INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 1338517

Solution to a bay design and production sequencing problem

Creswell, Steven Howard, M.S.

The University of Arizona, 1989

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

Solution to a Bay Design and Production Sequencing Problem

by

Steven Howard Creswell

A Thesis Submitted to the Faculty of the

DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN INDUSTRIAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1989

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Steve Cromwell

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Jeffrey B. Goldberg
Assistant Professor of Systems
and Industrial Engineering

9/29/89
Date

ACKNOWLEDGMENTS

I would like to express my deep appreciation to my advisor, Dr. Jeffrey Goldberg for his guidance and assistance. His suggestions and criticisms shaped many of the ideas contained within this thesis. I would also like to thank Dr. Ronald Askin for his assistance. He made many valuable suggestions that have been incorporated into the thesis.

Finally, I would like to express thanks to my wife, Leslie, for her love, support, and patience throughout my graduate work, as well as to my parents without whose encouragement my education would not have been possible.

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS - Continued

LIST OF ILLUSTRATIONS

Figure **Page**

Figures 1.1 and 1.2 courtesy of *Quad Systems Corporation*

LIST OF TABLES

ABSTRACT

This thesis addresses the problem of setting up a surface mount placement machine for production. The objective is to minimize the number of machine changeovers made during a production run consisting of a number of circuit cards. The solution to the problem involves two separate decisions. The first decision considers determining how to combine feeders together in "bays" or groups of feeders, and how to assign the bays to the circuit cards. The second decision considers the circuit card production sequence. A mathematical programming formulation is given, however, its solution is very difficult for problems of a realistic size. Several heuristic approaches are suggested and used to solve actual and test problems. The heuristic for bay design uses clustering techniques used in Group Technology while the sequencing problem is solved using heuristics based on solution techniques for the Traveling Salesman problem.

CHAPTER 1

INTRODUCTION

1.1 Overview

Surface Mount Technology (SMT) circuit cards and components are becoming increasingly prevalent in electronic assemblies. Due to their higher density and smaller size they are especially advantageous in complex electronic assemblies such as computer and defense products. Those products are often assembled in high-mix, low-volume manufacturing environments.

One of the primary steps in the production process is the placement of the electronic components onto the circuit cards. An SMT placement machine is shown in Figure 1.1. There is a wide variation in the types of SMT placement equipment available. The placement machine pictured is representative of "mid-range" equipment specifically designed for maximum flexibility and accuracy rather than processing speed. This type of machine is typically used in applications where a wide range of component types are being used in relatively low rate production. A single placement machine may be used to assemble many different products using many different component types. Therefore, it is desirable to have the ability to rapidly and simply change the machine setup between products.

1.2 Problem Description

The situation considered in this thesis is the task of setting up a single SMT placement machine for a production run consisting of a number of different circuit cards. The machine places components onto the circuit cards using feeders which are arranged around the perimeter of the machine. The feeders are grouped into "bays", which are sets of feeders sharing a common support and attachment base to the machine. A bay of feeders is shown in Figure 1.2. The feeders shown are in plastic stickpacks attached to a vibratory base.

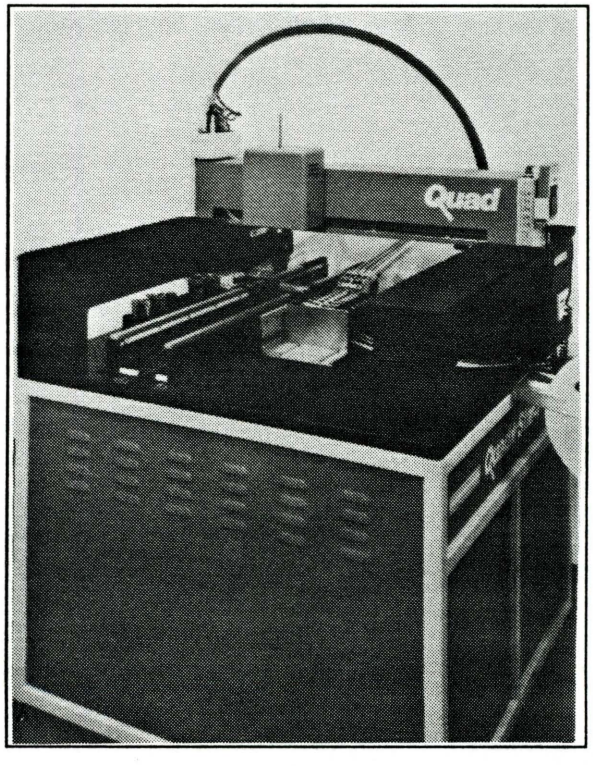


Figure 1.1 SMT Placement Machine

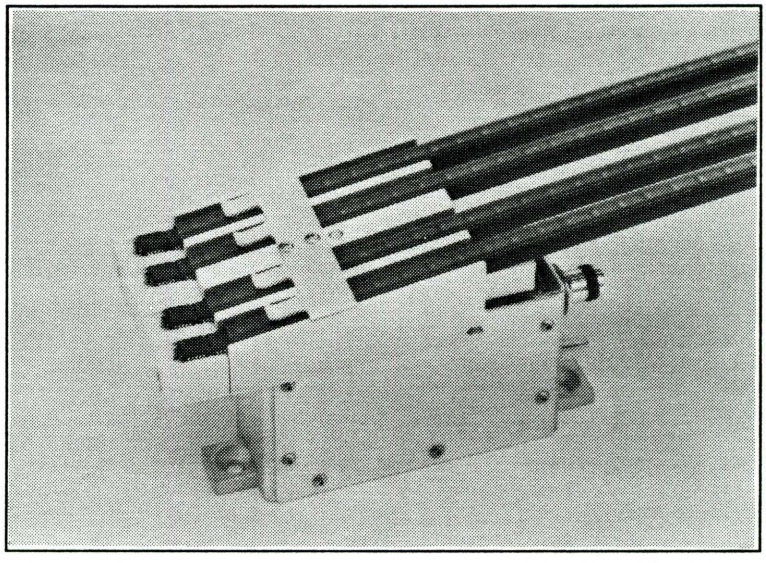


Figure 1.2 Typical Bay of Feeders

A typical production run will occur as follows - The machine is set up for the first circuit card in the sequence by attaching the bays containing all feeders required for that card. One or more of the same circuit card is assembled. The machine is then set up for each card in the sequence by removing unneeded bays and attaching the required bays to the machine. It is assumed that the machine has a large enough bay capacity to satisfy the feeder requirements for any single card. A feeder can be used on more than one bay, but it is assumed that the bays

are formed prior to the production run and individual feeders are not changed during the run.

Passive components such as capacitors and resistors are fed in tape feeders. The components are contained in a roll of tape which is incrementally advanced, moving the components to a pickup point. Larger components including ceramic chips and integrated circuits are generally fed in plastic stickpacks. The stickpacks are attached to a vibratory base at an angle and then gently vibrated into the pickup position. These two types of feeders are not mixed on the same bay. The number of feeders that will fit on a single bay depends on the size of each component. There is also an upper limit on the number of bays that may be placed on the machine since there is limited space around the perimeter of the machine.

Since many components are used on more than a single card, a bay may have to be removed from and reattached to the machine several times during the production run. It is desirable to form the bays and determine the production sequence so that the number of changeovers is as small as possible. This will minimize machine downtime, operator intervention, and reduce the probability of a changeover error. The solution to this problem involves two types of decisions. The first is to determine the assignment of feeders to bays and the assignment of bays to circuit cards. This is referred to as the bay design problem. The second

decision is to find the optimal production sequence in terms of minimizing the number of bay changeovers. This is referred to as the sequencing problem. In the remainder of the thesis the specific problem described above will be referred to as the SMT setup problem.

1.3 Potential Savings

The implementation and use of the methods described in this thesis will involve some investment of time by supervision or production control personnel. It is appropriate to question whether the potential advantages of using the methods justify that investment of time. Consider the alternative method of operation. Changing the machine setup for a different circuit card can be done in one of two ways. The first possibility is that each circuit card will be assigned a dedicated set of bays. In that case the bays for the completed circuit card are all removed and the bays for the next circuit card are attached. While this may be suitable for a relatively small number of circuit cards, as that number increases this policy will result in a large investment in expensive tooling and the associated problem of controlling the bays. An error in attaching a bay to the machine will at best result in a wrong component being placed, and damage to the placement head, components, and circuit card at worst.

A more likely method for setting the machine up is changing individual feeders in the bays. Most components come in standard body widths so it is

possible to remove one component stickpack and replace it with another of the same size. Problems with this method include the amount of time involved in changing individual feeders and the possibility of inserting the wrong component.

1.4 Organization

This thesis will be organized as follows. The literature related to the SMT setup problem is reviewed in Chapter 2. Production sequencing problems have been widely studied while there has been less research done on combining the problem with machine setup. Group Technology clustering problems are also reviewed as they relate to the methods developed in this thesis. A mathematical model of the problem is developed in Chapter 3. The model is a binary integer formulation. The nonconvexity of the continuous relaxation of the problem is also examined. This shows the difficulty of determining an optimal solution to the model. In Chapter 4 heuristic solution methods are developed. There are two primary heuristics, one for bay design and the other for determination of the production sequence. The heuristics have been coded in Pascal. In Chapter 5 computational results for the heuristics are presented. The heuristics were applied to both actual data and to randomly generated test problems. Conclusions and directions for further work are given in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

There have been a number of papers written on situations related to the SMT setup problem. Very similar topics are considered in papers published by Bard ([3], 1988) and Tang and Denardo ([15], 1988) & ([16], 1988). These papers address the problem of scheduling jobs on a flexible manufacturing cell which uses an automatic tool interchange mechanism. Integral to scheduling the jobs, the papers consider the assignment of tools to the tool interchange mechanism. Several of the ideas developed in these papers have been used in this thesis.

