OPTIMIZATION OF THE ASSIGNMENT OF PRINTED CIRCUIT CARDS TO ASSEMBLY LINES IN ELECTRONICS ASSEMBLY

by

Sudeer Bhoja

Thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
In
Industrial and Systems Engineering

Dr. Kimberly Ellis, Chair

Dr. William Sullivan

Dr. John Kobza

September 28, 1998
Blacksburg, Virginia

Keywords: Process Planning, Line Assignment, Printed Circuit Card Assembly,
Line Balancing, Card Grouping

OPTIMIZATION OF THE ASSIGNMENT OF PRINTED CIRCUIT CARDS TO ASSEMBLY LINES IN ELECTRONICS ASSEMBLY

Sudeer Bhoja

(ABSTRACT)

The focus of this research is the line assignment problem in printed circuit card assembly systems. The line assignment problem involves the allocation of circuit card types to an appropriate assembly line among a set of assembly lines with the objective of reducing the total assembly time. These circuit cards are to be assembled in a manufacturing facility, capable of simultaneously producing a wide variety of printed circuit cards in different production volumes. A set of component types is required for each printed circuit card. The objective is to assign the circuit cards to the assembly line such that the total assembly time, which includes the setup time as well as the processing time required for all card types in a set, is minimized.

The focus of this research is to develop an algorithmic strategy for addressing this problem in electronics assembly. This problem involves considering several interrelated decision problems such as assigning printed circuit cards to assembly lines, grouping circuit cards into families to reduce the number of setups, and assigning component types to machines to balance workload. The line assignment models are formulated as large scale mixed integer programming problems and are solved using a branch-and-bound algorithm, supplemented by techniques for improving the solution time. The models and solution approaches are demonstrated using industry representative data sets and can serve as useful decision support tools for process planning engineers.

ACKNOWLEDGMENTS

Dr. Kimberly Ellis provided the overall guidance for this research effort. She gave me enough freedom and latitude for doing this research and always had a patient ear and time to listen. Throughout the course of this research she has shown enormous patience and understanding and I look forward to joint research endeavors with her in the future. I would also like to thank all members of my committee, Dr. John Kobza, and Dr. William Sullivan, for their time, comments, and support during the course of this research.

Lucent Technologies, Columbus, Ohio provided the financial support, with Dr. Ellis as the Principal Investigator. I am pretty sure that this effort would not have been possible without their generous support. I would like to thank Brian Lassahn, and Jim Steinbugl for their support during the course of this thesis. Jim has been extremely helpful in sharing the data and also gave me helpful Access tips.

Beyond those directly involved, I would also like to express special thanks to Dr. W. J. Fabrycky. I have worked for him on a couple of projects and I always found him to be an amazing human being. He has helped me in more ways than I can enumerate. Also, thanks to all my professors at REC and Virginia Tech who have made this possible.

I would also like to thank my family members and friends, Sujith, Avani, Ashani, Bharadwaj, Mohan, Rangan, Bharat, Arief, Hongjiew, John, and Jay. I would also thank my parents for their love and support. Finally and most importantly, my brother, Sudeep deserves tribute for his encouragement and support throughout my stay at Virginia Tech.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATION

i	index for component types
j	index for card types
k	index for machines
g	index for groups
l	index for lines
$\mathbf{A}$	set of component types
$\mathbf{B}$	set of card types
$\mathbf{F}$	set of groups
$\mathbf{M}$	set of machines
$\mathbf{L}$	set of assembly lines
a_{ij}	parameter which indicates if component type i is required on card type j
b_{jl}	parameter which indicates if card type j can be assigned to line l
h_{ikl}	parameter which indicates if component type i can be placed by machine k in line l
d_{ij}	quantity of component type i required on card type j
f_{kl}	time to change one feeder on machine k in line l
c_{kl}	constant time obtained by regression analysis on machine k in line l
t_{ikl}	time to place component type i by machine k in line l
q_j	production requirements for card type j
s_{ikl}	number of slots used by component i on machine k in line l
S_{kl}	total number of feeder slots available on machine k in line l
C_l	total available capacity in line l
V_{jgl}	binary variable which indicates if card type j is assigned to group g in line l
X_{ikgl}	binary variable which indicates if component type i is staged on machine k

	for group *g* in line *l*
Y_{ijkgl}	integer variable that indicates the quantity of component type *i* placed on card type *j* by machine *k* for group *g* in line *l*
PT_{jl}	total processing time for card type *j* in line *l*
ST_{jl}	total setup time required for card type *j* in line *l*
$flow_{jl}$	total assembly time for card type *j* in line *l*
X_{ijkl}	binary variable which indicates if component type *i* is staged for card type *j* on machine *k* in line *l*
ST_{gl}	setup time for group *g* in line *l*

CHAPTER I

INTRODUCTION

1.1. OVERVIEW OF ELECTRONICS ASSEMBLY

The electronics industry has experienced unprecedented growth in the last few decades and continues to rank as an important industry as the twenty-first century approaches. According to the U.S. Bureau of Labor Statistics (1995), the electronics industry employs 2.6 million people in the U.S., making it the nation's largest employer [32]. According to SEMATECH, a non-profit research and development consortium of U.S. semiconductor manufacturers, the electronics industry has become larger than the automotive, aerospace, and steel industries combined [32].

Printed circuit cards are vital components of practically every modern electronics system, from computers to communication satellites. Efficient use of valuable resources to assemble these printed circuit cards will enable electronics companies to remain competitive in the next millenium. If the electronic manufacturing base thrives, the supporting industries that provide equipment, materials, and labor also benefit.

Managing printed circuit card assembly systems requires expertise in a variety of areas including product design, process planning, production operations, material flow/facilities layout, and production planning and control. A computer network, which interconnects all the manufacturing activities of the plant, would enable the flow of information between the different activities in electronics assembly. The output of a higher level activity is sent to the next lower level activity. Feedback information is sent back to the higher level activities and changes are made to the system if necessary. A high level summary of the flow of information between different activities in electronics assembly is captured in Figure 1-1.

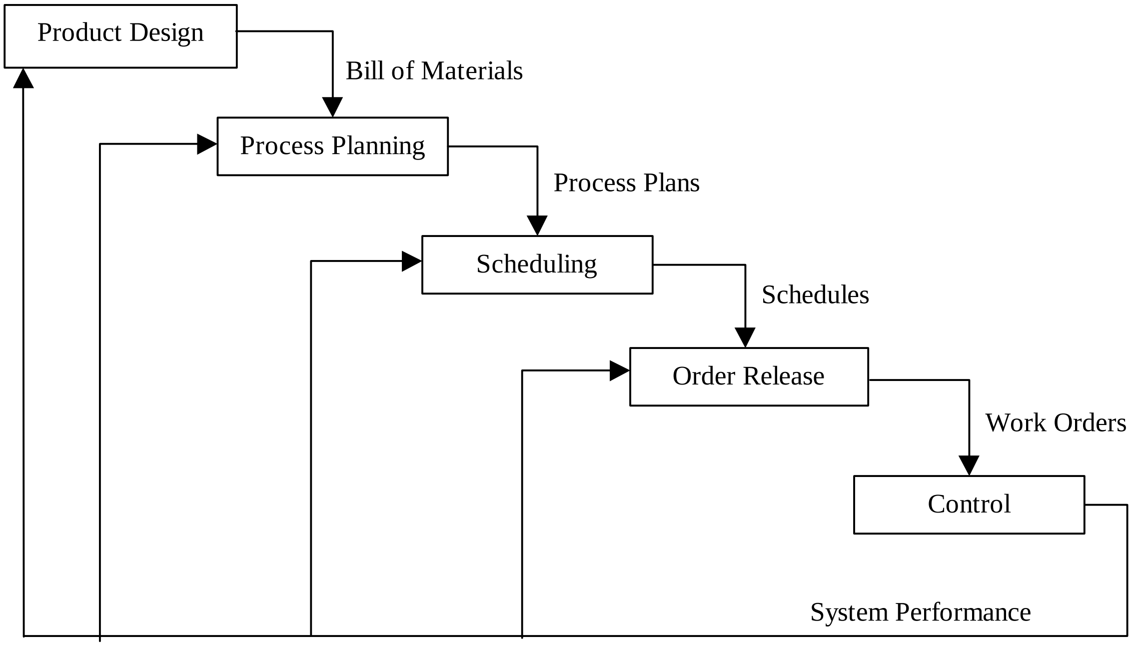

Figure 1-1. Different Activities in a Manufacturing System

1.2. PROCESS PLANNING IN ELECTRONICS ASSEMBLY

Among the different activities in electronics assembly, process planning is particularly important since it determines the efficiency of expensive and highly utilized resources. In general, process planning entails the specification of the operations required for converting raw material into finished products. Process planning requires an in-depth understanding of the functional needs of the product as specified by the design engineers as well as knowledge about the availability and capability of production operations in the manufacturing facility.

Based on the product design information, process planning determines the manufacturing processes and processing sequences needed to produce the final product under a set of operating constraints. In electronic assembly systems, this is a complex function due to the large number of components to be assembled, the diversity of card types, and the variety of assembly machine technologies [14].

Process planning in electronics assembly involves addressing two closely related issues: *process optimization* and *setup management* [26]. The primary focus of process optimization is processing time efficiency while setup management refers to the planning decisions regarding the organization and allocation of resources and products in the assembly system. Table 1-1, adapted from Ellis *et al.* [15], summarizes the various decision problems required for setup management and process optimization decisions.

Table 1-1. Process Planning Decisions [15]

- **Process Optimization**
  - Component Allocation
  - Feeder Assignment
  - Placement Sequencing
- **Setup Management**
  - Line Assignment
  - Strategy Selection
  - Card Grouping
  - Card Sequencing

The process planning decisions in printed circuit card assembly systems are complex and are highly interrelated. The interrelationships among the various process planning decisions are illustrated in Figure 1-2. The higher level setup management decisions are represented by the rings on the outside while the inner rings represent process optimization decisions. The setup strategy selection problem shown at the outermost level involves determining an assembly line operating policy. The setup strategy often determines the nature of the other decision problems in process planning. The next level involves grouping of cards into families, sequencing the cards in an assembly line, assigning circuit cards to assembly lines, and grouping machines into workcells. The component allocation decision problem involves assigning component types to machines with the objective of balancing the line. Based on the assignment of component types to machines, the feeder assignment and placement sequencing decisions at the machine level are determined.

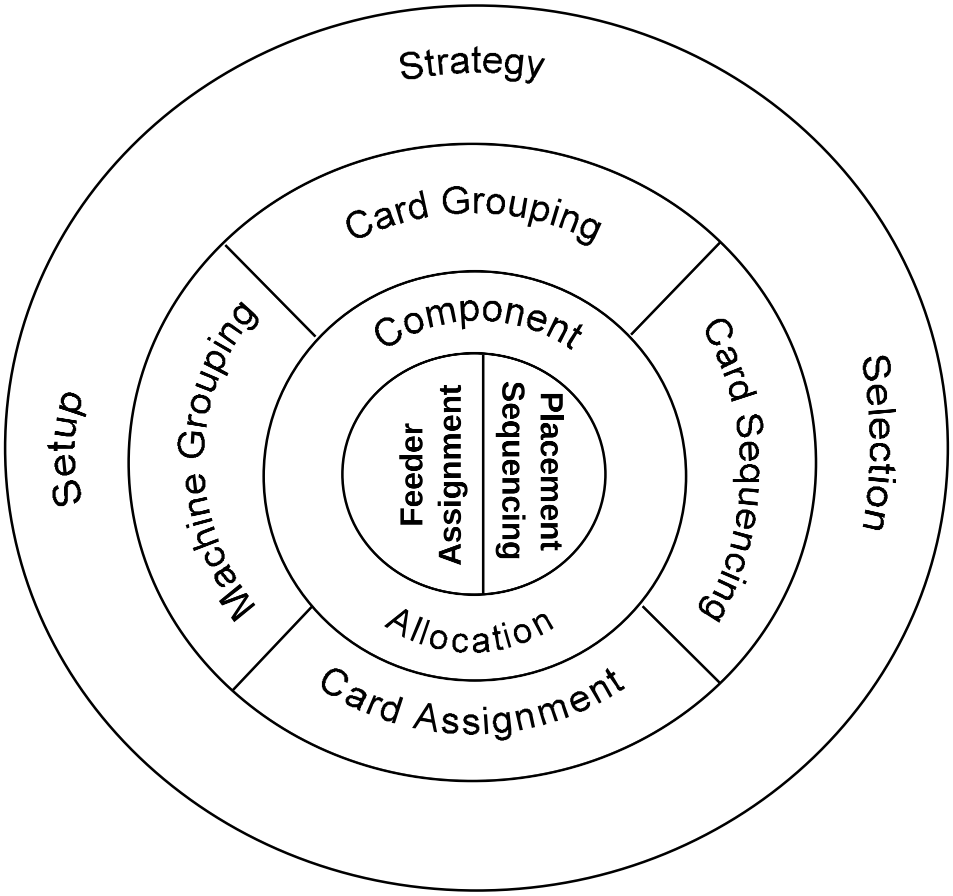

Figure 1-2. Process Planning Decisions in Printed Circuit Card Assembly [14]

1.3. PRINTED CIRCUIT CARD ASSEMBLY SYSTEMS

The purpose of a printed circuit card assembly system is to produce functional and reliable circuit cards in different production quantities during a specified production period. Efficient use of the assembly equipment will result in a strong impact on the overall performance of the assembly process. Figure 1-3 illustrates a functional model for a printed circuit card assembly system. The function of the system is to manufacture printed circuit cards. Materials and information needed to produce the part spectrum serve as inputs to the assembly system. Materials consist of bare circuit cards, electronic components on reels, solder, fluxes, and other supplies necessary to produce printed circuit cards. Information includes printed circuit card design and layout, production plans, and available labor information. Constraints include cost and customer due dates to manufacture these circuit cards. Some of the mechanisms to convert inputs to outputs include manufacturing methods and engineering algorithms in a printed circuit card manufacturing facility. The assembled circuit cards, defective cards, and feedback information on manufacturing resource utilization and product quality serve as outputs from the printed circuit card assembly system.

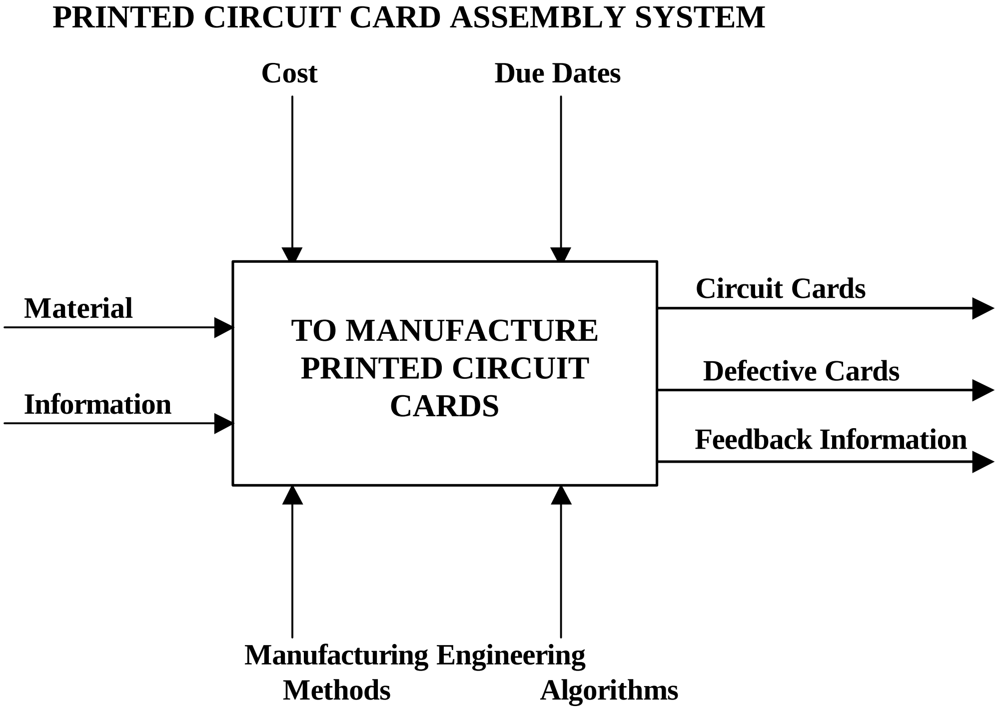

Figure 1-3. Printed Circuit Card Assembly System Illustration

In printed circuit card assembly, a blank printed circuit card is populated with electronic components (resistors, capacitors, ICs) by attaching the components to the card. In the past, the components were placed on the card by inserting the electric leads through pre-drilled holes in the card. A wave soldering process to secure an electric connection followed this process. This technology still exists and is referred to as Through-Hole Technology. Another approach, Surface Mount Technology (SMT) has become very popular in printed circuit card assembly. With this approach, the components are directly attached to the surface of the card. Solder paste is screened onto the card and components are placed on the solder paste by a pick and place operation. This is followed by an inspection operation. The card is then sent to an oven where the solder is reflowed at high temperatures. Finally, the card is subjected to cleaning followed by a testing operation. This approach eliminates the need to drill holes in the card and allows for much smaller components and higher component density. An example of a printed circuit card assembly line is shown in Figure 1-4, adapted from Johnson [24].

