1	1	1	1	1
C	9	3	1	1	1	1	1	1	1	1
D	8	4	1	1	1	1	1	1	1	1
E	7	5	1	1	1	1	1	1	1	1
F	6	5	2	1	1	1	1	1	1	1
G	5	5	3	1	1	1	1	1	1	1
H	4	4	4	2	1	1	1	1	1	1
I	2	2	2	2	2	2	2	2	2	2

Table 15.

Problem SetM3, Total production $D_T = 100$, $m = 15$

Problem name	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}
A	40	40	8	1	1	1	1	1	1	1	1	1	1	1	1
B	35	35	10	5	5	1	1	1	1	1	1	1	1	1	1
C	30	30	15	10	5	1	1	1	1	1	1	1	1	1	1
D	25	25	20	15	5	1	1	1	1	1	1	1	1	1	1
E	20	20	20	10	10	1	1	1	1	1	1	1	1	1	1
F	20	20	15	15	10	6	6	1	1	1	1	1	1	1	1
G	15	15	15	10	10	10	10	5	4	1	1	1	1	1	1
H	15	15	10	10	10	10	10	10	4	1	1	1	1	1	1
I	7	7	7	7	7	7	7	7	7	7	6	6	6	6	6

Appendix B

Table C1.1 **Value of Pearson's R**

	Level of Significance for One-Side H1			
	0.05	0.025	0.01	0.005
	Level of Significance for Two-Side H1			
d.f.	0.1	0.05	0.02	0.01
2	0.9000	0.9500	0.9800	0.9900
3	0.8054	0.8783	0.9343	0.9587
4	0.7293	0.8114	0.8822	0.9172
5	0.6694	0.7545	0.8329	0.8745
6	0.6215	0.7067	0.7887	0.8343
7	0.5822	0.6664	0.7498	0.7977
8	0.5494	0.6319	0.7155	0.7646
9	0.5214	0.6062	0.6851	0.7079
10	0.4973	0.5760	0.6581	0.7079
11	0.4762	0.5529	0.6339	0.6835
12	0.4575	0.5342	0.6120	0.6614
13	0.4409	0.5139	0.5923	0.6411
14	0.4259	0.4973	0.5442	0.6226
15	0.4124	0.4821	0.5577	0.6055
16	0.4000	0.4683	0.5425	0.5897
17	0.3887	0.4555	0.52185	0.5751
18	0.3783	0.4438	0.5155	0.5614
19	0.3687	0.4329	0.5034	0.5487
20	0.3598	0.4227	0.4921	0.5368
25	0.3233	0.3809	0.4451	0.4869
30	0.2960	0.3494	0.4093	0.4487
35	0.2746	0.3246	0.3819	0.4182
40	0.2573	0.3044	0.3578	0.3932
45	0.2428	0.2875	0.3384	0.3721
50	0.2306	0.2732	0.3218	0.3541
60	0.2108	0.2500	0.2948	0.3248
70	0.1954	0.2319	0.2737	0.3017
80	0.1829	0.2172	0.2565	0.2830
90	0.1726	0.2050	0.2422	0.2673
100	0.1638	0.1946	0.2301	0.2540