The general problem of scheduling jobs, without respect to the tools required, has also been extensively studied. This problem can be formulated as a Hamiltonian cycle problem. The Hamiltonian cycle problem is commonly referred to as the Traveling Salesman problem (TSP) and is one of the most widely studied topics in operations research. The amount of setup between jobs can be modeled as a "distance", allowing the sequencing problem to be solved using TSP solution techniques. A wide variety of optimal and heuristic methods have been suggested in the literature.

The Bard and Tang & Denardo papers primarily address the sequencing of jobs. The tool assignment problem is a relatively easily solved subproblem of the sequencing problem. In addition to determining the production sequence and tool (bay) assignment strategy, the SMT setup problem requires that feeders be grouped into bays before they can be assigned to the machine. Ahmadi, Grotzinger, and Johnson ([1], 1988) consider a problem in which feeders are partitioned into two groups to minimize downtime during the assembly of an SMT circuit card. This last paper concentrates on decisions associated with processing a single circuit card rather than a series of circuit cards.

The literature on Group Technology (GT) contains a number of concepts that are applicable to the SMT setup problem. The feeder grouping required in the SMT setup problem is very similar to clustering problems considered in the GT literature. In Group Technology, machines are grouped to form machine cells based on the similarity of the parts they process. These grouping techniques may be applied to grouping feeders based on the number of common circuit cards on which they are used. The clustering problem has been well studied in papers by King ([6], 1980), Stecke ([14], 1986), and Kumar et al ([7], 1986) among many others.

2.1 Sequencing Problems

The TSP or Hamiltonian cycle problem involves finding the shortest complete circuit connecting a group of cities such that every city is visited once. It is a well known NP-Complete problem. A variation on the TSP is the Hamiltonian path problem in which the shortest path is found between an initial and final city. The Hamiltonian path problem is also NP-Complete. (Garey & Johnson [4], 1979) Either of these formulations can be used to model a production sequencing problem; the choice between them depends on the particular situation. The TSP can be formulated as follows:

Formulation

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij} X_{ij} \qquad \text{(TSP1)}$$

Subject To:

$$\sum_{j=1}^{N} X_{ij} = 1 \qquad \forall\ i = 1, .., N \qquad \text{(TSP2)}$$

$$\sum_{i=1}^{N} X_{ij} = 1 \qquad \forall\ j = 1, .., N \qquad \text{(TSP3)}$$

$$\sum_{i \in S}\sum_{j \in S} X_{ij} \leq |S| - 1 \qquad \forall\ S \subset V, S \neq \emptyset \qquad \text{(TSP4)}$$

$$X_{ij} \in \{0,1\} \qquad \text{(TSP5)}$$

The TSP is formulated by defining the problem on a directed graph. The cities are modeled as vertices in the graph while the distances between cities are represented by arcs or edges connecting the vertices. In the formulation above there are N vertices in graph V. Variable $X_{ij} = 1$ if that arc is included in the tour, $X_{ij} = 0$ otherwise. The C_{ij}'s specify the weight or distance of each arc. S is any subset of the vertices in V. Constraints (TSP2) and (TSP3) ensure that exactly one arc enters and leaves every vertex in the graph. Constraints (TSP4) are the subtour elimination constraints.

There has been a great deal of work done on solution techniques for the TSP. A good survey of the problem and the best solution techniques is contained in Lawler ([10], 1985). Lawler also contains a good survey of heuristic solution methods for the TSP. Most heuristics assume "well-behaved" Traveling Salesman Problems in which the distances between cities are symmetric and follow the triangular inequality. The triangular inequality can be expressed as follows:

$$C_{ik} \leq C_{ij} + C_{jk} \qquad \forall \; i, j, k \in \{1,..,N\}$$

Several of the heuristic methods are based on first finding the minimum weight spanning tree of the graph. The minimum weight spanning tree is the set of edges which connect all vertices in the graph with the smallest cumulative

distance. The most successful heuristic methods also incorporate improvement techniques which find locally optimal solutions. The best of these improvement techniques involves interchanging edges of the graph and is described in Kernighan & Lin ([5], 1973).

2.2 Machine Setup Problems

Tang and Denardo have written two companion papers considering scheduling jobs on a flexible manufacturing cell (N/C machine) ([15], 1988) & ([16], 1988). Each job requires a specific set of tools to be present on the machine during processing. Besides scheduling jobs, the problem also involves determining a strategy for assigning machine tools to a multiple position tool interchange mechanism. The papers develop methods to determine both the production sequence and the tool assignments for the machine. This problem will be referred to as the N/C setup problem.

Both papers consider the same problem using different objective functions. In the first paper the objective considered is minimizing the total number of individual tool changes made when processing a sequence of jobs. The second paper considers minimizing the number of instances in which tools are changed. The objective in the first paper is more similar to the objective in the SMT setup problem.

Assume there are N jobs to be processed. Those jobs require a total of M different tools which must fit into C available tool slots. The tool to job requirements are given in an M x 1 vector A_j. The variables P_n are used to count the number of tool switches made after job n. There are two types of decision variables. The first is the sequence in which to process the jobs, the second is the set of tools to place on the machine during processing. Variable $X_{jn} = 1$ if job j is assigned to the nth position in the sequence, $X_{jn} = 0$ otherwise. Let W_n be the M x 1 vector which describes tool assignments for the machine when processing the n^{th} job. The formulation used in the first paper is given below.

Formulation

$$\text{Minimize} \quad \sum_{n=1}^{N-1} eP_n \tag{TD1}$$

Subject To:

$$eW_n = C \qquad \forall \; n = 1, .., N \tag{TD2}$$

$$X_{jn}A_j \leq W_n \qquad \forall \; n = 1, .., N \tag{TD3}$$

$$\sum_{j=1}^{N} X_{jn} = 1 \qquad \forall \; n = 1, .., N \tag{TD4}$$

$$\sum_{n=1}^{N} X_{jn} = 1 \qquad \forall \; j = 1, .., N \tag{TD5}$$

$$P_n \geq W_{n+1} - W_n \qquad \forall \; n = 1, .., N-1 \tag{TD6}$$

$$P_n \geq 0 \qquad \forall\, n = 1, .., N\text{-}1 \tag{TD7}$$

$$W_n \text{ is a 0/1 vector } \;\&\; X_{jn} \in \{0,1\} \tag{TD8}$$

Constraints (TD2) require that all C tool slots be full. Constraints (TD3) require that the tools assigned to the machine during the processing of any job contain all tools required for that job. Constraints (TD4) and (TD5) require that each job be assigned one position in the sequence and that each position in the sequence be assigned one job. The counting variable, P_n, is defined in constraints (TD6) and (TD7), while constraints (TD8) give the binary constraints for all decision variables.

The objective (TD1) corresponds to minimizing the number of bay changes in the SMT setup problem when a "tool" in the N/C setup problem is considered analogous to a "bay" of feeders in the SMT setup problem. All tools or bays required for the first job must be present on the machine at the beginning of the production run and tools or bays are switched between jobs when necessary. In both the N/C setup and the SMT setup problems there is a limit on the number of tools or bays that may be placed on the machine at one time. The objective is to find the production sequence and tool or bay assignments which require the fewest number of changeovers. The SMT setup problem requires the additional complication of designing bays by assigning appropriate feeders to each bay.

Tang & Denardo develop a heuristic method for determining the production sequence based on solving Hamiltonian path problems. They also introduce a tool change policy known as KTNS or Keep Tool Needed Soonest, which is proven to be an optimal policy. This policy dictates that excess tool capacity on any job be filled with the tools which are needed soonest by upcoming jobs in the production sequence. They also show that determination of the tool replacement strategy is an easily solved subproblem of the sequencing problem.

Bard ([3], 1988) has also addressed the N/C setup problem using the same objective considered in the first Tang and Denardo paper (TD1). He offers a mathematically rigorous examination of the same sequencing problem and generates solutions using Langrangian relaxation techniques.

2.3 Group Technology Grouping Problems

As illustrated above, the N/C setup problem is identical to the SMT setup problem once the bays have been formed. The problem of grouping feeders into bays is very similar to the problem of grouping machines into cells in Group Technology. Machine groups are formed based on the similarity of the parts produced on the machines. In the same sense, feeders should be grouped into bays based on the similarity of the cards on which they are used.

There are a number of successful methods described in the literature for forming machine groups. One of the first was suggested by King ([6], 1980). He forms a 0/1 matrix which describes the machine to component requirements. An element of this machine/component matrix (MCM) is set to one, $x_{ij} = 1$, if component j is processed on machine i. King describes a method of sorting the rows and columns of the matrix so that natural groups are formed. There are a number of other papers, including Kumar et al ([7], 1986) and Stecke ([14], 1986), which describe other successful methods for performing this grouping. King's procedure is described as follows:

Step 1 The pattern of cell entries in each row is read as a binary word. The rows are rearranged in decreasing binary order.

Step 2 The pattern of cell entries in each column is read as a binary word. The columns are rearranged in decreasing binary order.

Step 3 If neither the columns or rows was rearranged in Step 1 or Step 2, then go to Step 4, otherwise, go to Step 1.

Step 4 Stop.

In the same paper King describes a method to break the rows (machines) of the matrix into machine groups called the dendogram approach. Similarity coefficients between adjacent rows in the sorted matrix are used to determine where breaks should be made to form groups. This method was first described in McCauley ([9], 1972). A Jacard similarity coefficient between two machines is defined as the number of components processed on both machines divided by the

total number of components processed on either machine. See Section 4.1.1 for a mathematical description. The dendogram procedure determines groupings using these similarity coefficients. Some initial machine ordering is assumed. Initially, all machines are considered as a single group. The machines are broken into groups by making a break at successively higher values of similarity coefficients. Figure 2.1 illustrates the concept.