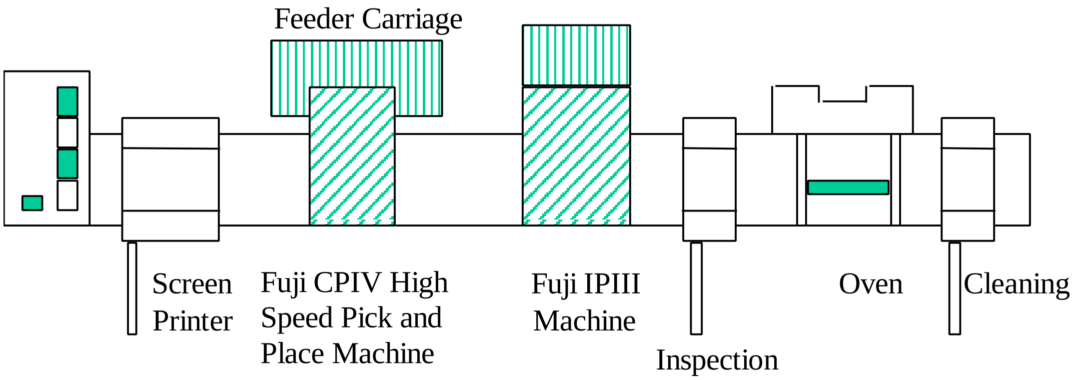

Figure 1-4. Printed Circuit Card Assembly Line [24]

The pick and place machines place the components onto the printed circuit card. A variety of assembly machines with different specifications are used for the pick and place operation, and these machines are often expensive to acquire. Figure 1-5, adapted from McGinnis *et al.* [27], is a schematic of a machine used to place components on a circuit card. For this machine type, the feeder carriage holds the feeders with the reels of component, the transporting mechanism retrieves components from the feeder carriage using a suction mechanism and places them on the circuit card, and the assembly locator moves the circuit card into position for the placement operations. Many of these actions are accomplished concurrently.

Developing a process plan for individual machines is often difficult. Developing a process plan for systems with multiple machines in an assembly line and multiple assembly lines in the facility is even more difficult. The focus of this research is on the higher level process planning problem of assigning circuit cards to assembly lines. Important characteristics of the pick and place machines are incorporated in the solution approach.

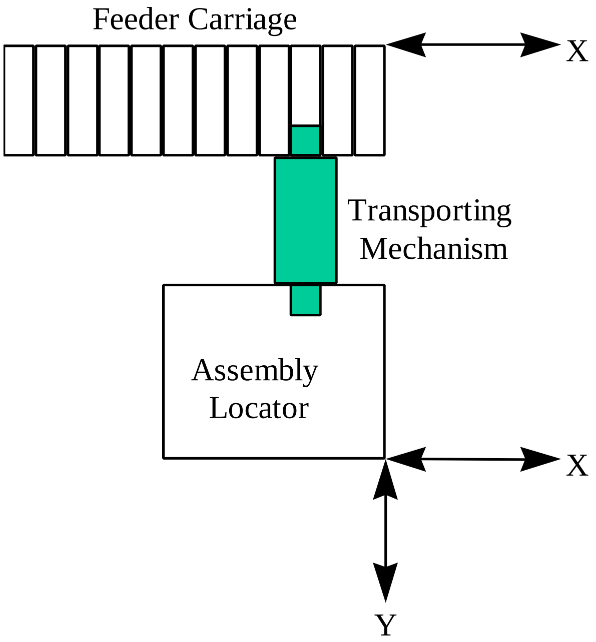

Figure 1-5. Schematic of Assembly Machine [27]

1.4. RESEARCH OBJECTIVES

The goal of this research is to make a contribution to the ongoing evolution of the field of printed circuit card assembly. This goal is achieved through the use of algorithmic strategies to address issues in the area of process planning. Specific objectives of this research include:

1. Definition and formalization of the line assignment problem;
2. Understanding the nature of underlying interrelated decision problems in printed circuit card assembly;
3. Conducting a literature review on the latest research in the area of printed circuit card assembly;
4. Development of optimization models for solving the line assignment problem by incorporating the lower level decision problems in printed circuit card assembly;

5. Delineation of a structured and disciplined approach for making decisions related to the assignment of each circuit card type to an appropriate assembly line;
6. Implementation of the solution approach through case study applications at a large international network wireless firm, and
7. Establishment of a baseline for the future development of this field of study.

1.5. BENEFITS FROM THIS RESEARCH

The proposed model and automated solution approach for addressing the line assignment problem offers significant opportunity for electronic assembly industries to improve productivity and throughput of assembly lines. The proposed solution approach to the problem provides optimal or near-optimal line assignments in a reasonable amount of time. An automated solution approach reduces the effort required by engineers to generate a process plan and allows a faster response to changes in production requirements. Also, increased throughput may reduce the need for additional capital expenditures for expensive assembly equipment [14].

1.6. ORGANIZATION OF THESIS

The thesis is organized into seven chapters and a supporting appendix as shown in Figure 1-6. A formal problem statement is presented in Chapter II. Chapter III provides a review of the research literature related to this area. The mathematical models and solution approaches for the line assignment problem are presented in Chapter IV. The solution techniques used to solve the mathematical models are described in Chapter V. Case study results using industry representative data are presented and analyzed in Chapter VI. Finally, Chapter VII summarizes the research contributions and areas of potential research to address different process planning issues in printed circuit card assembly.

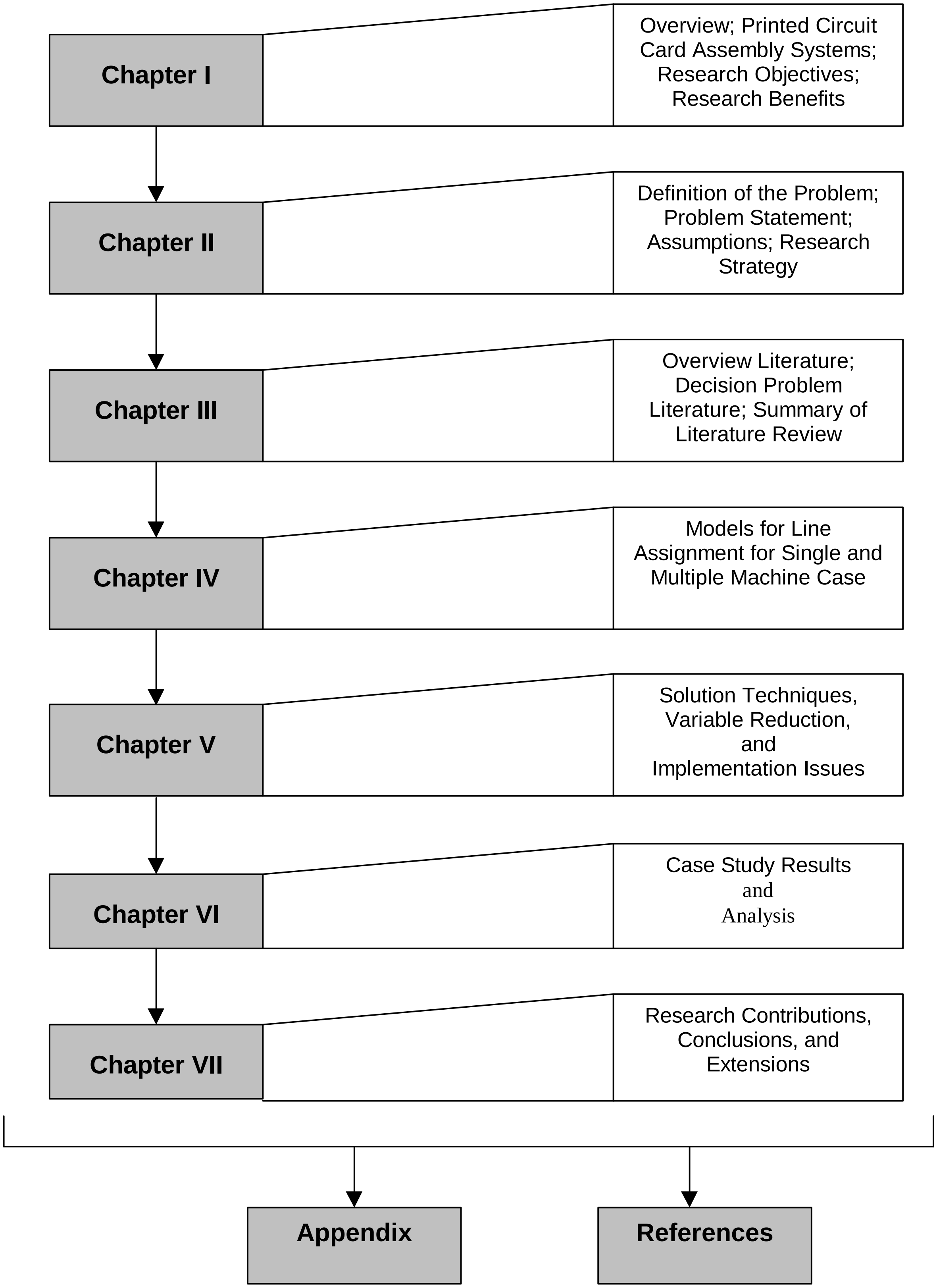

Figure 1-6. Organization of this Thesis

CHAPTER II
PROBLEM DESCRIPTION

The primary emphasis of this research is on addressing the line assignment problem. The purpose of this chapter is to formally define the line assignment problem and illustrate the interrelationships among other process planning decision problems. The relationships or similarities between these problems and other problems within industrial engineering and operations research are highlighted. Finally, the research strategy to address the line assignment problem is discussed.

2.1. PROBLEM OVERVIEW

According to Ellis [14], process planning in printed circuit card assembly systems involves addressing the following decision problems:

- Selecting the appropriate setup strategy;
- Assignment of printed circuit card cards to assembly lines;
- Grouping of printed circuit cards into families;
- Sequencing the production of the circuit packs on the assembly lines;
- Allocating component types to assembly machines;
- Determination of component placement sequence;
- Assignment of component types to specific feeder carriage slots.

In this research, the line assignment problem is addressed along with the related card grouping and component allocation problems. These problems are complex in nature due to the amount of information involved and the interrelationships with the other process planning decisions. Figure 2-1, adapted from Ellis [14], shows the interrelationships among various process planning decisions.

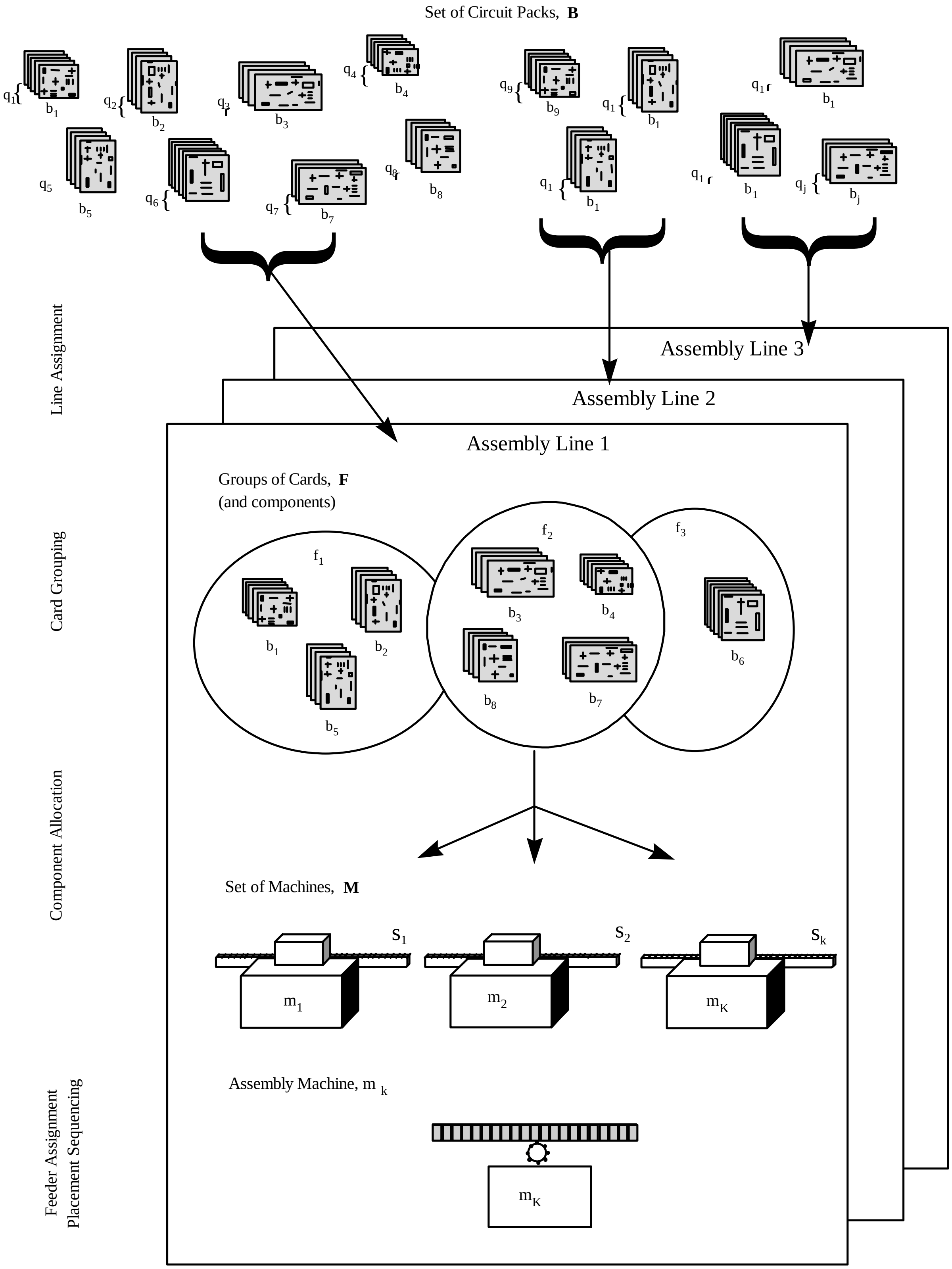

Figure 2-1. Relationships Among Various Process Planning Decisions [14]

2.2. PROBLEM STATEMENT

The line assignment problem involves the assignment of circuit cards to an appropriate assembly line with the objective of reducing the total assembly time. The total assembly time includes the setup time for each of the required setups as well as the processing time required for each card type. Each of the circuit cards has required component types as specified by the design information. In addition, each of the circuit cards has required production quantities as specified by the production planning information. These circuit cards are to be assembled in a production facility, which is composed of multiple assembly lines with different capabilities and production capacities. The problem is to assign each of the circuit packs to an assembly line in order to maximize throughput and meet demand requirements. Figure 2-2, adapted from Ellis [14], summarizes the circuit cards to line assignment problem.

Line Assignment Problem

Given:

A set of circuit card types characterized by
    a set of required component types and placement locations;
    the production quantities;
to be produced with
    a set of automated assembly lines during a specified planning horizon;

Determine:

The assignment of each circuit card type to a specific assembly line such that
    the component staging requirements are met;
    the production quantities are satisfied; and
    the feeder and production capacities of the assembly lines are not exceeded.

Objective:

Minimize the **total assembly time** where the assembly time includes both setup time and processing time.

Figure 2-2. Line Assignment Problem [14]

The line assignment problem consists of a set of component types, **A**, a set of card types, **B**, a set of assembly lines, **L**, and a set of machines, **M**. A subset of component types $\mathbf{A_j} \subseteq \mathbf{A}$ is to be placed on card type *j*. The number of placements of component type *i* on card type *j* is represented as d_{ij}. The components $\mathbf{A_j}$ will be placed by machines $\mathbf{M_l} \subseteq \mathbf{M}$ in line *l* for each card type *j*. Each component type *i* occupies a finite staging capacity s_{ikl} on the feeder carriage of machine *k* in line *l*. The total staging capacity on machine *k* in line *l* is equal to S_{kl}. The time to place component type *i* using machine *k* in line *l* is indicated by t_{ikl}. A production demand q_j is required for each card type *j* during time horizon *t*. The following notation summarizes the known information:

$\mathbf{A}$	a set of component indices	$\mathbf{A} = \{1, 2, .., i, \ldots, I\}$
$\mathbf{B}$	a set of card indices	$\mathbf{B} = \{1, 2, .., j, \ldots, J\}$
$\mathbf{M}$	a set of machine indices	$\mathbf{M} = \{1, 2, .., k, .., K\}$
$\mathbf{L}$	a set of line indices	$\mathbf{L} = \{1, 2, .., l, \ldots, L\}$
$\mathbf{F}$	a set of group indices	$\mathbf{F} = \{1, 2, .., g, \ldots, G\}$
$\mathbf{A_j} \subseteq \mathbf{A}$	indexed set of components required for card *j*	
$\mathbf{M_l} \subseteq \mathbf{M}$	indexed set of machines in line *l*	
$\mathbf{t}$	time horizon for planning	
$\mathbf{d_{ij}}$	quantity of component type *i* used on card type *j*	
$\mathbf{q_j}$	production requirements for card type *j*	
$\mathbf{s_{ikl}}$	number of feeder slots required for component type *i* on machine *k* in line *l*	
$\mathbf{S_{kl}}$	total number of feeder slots available on machine *k* in line *l*	
$\mathbf{t_{ikl}}$	time to insert/place component type *i* on machine *k* in line *l*	

The assignment of circuit cards to assembly lines is to be accomplished by considering the lower level decision problems. The circuit card family designated by *g* in assembly line *l* should contain at least one card type *j* from the set of cards **B,** such that $\mathbf{B_{gl}} \subseteq \mathbf{B}$. Each card type *j* is assigned to exactly one line *l* and appears in exactly one family. The corresponding components $\mathbf{A_j}$ need to be assigned to at least one machine *k* in line *l* during the processing of this group *g*. $\mathbf{A_{jkgl}} \subseteq \mathbf{A}$ represents the component types required by card type *j* that are placed by machine *k* for group *g* in line *l*. The binary variable X_{ikgl} would indicate if component type *i* is assigned to machine *k* for group *g* in line *l*. The feeder slot capacity, S_{kl}, of machine *k* in line *l* must not be exceeded by the component types *i* assigned to the machine k during any group g in line l, such that $\sum_i s_{ikl} X_{ikgl} \le S_{kl}$. The binary variable V_{jgl} indicates if card type *j* is assigned to group *g* in line *l*. The integer variable Y_{ijkgl} indicates the total number of component types *i* placed on card type *j* by machine *k* for group *g* in line *l*. All the required components for card type *j* must be placed by some machine *k* in $\mathbf{M_l}$, such that $\sum_k Y_{ijkgl} = d_{ij}$. The following decision variables summarize the decision requirements:

$\mathbf{X_{ikgl}}$	$\begin{Bmatrix} 1 \text{ if component type } i \text{ is staged on machine for group } g \text{ in line } l \\ 0 \text{ otherwise} \end{Bmatrix}$
$\mathbf{V_{jgl}}$	$\begin{Bmatrix} 1 \text{ if board type } j \text{ is assigned to group } g \text{ in line } l \\ 0 \text{ otherwise} \end{Bmatrix}$
$\mathbf{Y_{ijkgl}}$	$\begin{Bmatrix} \text{Quantity of component } i \text{ placed on card type } j \text{ by machine } k \\ \text{for group } g \text{ in line } l \end{Bmatrix}$
$\mathbf{B_g} \subseteq \mathbf{B}$	indexed set of components required for card type *j*
$\mathbf{A_{jkgl}} \subseteq \mathbf{A}$	indexed set of component types that are placed by machine *k* on card type *j* for group *g* in line *l* $\mathbf{A_{jkgl}} = \{i: \mathbf{X_{ikgl}} = 1 \text{ and } i \in \mathbf{A_j}\}$

The objective is to minimize the total assembly time for the set of cards in **B**. The total assembly time is the sum of setup time for each of the groups and the processing time for each of the cards in each of the groups. If ST_{gl} indicates the setup time for group *g* in line *l* and PT_{jl} is the processing time for card type *j* in line *l*, then the total assembly time can be expressed as $\sum_l \sum_g ST_{gl} + \sum_j \sum_l q_j \, PT_{jl}$.

2.3. ASSUMPTIONS

The following assumptions were included in addressing this problem:

1. The time spent in setup is proportional to the number of feeder changes.
2. Processing cannot occur while the feeders are being changed.
3. The feeder carriage can accommodate any combination of feeders.

4. There is a set of J card types to be processed during a time period and these card types are known at the beginning of the time period.
5. All the placement machines in the assembly line are setup concurrently, and the machine that requires the largest setup time determines the setup time.
6. The placement machine which requires the largest processing time (the bottleneck machine) dictates the processing time.
7. All the card types that need to be processed are known at the beginning of a time period and can be processed during this period.

2.4. RELATION WITH OTHER PROBLEMS IN LITERATURE

The line assignment problem is similar to the *generalized assignment problem* [6] [7] wherein a set of operators are assigned to a set of machines to minimize total cost. The circuit cards can be viewed as a set of operators in a facility, while the assembly lines can be viewed as machines. The cost of assigning operators to different machines can be viewed as cost of assigning cards to assembly lines.

The card grouping problem is analogous to an *uncapacitated facility location problem* [13][15] wherein service facilities are provided for a set of clients. The printed circuit cards can be viewed as clients while the groups are service facilities. The processing time for each card type in a group represents the distance from a client to a service facility, while the setup time for a group can be viewed as the cost of building a new service facility [13].

The component allocation problem is a special case of the *mixed-model assembly line balancing problem* or a *bin packing problem* [3][13] and involves assigning component types to machines in an attempt to balance workloads across different machines [13]. The number of feeder slots on the feeder carriage is analogous to the capacity of a bin. The number of slots that a component type occupies in the feeder carriage is analogous to the size of a unit in the bin. The time to place a component type by a placement machine is analogous to the weight of a unit. In this case, there are multiple bins to pack (machines) and the objective is to balance the weight across the bins [13].

The feeder assignment and placement sequencing problems are comparable to the *quadratic assignment problem* (QAP) [15] and the *travelling salesman problem* (TSP) [6]. Feeder assignment involves assigning component feeders in the form of reels to feeder locations, and placement sequencing problems specifies the sequence in which each component is placed on the circuit card by the surface mount machine. Generally, the feeder assignment problems and placement sequencing problems are decoupled and solved using QAP and TSP heuristics [6].

2.5. RESEARCH STRATEGY

The research strategy for addressing the circuit card to line assignment problem was highlighted by Ellis [14] in a project proposal submitted to a large international network wireless firm. It consists of the following steps:

1. Refine the problem statement through discussions and site visits with the process planning engineers;
2. Develop optimization models which capture the important characteristics of the line assignment problem in printed circuit card assembly;
3. Develop a solution approach appropriate for the optimization model, based on the detail and sophistication of the resulting optimization models;
4. Implement the solution approach through industry representative data sets using a combination of CPLEX optimization software and C++ programs on a Sun UltraSPARC2 Workstation; and
5. Evaluate the optimization models using the results obtained;

CHAPTER III

LITERATURE SURVEY

The line assignment problem is an integral decision in process planning and is closely related to other decision problems. The related decision problems include setup strategy selection, card grouping, component allocation, feeder assignment and placement sequencing. A summary of the literature for these decision problems is presented in this chapter. The chapter concludes with a summary of literature review and the relationship of the published literature to the current research.

3.1. DECISION PROBLEM LITERATURE

The research by Ammons *et al.* [4] provides an overview of the decision problems in process planning. The authors identify the levels of decision problems involved in process planning in printed circuit card assembly. For each level of the hierarchy, the following procedures are identified:

- Setup management strategies,
- Line Assignment,
- Card Grouping,
- Card Sequencing,
- Component allocation, and
- Feeder assignment and placement sequencing.

Computer-aided Process Planning Tools (CAPP) provide an effective means to capture the relationships for implementing the algorithmic strategies in the decision making process. McGinnis *et al.* [27] illustrate the relationships between functions of process planning, production planning, production scheduling, and shop floor control.

They emphasize the need for an algorithmic strategy for making higher-level decisions such as assignment of cards to assembly lines and grouping circuit cards into families.

The literature is organized according to the hierarchy suggested by Ammons *et al.* [4]. First, literature dealing with setup management strategies in printed circuit card assembly systems is addressed. Next, the articles on the line assignment problem and the card grouping problem are discussed. The literature related to the component allocation problem is presented in the next section. Finally, the literature focusing on feeder assignment and placement sequencing problems is presented.

3.1.1. Setup Strategy Selection

Manufacturers currently employ a variety of setup strategies in printed circuit card assembly. The papers related to setup management strategies and setup optimization are summarized in this section.

A classification of setup strategies was presented by McGinnis *et al.* [27], where the setup strategies are classified as follows:

1. Single setup strategy
    a) unique setup
    b) family setup
2. Multi-setup strategy
    a) decompose and sequence
    b) partition and repeat

The single setup strategy requires only one setup of the production line to assemble a given set of card types. If the set contains only a single card type, then the strategy is referred to as *unique setup strategy*. With the unique setup strategy, each circuit card requires its own setup. If the set contains multiple card types, then the strategy is referred to as *family setup strategy*. With family setups, the circuit cards are grouped into families. All the circuit cards in the same family can be processed in one setup. In the case of a unique setup strategy, maximum production rate can be achieved for a set of card types in the schedule by optimizing the line for each card type. However, this may require long downtimes since the line is reset between card types. In the case of a family setup strategy, all the circuit cards in the group are processed in one setup, thus reducing setup time on the assembly line. However, the line may be less efficient for assembling a given card type with a family setup as compared to unique setup [13]. When choosing among unique and family setup strategies there is a tradeoff between increased availability of the assembly line and reduced line efficiency for individual card types in the production schedule.

The multi-setup strategy requires more than one setup of the production line to complete the assembly of a set of circuit card types, often due to the limited feeder capacity on the surface mount machine. In the case of *decompose and sequence strategy*, a family of card types is divided into subfamilies and are then sequenced to minimize the incremental setups between subsets [13]. In the case of *partition and repeat strategy*, “the set of components required for the family of card types is partitioned into subsets such that the assembly line has enough staging capacity for each subset of components [13]”. Later, the assembly line is configured to process each subset of components without exceeding feeder capacity.

The setup management strategies used in different divisions of Hewlett-Packard are summarized by Johnson [24]. The Spokane division produces a high mix of cards in small lots. Each time a batch of cards is completed, the line must undergo a setup for the next card type. This approach is related to the single setup strategy. The Andover installation of Hewlett-Packard produced over 100 card types in medium batches (25-50). At the beginning of each shift, the production sequence is determined. Between each card produced the software would inform the operator which feeders to remove and which ones to add. This is similar to the multi-setup strategy (partition and repeat). The Roseville Network Division of Hewlett-Packard separated the cards into four production families. On the high speed pick and place machine, they segmented the feeder bank into two halves. The most common components are dedicated to the first half while the second half were based on the family. Within a family they do not have to make any changes, then for a new family, only the second half of the feeder bank needs to be changed. The Puerto Rico division had redesigned the feederbank into two halves, which could be completely removed in a few minutes. The idea is to move the setup offline but is constrained by the cost of a spare feederbank and the cost of extra feeders.

Maimon *et al.* [26] look at two scheduling methods that can reduce the set-up in printed circuit cards assembly. The Grouped Set-Up (GSU) method is based on the idea that groups of printed circuit cards are produced in two stages. First, the common components (i.e., components that are shared among two or more printed circuit card types in the group) are setup on the machine for the whole group. In the next stage, referred to as residual setup and assembly, the assembly of the remaining components on each printed circuit card type is performed sequentially. The second approach, Sequence Dependent Scheduling (SDS), is based on the idea that printed circuit card types should be sequenced such that the follower printed circuit card will have a maximum of common components with the current printed circuit card, thus eliminating much of the setup between them. The authors compare these methods in terms of line throughput, average work-in-process level, and implementation complexity.

The setup strategy selection problem has also been addressed by Ellis [13]. A solution approach is developed for determining whether a card type should be assembled using a unique setup or family setup strategy. The research addresses card grouping, component allocation, and machine optimization decisions. The lower level machine configuration issues are captured through an empirically based processing estimator function. This estimator function is developed experimentally for specific placement machine type and machine optimization software. This function is incorporated with the higher level decision problems of component allocation and card grouping and is modeled as a large scale linear mixed integer program. The setup strategy approach is limited to single setup strategies and is solved using a branch-and-bound algorithm for the single machine and the multiple machine case. The methodology is demonstrated through application to several manufacturing scenarios by applying the results to several industrial case studies.

3.1.2. Line Assignment

The line assignment problem has a problem structure similar to a generalized assignment problem. The generalized assignment problem involves assigning tasks to machines with the objective of minimizing total cost (time) and is a special case of the minimum cost network flow problem. The cost of assigning a task to a machine is represented by the flow on arcs. The generalized assignment problem is modeled as a linear integer problem by several researchers [7][16][29].

In the case of line assignment problem, the circuit cards to be processed can be viewed as tasks while the assembly lines represent the machines. The total assembly time, which includes setup and processing time, is represented by the flow on the arcs. The total assembly time, however, is not known in advance. In addition, assembly line capacities should not be exceeded when assigning circuit cards to lines and certain card types need to be processed together in an assembly line. Thus, the line assignment problem has some similarities to the generalized assignment problems but also has additional complexity. No literature on the line assignment problem in printed circuit card assembly was found.

3.1.3. Card Grouping

The objective of the grouping problem is to group circuit cards into families to reduce the number of setups in printed circuit card assembly systems. Concepts arising from grouping and loading problems in flexible manufacturing systems (FMS) and cellular manufacturing can be modified for addressing the card grouping problem in printed circuit card assembly [6][31][34]. Many of these concepts are adapted to card grouping in printed circuit card assembly.

Bhaskar and Narendran [8] use the group technology approach to group the printed circuit cards for setup reduction. In their research, a set of printed circuit cards is to be assembled on a single machine to minimize setup time. This problem is formulated as a mathematical model and is suggested to be NP-hard. The printed circuit cards and the components are then grouped on the basis of a similarity measure, called the cosine similarity measure. A network is then constructed with printed circuit cards as nodes and the similarities between them as the weights of the arcs. A heuristic based on a maximum spanning tree approach is used for grouping a set of printed circuit cards. Initially, all the printed circuit cards are placed in a single group, resulting in an infeasible solution that violates the capacity constraint. An iterative procedure is used to increase the number of groups while checking for the capacity constraint at each iteration.

Askin *et al.* [5] analyze a component assignment and grouping problem independently of each other. They develop a component assignment algorithm to balance workload among different machines subject to feeder capacity. Once components are assigned, the similarity of processing time on each machine is used to join cards into groups. Three procedures are proposed for allocating components to machines and subsequently scheduling cards on machines in order of shortest processing time (SPT). The performance of the three procedures is compared by using estimated daily, two-day, and weekly production requirements for an actual cell consisting of five machines.

Hillier and Brandeau [22] investigate how to assign cards and components to machines and a manual assembly process so as to minimize cost. The manual process is assumed to be uncapacitated while the insertion machines are capacitated. An optimal solution is developed for a single-machine case by exploiting the problem structure. The algorithm uses a linear relaxation to obtain lower bounds, and Lagrangian relaxation to obtain feasible solutions and upper bounds. For the multiple machine case, starting with the fastest machine (defined as the machine that could produce the entire set of cards the fastest), the one-machine problem is solved. The cards assigned to the first machine are then removed from further consideration. Then, the single machine problem is solved again with the second fastest machine. The process is repeated until all the machines have been considered. The remaining unassigned cards are then assigned to the manual process. This research has been inspired by an application at Hewlett-Packard, where the cards were produced in low volumes and machines were not a bottleneck. The models proposed by the authors do not attempt to balance the workloads across machines.

Dino and Marco [12] analyze a grouping problem using the Cellular Manufacturing and Group Technology (GT) concepts in printed circuit card assembly. The authors attempt to minimize the number of setups and maximize the workload balance across different machines. A heuristic is developed and is tested using a simulation model. This heuristic allows the user to define the upper and lower utilization rates of the assembly machine. The heuristic is implemented using data sets from a real printed circuit card assembly system.