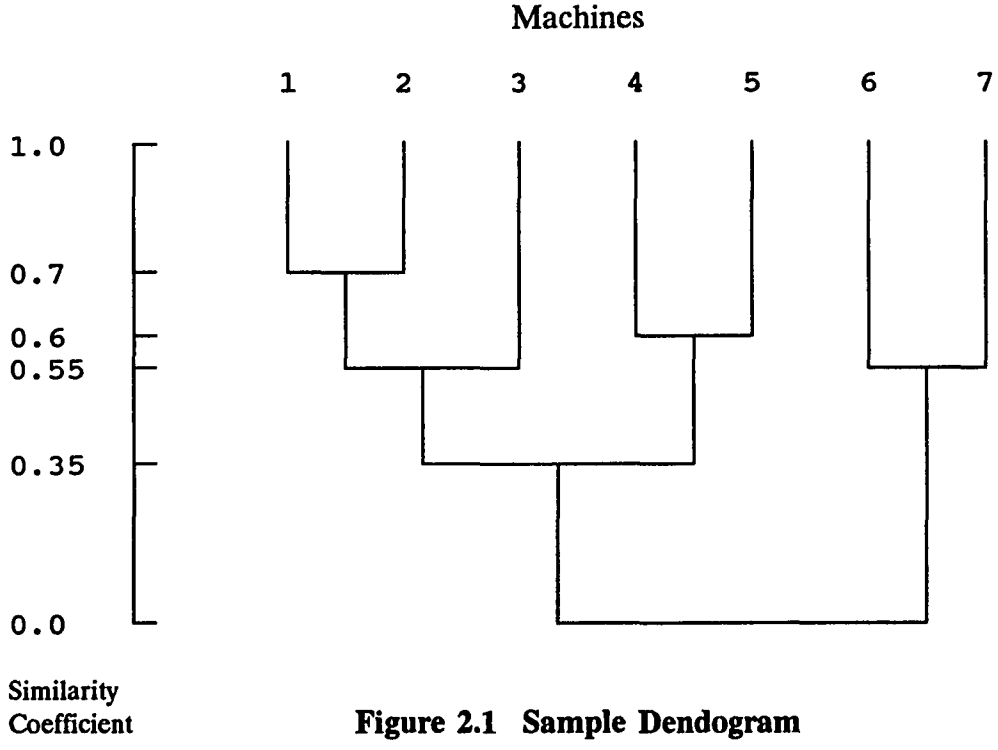


Figure 2.1 Sample Dendogram

At a similarity level of 0.0 all machines are in a single group. At similarity levels between 0.0 and .35 there are two groups of feeders: the first with machines

1, 2, 3, 4, and 5, and the second group with machines 6 and 7. Note that at a similarity level of 1.0 all machines are in a separate group.

These same techniques can be used in the SMT setup problem to form bays by grouping feeders together. A matrix can be formed in which the circuit cards correspond to components and feeders correspond to machines in the MCM. This feeder/circuit card matrix will be referred to as the FCM.

CHAPTER 3

MODEL DEVELOPMENT

3.1 Overview & Notation

The objective function that will be used when developing the formulation is minimizing the number of bay changeovers during a production run. There are two sets of decision variables that will affect this objective. First are the decisions concerning the feeder to bay and bay to card assignments. Second are the decisions associated with sequencing the circuit cards. The formulation developed in this chapter addresses the first set of decisions since the sequencing problem cannot be solved until the bay design decisions have been made. Either of the methods discussed in Chapter 2 (Bard [3] or Tang & Denardo [15]) could be used to solve the sequencing problem once the bays have been formed. It was intended that the bay formation problem and the sequencing problem could be solved iteratively to generate a good overall solution.

The formulation assumes there are N circuit cards to be processed. Those circuit cards require a total of M different feeders which are assigned to P bays. The formulation requires the following notation:

Variables

X_{ij} - equals 1 if feeder i is assigned to bay j, 0 otherwise

Y_{jk} - equals 1 if bay j is assigned to card k, 0 otherwise

Parameters

T_k - the set of feeders required by card k

S_i - the linear space required for feeder i

Q - bay capacity for the machine (# bays)

L_j - feeder space capacity for bay j

3.2 Assumptions

The first assumption is that the bay design formulation is solved using a predetermined production sequence. This sequence can be the solution from a heuristic method or could even be a random sequence. It is intended that the bay design model will be solved iteratively with the sequencing model. Starting with an initial sequence, the solution to the bay design model will be the best possible bay design using that sequence. Then, using the new bay design, the sequencing problem could be resolved, hopefully generating a better overall solution.

The second assumption is that a feasible solution is always possible. No circuit card will require more feeders than can fit on Q bays. The number of bays specified in the model, P, should be large enough to allow a feasible solution for

all circuit cards. Note that the final solution to the model may use only some subset of those bays and that an upper limit on *P* can be found easily by assigning each circuit card its own distinct set of feeders and bays.

A single changeover is defined as the combination of removing one bay of feeders and placing another on the machine. Due to the result of Tang & Denardo, that the KTNS policy is optimal, the placement machine will always be fully loaded with bays. Whenever a bay is removed, another will always replace it. The bay design formulation developed here will always generate a solution which satisfies the KTNS policy.

3.3 Formulation

Formulation

$$\text{Minimize} \quad \sum_{k=1}^{N-1} \sum_{j=1}^{P} |Y_{j,k+1} - Y_{jk}| \qquad \text{(BD1)}$$

Subject To:

$$\sum_{j=1}^{P} (X_{ij}\, Y_{jk}) \geq 1 \qquad \forall\, k = 1, .., N \text{ and } (i \in T_k) \qquad \text{(BD2)}$$

$$\sum_{i=1}^{M} (X_{ij}\, S_i) \le L_j \qquad \forall\, j = 1, .., M \tag{BD3}$$

$$\sum_{j=1}^{P} (Y_{jk}) \le Q \qquad \forall\, k = 1, .., N \tag{BD4}$$

$$X_{ij}, Y_{jk} \in \{0,1\} \tag{BD5}$$

The objective (BD1) counts the number of bay changeovers required when moving through the production sequence. Constraints (BD2) require that the required feeders for each card be assigned to at least one bay assigned to card *k*. Constraints (BD3) restrict the size of feeders assigned to any bay to be less than the space available on the bay. Constraints (BD4) limit the number of bays assigned to the machine for any card to be less than the machine limit.

3.4 Commentary

The SMT Setup problem involves two separate sets of decisions: the TSP sequencing problem and the bay design problem. The bay design formulation developed above is a non-linear, binary integer formulation. The TSP is a well known NP-Hard problem. Direct solution of the combined problem would be very difficult for problems of a large (and realistic) size. The bay design formulation might still be used in a branch and bound procedure to generate

bounds on the solution. In order for these bounds to be valid, the continuous relaxation of the problem would have to be a convex programming problem.

The objective (BD1) could easily be transformed to a linear function similar to (TD1), and is therefore a convex function. It must be determined whether the constraints form a convex region. By inspection constraints (BD3)-(BD4) are linear functions and are therefore convex. If (BD2) generates a convex feasible set, then the relaxed problem could be solved by nonlinear programming code for convex problems. A lower bound on the solution to the bay design problem could then be found. Unfortunately, the feasible set for constraints (BD2) can be shown to be nonconvex. The proof of nonconvexity is given in Appendix A.

CHAPTER 4

HEURISTIC METHOD DEVELOPMENT

As discussed in the previous chapter, finding an optimal solution to the SMT setup problem is a very difficult task. A solution involves two sets of decisions; determining the production sequence for the circuit cards, and specification of the bay design. Due to the difficulty of directly solving the mathematical models, heuristic methods have been emphasized in this thesis. The heuristic methods developed involve a series of steps addressing those decisions separately.

The methods used to determine the bay design are derived from GT clustering techniques. The feeder/circuit card matrix (FCM), which is a binary matrix describing the feeder to circuit card requirements, is sorted to obtain an initial feeder ordering. The result of this step is an FCM with feeders and circuit cards clustered into groups. The feeders are then assigned to bays in sequential order using a method based on the dendogram approach used in forming GT machine groups. Assuming that the bays designed in the dendogram step represent a feasible solution, KTNS distances can be calculated for any given sequence, allowing the production sequence to be determined. A complete description of the bay design procedure is given in Section 4.1.

It is possible that the bay design step will not generate a feasible solution or that a better solution is attainable. Therefore, an improvement step is incorporated to attempt to find a better (and feasible) solution to the bay design problem. This is accomplished by revising the sorted FCM to generate more tightly clustered groups. A selected feeder is duplicated and its circuit card assignments split between the existing and duplicated feeder. The bay design procedure is then repeated using the augmented FCM. This improvement step is repeated a number of times finding the best production sequence each time a feasible bay design is found. A complete description of this procedure is given in Section 4.2.

In order to use heuristic TSP solution techniques to solve the scheduling portion of the problem, a distance matrix must be determined which specifies the "distance" between any two circuit cards. These distances in the matrix will be a function of the amount of setup required between the cards. Since the amount of setup required is dependent on the design of the bays, the bays must be designed before the sequencing problem can be solved. The sequencing problem heuristic is discussed in Section 4.3.

A flowchart for the combined heuristic is shown in Figure 4.1.