Daskin *et al.* [11] analyze a card grouping problem for a single machine case using the 'partition and repeat' strategy for managing printed circuit board assembly resources. A mathematical formulation is provided and is shown to be NP-complete in nature. A heuristic model for the printed circuit card grouping problem is proposed for the case where printed circuit card is loaded once in an optimal solution. A branch and bound algorithm is developed by addressing issues such as variable branching strategies, node selection strategies, solution bounding at each node, and the application of the heuristic model in conjunction with the algorithm. Computational results of this algorithm are given for four test problems.

A number of other researchers have also addressed the card grouping problem including Shtub and Maimon [33] and Hashiba and Chang [20]. Shtub and Maimon [33] address the card grouping problem by developing similarity measures between each card type. They develop a solution approach for card grouping based on these similarity measures for a single machine or multiple machines treated independently. Hashiba and Chang [20] propose a method for the card grouping problem for multiple machine case by aggregating all the machines into a single "super machine" using group technology methods. The objective is to reduce the total setup time for a set of card types.

3.1.4. Component Allocation

For a set of cards, the component allocation decision distributes component types, and the corresponding workload, among available placement machines. Component allocation in printed circuit card assembly systems is a special case of classical mixed-model assembly line balancing and involves assigning component types to various machines to balance the workload among different machines.

Ammons *et al.* [3] address the problem of component allocation to balance workload across two or more machines. The objective is to balance, for every card type, a combination of processing time and setup time. A mathematical formulation is developed and two alternative solution approaches to solve this problem are presented. The first method involves using a list-processing based heuristic, which minimizes the difference between minimum and maximum workload of the machines placing the components. The second method is based on solving a linear mixed integer program using a mixed integer programming solver. Industrial case study results are presented for each approach.

Hillier and Brandeau [21] investigate how to assign the cards and components to machines and a manual assembly process at Hewlett-Packard company so as to minimize cost while at the same time balancing workloads across machines. A binary integer formulation that minimizes total setup and component insertion cost is presented. The authors present a mathematical formulation and develop a heuristic and optimal method for solving this problem. The optimal method uses branch and bound, with the heuristic applied to generate upper bounds. Computational results are presented and the optimal method was shown to be effective for small and medium-sized problems.

Ahmadi and Kouvelis [2] solve a staging problem on a dual delivery pick and place machine. The staging problem involves the allocation of component feeders to its two-feeder carriers along with the corresponding sets of vacuum nozzles to dispense the components. An integer programming formulation of the staging problem for a single card type is presented. The formulation belongs to the class of NP-complete problems and a Lagrangian relaxation scheme is used to obtain upper bounds for the branch and bound algorithm used to solve the single product staging problem. An algorithm that builds upon the results of the single product staging problem is used to solve the multiple product staging problem. Computational results are presented for the single and multiple product staging problems.

A number of other researchers have addressed the component allocation problem. This research is described in Ammons *et al.* [3] and is categorized by setup strategy.

3.1.5. Feeder Assignment and Placement Sequencing

The feeder assignment and placement sequencing problems involve decisions at the lowest level. Feeder assignment involves assigning component types to locations on the feeder carriage of the assembly machine, and placement sequencing involves specifying the sequence that the components are placed on the circuit card by the assembly machine. An overview of the feeder assignment and placement sequencing decision problems is provided by McGinnis *et al.* [27]. The form and complexity of these problems is a function of the type of assembly machine under consideration.

Crama *et al.* [10] investigate the feeder rack assignment problem, the placement sequencing problem, and the component retrieval plan problem arising at a Philips manufacturing plant. For each component on the card, a component retrieval plan provides information on the feeder location from which the component is to be retrieved. The Fuji CP-IV, a surface mount assembly machine, is investigated and the feeder rack assignment problem is solved using a heuristic that takes into account as much as possible the individual card type characteristics. The placement sequencing and the component retrieval problem are solved using constructive heuristics and local search methods. The solution procedure shows significant improvement in terms of makespan when tested on real life instances arising at Philips NV.

The solution for the feeder slot assignment problem for a given component placement sequence is developed by Moyer and Gupta [28]. The problem is modeled as a quadratic assignment problem and is found to be NP-complete. Two different heuristic approaches are proposed to minimize the feeder travel distance. A comparative analysis between the two heuristics is presented and the authors summarize the attributes of each heuristic.

Ahmadi *et al.* [1] look at the feeder assignment problem for a dual delivery pick-and-place machine. The authors formulate this problem as a mathematical program and establish this problem to be NP-complete. Heuristics are developed with the objective of minimizing the changes in the feeder carrier direction and total movement. The method has been implemented at the IBM plant in Austin, Texas, and has achieved 7 to 8% reductions in cycle time.

Leipala and Nevalainen [25] look at the placement sequencing problem on a Panasert RH machine in electronics assembly. The placement sequencing problem is solved as an asymmetric traveling salesman problem. A mathematical formulation is presented and heuristics are developed for solving the asymmetric traveling salesman problem.

A number of other researchers have addressed the feeder assignment or placement sequencing problems. This research is described by McGinnis *et al.* [27] and is categorized by the type of placement machine considered.

3.2. SUMMARY OF LITERATURE REVIEW

The literature described in this chapter includes articles on setup management, line assignment, card grouping, component allocation, and feeder assignment and placement sequencing. The focus of this research is to develop a solution approach for solving the assignment of circuit cards to assembly lines with the objective of minimizing the total assembly time including setup time and processing time. Several researchers have conducted research on solution approaches for the various related decision problems, however no literature was found that addresses line assignment problem in printed circuit card assembly. This research addresses the line assignment problem by incorporating the lower level decision problems of card grouping and component allocation in printed circuit card assembly.

CHAPTER IV

LINE ASSIGNMENT PROBLEM MATHEMATICAL MODELS

The approach for addressing the line assignment problem is to develop mathematical models that capture the important characteristics of the line assignment problem and the related problems of card grouping and component allocation. The line assignment problems are modeled as large scale linear mixed integer programming problems in this chapter. The mathematical models are presented in this chapter for both single and multiple machine problems, assuming single setup strategy approach.

4.1. SINGLE MACHINE LINE ASSIGNMENT PROBLEMS

The model illustrated in this section captures the problem of assigning circuit cards to assembly lines and grouping circuit cards into families, assuming a single assembly machine in each line. The objective is to minimize the total assembly time for all the cards which includes both setup time and processing time. The processing time is assumed to be independent of processing times of the other cards assigned to the same family. Ellis [13] addresses the single machine card grouping problems for single setup strategies. These concepts are extended to handle the line assignment decisions in printed circuit card assembly for a single machine case.

The system can be described as follows: J printed circuit cards are to be assembled in L lines. They require I different types of components. Each machine in a line has a finite staging capacity, which is equal to the total number of slots S_l, and a finite production capacity, represented by C_l. Each component requires slot capacity on the assembly machine in line l, as represented as s_{il}. This problem is formulated as a 0-1 linear mixed integer programming model with the following parameters and decision variables:

Parameters

i = index for component types; i = 1, 2, ………I

j = index for card types; j = 1, 2, ………J

g = index for groups;

l = index for lines;

q_j = production quantity of card type *j*;

$$a_{ij} = \begin{cases} 1 \text{ if component type } i \text{ is used on card type } j \\ 0 \text{ otherwise} \end{cases}$$

$$b_{jl} = \begin{cases} 1 \text{ if card type } j \text{ can be processed in line } l \\ 0 \text{ otherwise} \end{cases}$$

f_l = time to change one feeder on machine in line *l*;

c_l = constant parameter obtained from regression analysis used for estimating cycle times for a card type in line *l*;

d_{ij} = quantity of component type i used on card type *j*;

s_{il} = number of feeder slots required for component type *i* on machine in line *l*;

S_l = total number of feeder slots available on machine in line *l*;

C_l = available production capacity in line *l*;

t_{il} = time to insert/place component type *i* on machine in line *l*;

ST_{gl} = setup time for group *g* in line *l*;

PT_{jl} = processing time for card type *j* in line *l*;

Decision Variables

$$X_{igl} = \begin{cases} 1 \text{ if component type } i \text{ is staged on machine for group } g \text{ in line } l \\ 0 \text{ otherwise} \end{cases}$$

$$V_{jgl} = \begin{cases} 1 \text{ if board type } j \text{ is assigned to group } g \text{ in line } l \\ 0 \text{ otherwise} \end{cases}$$

The problem formulation is shown in Figure 4-1. The objective function (4-1) corresponds to the total assembly time for the set of cards to be produced. The setup time for each group *g* assigned to line *l* is modeled as a linear function of the number of components that are setup on the machine and is captured by (4-2). The processing time for card *j* is the time to insert all components plus a constant time required to adjust the card as it enters the machine and is captured by (4-3). Constraint (4-4) ensures that each card type *j* is assigned to exactly one group in one line. Constraint (4-5) ensures that all components required for a printed circuit card are mounted on the assembly machine, when the printed circuit card is loaded. Sometimes certain card types cannot be produced on an assembly line due to the characteristics of the circuit card, the features of the testing equipment, the lack of a glue dispenser, or other reasons. If card type *j* cannot be assigned to line *l* (such that $b_{jl} = 0$), then the corresponding V_{jgl} variables are not included for constraints (4-3), (4-4), and (4-5). Constraint (4-6) corresponds to the feeder capacity constraint, which ensures that the total number of feeder slots required for all the components in group *g* assigned to line *l* cannot exceed the assembly machine feeder carriage capacity. Constraint (4-7) is the line capacity constraint, and ensures that the total setup time and the total processing time for cards to be assembled in the line does not exceed production capacity. Finally, constraint (4-8) ensures that the decision variables are binary decision variables.

LINE ASSIGNMENT FOR SINGLE MACHINE CASE

Minimize

$$\sum_{l}\sum_{g} ST_{gl} + \sum_{l}\sum_{j} q_j PT_{jl} \qquad (4\text{-}1)$$

Subject to:

$$ST_{gl} - \sum_{i} f_l X_{igl} \geq 0 \qquad \forall g, l \qquad (4\text{-}2)$$

$$PT_{jl} - \sum_{i} d_{ij} t_{il} V_{jgl} - c_l V_{jgl} \geq 0 \qquad \forall j, g, l \qquad (4\text{-}3)$$

$$\sum_{l}\sum_{g} V_{jgl} = 1 \qquad \forall j \qquad (4\text{-}4)$$

$$a_{ij} V_{jgl} \leq X_{igl} \qquad \forall i, j, g, l \qquad (4\text{-}5)$$

$$\sum_{i} s_{il} X_{igl} \leq S_l \qquad \forall g, l \qquad (4\text{-}6)$$

$$\sum_{g} ST_{gl} + \sum_{j} PT_{jl} \leq C_l \qquad \forall l \qquad (4\text{-}7)$$

$$X_{igl}, V_{jgl} \in \{0, 1\} \qquad \forall i, j, g, l \qquad (4\text{-}8)$$

Figure 4-1. Line Assignment Problem for Single Machine

In this formulation, G is a parameter and is initially set to J. When no grouping is possible, the number of card groups is equal to number of printed circuit cards that needs to be processed. For a version of this problem with 20 circuit card types, 284 component types, and 4 lines, the model contains 9398 constraints and 8299 variables. These kinds of problems are shown to be NP-complete in nature [18]. A solution approach is presented in the subsequent chapters to solve these problems in a reasonable amount of time.

4.2. MULTIPLE MACHINE LINE ASSIGNMENT PROBLEMS

The single machine line assignment problem is extended to handle the case when there are multiple machines in each line. Two models are presented which address the multiple machine line assignment problem by considering the lower level decision problems. The problem of assigning circuit cards to assembly lines, grouping circuit cards into families, and assigning components to assembly machines is captured by the first model. This model allows cards to be grouped together in families for processing. The second model captures the decision problems of assigning circuit cards to assembly lines and allocating component types to machines to balance workload. This model assumes each circuit card is processed with a unique setup. The multiple machine line assignment problems are formulated as linear mixed integer programming models with the following parameters and decision variables:

Parameters

i = index for component types; $i = 1, 2, \ldots\ldots\ldots I$

j = index for card types; $j = 1,2, \ldots\ldots\ldots J$

k = index for machines;

g = index for groups;

l = index for lines;

q_j = production quantity of card type j;

$$a_{ij} = \begin{Bmatrix} 1 \text{ if component type } i \text{ is used on card type } j \\ 0 \text{ otherwise} \end{Bmatrix}$$

f_{kl} = time to change one feeder on machine k in line l;

c_{kl} = constant parameter obtained from regression analysis used for estimating cycle times for a card type on machine k in line l;

$$b_{jl} = \begin{Bmatrix} 1 \text{ if card type } j \text{ can be processed in line } l \\ 0 \text{ otherwise} \end{Bmatrix}$$

$$h_{ikl} = \begin{Bmatrix} 1 \text{ if component type } i \text{ can be placed by machine } k \text{ in line } l \\ 0 \text{ otherwise} \end{Bmatrix}$$

d_{ij} = quantity of component type i used on card type j;

s_{ikl} = number of feeder slots required for component type i on machine k in line l;

S_{kl} = total number of feeder slots available on machine k in line l;

C_l = available production capacity in line l;

t_{ikl} = time to insert/place component type i on machine k in line l;

ST_{jl} = setup time for card j in line l;

PT_{jl} = processing time for card type j in line l;

Decision Variables

$$X_{ijkl} = \begin{cases} 1 & \text{if component type } i \text{ is staged on machine } k \text{ for card type } j \text{ in} \\ & \text{line } l \\ 0 & \text{otherwise} \end{cases}$$

$$V_{jgl} = \begin{cases} 1 & \text{if board type } j \text{ is assigned to group } g \text{ in line } l \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{ijkgl} = \begin{cases} \text{Quantity of component type } i \text{ placed on card type } j \text{ by machine } k \\ \text{for group } g \text{ in line } l \end{cases}$$

4.2.1. Multiple Machine Case with Card Grouping

The line assignment model for multiple machines with card grouping considers the line assignment, card grouping, and component allocation decision problems. This model incorporates the processing times of cards and is appropriate when multiple machines in series are used to assemble a set of cards and the processing times for a card type is not influenced by the other cards in a group. This model is applicable for both single and double sided cards and takes the line and machine requirements for the card types into account. This is an extension of a multiple machine card grouping model developed by Ellis [13] for a single assembly line.

The mixed integer problem formulation is shown in Figure 4-2. The objective function minimizes the total assembly time for the set of cards, including setup time and processing time as shown in (4-9). The setup time for each group g assigned to line l is modeled as a linear function of the number of components setup on different machines in each line and the machine that takes the maximum setup time dictates the setup time as represented by (4-10). The processing time for card *j* in line *l* is the minimum time to insert all components plus a constant time required to adjust the card as it comes into the machine. The machine that takes the maximum time dictates the processing time for card type *j* in line *l* and is captured by (4-11). Constraint (4-12) ensures that each card type *j* can be assigned to exactly one group *g* in one line *l*. The total quantity of component *i* required on card type *j* must be placed by the set of machines in line *l*, as expressed by constraint (4-13). Constraint (4-14) ensures that component *i* placed on card type *j* by machine *k* in line *l* for group *g* is staged on machine *k* in line *l*. The total number of feeder slots required for the components in group *g* assigned to line *l* cannot exceed the assembly machine feeder carriage capacity as represented by constraint set (4-15). Constraint set (4-16) ensures that the capacity of the line is not exceeded. Finally, constraint set (4-17) ensures that the decision variables are integer variables. This is the first known model to address line assignment, card grouping, and component allocation to balance workload decision problems.

Sometimes certain card types cannot be produced on all lines (such that $b_{jl} = 0$) due to the characteristics of the circuit card, the features of the testing equipment, the lack of glue dispenser for double sided cards, or other reasons. The corresponding V_{jgl} and Y_{ijkgl} variables are not included in the model when the problem is generated. This model also accounts for the fact that certain machines cannot place certain components. For example, some of the larger components such as integrated circuits cannot be placed by a chip shooter machine such as Fuji CP IV. If component type *i* cannot be placed by machine *k* in line *l* such that $h_{ikl} = 0$, then the corresponding X_{ikgl} and Y_{ijkgl} variables are not included in the model.

For a version of this problem with 1783 components, 33 cards, and 9 lines, the model contains 1,244,881 integer variables and 994,360 constraints. An algorithmic solution approach needs to be developed to handle these large size problems. A flexible solution approach that can improve the computational time required to solve these models may be developed as an extension to this research effort.