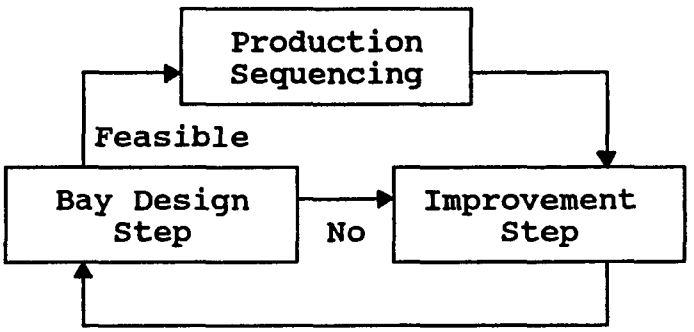


Figure 4.1 General Heuristic Procedure

4.1 Bay Design

There are two separate steps involved in designing the bays. The first is sorting the FCM to obtain a feeder ordering while the second is the actual assignment of the feeders to bays. After feeder assignment, feasibility can be determined. Note that the bay design may have one or more circuit cards assigned to more than the maximum number of bays that can fit on the machine. If this is the case, the improvement step is employed in an attempt to generate a feasible bay design. One or more of these improvement steps might be necessary to generate a feasible bay design. Even if the bay design is feasible, the improvement is used to generate better solutions.

4.1.1 Matrix Sorting

The initial method used to sort the FCM was the binary clustering technique described in King ([6], 1980). The rows and columns of the FCM are sorted iteratively until no further sorting is possible. King shows that the algorithm

will converge in a finite number of iterations. One serious shortcoming of the method is that it does not converge to a unique solution. The final matrix obtained depends on the initial ordering of the rows and columns.

An alternate method of sorting the FCM was developed by applying TSP solution techniques. This technique has the advantage of converging to a unique solution and was found to generate considerably better solutions than binary clustering. Comparative results and a description of the measures used to compare the methods are given in Chapter 5. A two-step approach is used to sort the FCMs. The method is described as follows:

Step 1 A distance matrix is calculated for the rows (feeders) of the FCM using one of the distance metrics defined below.

Step 2 A heuristic tour is generated using the modified spanning tree heuristic. This heuristic is described in Appendix B.

Step 3 A 2-Optimal Hamiltonian path is found by performing interchanges on the tour found in Step 2. The Hamiltonian path method is described in Appendix C.

Step 4 The rows of the FCM are then sorted in the order of the Hamiltonian path found in Step 3.

Steps 1-4 are then repeated for the columns of the FCM.

In order to apply this technique to sorting the FCM, a distance matrix must be defined. Several distance metrics were considered:

Define:

T_i = total # of '1' elements in row (or column) i

C_{ij} = # of '1' elements common to rows (or columns) i & j

D_{ij} = computed distance metric

Metric #1 The number of '1' elements common to any two rows (or columns), C_{ij}, can be counted easily. Let $N = \max C_{ij} \; \forall \; i, j$. Two rows with no common elements will be assigned a distance of N, while two rows with N common elements will be assigned a distance of 0. So, define

$$D^1_{ij} = N - C_{ij}$$

Metric #2 If feeders could be changed individually, this metric would represent the number of feeder changeovers between any two cards. Given any two circuit cards and the feeders required by the first card, this metric is defined as the number of feeders removed after the first card's production plus the number of new feeders required. A similar calculation (but not a similar interpretation) can be made for the distance between any two feeders. So, define

$$D^2_{ij} = T_i + T_j - 2C_{ij}$$

Metric #3 A Jacard similarity coefficient can be defined between any two rows (or columns) in the FCM. This similarity coefficient is in the range (0,1) and is defined as follows:

$$D^3_{ij} = 1-(C_{ij} / (T_i + T_j - C_{ij}))$$

The distance matrix is formed for both the rows and columns of the FCM. Using that distance matrix TSP solution techniques can be used to sort the rows and columns of the FCM. An unsorted FCM will be sorted twice - once

rearranging the rows and a second time rearranging the columns. The distance matrix for the rows is unaffected by the order of the columns and vice versa. The column sort is not strictly necessary for forming bays but is performed to provide an initial sequence for the circuit card sequencing problem. It also allows direct comparison of the TSP sorting techniques to the binary clustering technique since the columns are sorted in that method.

In practice, a Hamiltonian path problem was solved rather than the TSP. There are two reasons why a path is more appropriate than a cycle. First, when sorting the rows (feeders) of the FCM there is no purpose in minimizing the "distance" between the first and last feeder since those feeders will always be assigned to different bays. Secondly, a path is a more appropriate representation of the sequencing problem being modelled. There must be some time allowed at the end of a production run to replenish the feeders. Also, a run might involve a subset or superset of the circuit cards assembled in the previous run.

The Hamiltonian path problems were solved using the following method. Using a distance matrix defined by one of the metrics described above, a tour was generated using a modified spanning tree heuristic which generates "good" solutions with relative simplicity. A complete description of the method is given in Appendix B. A "good" Hamiltonian path was found by dropping one of the edges of the heuristic tour and applying a 2-Opt interchange procedure. The two

different variations of 2-Opt used are based on Lin & Kernighan's interchange techniques. The Hamiltonian path procedure is described fully in Appendix C. Comparative data on the sorting methods is given in Chapter 5.

4.1.2 Feeder to Bay Assignment

Given a sorted FCM, the individual feeders must be assigned to bays. The methods used in this step are based on clustering techniques used in Group Technology.

The simplest method to form bays is to add feeders to bays in the order in which they have been sorted in the FCM. Feeders are added to a bay until the bay is full, then a new bay is started. The shortcoming with this approach is that feeders with relatively little similarity may be placed on the same bay while other very similar feeders may be separated because of the feeder ordering. The FCM sorting step was meant to form natural groups of feeders. These groups are not necessarily recognized using this simple method.

An alternative method is to use similarity coefficients to determine where to make a "break" between two feeders. A "break" between two feeders forces those feeders to be assigned to different bays. A similarity vector can be formed by calculating the similarity coefficient between every adjacent pair of feeders in the sorted FCM. The similarities will fall in the range (0,1), a '0' signifying the

two feeders are used on no common cards, and a '1' signifying the feeders are used on exactly the same cards. This similarity coefficient is identical to D^3_{ij} in Metric #3.

The bay formation procedure is described as follows:

Step 1 The similarity vector is formed by calculating a similarity coefficient between every adjacent pair of feeders in the FCM. All feeders are considered as a single group for Step 2.

Step 2 Bays are formed in each feeder group by sequentially adding feeders to bays until the bays are full.

Step 3 The feasibility of the bay design is assessed.

Step 4 The slack (empty space) on any bay is filled with feeders required by the most infeasible cards determined in Step 3.

Step 5 The (next) lowest similarity is found. A break is forced at that point forming two separate groups of feeders. Go to Step 2.

At higher levels of similarity there will be quite a few forced breaks. The method that was used to solve problems also considered subgroups of those breaks. For example, there might be 5 breaks required at a similarity level of .35. The procedure considered solutions using all 5 breaks, any combination of 4 of those breaks, any combination of 3 of those breaks, etc. Due to computation time constraints, the number of combinations was restricted. No more than 10 breaks were considered with subgroups of (P-4), (P-3), (P-2), and (P-1). (P is the number of breaks)

Step 4 in the above procedure involves filling slack existing on any bay. It is possible that some slack will be left on the last bay formed in each group since a new bay is started with every new group. Filling this slack might very well resolve an infeasibility or reduce the total number of bays assigned. This is accomplished by assessing the feasibility of the bay design before the slack is filled. Circuit cards are ranked in descending order by the number of bays assigned. Starting with the card with the highest number of bay assignments, feeders are reassigned to the slack positions if the reassignment will reduce the number of bay assignments for that card.

The bay formation procedure described above will consider a number of different bay designs. The different bay designs can be compared using two measures. The first, and most important, is the resulting number of infeasible cards. The second is the total number of bays assigned to all circuit cards. This number is the sum of the number of bays assigned to each individual circuit card. The "best" bay design in terms of these measures is retained for use in the sequencing portion of the problem. An important point should be emphasized pertaining to the bay design step - *a feasible solution may not be found.* In many of the test cases and with real data, the best bay design found in this step still contained one or more circuit cards with an infeasible number of bay assignments.

The improvement step used for finding a feasible bay design is described in Section 4.2.

The bay design procedure worked well in forcing breaks between dissimilar feeders. A stopping condition is not mentioned but it is obvious that at higher levels of similarity there will be many forced breaks. This results in individual feeders and small groups of feeders being assigned to bays. It is clear that having slack space on the bays is nonoptimal. Even when this slack is filled (Step 4), the solutions were found to deteriorate at higher levels of similarity.

The results of the bay design procedure can be summarized by another matrix describing circuit card to bay assignments. This bay/card matrix (BCM) is used to determine the production sequence. Summing the elements of the BCM in column j gives the total number of bays assigned to circuit card j. The feeder to bay assignments are feasible only if all the column sums are less than the maximum number of bays allowed on the machine at one time.

4.2 Improvement Step

As discussed in the previous section, the bay formation step will not always result in a feasible bay design. Even if a feasible bay design is found, a better solution may be possible. The improvement step is based on duplicating feeders to create a more tightly clustered structure in the FCM. The matrix sorting step

may result in sorted matrix which still contains some "out of place" elements, as shown in Figure 4.2. One or more feeders may include circuit card assignments which do not "fit" well with the feeders adjacent to it. In Figure 4.2, feeder #13 has two distinct sets of circuit card assignments. The assignment of

		Circuit Cards														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0
	3	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
	4	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0
	5	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
F	6	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
e	7	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
e	8	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
d	9	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0
e	10	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
r	11	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
s	12	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	13	0	0	0	0	0	0	1	1	1	0	0	1	1	1	1
	14	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0
	15	0	0	1	0	0	0	0	1	1	1	0	0	0	0	0
	16	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1
	17	0	0	0	0	1	0	0	0	0	1	0	1	1	1	1
	18	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1

Figure 4.2 FCM Prior to Duplication Step

feeder #13 to cards 7, 8, and 9 is a good fit since the feeders adjacent to it are assigned to same cards. But, the assignment of feeder #13 to cards 12, 13, 14, and 15 is not a good fit since no other feeders near feeder #13 are required by those cards. The bay to which feeder #13 is assigned will have to be attached to the machine for that single feeder when processing cards 12, 13, 14, and 15.

Duplicating feeder #13 and splitting the circuit card assignments would allow the duplicated feeder to be resorted resulting in a more tightly clustered FCM. This step may reduce the number of infeasible cards and the total number of bays assigned. A description of this duplication step is as follows:

Step 1 Candidate feeder/card groups are identified for possible duplication. A candidate is any single feeder and group of cards which does not have more than a single other feeder/card assignment in the four closest vertical positions:

0 - position #1
0 - position #2
1 - Candidate Feeder/Card
0 - position #3
0 - position #4

Step 2 A similarity sum is computed for each candidate feeder and group of cards. That sum is found by computing and summing the similarity coefficient with all other feeders in the FCM.

Example Refer to Figure 4.2. Feeder #13 and cards 12, 13, 14, and 15 comprise one such feeder/card group candidate. That group has nonzero similarity with feeders 4, 16, 17 and 18:

$D_4 = 1/(4+5-1) = .125$
$D_{16} = 4/(4+7-4) = .571$
$D_{17} = 4/(4+6-4) = .667$
$D_{18} = 3/(4+5-3) = .50$

Similarity Sum = 1.863

The feeder/card group with the highest similarity sum will be duplicated. The feeder is duplicated and assigned to the cards in the feeder/card group. Following the duplication step, the feeders of the FCM are resorted to find the best position for the new, duplicated feeder. The columns of the FCM are not resorted. The reason for this is that resorting the cards was found to mix the matrix, nullifying the effects of the duplication. In practice this duplication procedure was found to work well for a number of iterations. In the initial stages the duplication step was able to find good candidates for duplication, but, as the matrix became more tightly clustered, this duplication step was not as beneficial.

4.3 Sequencing

The actual production sequencing problem can only be solved when the bay design step described in Section 4.1 has been completed, resulting in a feasible solution. When using the KTNS policy, a distance measure between two cards cannot be formed since KTNS distances are dependent on the sequence and are not symmetric. This fact precludes the use of the heuristic sequencing methods used in FCM sorting to find the production sequence. Instead, the production sequence was determined by finding a good starting sequence and applying 2-Opt to improve on that sequence.

Two methods were used to find the starting production sequence. The first starting sequence was found using the Hamiltonian sorting methods used in matrix

sorting. The distance matrix used was formed for the BCM using Metric #3 (similarity coefficients). The second starting sequence was the card ordering obtained in the FCM sorting step.

These two solutions can be compared by calculating the KTNS distance for both sequences. Using the best of the two sequences as a starting point, a 2-Opt solution is found by interchanging cards in the sequence. Because of the lack of symmetry in KTNS distances, the number of changeovers must be computed moving both forward and backward through any sequence.

4.4 Combined Heuristic

The combined heuristic starts by sorting both the rows and columns of the initial FCM. The dendogram procedure is then applied to find the best initial bay design. If a feasible solution is found the best production sequence is determined using the KTNS sequencing procedure. Next, the FCM is analyzed to identify a feeder to be duplicated in the improvement step. That feeder is duplicated and its circuit card assignments split between the existing and duplicated feeders. The feeders are then resorted and the procedure is repeated. If a feasible bay design is not found during any iteration then the sequencing step is skipped, going directly to the improvement step. A flow chart for the entire heuristic procedure is shown in Figure 4.3.

Note that beyond the initial sorting of the circuit cards in Step 1, the columns of the FCM are not sorted again. Even when a feasible bay design is found and a KTNS production sequence determined, the new sequence is not returned to the bay design portion of the problem as was suggested in Chapter 3. The reason for this is to preserve the tightly clustered structure of the FCM. But, after some number of iterations the procedure could be stopped and the best circuit card production sequence found could be used to restart the procedure. Results using this resequencing step are given in Chapter 5.

Program Flow Chart

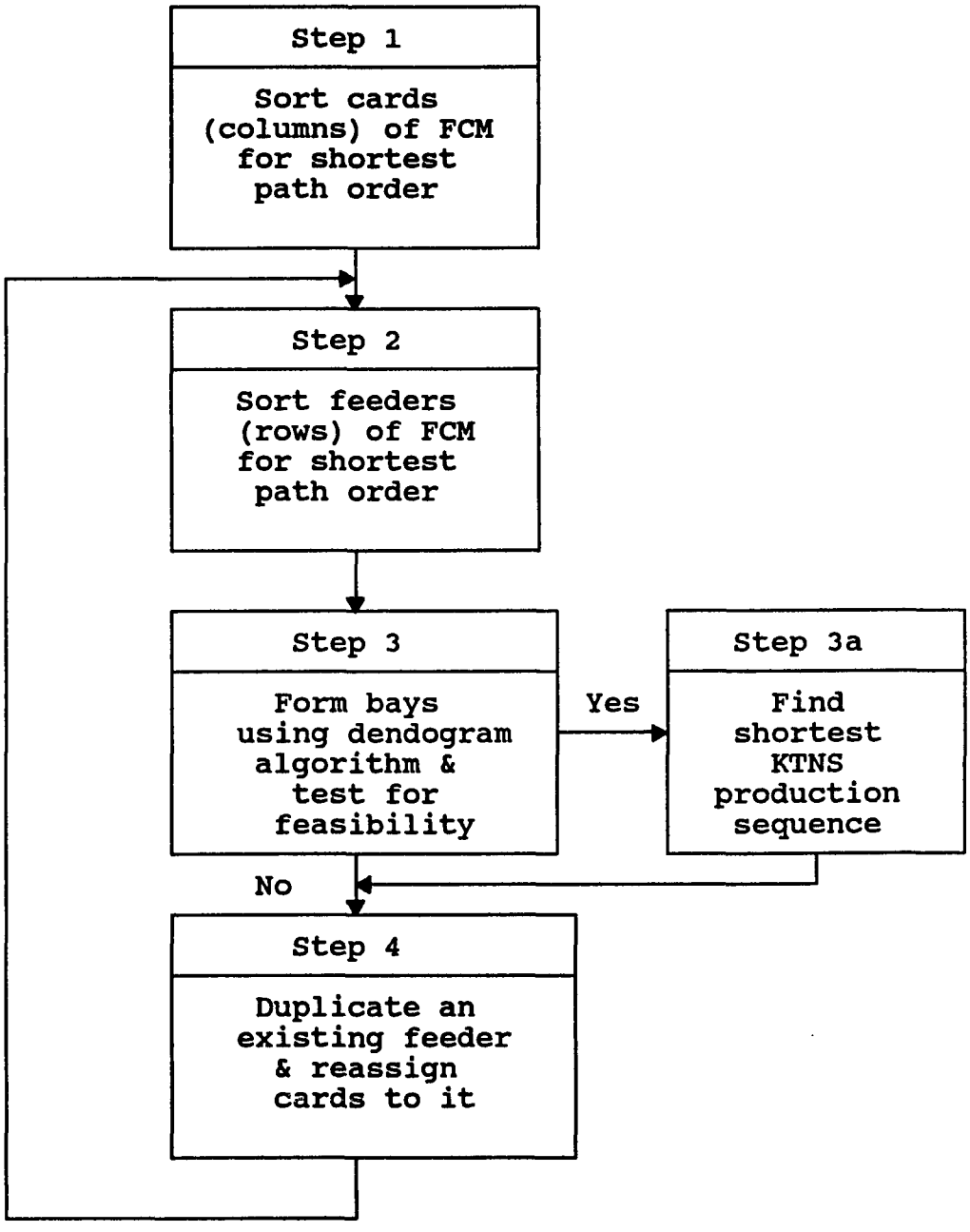


Figure 4.3 Detailed Heuristic Procedure

CHAPTER 5

COMPUTATIONAL RESULTS

The overall heuristic method used to solve the SMT setup problem involves several different steps. In each step a number of different methods and parameters were tested. The parameters were compared by applying them to 10 problems, denoted T1-T10. T1 is actual data while T2 - T10 are randomly generated test problems with characteristics similar to the actual data set. In this section we detail the results of these tests. The T1 data set involved 20 different circuit cards using 40 different feeders. The test problems were of the same size as the actual data set. The problem characteristics are summarized in Table 5.1.

Test Problem Characteristics

Problem #	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Total # Fdrs Assigned	221	227	197	226	205	210	217	231	231	261
Max. # of Fdrs/card	15	16	17	17	15	18	19	21	21	18
Ave. # of Fdrs/card	11.0	11.3	9.8	11.3	10.2	10.5	10.8	11.9	11.5	13.0

Table 5.1

The following assumptions were made when generating and solving the test problems:

1) Each feeder is assumed to have a size of 1.0 unit.
2) All feeders are of the same type (stickpacks).
2) Each bay is assumed to have a size of 4.0 units (holds 4 feeders).
3) The machine will hold a total of 6 bays (24 feeders).

5.1 Bay Design

As described in Chapter 4, the bay design problem involves two sets of decisions and procedures, matrix sorting and feeder to bay assignment. Each of the methods was tested using several different parameters. Comparative results from applying the different methods to the test problems are given.

5.1.1 Matrix Sorting

Two general methods were used to sort the feeder/circuit card matrices. The first is binary clustering and the second is Hamiltonian path sorting. Within Hamiltonian path sorting three different distance metrics were used.

In order to evaluate the quality of a sort procedure some form of measurement must be used. The most obvious measure is to apply the dendogram procedure to the sorted matrices and compare the total number of bays assigned and the number of infeasible cards. In general, a small total number of bays assigned and a small number of infeasible cards indicates a good

sort. But, these measures depend on the size and number of feeders in the FCM. Also, the breaks made in the FCM will not necessarily recognize the clusters that have been formed in the sort, so the measures can exhibit inconsistent behavior. It is desirable to use a measure which is independent of the feeder sizes and the breaks made in forming bays.