LINE ASSIGNMENT FOR MULTIPLE MACHINE CASE
(With Card Grouping)

Minimize $$\sum_{l}\sum_{g} ST_{gl} + \sum_{j}\sum_{l} q_j . PT_{jl} \qquad (4\text{-}9)$$

Subject to:

$$ST_{gl} - \sum_{i} f_{kl} X_{ikgl} \geq 0 \qquad \forall k, g, l \qquad (4\text{-}10)$$

$$PT_{jl} - \sum_{i} t_{ikl} Y_{ijkgl} - c_{kl} V_{jgl} \geq 0 \qquad \forall j, k, g, l \qquad (4\text{-}11)$$

$$\sum_{l}\sum_{g} V_{jgl} = 1 \qquad \forall j \qquad (4\text{-}12)$$

$$d_{ij} V_{jgl} \leq \sum_{k} Y_{ijkgl} \qquad \forall i, j, g, l \qquad (4\text{-}13)$$

$$Y_{ijkgl} \leq d_{ij} X_{ikgl} \qquad \forall i, j, k, g, l \qquad (4\text{-}14)$$

$$\sum_{i} s_{ikl} X_{ikgl} \leq S_{kl} \qquad \forall k, g, l \qquad (4\text{-}15)$$

$$\sum_{g} ST_{gl} + \sum_{j} PT_{jl} \leq C_l \qquad \forall l \qquad (4\text{-}16)$$

$$X_{ikgl}, V_{jgl} \in \{0, 1\} \qquad \forall i, j, k, g, l \qquad (4\text{-}17)$$

$$Y_{ijkgl} \geq 0, \text{integer}$$

Figure 4-2. Line Assignment Problem for Multiple Machines with Card Grouping

4.2.2. Multiple Machine Case for Unique Setups

The multiple machine line assignment model for unique setup strategies considers the line assignment and component allocation problems. The mixed integer problem formulation for this problem is shown in Figure 4-3. This model is applicable for both single and double sided cards and takes the line requirements of the card types into account.

The objective function minimizes the total assembly time for the set of cards, including setup time and processing time as shown in (4-18). The setup time for each card *j* assigned to line *l* is modeled as a linear function of the number of components setup on different machines in each line and the machine that takes the maximum setup time dictates the setup time as represented by (4-19). The processing time for card *j* assigned to line *l* is the time to insert all components plus a constant time required to adjust the card as it enters into the machine. The constant time required for each card type *j* on machine *k* in line *l* is provided by the industrial partner and is obtained by regression analysis for a particular machine type in each line. The machine that requires the largest processing time determines the processing time for card type *j* in line *l* and is captured by (4-20). Constraint (4-21) is applicable when a double sided card is processed as two separate single sided cards. This is further described in the following paragraph. Constraint (4-22) ensures that each card type *j* is assigned to exactly one line. The total quantity of component *i* required on card type *j* must be placed by the set of machines in line *l* and is captured by constraint (4-23). Constraint (4-24) ensures that component *i* placed on card type *j* by machine *k* in line *l* is staged on machine *k* in line *l*. The total number of feeder slots required for card type *j* assigned to line *l* cannot exceed the assembly machine feeder carriage capacity as represented by constraint set (4-25). Constraint set (4-26) ensures that the capacity of the line is not exceeded. Finally, constraint set (4-27) ensures that the decision variables are integer variables. This is the first known model to address line assignment and component allocation for unique setups.

Double sided cards have component types on both the front and the back side of the card. The front and back side components are differentiated in the input data (see Appendix A). The industrial partner handles them in two different ways.

Sometimes the double sided cards are processed on an assembly line with the capability to populate the back side of the card, flip the card, and then populate the front side of the card. The components on the backside of the card are allocated to the machines that precede the card flipper ($h_{ikl} = 0$ for these components for the machines that follow the flipper). The components on the front side of the card are allocated to machines that follow the card flipper ($h_{ikl} = 0$ for these components for the machines that precede the flipper). This ensures that the components on both sides of the card are allocated to the appropriate assembly machines.

Sometimes the double sided cards are processed as two separate cards. This is usually done when the assembly lines with the card flippers are not capable (for other reason) of processing a specific double sided card. In this case the double sided card is processed as two separate single sided cards. For the model, card type j is split into two separate cards j_1 and j_2, that are batched and processed once for the back side components and once for the front side components. Constraint (4-21) ensures that these cards are assigned to the same line.

Certain card types cannot be produced on some of the assembly lines (such that $b_{jl} = 0$) due to lack of testing equipment, etc. The corresponding V_{jl} and Y_{ijkl} variables are not included in the model when the problem is generated. This model also accounts for the fact that certain machines can only place certain components. For example, some larger components can be placed by a Fuji IP or a Fuji QP placement machine, but may not be placed by a Fuji CP placement machine. If component type i cannot be placed by machine k in line l such that $h_{ikl} = 0$, then the corresponding X_{ijkl} and Y_{ijkl} variables are not included in the model.

An industry version of this problem with 2739 components, 71 cards, and 9 lines contains 244,411 variables and 143,132 constraints. A solution approach is presented in the next chapter to solve these models in a reasonable amount of time.

LINE ASSIGNMENT FOR MULTIPLE MACHINE CASE

(With Unique Setups)

Minimize $$\sum_{l}\sum_{j} ST_{jl} + \sum_{j}\sum_{l} q_j\, PT_{jl} \qquad (4\text{-}18)$$

Subject to:

$$ST_{jl} - \sum_{i} f_{kl} X_{ijkl} \geq 0 \qquad \forall j, k, l \qquad (4\text{-}19)$$

$$PT_{jl} - \sum_{i} t_{ikl} Y_{ijkl} - c_{kl} V_{jl} \geq 0 \qquad \forall j, k, l \qquad (4\text{-}20)$$

$$V_{j_1 l} \leq V_{j_2 l} \qquad \forall j_1, j_2, l \qquad (4\text{-}21)$$

$$\sum_{l} V_{jl} = 1 \qquad \forall j \qquad (4\text{-}22)$$

$$d_{ij} V_{jl} \leq \sum_{k} Y_{ijkl} \qquad \forall i, j, l \qquad (4\text{-}23)$$

$$Y_{ijkl} \leq d_{ij} X_{ijkl} \qquad \forall i, j, k, l \qquad (4\text{-}24)$$

$$\sum_{i} s_{ikl} X_{ijkl} \leq S_{kl} \qquad \forall j, k, l \qquad (4\text{-}25)$$

$$\sum_{j} ST_{jl} + \sum_{j} PT_{jl} \leq C_l \qquad \forall l \qquad (4\text{-}26)$$

$$X_{ijkl}, V_{jl} \in \{0, 1\} \qquad \forall i, j, k, l \qquad (4\text{-}27)$$

$$Y_{ijkl} \geq 0, \text{integer}$$

Figure 4-3. Line Assignment Problem for Multiple Machines with Unique Setup

4.3. SUMMARY OF MATHEMATICAL MODELS

Mathematical models for the line assignment problem are presented and described in this chapter. The models are classified as single machine and multiple machine line assignment model.

The single machine line assignment problem can be used when there is only one machine in all the assembly lines or when multiple machines can be aggregated into a single machine. The printed circuit cards are assigned to assembly lines and card grouping is done to reduce the number of setups in each line. This model could also be used to obtain an initial assessment of the line assignment decision problem for a multiple machine problem.

The multiple machine line assignment problems are applicable when there is more than one machine in any of the assembly lines. The model with *card grouping* assigns circuit cards to assembly lines, groups cards into families, and allocates components by considering the effects of balancing the line. The model with *unique setups* assigns circuit cards to assembly lines and allocates components to different machines to balance workload.

The solution to these models provides valuable information to process planning engineers in printed circuit card assembly. The solution approaches to these models and case study results are presented in the following chapters.

CHAPTER V

LINE ASSIGNMENT PROBLEM SOLUTION APPROACH

The solution approaches for the line assignment problem are discussed in this chapter. An overview of the methods commonly employed to solve integer programming problems is presented in the first section. The primary method employed in this research is the branch-and-bound approach which is augmented in several ways to improve the solution process. The specific solution approaches employed to solve the line assignment problems are discussed in this chapter.

5.1. SOLVING INTEGER PROGRAMMING PROBLEMS

The following methods are commonly employed to solve integer programming problems [29]:

- Branch and Bound Algorithm
- Cutting Plane Techniques
- Heuristic Methods

5.1.1. Branch and Bound Algorithm

The following steps are employed to solve the integer programming problem using the branch and bound approach:

1. Solve the linear relaxation of the problem using a corresponding linear programming algorithm (primal simplex, dual simplex, network simplex, and barrier methods [7]).
2. If the solution is an integer feasible solution, then optimal solution may have been obtained and procedure may stop.

3. Otherwise, create two new subproblems by branching on a fractional variable.
4. Choose an active subproblem and branch on the fractional variable. A subproblem is not active in the following cases:
    - All variables in the solution are integer variables.
    - The subproblem is infeasible.
    - The subproblem is fathomed by a bounding argument.
5. Go to Step 1 until there are no active subproblems and print optimal solution.

5.1.2. Cutting Plane Techniques

Cutting plane methods require adding constraints to the linear programming problem until the optimal basic feasible solution takes on integer variables. The special type of constraint that will be added without modifying the problem is referred to as a 'cut'. The 'cut' should satisfy the following criteria:

- All integer feasible solutions are feasible for the cut, and
- The current fractional solution is not feasible for the cut.

The cuts could either be generated from the linear programming tableau, referred to as Gomory cuts, or by looking at the structure of the problem. Advanced treatment of the cutting plane techniques is provided by Nemhauser and Wolsey [29].

5.1.3. Heuristic Methods

For many optimization problems, a good but not necessarily optimal solution is sufficient for the final solution. Heuristic methods are often used to generate good, but not necessarily optimal, solutions quickly. These methods are generally problem-specific and are based on certain general rules. For example, heuristic methods used to obtain the bounds may enhance the branch and bound process. Figure 5-1, adapted from Askin and Standridge [6], shows the tradeoff between heuristic and optimization methods.

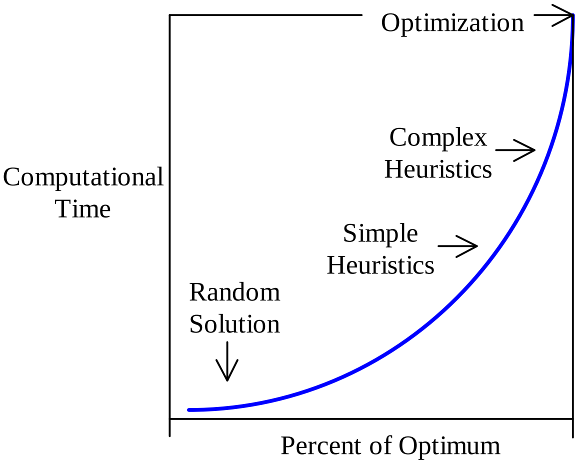

Figure 5-1. Heuristic versus Optimization Trade-off [6]

5.2. SINGLE MACHINE LINE ASSIGNMENT PROBLEMS

The single machine line assignment problem is presented in Section 4.1. For this problem, there is only one machine in each of the assembly lines. The following techniques were used to improve the branch-and-bound technique:

- Reduction of integer variables during problem generation
- Relaxing the integrality constraints on X_{ikl} variables
- Implementing priority orders for branching
- Using optimality tolerances
- Using special branching strategies to take advantage of Special Ordered Sets (SOSs)

Geerdink [18] and Ellis [13] describe the variable reduction technique for solving similar kinds of problems. They consider limiting the number of groups to which a card type can be assigned. For a production schedule that contains a set of m card types and l assembly lines, the number of assignment variables (V_{jgl}) is equal to m^2l if all the card types are considered for every group in each line. Using the variable reduction technique, the card types are limited to groups, which are numbered less than or equal to that particular card type as represented by equation 5-1. The resulting variable reduction for a set of 3 card types to be assembled by 3 assembly lines is illustrated in Figure 5-2.

Card Type 1	**Card Type 2**	**Card Type 3**
V_{111}, V_{112}, V_{113}	V_{211}, V_{212}, V_{213}	V_{311}, V_{312}, V_{313}
V_{121}, V_{122}, V_{123}	V_{221}, V_{222}, V_{223}	V_{311}, V_{322}, V_{323}
V_{131}, V_{132}, V_{133}	V_{231}, V_{232}, V_{233}	V_{331}, V_{332}, V_{333}

Card Type 1	**Card Type 2**	**Card Type 3**
V_{111}, V_{112}, V_{113}	V_{211}, V_{212}, V_{213}	V_{311}, V_{312}, V_{313}
	V_{221}, V_{222}, V_{223}	V_{311}, V_{322}, V_{323}
		V_{331}, V_{332}, V_{333}

Figure 5-2. Comparison of the Number of Card Assignment Variables

The number of assignment variables (V_{jgl}) is reduced to $\left(m\left(\frac{m+1}{2}\right)\right) * l$ which is a significant reduction of variables for a large set of card types and assembly lines. During the problem generation phase, eliminating redundant solution points reduces the solution space. The assignment variables are generated only for the group whose number is less than or equal to the card type number such that:

$$\sum_{j} V_{jgl} = 1 \qquad \forall j, g, l \qquad \text{where } j \geq g. \tag{5-1}$$

In addition, the integrality requirements for the X_{igl} variables are relaxed to further reduce the number of integer variables. With this formulation, the assignment variables X_{igl} are binary if the V_{jgl} variables are binary. Hence, the binary constraints on X_{igl} variables are relaxed, and only the binary constraints for V_{jgl} variables are included in the model.

In many optimization problems, a good but not necessarily optimal solution is sufficient for the final solution. This model is an example of these types of problems where the solution time to achieve optimality is significant. When a good answer suffices, the branch-and-bound process is terminated through the use of optimality tolerances [29].

This tolerance is the difference between the objective value of the best integer solution and the solution value of the linear programming relaxation of the best node remaining. This reduces the number of nodes that are evaluated in the solution process and decreases the solution time.

Special Ordered Sets (SOSs) of Type 3 are identified to improve performance of the branch and bound approach. SOS of Type 3 is defined as the set of variables appearing in an equality constraint with all binary variables with +1 or –1 coefficients and right hand side value of 1 – (number of –1 coefficients). Constraint set (4-4) is identified as a Special Ordered Set of Type 3. SOS branching successively partitions the set into two subsets, and the weights (orders) are used to decide where to split the set. The CPLEX SOSSCAN option with mixed integer problems automatically assigns the weights (orders) to individual SOS members.

The single machine line assignment problem is implemented using the commercially available CPLEX Mixed Integer Solver 6.0 [9] on an Sun UltraSPARC2 Workstation. The results and computational requirements using this approach are presented in the following chapter.

5.3. MULTIPLE MACHINE LINE ASSIGNMENT PROBLEMS

The integer programming models for the multiple machine line assignment problems are presented in Section 4.2. The solution techniques for the multiple machine line assignment problems with card grouping and the multiple machine line assignment problems with unique setups are described in the following sections.

5.3.1. Multiple Machine Case with Card Grouping

The multiple machine line assignment problem, described in Section 4.2.1, assigns circuit cards to assembly lines, groups cards into families, and allocates components to balance the workload among different machines. This model was implemented using CPLEX Mixed Integer Solver 6.0 [9] on a Sun UltraSPARC2 Workstation. A primal heuristic was also developed using data structures provided by MINTO (Mixed INTeger Optimizer) to obtain upper bounds for the branch and bound process. For realistic size problems, however, the solution times to solve these models were prohibitive. Additional techniques to enhance the branch and bound approach need to be developed to improve the solution approach. This is suggested as one of the areas for future study.

5.3.2. Multiple Machine Case for Unique Setups

The multiple machine line assignment problem for unique setups, described in Section 4.2.2, assigns circuit cards to assembly lines and allocates components to balance the workload among different machines. For this problem, the following techniques are used to improve the solution approach:

- Decomposing the problem into two sub-problems
- Relaxing the integrality requirements on Y_{ijkl} variables
- Using optimality tolerances
- Varying performance parameters to improve MIP performance

The problem is decomposed into two subproblems without any effect on solution quality due to the unique structure of the problem. Decomposing the problem significantly improves the solution time during the branch and bound process. The first subproblem is the component allocation problem for unique setup. The component allocation problem is generated for each assembly line and includes all the card types that are feasible for the line. The assembly lines are assumed to be uncapacitated. The solution to the models provides the assembly time (the sum of the setup time and processing time) for each card type on each line. The assembly time can be viewed as the cost of assigning the card type to an assembly line. This information is then used in the second subproblem, the network flow problem. This network flow problem is an extension of the generalized assignment problem through the addition of capacity constraints. Assembly line capacities should not be exceeded when assigning circuit cards to assembly lines.

The mathematical formulation for the component allocation model is shown in Figure 5-3. The objective function minimizes the total assembly time for the set of cards, including setup time and processing time as shown in (5-1). The setup time for each card *j* is modeled as a linear function of the number of components setup on different machines and the machine that takes the maximum setup time dictates the setup time as represented by (5-2). The processing time for card *j* is the time to insert all components plus a constant time required to adjust the card as it enters into the machine. The constant time required for each card type *j* on machine *k* is provided by the industrial partner and is obtained by regression analysis for a particular machine type in each line. The machine that requires the largest processing time determines the processing time for card type and is captured by (5-3). The total quantity of component *i* required on card type *j* must be placed by the set of machines and is captured by constraint (5-4). Constraint (5-5) ensures that component *i* placed on card type *j* by machine *k* is staged on machine *k*. The total number of feeder slots required for card type *j* cannot exceed the assembly machine feeder carriage capacity as represented by constraint set (5-6). Constraint set (5-7) is an optional constraint and ensures that all the placements of a particular component type *i* on card type *j* is placed by only one machine *k*. Finally, constraint set (5-8) ensures that the decision variables are integer variables.

Figure 5-4 shows the network representation of the network flow problem. The flow on the arcs for card type *j* in line *l* is equal to $ST_{jl} + PT_{jl}$. The mathematical formulation for the network representation of the assignment problem is illustrated in Figure 5-5. The objective is to minimize the total assembly time for all card types in the production schedule as represented by (5-9). Constraint (5-10) ensures that card type *j* can be assigned to only one line. Constraint (5-11) ensures that the line capacity is not exceeded. Constraint (5-12) is applicable when a double sided card is processed as two single sided cards. In this case, card type *j* is split into two separate single cards, j_1 and j_2, that are batched and run through once for all topside and once on the bottom side. This constraint ensures that card j_1 and j_2 are assigned to the same line. Constraint (5-13) ensures that the decision variables are binary integer variables. The results and computational requirements using this approach are presented in the following chapter.

COMPONENT ALLOCATION MODEL

(For Unique Setups)

Minimize $\sum_{j} ST_j + \sum_{j} q_j PT_j$ (5-1)

Subject to:

$ST_j - \sum_{i} f_k X_{ijk} \geq 0 \quad \forall j, k$ (5-2)

$PT_j - \sum_{i} t_{ik} Y_{ijk} - c_k \geq 0 \quad \forall j, k$ (5-3)

$\sum_{k} Y_{ijk} \leq d_{ij} \quad \forall i, j$ (5-4)

$Y_{ijk} \leq d_{ij} X_{ijk} \quad \forall i, j, k$ (5-5)

$\sum_{i} s_{ik} X_{ijk} \leq S_k \quad \forall j, k$ (5-6)

$\sum_{k} X_{ijk} = 1 \quad \forall i, j$ (Optional) (5-7)

$X_{ijk} \in \{0, 1\} \quad \forall i, j, k$ (5-8)

$Y_{ijk} \geq 0$, integer

Figure 5-3. Component Allocation Problem with Unique Setup.

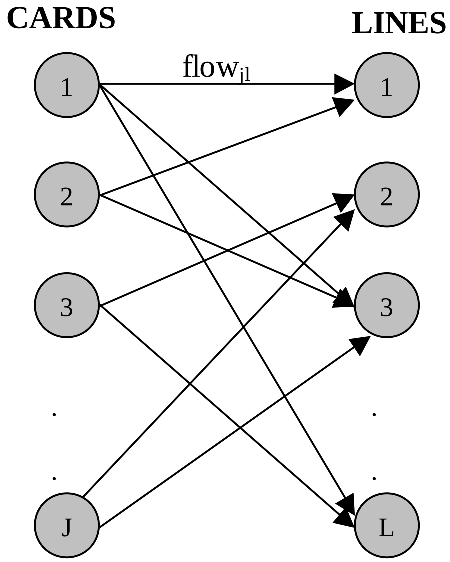

Figure 5-4. Network Representation

NETWORK FLOW MODEL

(of Line Assignment Problem)

Minimize $$\sum_{l}\sum_{j} \text{flow}_{jl} V_{jl} \tag{5-9}$$

Subject to:

$$\sum_{l} V_{jl} = 1 \qquad \forall\, j \tag{5-10}$$

$$\sum_{j} \text{flow}_{jl} V_{jl} \le C_l \qquad \forall\, l \tag{5-11}$$

$$V_{j_1 l} \le V_{j_2 l} \qquad \forall j_1, j_2, l \tag{5-12}$$

$$V_{jl} \in \{0, 1\} \qquad \forall\, j, l \tag{5-13}$$

Figure 5-5. Network Flow Model of Line Assignment Problem.

For the component allocation problem, the optional constraint (5-7) is omitted which allows the component types to be assigned to more than one machine. In addition, integrality requirements on the Y_{ijk} variables are relaxed to improve the speed of the branch and bound process in the component allocation representation shown in Figure 5-3. Based on the values of the Y_{ijk} variables in the final solution, these variables are then rounded using the rounding heuristic shown in Figure 5-6. The following list summarizes the variable notation used with the rounding heuristic routine.

Parameters

i = index for component types;

j = index for card types;

k = index for machines;

l = index for lines;

f_{kl} = time to change one feeder on machine *k* in line *l*;

c_{kl} = constant regression parameter for machine *k* in line *l*;

d_{ij} = quantity of component type *i* used on card type *j*;

t_{ikl} = time to insert component type *i* on machine *k* in line *l*;

b_{jl} = $\left\{\begin{array}{l} 1 \text{ if card type } j \text{ can be processed in line } l \\ 0 \text{ otherwise} \end{array}\right\}$

h_{ikl} = $\left\{\begin{array}{l} 1 \text{ if component type } i \text{ can be placed by machine } k \text{ in line } l \\ 0 \text{ otherwise} \end{array}\right\}$

X_{ijkl} = $\left\{\begin{array}{l} 1 \text{ if component type } i \text{ is staged on machine } k \text{ for card} \\ \quad \text{type } j \text{ in line } l \\ 0 \text{ otherwise} \end{array}\right\}$

Y_{ijkl} = the solution values obtained from the component allocation representation;

R_{ijkl} = the revised rounded values assigned by the rounding routine;

Sum_{ijl} = the sum of all rounded values for component type *i* on machine *k* in line *l*;

ST_{jkl} = setup time for card type j on machine k in line l;

PT_{jkl} = processing time for card type j on machine k in line l;

flow_{jl} = total assembly time for card type j on machine k in line l;

difference = difference between rounded and actual values.

```
1. Calculate rounded values from final solution such that
        $R_{ijkl} = \left| Y_{ijkl} + 0.5 \right|^{-}$            $\forall$ i, j, k, l
2. Calculate the sum for each component i on card type j in line l such that
        $Sum_{ijl} = \sum_{k} R_{ijkl}$            $\forall$ i, j, l
3. Adjust the rounded values to ensure that $Sum_{ijl} = d_{ij}$
    For each component type i on card type j in line l
        (a)    difference = $Sum_{ijl} - d_{ij}$
        (b)    If difference > 0, decrease $R_{ijkl}$ values
                   For each machine k
                   If $R_{ijkl} > 0$ & difference > 0
                       $R_{ijkl} = R_{ijkl} - 1$
                       difference = difference - 1
        (c)    If difference < 0, increase $R_{ijkl}$ values
                   For each machine k
                   If $h_{ikl} = 1$ & difference < 0
                       $R_{ijkl} = R_{ijkl}$ + difference
4. Find the processing time for card type j on machine k in line l such that
        $PT_{jkl} = \sum_{i} t_{ikl} R_{ijkl} + (c_{kl})(b_{jl})$            $\forall$ j, k, l
5. Calculate the maximum processing time for card type j in line l such that
        $PT_{jl} = \max_{k \in K} \{PT_{jkl}\}$            $\forall$ j, l
6. For all values of $R_{ijkl}$ greater than zero set
        $X_{ijkl} = 1$            $\forall$ i, j, k, l
7. Find the setup time for card type j on machine k in line l such that
        $ST_{jkl} = \sum_{i} X_{ijkl} * f_{kl}$            $\forall$ j, k, l        {Continued}
```

Figure 5-6. Rounding Heuristic Routine

8. Calculate the maximum setup time for card type *j* in line *l* such that

$$ST_{jl} = \max_{k \in K} \{ST_{jkl}\} \qquad \forall\ j, l$$

9. Calculate the total assembly time for card type *j* in line *l* such that

$$flow_{jl} = PT_{jl} + ST_{jl} \qquad \forall\ j, l$$

10. Pass all the values of $flow_{jl}$ to the network model.

Figure 5-6. Rounding Heuristic Routine (Continued)

Including the optional constraint $\sum_k X_{ijk} = 1\ \forall i, j$ in Figure 5-3 ensures that all the placements of a particular component type *i* on card type *j* are placed by only one machine in a line. This also ensures that the values of Y_{ijk} are integer in the final solution.

Optimality tolerances are specified and this results in the reduction of the number of nodes that are evaluated in the solution process. Identification of special ordered sets, specifying the branching on a variable, and using a root heuristic at the root node to obtain the first integer solution are explored to improve the solution process. The root heuristic fixes variables that are close to integral values and then proceeds with a restricted branch and bound. The models and solution approaches are built into the line assignment optimizer used in conjunction with CPLEX 6.0 on a Sun UltraSPARC2 Workstation.

5.4. SUMMARY OF SOLUTION APPROACH

The integer programming techniques employed to solve the line assignment problem have been described in this chapter. The primary technique employed is the branch-and-bound approach with linear programming relaxations. The branch-and-bound approach is enhanced through the use of several improvement techniques, which are described in this chapter. These techniques include reduction of the number of integer variables, relaxing the integrality constraints on some variables, assigning priority branching for certain variable types, activation of root heuristic to obtain the first integer solution, and use of optimality tolerances. For several industry representative data sets, case study results and computational requirements are presented in the following chapter.

CHAPTER VI

CASE STUDY RESULTS AND ANALYSIS

The models and solution approaches described in the preceding chapters provide valuable approaches for analyzing the line assignment problem. This chapter describes the application and analysis of these solution approaches. Several case studies and examples are included to illustrate the effectiveness of these methods to different production scenarios. In addition, computational requirements on these problems are addressed.

6.1. SINGLE MACHINE LINE ASSIGNMENT PROBLEMS

The single machine line assignment model and solution approaches developed in Chapter IV and V are applied to several industry representative data sets. The purpose of this case study is to illustrate the effectiveness of this model when there is only one machine in a assembly line. These data sets are used to evaluate the computer processing time required to solve these models to near optimality using optimality tolerances.

6.1.1. Case Study Description

In this case study, three data sets which range in size from 10 to 20 card types are used. The number of card types, the number of unique components, and the number of lines for each of the data set are shown in Table 6-1. Appendix B contains additional information on these data sets. The number of unique components and the number of component placements for each card type are summarized in Appendix B for data sets S1, S2, and S3.

Table 6-1. Summary of Data Sets for Single Machine Case Study

Data Set	**Unique Circuit Card Types**	**Total Unique Component Types**	**Number of Lines**
S1	20	285	4
S2	20	320	3
S3	10	220	3

These data sets are solved using CPLEX Mixed Integer Programming Software [9] on a Sun UltraSPARC2 Workstation. An optimality tolerance of 5% was used for all the data sets. The computational times required for solving these data sets are shown in Table 6-2. The assignment of circuit cards to lines and groups of circuit cards to families are summarized in Table 6-3 for data set S3. For this data set, circuit cards 5 and 7 are grouped together as a family and processed in one setup in line 1. Circuit cards 1, 2, 3, 9, and 10 are grouped as a family on line 2. Circuit card 8 is processed by itself on line 2. Circuit card 4 and 6 belong to the same family and are processed in one setup in line 3.

Table 6-2. Computational Time for Single Machine Line Assignment Problems

Data Set	**CPU Time (secs)**
S1	24032
S2	24001
S3	76

Table 6-3. Computational Results for S3 Data Set

Assembly Line	Card Groups
1	{5, 7}
2	{1, 2, 3, 9, 10} {8}
3	{4,6}

6.1.2. Summary of Single Machine Case Study Results

The results indicate that the model and the solution approaches developed in this research for addressing the assignment of circuit cards to assembly lines with single machine are effective. The data sets used for this case study are representative of industry circuit card information.

6.2. MULTIPLE MACHINE LINE ASSIGNMENT PROBLEMS

The models and solution approaches developed in Chapter IV and Chapter V for multiple machine line assignment problems are studied for different production scenarios in printed circuit card assembly. The purpose is to evaluate the effectiveness of these models when there is more than one machine in any one or all the lines. Industry representative data sets are used to demonstrate the computer processing time required for solving these models to near optimality using optimality tolerances.

6.2.1. Multiple Machine Case for Unique Setups

The multiple machine line assignment model for unique setups assigns cards to lines and allocates components to machines to balance workload. For this case study, nine assembly lines are available for processing the circuit cards. The assembly lines contain a variety of equipment including Fuji CP, Fuji IP, and Fuji QP placement machines. Lines 1, 2, and 3 have four placement machines as well as a card flipper. These lines can process both single and double sided cards. Line 4 has two placement machines, while lines 5, 6, 7, 8, and 9 have 3 placement machines respectively. The lines are operated for two shifts per day with 7.5 hours per shift for seven days in a week. The card data sets range in size from 64 card types to 88 card types. The industrial partner provided the data sets based on weekly schedules using the relational data structure presented in Appendix A. A summary of the three data sets (M1, M2, and M3) used for this case study is shown in Table 6-4. Additional information on these data sets can be found in Appendix C, which summarizes the individual characteristics of the card types in the data set.

Table 6-4. Summary of Data Sets for Multiple Machine Case With Unique Setups

Data Set	Unique Circuit Card Types	Total Unique Component Types	Number of Lines
M1	64	2520	9
M2	74	2843	9
M3	88	3200	9

The decomposed models described in Section 5.3.2 are solved for data sets M1, M2, and M3 using the Line Assignment Optimizer on a Sun UltraSPARC2 Workstation. Additional information on the Line Assignment Optimizer can be found in Appendix A. An optimality tolerance of 2% was used for all the data sets. Computational results are summarized in Table 6-5 for each of these data sets.

Table 6-5. Computational Results for Multiple Machine Case with Unique Setups

Data Set	Total Assembly Time	CPU Time (secs)
M1	399.20 hours	203
M2	433.42 hours	233
M3	589.82 hours	386

The utilization of each line for data sets M1, M2, and M3 are illustrated in Figure 6-1. Lines 1, 2, and 3 are the fastest lines and are almost filled to capacity. The remaining lines are under-utilized.