A more consistent measure is described as follows. Each row and column of the FCM contains one or more '1' elements. Before sorting, these elements are spread out across the entire FCM. The purpose of the sorting step is to bring these nonzero elements together into clusters. The quality of the sort can be measured by counting the number of '1-groups' in every row and column. A 1-group is one or more adjacent 1's not separated by a 0, and thus may be a single '1' element. Consider the following example:

	1	2	3	4	5	6	7	8	9	10	
1	0	0	1	0	1	0	0	1	0	1	4
2	1	0	0	1	0	0	1	0	1	0	4
3	0	0	1	0	1	0	1	0	1	1	4
4	0	1	0	0	0	1	0	1	0	0	3
5	0	0	1	0	1	0	0	0	0	1	3
6	0	1	0	0	0	1	0	0	0	0	2
7	0	1	0	0	0	1	0	1	0	1	4
8	1	0	1	1	0	0	1	0	1	0	4
9	0	0	0	0	1	0	0	1	0	1	3
10	0	0	1	0	1	0	0	1	0	1	4
	2	2	4	2	4	2	2	4	2	5	64

Figure 5.1 Unsorted FCM

In this unsorted matrix the number of 1-groups is given at the end of each row and column. The total score for the matrix is the sum of all row and column scores. This score is computed again after the matrix has been sorted to measure the effectiveness of the sort:

	4	1	7	9	3	5	10	8	2	6	
2	1	1	1	1	0	0	0	0	0	0	1
8	1	1	1	1	1	0	0	0	0	0	1
3	0	0	1	1	1	1	1	0	0	0	1
5	0	0	0	0	1	1	1	0	0	0	1
1	0	0	0	0	1	1	1	1	0	0	1
10	0	0	0	0	1	1	1	1	0	0	1
9	0	0	0	0	0	1	1	1	0	0	1
7	0	0	0	0	0	0	1	1	1	1	1
4	0	0	0	0	0	0	0	1	1	1	1
6	0	0	0	0	0	0	0	0	1	1	1
	1	1	1	1	1	1	1	1	1	1	20

Figure 5.2 Sorted FCM

This sorted matrix illustrates the best possible case in which the FCM was sorted resulting in a single 1-group in every row and column. The lower limit on the score for any matrix consisting of m rows and n columns is $m + n$.

Comparative data on seven different sorting methods is given in Tables 5.1, 5.2, and 5.3. Table 5.1 gives the results in terms of the '1-group' measurement

described above. In Tables 5.2 and 5.3 the different methods are compared after forming bays from the sorted matrices (Step 3 in Figure 4.3). Table 5.2 gives the total number of bays formed, while Table 5.3 gives the resulting number of infeasible cards. It was not possible to compare the methods using the resulting number of bay changeovers since many of the sorting methods did not generate feasible bay designs. It is apparent, though, that the sorting methods which generated the smallest number of '1-groups' also generated a small number of infeasible cards. The sorting methods are described as follows:

KBC - King's binary clustering algorithm

D1E - TSP & Hamiltonian path sorting using Metric #1 and H.P. method #1

D1V - TSP & Hamiltonian path sorting using Metric #1 and H.P. method #2

D2E - TSP & Hamiltonian path sorting using Metric #2 and H.P. method #1

D2V - TSP & Hamiltonian path sorting using Metric #2 and H.P. method #2

D3E - TSP & Hamiltonian path sorting using Metric #3 and H.P. method #1

D3V - TSP & Hamiltonian path sorting using Metric #3 and H.P. method #2

of 1-Groups

Problem	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg.
Unsorted	308	263	255	256	237	264	242	264	293	260	264.2
KBC	213	198	156	158	152	139	202	139	170	147	167.4
D1E	158	175	135	153	117	122	160	137	148	148	145.3
D1V	139	170	128	134	126	125	148	122	147	137	137.6
D2E	141	167	131	126	112	112	147	113	140	130	131.9
D2V	142	173	131	126	113	114	151	118	143	140	135.1
D3E	146	173	127	146	131	114	161	120	139	140	139.7
D3V	137	162	117	124	111	105	141	111	135	137	128.0

Table 5.2

Total # of Bays Assigned

Problem	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg.
Unsorted	132	153	137	150	134	122	132	153	140	148	140.1
KBC	100	114	96	104	97	83	104	106	99	107	101.0
D1E	104	111	91	105	83	82	92	102	101	99	97.0
D1V	101	116	90	105	94	94	100	97	100	108	100.5
D2E	94	111	82	99	85	76	88	94	93	103	92.5
D2V	96	107	87	87	84	76	95	94	94	105	92.5
D3E	98	107	87	96	94	83	91	88	94	106	94.4
D3V	96	110	87	98	85	76	88	96	94	106	93.6

Table 5.3

# of Infeasible Cards

Problem	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg.
Unsorted	14	14	12	16	12	9	13	16	12	15	13.3
KBC	5	6	2	6	4	2	7	2	4	4	4.2
D1E	2	2	2	3	0	0	2	1	6	2	2.0
D1V	0	7	4	3	3	0	5	5	5	3	3.5
D2E	2	5	2	0	0	0	1	2	4	3	1.9
D2V	0	3	2	0	0	1	3	3	3	2	1.7
D3E	1	2	0	2	2	0	3	3	3	0	1.6
D3V	1	6	0	2	0	0	0	4	4	2	1.9

Table 5.4

Of the seven different sorting methods tested three were clearly inferior to the other methods; King's binary clustering, and both methods using Metric #1. In each table these methods had the worst averages. As discussed in Section 4.1, the binary clustering technique is sensitive to the starting ordering of the rows and columns. Even applying binary clustering to a matrix presorted by the best of the other methods, the sorting results were still inferior.

Of the three measures used to compare the sorting methods, the first two were considered to be preferable to the third. The scores generated from both the '1-groups' measure and the 'Total # of Bays Assigned' measure exhibited a wide range of values. For the '# of Infeasible Cards' measure there is a lower limit on the value of a score, i.e. 0. This lower limit was achieved in many of the

sorts. The '# of Infeasible Cards' measure is also dependent on the bay limit for a particular placement machine. Therefore, the first two measures were considered more credible. The top three sorting methods in both Tables 5.1 & 5.2 were the D2E, D2V, and D3V methods. These three sorting techniques were used for testing in the combined heuristic procedure.

5.1.2 Bay Formation

As described in Chapter 4, feeders must be assigned to bays after the FCM has been sorted. This was accomplished by sequentially assigning feeders to bays from each of the groups formed in the dendogram procedure.

The first set of decisions in the bay formation procedure involved determining how to break the feeders into the groups from which the bays were formed. As the similarity level is increased in the dendogram procedure, more breaks will be forced. As the number of forced breaks increases, the quality of the bay designs generally deteriorated. In the test problems that were solved the best bay design found involved 10 or fewer breaks in virtually every case. Therefore, only the first ten breaks were considered when forming feeder groups.

Subgroups of the set of forced breaks were also considered. Because of computational considerations the number of subgroups considered was limited. Subgroups of size (P-4), (P-3), (P-2), and (P-1) were considered.

The last set of decisions involved in the bay formation step involved the selection of the "best" bay design. The most important criterion for selecting a bay design is that the design be feasible, i.e. - no circuit card is assigned to more the maximum number of bays allowed on the machine. Assuming that two feasible bay designs are being compared, there are two possible selection criteria. The first is to select the bay design having the smallest total number of bay assignments. The second possibility is to select the bay design having the smallest number of bays. The second measure was more effective in selecting good bay designs. Comparative data on these two selection criteria is given in Section 5.4.

5.2 Improvement Step

The improvement step used to generate a feasible (or improved) bay design was the most difficult step in the overall heuristic. Although the objective was clear - improve the bay design to reduce the number of changeovers, there was not a obvious method to accomplish that objective. There were two decisions involved in this step. The first was selecting the feeder to be duplicated while the second was selecting the set of cards to be assigned to the duplicated feeder.

The feeder/card groups were selected by examining the structure of the sorted FCM. Intuitively, any feeder/card element ($x_{ij} = 1$) that was separated from any clusters should be considered a candidate for duplication. A single

feeder may be assigned to several cards for which this is true. That feeder and set of cards are considered a candidate feeder/card group. An exact definition of when an element is "in a cluster" is difficult to quantify. In any column of the FCM, a candidate feeder/card would be any '1' element in the column without any vertically adjacent '1' elements. Since bays typically hold 4 feeders, the feeders one and two positions away in the column were also considered. The overall heuristic method seemed relatively insensitive to this selection process.

Given several different feeder/card group candidates some measure must be used to select one of the feeders for duplication. Several different measures were considered and tested. In each case the augmented FCM was resorted and evaluated by calculating the number of '1-groups' in the sorted matrix. The following measures were considered:

1. A similarity coefficient between the candidate feeder/card group and each row in the FCM can be calculated. Those similarities are summed for each candidate feeder/card group.
2. Same measure as above except that a modified similarity coefficient defined as follows is used:

$$D_{ij} = 1-(C_{ij} / \max(T_i, T_j))$$

3. The highest single similarity value between the candidate feeder/card group and another row in the FCM.

Measure #1 performed considerably better than the other two measures.

5.3 Sequencing

The production sequencing portion of the problem was relatively simple compared with the bay design and improvement steps. As the combined heuristic advanced through a series of iterations, the sequencing problem was solved every time a feasible bay design was found.

The TSP solution techniques used in FCM sorting could not be used directly since a valid distance matrix could not be formed. KTNS distances are sequence dependent and are not symmetric. Therefore, finding a solution must involve some form of enumeration of the possible solutions. As an alternative to complete enumeration, a 2-Opt interchange technique was applied.