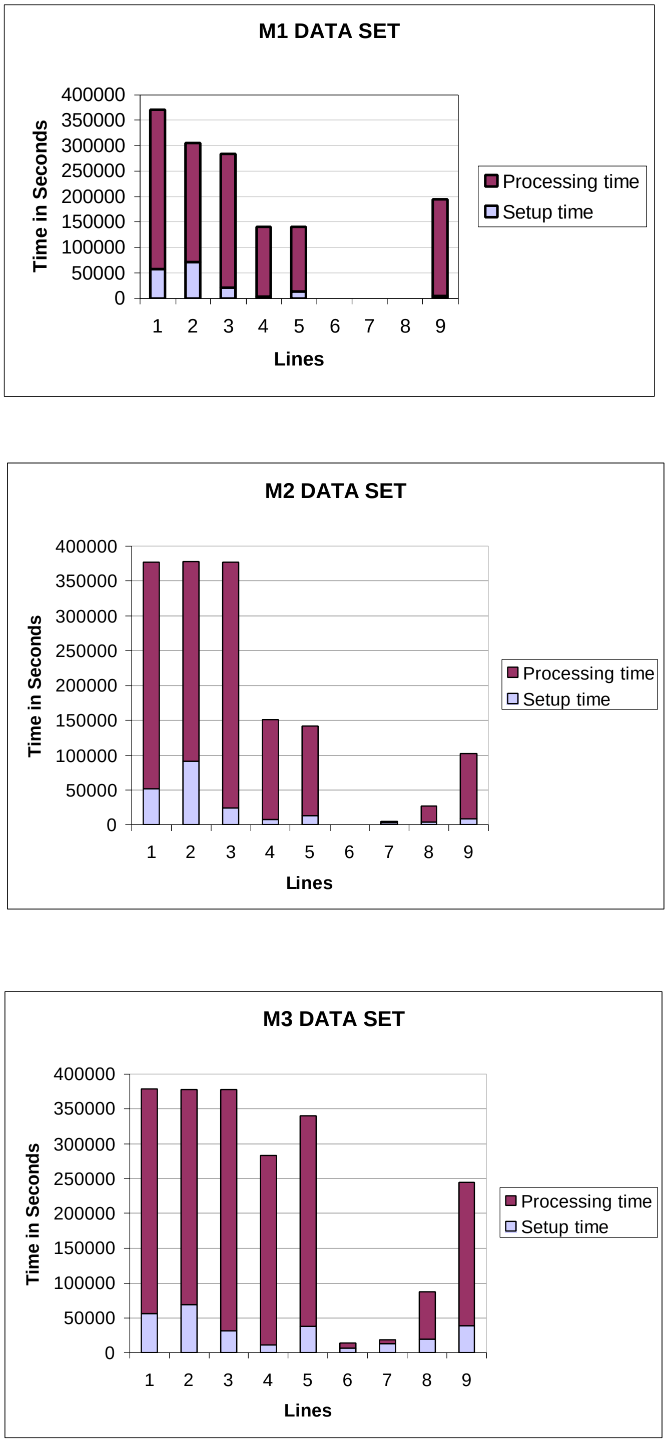

Figure 6-1. Line Utilization Results for Unique Setups

The component allocation results for two card types in data set M1 are illustrated in Figure 6-2. Card types 12 and 15 are assigned to line 1. Line 1 has four different machines, the first two machines are Fuji CPs, the third machine is a Fuji IP, and the fourth machine is a Fuji QP. Card 12 is well balanced across different machines in line 1. Card 15 has many larger components that can only be placed by the Fuji IP and Fuji QP placement machines (machines 3 and 4). Due to the types of components on this card, a completely balanced workload is not possible across all four machines. The workload for card 15, however, is well balanced across machines 3 and 4.

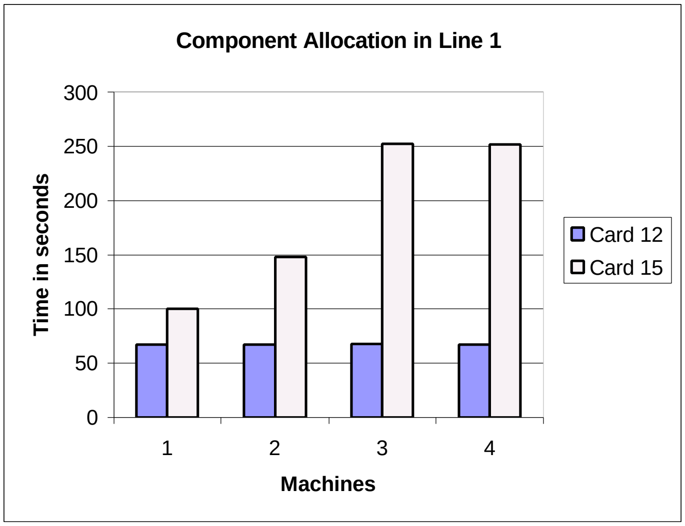

Figure 6-2. Component Allocation for Unique Setups

The multiple machine line assignment model described in Section 4.2.2 (without decomposition) was compared with the decomposed solution approach described in Section 5.3.2. The purpose of the comparison was to investigate the effectiveness of the decomposed approach. The smallest and largest data sets M1 and M3 were selected for this comparison. Both models were solved with the Y_{ijkl} variables relaxed and the optional constraint ($\sum_{k} X_{ijk} = 1 \quad \forall\, i, j, l$) was not included. CPLEX Mixed Integer Programming Software [9] was used to solve the model without decomposition on a Sun UltraSPARC2 Workstation. The problems were solved with an optimality tolerance of 2%. The assembly times and the computational time for both the data sets are summarized in Table 6-6 for both the decomposed and the full model.

For the M1 data set, the full model took 5,774 seconds to solve which was significantly longer than the 203 seconds for the decomposed model. The resulting assembly times are 385.50 hours for the full model and 399.20 hours for the decomposed model which is a difference of 3.53%. The primary reason for the difference in the assembly times is the application of rounding heuristic. For the full model, the rounding heuristic described in section 5.3.2. is applied after the model is solved. The rounding heuristic ensures that the required components for each card type are assigned to a placement machine in the assembly line. After the rounding heuristic is applied, however, the total assembly time (setup and processing time) may slightly exceed the capacity of the lines. For the decomposed model, the rounding heuristic is applied after the component allocation subproblems are solved. Again, the rounding heuristic ensures that the required components for each card type are assigned to a placement machine in the assembly line. After the rounding heursitic is applied, however, the total assembly is calculated and then used as input to the network flow model. This approach ensures that the capacity of the lines are not exceeded, but also results in slightly longer assembly times.

For the M3 data set, a solution was not found for the full model. The solution process was terminated due to lack of memory after 21,928 seconds (approximately 6 hours). This problem arises when the branch and bound tree becomes so large that insufficient memory is available to solve a linear programming subproblem. For the decomposed model, a solution was found in 386 seconds (approximately 7 minutes), thus demonstrating the effectiveness of the decomposed approach for large data sets.

Table 6-6. Comparison of Solution Approaches for Multiple Machine Line Assignment Problem with Unique Setups

Data Set	**Full Model**		**Decomposed Model**	
	Assembly Time	CPU Time **(seconds)**	Assembly Time	CPU Time **(seconds)**
M1	385.50 hrs	5774	399.20 hrs	203
M3	?	> 21928	589.82 hrs	386

6.2.2. Summary of Multiple Case Study Results

These case study results indicate that the models and the solution approaches developed in this research for addressing the multiple machine line assignment problems are effective. The decomposed solution approach ensures that good solutions are obtained in a few minutes. The models and solution approaches can be employed as decision tools by process planners and production planners under different production scenarios. The development of a solution approach for the multiple machine line assignment problem with card grouping is an area for further research.

6.3. SUMMARY OF CASE STUDY RESULTS AND ANALYSIS

The case study results demonstrate the effectiveness of the mathematical models and the solution approaches developed for single and multiple machine line assignment problems. A solution approach for making higher-level decisions such as assignment of cards to assembly lines by incorporating lower level decisions such as component allocation to balance workload is demonstrated. The models and solution approaches can be integrated and implemented as part of a decision support system for planning the assembly of printed circuit cards.

CHAPTER VII

RESEARCH SUMMARY AND CONCLUSIONS: CONTIBUTIONS, AND POSSIBLE EXTENSIONS

The uniqueness of this research and specific research contributions are presented this chapter. In addition, some observations regarding the applicability of the research and a discussion of future extensions are included.

7.1. UNIQUENESS OF THIS RESEARCH

The line assignment problem involves considering several underlying decision problems including card grouping and component allocation decision problems. Machine and line characteristics are important when making the decision of assignment of circuit cards to assembly lines. Certain card characteristics need to be considered when assigning cards to a line and component types to a machine. For example, a double sided card is processed differently than a single sided card. This research has successfully incorporated constraints that may be faced by a process planning engineer at the line level, machine level, and individual card level.

A disciplined and structured methodology and a computer decision support tool is essential to make pertinent process planning decisions in printed circuit card assembly. This research undertaking is unique, not only in development of solution approaches to handle the line assignment problem under different production scenarios, but also in its attempt to formalize the solution process from obtaining the input data and integrating the output data with a relational database.

7.2. BENEFITS TO INDUSTRY

The models and solution approach augment the existing decision process by providing near-optimal line assignments. It offers the opportunity to improve the productivity and throughput of the assembly lines by facilitating efficient use of valuable resources. The benefits to industry by the way of this research effort include:

- Automated solution approach to handle circuit card to line assignment.
- Improved line or workload balancing through efficient allocation of component types to assembly machines.
- Reduced setup time through grouping of cards to families.
- Near-optimal solutions in a matter of few minutes.
- Reduced engineering effort to generate a process plan, thereby decreasing the lead time to produce printed circuit cards.
- Flexible approach that can handle new line configurations and address varying production requirements.

7.3. SUMMARY OF RESEARCH CONTRIBUTIONS

Given its unique emphasis on the line assignment problem and integration with a relational database structure, this research adds to the body of knowledge in process planning for electronics assembly. In support of the above statement, specific and unique contributions of this research are enumerated as follows:

- *Definition and formalization of the line assignment problem.* The line assignment problem was defined and then the interrelationships among other process planning decision problems were illustrated in Chapter II.
- *Summary of literature and relationship of published literature to the current research.* The latest research conducted by researchers on different issues pertaining to assignment of cards to lines, setup management strategies, card grouping, component allocation, and feeder assignment and placement sequencing is highlighted in Chapter III.
- *Development of mathematical models.* Three different mathematical models were developed to address the line assignment problem in printed circuit card

assembly in Chapter IV. The single machine line assignment models are applicable when there is only one assembly machine in each line. The model groups circuit cards into families and assigns them to a line. The model could also be used for an initial assessment of the line assignment problem for multiple machines, with the machines aggregated as one single machine. The multiple machine models are applicable when a set of card types are to be assembled on a set of assembly lines, with a series of assembly machines in a line. Two versions of the multiple machine models are presented: one for group setups and one for unique setups. The model with card grouping included is the most comprehensive model and assigns cards to lines, groups circuit cards into families, and assigns component types to machines. The version with unique setups assumes one setup for each card type and addresses both line assignment and component allocation.

- *Development of solution approach.* The primary technique used to solve the mixed integer programming models is the branch-and-bound technique. The branch-and-bound approach is enhanced through the use of several improvement techniques. The number of integer variables are reduced during problem generation, the integrality constraints on some variables are relaxed, branching priority is assigned to certain variable types, a root heuristic is used to obtain the first integer solution, and optimality tolerances were explored to improve the solution process. These techniques are discussed for both the single and multiple machine line assignment models in Chapter V.
- *Implementation of models and solution approach using industry representative data sets.* All the models and solution approaches are implemented using CPLEX Mixed Integer Programming Software on a Sun UltraSPARC2 Workstation. The models and solution approaches are demonstrated using industry representative data sets. The case study results and computational requirements indicate good solutions are obtained in a reasonable amount of time in Chapter VI.

7.4. POSSIBLE RESEARCH EXTENSIONS

Several extensions to this research have been identified during the process of conducting this research. The single and multiple machine models could be extended to handle multiple time periods in a production schedule. The models could include labor information, test times, and inventory information to enhance the applicability of line assignment models.

An effective solution approach needs to be developed to handle multiple machine line assignment models with card grouping. The initial approach investigated in this research resulted in computational times that were prohibitive. Additional investigation of the characteristics of these classes of problems may improve the solution approach.The line assignment models developed in this research can be extended to include multisetup strategies described in Chapter III. The inclusion of this class of setup strategies may add further complexity to the problem based on the type of strategy used. In developing these models and solution procedures, the impact of lower level decision problems needs to be considered.

A hierarchical approach to production planning and control and its relationship with process planning needs to be delineated for electronics assembly systems. The literature currently published mostly does not consider the effects of lower level decision problems.

This research has demonstrated that proper consideration of line assignment decision problems could enhance productivity of printed circuit card assembly systems. Additional techniques may be developed to reduce the complexity of these models and improve the computational time required to solve these problems. The area of process planning in electronics assembly has a lot of potential to further improve the productivity of electronic assembly systems.

7.5. CONCLUSIONS

The conclusions from this research on the line assignment decision problem are summarized as follows:

1. Assigning a circuit card to an assembly line involves considering underlying decision problems including strategy selection, card grouping, card sequencing, component allocation, and feeder assignment/placement sequencing problems. The interrelationships among line assignment, card grouping, and component allocation are demonstrated in this research.
2. The line assignment problem can be modeled as a large scale mixed integer programming problem. In this research effort, models were presented for both single and multiple machine cases.
3. The branch and bound approach was effective in addressing the single machine line assignment models for single machines with card grouping and multiple machines with unique setups. Further research on the line assignment model for multiple machines with card grouping is suggested.
4. Solutions for the line assignment and component allocation problem can be obtained in a reasonable amount of time as demonstrated by case study results based on industry representative data.
5. An automated approach in the form of a computer based decision support tool needs to be used with process planning systems in electronics assembly. A decision support tool that considers the integration of input and output requirements increases the applicability of the solution approach.

This research has demonstrated that the proper consideration of line assignment decisions can improve productivity in printed circuit card assembly systems. Additional research in the area of line assignment and related process planning issues has the potential to further improve the productivity of electronic assembly systems.

REFERENCES

[1] J. Ahmadi, R. Ahmadi, H. Matsuo, and D.Tirupati, “Component Fixture Positioning/Sequencing for Printed Circuit Board Assembly with Concurrent Operations,” *Operations Research*, Vol. 43, No.3, pp. 444-457, 1995.

[2] R. H. Ahmadi, and P. Kouvelis, “Staging Problem of a Dual Delivery Pick-And-Place Machine in Printed Circuit Card Assembly,” *Operations Research*, Vol. 42, No. 1, pp. 81-91, 1994.

[3] J. C. Ammons, M. Caryle, L. Cranmer, G. DePuy, K. P. Ellis, L. F. McGinnis, C. A. Tovey, and H. Xu, “Component Allocation to Balance Workload in Printed Circuit Card Assembly Systems,” *IIE Transactions*, Vol 29, pp 265-275, 1997.

[4] J. C. Ammons, M. Caryle, L. Cranmer, G. DePuy, K. P. Ellis, L. F. McGinnis, C. A. Tovey, and H. Xu, “Computer-Aided Process Planning in Printed Circuit Card Assembly,” *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, Vol. 16, No. 4, pp. 370-376, 1993.

[5] R. G. Askin, M. Dror, and A. J. Vakharia, “ Printed Circuit Board Family Grouping and Component Allocation for a Multimachine, Open-Shop Assembly Cell,” *Naval Research Logistics*, Vol. 41, pp 587-608, 1994.

[6] R. G. Askin and C. R. Standridge, “*Modeling and Analysis of Manufacturing Systems*”, John Wiley & Sons, Inc., 1993.

[7] M. S. Bazarraa, J. J. Jarvis and H. D. Sherali, “*Linear Programming and Network Flows*”, John Wiley & Sons, Inc., 1990.

[8] G. Bhaskar, and T. T. Narendran, “ Grouping PCBs for Setup Reduction: a Maximum Spanning Tree Approach,” *International Journal of Production Research*, Vol. 34, No. 3, pp. 621-632, 1996.

[9] *CPLEX Callable Library and Mixed Integer Library*, 6.0 ed, CPLEX Optimization, Inc, 1995.

[10] Y. Crama, O. E. Flippo, J. Klundert, and F. C. R. Spieksma, " The Assembly of Printed Circuit Boards: A case with Multiple Machines and Multiple Board Types," *European Journal of Operational Research*, Vol. 98, pp. 457-472, 1997.

[11] M. S. Daskin, O. Maimon, A. Shtub, and D. Braha, "Grouping components in printed circuit card assembly with limited component staging capacity and single card setup: problem characteristics and solution procedures," International Journal of Production Research, Vol. 35, No.6, pp.1617-1638, 1997.

[12] L. Dino, and P. Marco, "Cell Formation in PCB Assembly Based on Production Quantitative Data," *European Journal of Operational Research*, Vol. 69, No. 3, pp. 312-329, 1993.

[13] K. P. Ellis, " Analysis of Setup Management Strategies In Electronic Assembly Systems, " PhD thesis in Industrial and Systems Engineering, Georgia Institute of Technology, May 1996.

[14] K. P. Ellis, " Optimization of Assignment of PCBs to Assembly Lines, " Research proposal submitted to Lucent Technologies, Columbus, Ohio, June 1997.

[15] K. P. Ellis, L. F. McGinnis, and J. C. Ammons, " Setup Management Issues in Printed Circuit Card Assembly Systems," *7th International Conference on Flexible Automation and Intelligent Manufacturing*, June 25-27, 1997.

[16] R. L. Francis, L. F. McGinnis, and J. A. White, *Facility Layout and Location: An Analytical Approach*, 2nd edition. Englewood Cliffs: Prentice-Hall, Inc., 1992.

[17] *Functional description of MINTO, a Mixed INTeger Optimizer*, Version 2.3, Georgia Institute of Technology, 1996.

[18] J. Geerdink, " Sequencing Interdependent Jobs," MS thesis in Mathematics and Computing Science, Eindhoven University of Technology, Feb 1992.

[19] H. O. Gunther, M. Gronalt, and F. Piller, "Component kitting in semi-automated printed circuit board assembly," *International Journal of Production Economics*, Vol 43, pp. 213-226, 1996.