Two different methods were used to generate an initial sequence. The first was to use the existing card ordering from the sorted FCM. The second sequence was obtained by solving a TSP using a distance matrix formed by calculating similarity coefficients over the BCM. Dropping the largest similarity in the heuristic tour gives the starting sequence.

The number of changeovers required using a KTNS policy given any sequence is easily computed. Note that because of the lack of symmetry in KTNS distances, the number of changeovers must be computed moving both forward and

backward through the sequences. Using the best of the two starting sequences, a 2-Opt interchange procedure was used to find the best production sequence.

5.4 Combined Heuristic

Several different versions of the combined heuristic procedure were tested on ten problems. The problems are denoted T1 - T10. The three best FCM sorting techniques identified in Section 5.1 were combined with the two different bay design criteria described in Section 5.1, making a total of six different versions of the heuristic procedure.

The heuristic procedures were coded using Borland Turbo Pascal version 3.0. The test problems were run on one of three IBM and IBM-compatible personal computers. Two were 80386 based PC's while the third was 80286 based. In most cases the test problems were run through 15 iterations of the heuristic, with one iteration consisting of Step 2, Step 3, Step 3a, and Step 4 in Figure 4.3. A complete run of one version of the heuristic on one test problem took between 1-1/2 and 2-1/2 hours, averaging 2 hours. The number of test cases was restricted to 10 due to the computation time involved.

All six versions were applied to the first 5 data sets, T1 - T5. The best overall version was determined and applied to the remaining problems. Results

for problems T1 - T5 are given in Table 5.5. The different heuristic methods used are described as follows:

M1C1 - FCM sorting method D2E and bay selection criteria #1

M2C1 - FCM sorting method D2V and bay selection criteria #1

M3C1 - FCM sorting method D3V and bay selection criteria #1

M1C2 - FCM sorting method D2E and bay selection criteria #2

M2C2 - FCM sorting method D2V and bay selection criteria #2

M3C2 - FCM sorting method D3V and bay selection criteria #2

# of Bay Changeovers - Problems T1 - T5

Method	T1	T2	T3	T4	T5	Avg.
M1C1	16	15	10	13	8	12.4
M2C1	14	15	10	14	8	12.2
M3C1	14	17	9	12	8	12.0
M1C2	14	15	9	12	7	11.4
M2C2	11	15	10	12	7	11.0
M3C2	11	17	8	12	6	10.8

Table 5.5

The best method was M3C2 on average. Note that this method did not find the best solution in all cases. The solutions generated by this method were better than the other methods using either bay design selection criteria. Also note

that in every case bay design selection criterion #2 was superior to criterion #1. Based on this comparison method M3C2 was used to solve the remaining five test problems. The results for those test problems are given in Table 5.6.

Test Problems Results - Method M3C2

Problem #	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Aver
# of Bay Changeovers	11	17	8	12	6	10	6	-	18	14	11.3
Best Solution Iter. #	4	5	1	1	1	4	1	-	6	6	3.2

Table 5.6

The heuristic was able to find a feasible solution in all but one case, T8. The iteration number on which the best solution was found is also included. Note that the best solution was very often found early in the procedure when few feeders had been duplicated. The average of these numbers was 3.2; i.e 3.2 feeders had been duplicated from the original FCM.

As was mentioned in Section 4.4, the heuristic procedure can be stopped after some number of iterations and restarted using the best KTNS production sequence found. For each of the test problems the columns of the FCM were sorted to this order and the heuristic procedure was restarted with no duplicated feeders. This procedure was applied to all ten problems using method M3C2. Those results are given in Table 5.7.

Resequenced Test Results - Method M3C2

Problem #	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Aver
# of Bay Changeovers	11	17	8	12	5	11	6	-	18	14	11.3
Best Solution Iter. #	5	5	1	1	1	2	1	-	6	7	3.2

Table 5.7

The average performance of the resequenced problems was the same as the original test runs. In some cases the resequenced problems did find a better solution, but the improvement does not seem to justify the additional computation time.

CHAPTER 6

CONCLUSIONS

6.1 Discussion of Results

It is apparent from the sections on mathematical formulations for the bay design problem and the sequencing problem that the SMT setup problem is extremely difficult to solve. A suitable method was not found to solve the bay design problem to optimality for problems of a realistic size. The sequencing portion of the problem is also very difficult, being a well-known NP-hard problem.

It was demonstrated in the test problems that the key to finding a good solution to the SMT setup problem is finding a good bay design. In general, the bay designs which employ a small number of bays provided the best solutions even when using a nonoptimal production sequence. These bay designs were often found early in the procedure when only a small number of feeders in the FCM had been duplicated. The best solution was usually one of the first feasible bay designs found. In addition, resequencing the cards in the FCM using the best production sequence did not generally improve the solution.

6.2 Future Directions

The TSP methods used to sort the FCM's were considered to be very successful. Using the binary clustering technique, it was difficult to find a feasible bay design for many of the tested FCM's even after a number of improvement steps. By using the TSP sorting procedures it was possible to generate bay designs that were feasible or near feasible before duplicating any feeders. It seems unlikely that there is much room for improvement in the methods used to sort the FCM's.

The dendogram procedure used in bay formation was also considered to be successful. The method essentially enumerates many different possible bay designs, keeping the best. The most difficult step in the overall procedure was the improvement step. Although the goal in this step is clear - duplicate feeders in order to improve the bay design and reduce the number of changeovers, the method to accomplish this goal is not as clear. It is in this step that there seems to be the most room for improved methods.

APPENDIX A

CONSTRAINT BD2 NONCONVEXITY PROOF

In order to show that an inequality constraint forms a convex set, it must be shown that a convex combination of any two points satisfying the inequality will also satisfy the inequality. Assume

$$f(P^1) \geq \alpha \quad \text{and} \quad f(P^2) \geq \alpha$$

If it can be shown that $f(\lambda P^1 + (1-\lambda)P^2) \geq \alpha$ for $\lambda \in (0,1)$ then the problem is a convex programming problem. For constraints (BD2)

$$f(X,Y) = \sum_{j=1}^{P} (X_{ij}\, Y_{jk}) \geq 1 \qquad \forall\, k = 1, .., N \text{ and } (i \in T_k)$$

Assume that for two solutions:

$$\sum_{j=1}^{P} (X^1_{ij}\, Y^1_{jk}) \geq 1 \quad \text{and}$$

$$\sum_{j=1}^{P} (X^2_{ij}\, Y^2_{jk}) \geq 1$$

A convex combination of those two solutions is formed as follows

$$\sum_{j=1}^{P} (\lambda(X^1_{ij}, Y^1_{jk}) + (1-\lambda)(X^2_{ij}, Y^2_{jk}))$$

$$= \sum_{j=1}^{P} (\lambda X^1_{ij} + (1-\lambda)X^2_{ij})(\lambda Y^1_{jk} + (1-\lambda)Y^2_{jk})$$

$$= \sum_{j=1}^{P} \lambda^2 X^1_{ij}\, Y^1_{jk} + \lambda(1-\lambda)X^1_{ij}\, Y^2_{jk} + \lambda(1-\lambda)X^2_{ij}\, Y^1_{jk} + (1-\lambda)^2 X^2_{ij}\, Y^2_{jk}$$

$$= \lambda^2 \sum_{j=1}^{P} X^1_{ij}\, Y^1_{jk} + (1-\lambda)^2 \sum_{j=1}^{P} X^2_{ij}\, Y^2_{jk} + \lambda(1-\lambda) \sum_{j=1}^{P} (X^1_{ij}\, Y^2_{jk} + X^2_{ij}\, Y^1_{jk})$$

Assume that there is only one variable in each set of solutions which are nonzero:

$$X^1_{ij} = 1 \text{ and } Y^1_{ij} = 1 \text{ at } j = j^1$$

$$X^2_{ij} = 1 \text{ and } Y^2_{ij} = 1 \text{ at } j = j^2 \neq j^1$$

The following statements can then be made:

$$\sum_{j=1}^{P} (X^1_{ij}\, Y^2_{jk} + X^2_{ij}\, Y^1_{jk}) = 0$$

$$\sum_{j=1}^{P} (X^1_{ij}\, Y^1_{jk}) = 1$$

$$\sum_{j=1}^{P} (X^2_{ij}\, Y^2_{jk}) = 1$$

The convex combination of the two solutions may now be written as follows:

$$f(\lambda(X^1,Y^1) + (1-\lambda)(X^2,Y^2) = \lambda^2(1) + (1-\lambda)^2(1) + \lambda(1-\lambda)(0)$$
$$= 2\lambda^2 - 2\lambda + 1$$

This function is not strictly greater than 1 for all values of λ in (0,1); therefore constraints (BD2) do not form a convex set.

APPENDIX B

MODIFIED SPANNING TREE HEURISTIC

The TSP is generally considered to be one of the most difficult combinatorial problems. While even large problems can be solved to optimality, the amount of work required makes heuristic methods an attractive alternative. Many heuristic methods are described in the literature with a great variation in computational complexity. Some of the most simple methods include the nearest-neighbor method, convex hull methods, and edge interchange schemes. These methods do not provide a deterministic guarantee of optimality for the solutions generated. Several methods have been suggested which use the minimum weight spanning tree as the basis for finding a tour. These methods have the advantage of generating worst-case bounds for the heuristic tours found.

In order for the worst-case bounds to be applicable, the TSP problems must involve distances that are symmetric and follow the triangular inequality (Lawler, ([10], 1985)). The spanning tree methods can still be applied to problems which do not satisfy these assumptions, but the bounds may not be valid.

The spanning tree heuristics function as follows. A minimum weight spanning tree (MWST) is determined for the representative graph. Using one of

several different methods, the spanning tree is augmented to create an Eulerian graph. An Eulerian graph is a graph in which every vertex has even degree. A tour can be constructed from any Eulerian graph using the shortcut method. See Lawler ([10], 1985) for a complete description of these methods.

There are two methods described in Lawler ([10], 1985) for augmenting the spanning tree to create an Eulerian graph. The first is to simply duplicate every edge in the spanning tree. This method will be referred to as the doubling procedure. Any vertex initially having odd degree will have even degree in the doubled graph. The second method is to use Christofides' algorithm. In this method the vertices in the spanning tree with odd degree are identified. It can easily be shown that there will be a even number of these vertices. The minimum weight matching is calculated for those vertices. A matching is any set of edges such that all vertices are connected to at least one edge. The minimum weight matching on an even set of vertices will involve a single edge being attached to every vertex. Combining this matching with the spanning tree will create an Eulerian graph. Christofides' algorithm is the best known heuristic method in terms of the worst-case bounds generated.

The TSP methods used in this thesis use another method for generating the Eulerian graph. As was mentioned in the previous paragraph, it can be shown that there will always be an even number of odd-degreed vertices in the MWST.

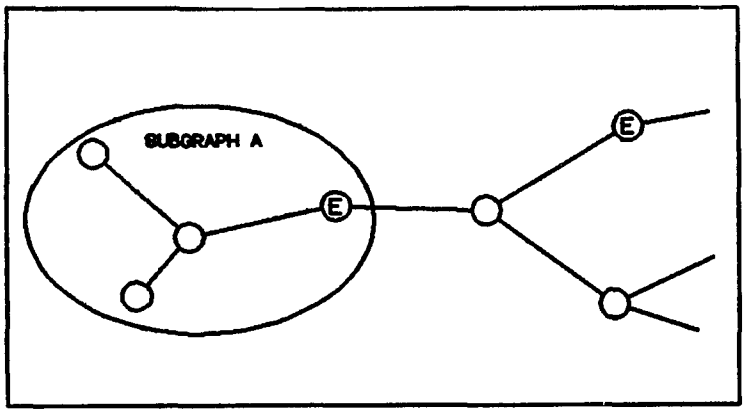
A second MWST is determined on the subgraph consisting of the set of odd-degreed vertices in the original spanning tree. The second spanning tree can be modified such that the resulting degree of all vertices in the subgraph is odd. Adding this modified spanning tree to the original spanning tree will result in an Eulerian graph. This Eulerian graph is then used to create a heuristic tour. This method is referred to as the modified spanning tree heuristic.

The modified spanning tree heuristic is based on finding a minimum weight spanning tree of the odd degreed vertices in the original spanning tree. This second spanning tree, denoted the odd spanning tree, is modified by dropping selective edges in the tree. The modification can always be done such that the degree of every vertex in the subgraph will have odd degree. This fact is proved as follows:

1. There will always be an even number of odd degreed vertices in the original spanning tree. This follows from the fact that each edge adds "2" to the cumulative degree of the tree. Therefore, the cumulative degree is an even number. In order for that total to be even there must be an even number of vertices with odd degree.

2. By the same argument there will be an even number of odd degreed vertices in the odd spanning tree. Since the number of vertices in the odd spanning tree is even, there will be an even number of even degreed vertices also.

3. A subgraph of the odd spanning tree, A, can be formed with a "pendant" even degreed vertex such that the subgraph contains only odd degreed vertices and the single even degreed vertex. The even degreed vertex is pendant in the sense that it is at the end of the path connecting it to other even degreed vertices.



4. The grouping above creates two subgraphs connected by a single edge. Both subgraphs have an even number of vertices. This is proved from the fact that the cumulative degree of subgraph A is odd. The single connecting edge contributes one to the cumulative degree and any other edges in A contribute 2 to the total. Since the total degree is odd and there is only one even vertex, there must be an odd number of odd degreed vertices in the subgraph A. Thus the total number of vertices in both subgraphs is even.

5. Deleting the edge connecting the subgraphs formed above separates the two subgraphs. The first subgraph is of odd degree for all vertices because the degree of the even vertex has been reduced by one and all other vertices were even to start with.

6. The subgraph containing the remaining vertices can be reduced by the steps above until two even vertices are adjacent to each other. Deleting that edge will make the degree of both vertices odd.

The minimum weight spanning tree is a lower bound on the distance of the optimal TSP tour. The proof of this follows from the fact that dropping any edge in the optimal tour will create a tree. The MWST is the smallest of all such trees. The optimal tour must then be strictly greater than the MWST. The worst-case bounds are derived from this fact. The doubling procedure can be shown to be

a 1-approximate method while Christofides' algorithm can be shown to be .5-approximate method. This means that in the worst case, the heuristic tours generated are guaranteed to be within 100% and 50% of the optimal tour, respectively. Note also that the MWST is a lower bound on the length of the Hamiltonian path.

While the doubling procedure is very simple, the minimum weight matching problem is a very complex procedure with complexity on the order $O(n^4)$ (n is the number of vertices in the graph). The modified spanning tree algorithm gives a performance guarantee that is somewhere between 50% and 100%. In the worst case, the Eulerian graph generated could be as large as that formed by the doubling procedure. In the best case, the Eulerian graph could be as small as that generated by Christofides' algorithm. The advantage to the modified spanning tree algorithm is that the complexity of the procedure is considerably less than Christofides' algorithm - on the order $O(n^2)$. When combined with an improvement procedure such as 2-Opt edge interchanging, the modified spanning tree heuristic generates very good tours. When used in the heuristic methods for the SMT setup problem, the procedure typically generated tours within 5%-20% of the length of MWST.

APPENDIX C

HAMILTONIAN PATH HEURISTIC

A Hamiltonian path sequence was used in both the FCM sorting and production sequencing portions of the SMT setup problem. The Hamiltonian path problems were solved by modifying heuristic tours generated by the modified spanning tree heuristic described in Appendix B. A path can be created from a valid tour by dropping any single edge. Two different variations of the procedure were used.

The first Hamiltonian path method (H.P. #1) is described as follows:

Step 1 An initial incumbent solution was determined by dropping the largest single edge in the heuristic TSP tour.

Step 2 Starting with the first edge, each edge in the heuristic tour is dropped to form a Hamiltonian path.

Step 3 For each of the N paths formed in Step 2, a single iteration of the 2-Opt interchange procedure is run.

Step 4 If any interchange was made in Step 3, return to Step 2 and repeat.

The second Hamiltonian path method (H.P. #2) is described as follows:

Step 1 An initial incumbent solution was determined by dropping the largest single edge in the heuristic TSP tour.

Step 2 A single iteration of the 2-Opt interchange procedure is performed on the heuristic TSP tour. At each interchange the largest edge in

the resulting tour is dropped and compared to the incumbent best solution.

Step 3 If any interchange was made in Step 2, return to Step 2 and repeat.

The two procedures both worked well in generating good Hamiltonian paths. The second method was preferred since it considered only n^2 interchanges, compared to n^3 in the first method. In addition, the second method inherently considers all solutions considered by the first method. In spite of this, the version of the SMT setup heuristic using H.P. #1 performed better than the version using H.P. #2 in some cases. In general, though, H.P. #2 was the better method.

REFERENCES

[1] Ahmadi, J., Stephen Grotzinger, and Dennis Johnson (1988), "Component Allocation and Partitioning for a Dual Delivery Placement Machine", *Operations Research*, Volume 36, No. 2, pp. 176-191.

[2] Askin, R.G. and A.J. Vakharia (1989), "Group Technology - Planning and Operation", Working Paper, University of Arizona, S.I.E Department.

[3] Bard, J.F. (1988), "A Heuristic for Minimizing the Number of Tool Switches on a Flexible Machine", *IIE Transactions*, Volume 20, No. 4, pp. 382-391.

[4] Garey, Michael R. and David S. Johnson (1979), *Computers and Intractibility*, W.H. Freeman, San Francisco.

[5] Kernighan, B.W., and S. Lin (1973), "An Effective Heuristic Algorithm for the Traveling Salesman Problem", *Operations Research*, Volume 21, No. 3, pp. 498-516.

[6] King, J.R. (1980), "Machine Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm, *International Journal of Production Research*, Volume 18, No. 2, pp. 213-232.

[7] Kumar, K. Ravi, Andrew Kusiak and Anthony Vannelli (1986), "Grouping of Parts and Components in Flexible Manufacturing Systems", *European Journal of Operational Research*, Volume 24, pp. 387-397.

[8] Mangasarian, O.L. (1988), "A Simple Characterization of Solution Sets of Convex Programs", *Operations Research Letters*, Volume 1, No. 1, pp. 21-26.

[9] McAuley, J. (1972), " Machine Grouping for Efficient Production", *Production Engineer*, Volume 51, pp. 51, 53.

[10] Lawler, E.L. et at Editors (1985), *The Traveling Salesman Problem*, Holt, Rinehart, and Winston, New York.

[11] Luenberger, David G. (1984), *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading Mass.

[12] Papadimitriou, C.H. and K. Steiglitz (1982), *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs, N.J.

[13] Rajagopolan, R. and J. Batra (1975), "Design of Cellular Production Systems - A Graph-Theoretic Approach", *International Journal of Production Research*, Volume 13, No. 6, pp. 567-579.

[14] Stecke, K.E. (1986), "A Hierarchical Approach to Solving Machine Grouping and Loading Problems of Flexible Manufacturing Systems", *European Journal of Operational Research*, Volume 24, pp. 369-378.

[15] Tang, C.S. and E.V. Denardo (1988), "Models Arising From a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches", *Operations Research*, Volume 36, No. 5, pp. 767-777.

[16] Tang, C.S. and E.V. Denardo (1988), "Models Arising From a Flexible Manufacturing Machine, Part II: Minimization of the Number of Switching Instants", *Operations Research*, Volume 36, No. 5, pp. 778-784.