[20] S. Hashiba, and T. C. Chang, “PCB Assembly Setup Reduction Using Group Technology,” *Computers and Industrial Engineering*, Vol 21, No. 1-4, pp. 453-457, 1991.

[21] M. S. Hillier, and M. L. Brandeau, “Cost Minimization and Workload Balancing in Printed Circuit Board Assembly,” Working Paper, Management Science Department, University of Washington, Seattle, Washington, 1996.

[22] M. S. Hillier, and M. L. Brandeau, “Optimal Component Assignment and Board Grouping in Printed Circuit Board Manufacturing,” Working Paper, Management Science Department, University of Washington, Seattle, Washington, 1996.

[23] S. Jain, M. E. Johnson, and F. Safai, “ Implementing Setup Optimization on the Shop Floor,” *OR Practice,* Vol 43, No. 6, pp. 843-851, 1996.

[24] M. E. Johnson, “Hewlett-Packard: Surface Mount Centers,” Owen Graduate School of Management, Teaching Case, Vanderbilt University, Nashville, TN, 1996.

[25] T. Leipala, and O. Nevalainen, “Optimization of the Movements of a Component Placement Machine,” *European Journal of Operational Research*, Vol. 38, pp. 167-177, 1989.

[26] O. Z. Maimon, E. M. Dar-El, and T.F.Carmon, “Set-up Saving Schemes for Printed Circuit Boards Assembly,” European Journal of Operational Research, Vol 70, pp 177-190, 1993.

[27] L. F. McGinnis, J. C. Ammons, M. Caryle, L. Cranmer, G. W. DePuy, K. P. Ellis, and C. A. Tovey, “Automated Process Planning for Circuit Card Assembly, “ *IIE Transactions*, Vol. 24, No. 4, pp. 18-30, May 1992.

[28] L. K. Moyer, and S. M. Gupta, “SMT Feeder Slot Assignment for Predetermined Component Placement Paths,” *Journal of Electronics Manufacturing*, Vol. 6, No. 3, pp. 173-192, 1996.

[29] G. L. Nemhauser and L. A. Wolsey, “*Integer and Combinatorial Optimization*”, John Wiley & Sons, Inc., 1988.

[30] D. Santos, J. Kane, F. Caballero, and K. Nagarajan, "On the Selection of Printed Circuit Board Assembly Line System," *Computers and Industrial Engineering*, Vol. 29, No. 1-4, pp. 591-595, 1995.

[31] H. Seifoddini, and M.Djassemi, "A New Grouping Measure for Evaluation of Machine-Component Matrices," *International Journal of Production Research*, Vol.34, No. 5, pp. 1179-1193, 1996.

[32] *Semiconductor manufacturing offers job with a great future*, Web source, http://www.4chipjobs.com/Great.shtml.

[33] A. Shtub and O. Maimon, "Role of Similarity Measures in PCB Grouping Procedures," *International Journal of Production Research*, Vol. 30, No. 5, pp 973-983, 1992.

[34] K. E. Stecke, "Formulation and Solution of Nonlinear Integer Production Planning Problems For Flexible Manufacturing Systems," *Management Science*, Vol. 29, No. 3, pp 273-288, Mar 1983.

[35] F. F. Suarez, M. A. Cusumano, and C. H. Fine, "An empirical study of manufacturing flexibility in printed circuit board assembly," *Operations Research*, Vol. 44, No. 1, pp. 223-243, Jan-Feb 1996.

[36] R. Jennings, *Using Access 97 Special Edition*, 2nd edition; Que Corporation, 1997.

[37] J. I. Van Zante-De Fokkert and E. H. P. Versteijnen, "A hierarchical planning approach for PCB assembly," Working Paper, Eindhoven University of Technology, Netherlands, 1997.

[38] M. R. Wilhelm and T. L. Ward, "Solving Quadratic Assignment Problems by Simulated Annealing," *IIE Transactions*, pp. 107-119, 1987.

APPENDIX A

INPUT REQUIREMENTS

RELATIONAL DATABASE DESIGN

As described in the previous chapters, the line assignment models are complex and require significant amount of data. For this research, the required data was structured using a relational database design and implemented using Microsoft Access 97 [36]. The process of relational database design and the relationships between tables is presented in this chapter. The integration between input and output data is summarized in the last section.

A.1. PROCESS OF DATABASE DESIGN

The following steps were undertaken during the process of designing this relational database model:

- Identify the entities (objects) and their attributes.
- Establish associations between objects.
- Develop the entity-relationship diagram.
- Create a preliminary data dictionary to define the tables.
- Establish the indexes based on associations between data objects.
- Design queries that create and modify data in the tables.
- Document the design of the database.

A.2. RELATIONAL DATABASE DESIGN

Entity-Relationship diagrams are used to represent the relationships between objects and to depict their behavior in the relational database design process. An entity is a unique representation of a real world object that is created by using the values of its attributes (significant property of real-world object) in computer-readable form. For the line assignment model, there are four basic entities. Table A-1, list the four entities and their primary attributes.

Table A-1. Entity Classes and Primary Attributes

Entity Class	Primary Attributes
1- LINE	Line_id
2- CARD	Card_id
3- MACHINE	Machine_id
4- COMPONENT	Component_id

An $IDEF_1$ relational data model was developed to represent relationships between different entities. A one-to-many relationship requires separate tables for each entity class and the link is provided via the primary key. A many-to-many relationship requires the creation of a new table through the concatenation of the primary keys of the original table. Primary keys are attributes, which help in uniquely identifying each record in a table. Figures A-1 and A-2 show the $IDEF_1$ representation and the relational data structure for the line assignment input model. The primary keys are shown underlined in Figure A-2.

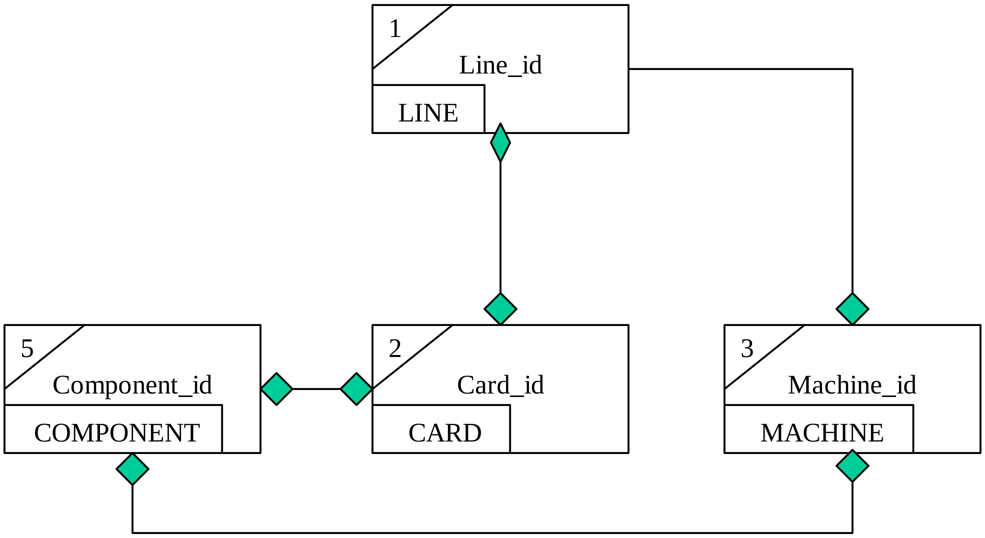

Figure A-1. Entity-Relationship Diagram

LINE(Line_id, Line_description);

MACHINE(Machine_id, Machine_description, Line_id, Total_slots);

CARD(Card_id, Card_description, Production_qty);

CARD_LINE(Card_id, Line_id, Feasible_assignment);

COMPONENT(Component_id, Component_description, Component_size);

COMPONENT_CARD(Card_id, Component_no, Component_qty, F/B)

COMPONENT_MACH(Component_id, Machine_id, Slots_used,

Time_to_place, Card_load_time, Time_to_change_feeder, Feasible_placement);

Figure A-2. Relational Data Structure

A.3. INTEGRATING INPUT AND OUTPUT DATA

The table structures were created in Microsoft Access and populated using actual industrial data. Queries are written in Structured Query Language (SQL) using the query design view of Microsoft Access 97 to compile the data in a specific format that can be used by the line assignment optimizer. Queries are generated from the tables described in the previous section. The input parameters for the line assignment model are obtained by running the queries on the tables as described in Table A-2.

Table A-2. Input for Line Assignment Model

Table	Attribute	Model Parameter
CARD	Production_qty	q_j
	Total record count	I
COMPONENT_CARD	Component_qty	d_{ij}
CARD_LINE	Feasible_assignment	b_{jl}
	Total record count	K
MACHINE	Total_slots	S_{kl}
	Card_load_time	c_{kl}
	Total record count	J
COMPONENT_MACH	Time_to_place	t_{ikl}
	Time_to_change_feeder	f_{kl}
	Slots_used	s_{ikl}
	Feasible_placement	h_{ikl}
LINE	Total record count	L

The results of the queries are automatically combined into a text file using the Visual Basic code implemented in Microsoft Access 97. This text file serves as the input data for the line assignment optimizer. The Line Assignment Optimizer interprets this text file and solves the line assignment problem. Within the line assignment optimizer decision support tool, a preprocessor reads the text file and generates the mathematical models, which are read into CPLEX. CPLEX solves the problem and generates a solution log file. A post processor written in C++ interprets this solution log file along with the original input data, and generates an output text file, which is imported back to Access 97. Several queries are used in Access to summarize the information from the Line Assignment Optimizer and generate line assignment plans for the assembly lines. Figure A-3 illustrates the entire structure of a decision support tool that can be used by process planning engineers in printed circuit card assembly. With the assistance of other researchers, this tool has been implemented and demonstrated for the industrial partner.

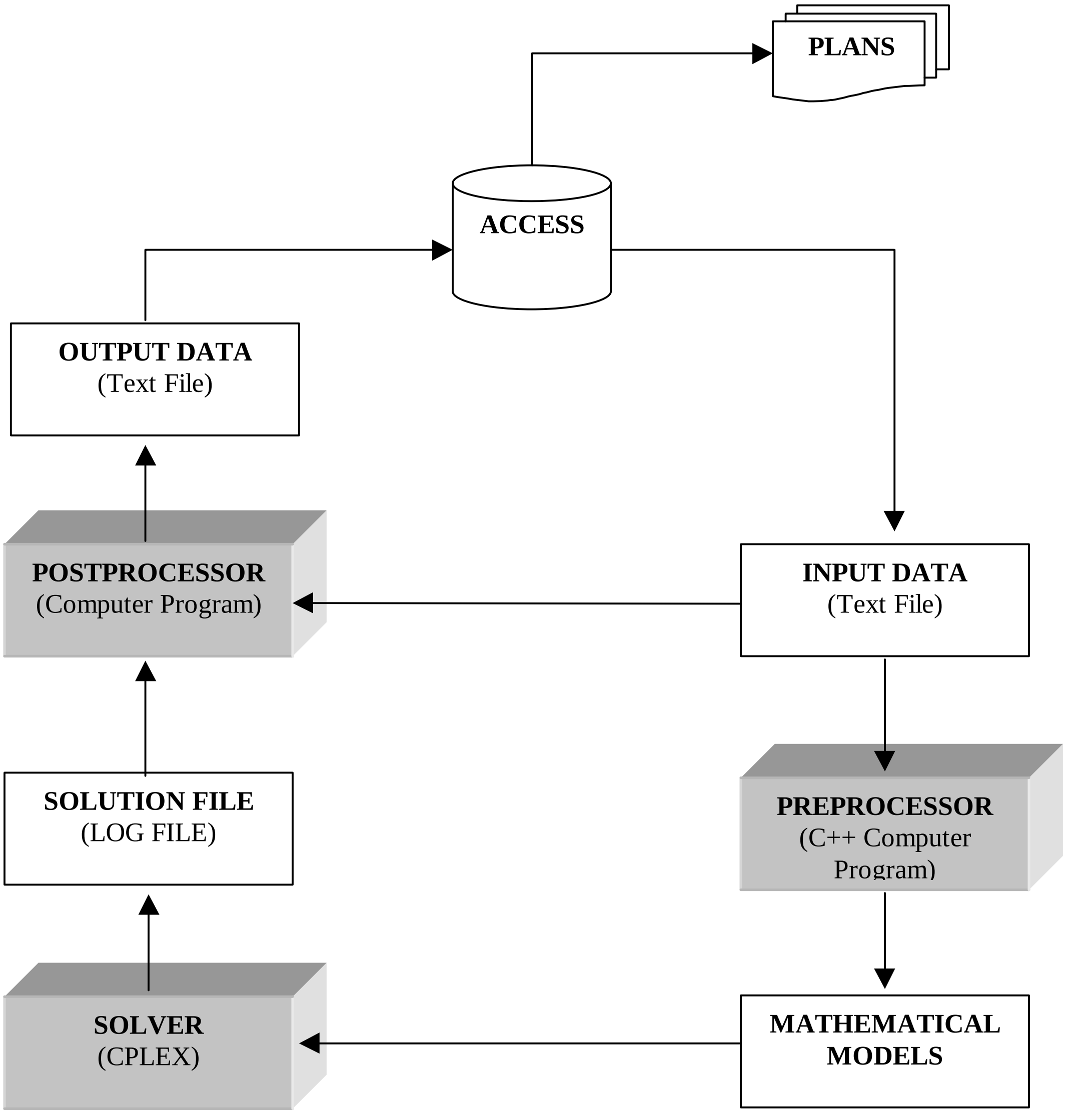

Figure A-3. Line Assignment Optimizer Decision Support Tool

APPENDIX B

DESCRIPTION OF SINGLE MACHINE DATA SETS

Table B -1. Description of Data Set S1

Card Identifier	Total Unique Components	Total Component Placements
1	16	94
2	25	70
3	17	84
4	53	296
5	20	63
6	7	97
7	8	46
8	47	111
9	6	23
10	38	252
11	19	78
12	18	75
13	13	43
14	13	43
15	9	19
16	51	312
17	37	185
18	37	80
19	26	81
20	22	102

Notes:

Total number of unique components	285
Total number of lines	4
Production Volume Range	5 to 960

Table B -2. Description of Data Set S2

Card Identifier	Total Unique Components	Total Component Placements
1	30	108
2	46	98
3	38	168
4	67	310
5	34	77
6	7	97
7	8	46
8	47	111
9	6	23
10	38	252
11	19	78
12	18	75
13	13	43
14	13	43
15	9	19
16	51	312
17	37	185
18	37	80
19	26	81
20	22	102

Notes:

Total number of unique components	320
Total number of lines	3
Production Volume Range	10 to 960

Table B -3. Description of Data Set S3

Card Identifier	Total Unique Components	Total Component Placements
1	110	198
2	88	220
3	121	352
4	88	264
5	4	45
6	22	297
7	11	132
8	11	165
9	44	44
10	22	44

Notes:

Total number of unique components	220
Total number of lines	3
Production Volume Range	5 to 74

APPENDIX C

DESCRIPTION OF MULTIPLE MACHINE DATA SETS

Table C-1. Description of Data Set M1

Card ID	Total Unique Components	Card ID	Total Unique Components	Card ID	Total Unique Components
1	44	27	44	53	61
2	77	28	12	54	40
3	49	29	101	55	45
4	86	30	70	56	80
5	91	31	119	57	138
6	9	32	197	58	76
7	51	33	107	59	92
8	4	34	44	60	99
9	24	35	43	61	109
10	20	36	26	62	162
11	24	37	74	63	67
12	13	38	81	64	72
13	90	39	50		
14	51	40	68		
15	26	41	78		
16	67	42	43		
17	45	43	55		
18	53	44	45		
19	56	45	65		
20	61	46	72		
21	11	47	61		
22	170	48	52		
23	31	49	34		
24	66	50	64		
25	67	51	43		
26	86	52	44		

Notes:

Total number of cards in schedule	64
Total number of unique components	2520
Total number of lines	9
Production volume range	3 to 3385

Table C-2. Description of Data Set M2

Card ID	Total Unique Components	Card ID	Total Unique Components	Card ID	Total Unique Components
1	44	27	29	53	35
2	77	28	11	54	45
3	49	29	97	55	65
4	86	30	96	56	72
5	11	31	148	57	61
6	9	32	204	58	52
7	51	33	31	59	34
8	4	34	66	60	64
9	4	35	67	61	43
10	24	36	86	62	44
11	20	37	44	63	61
12	24	38	12	64	40
13	13	39	101	65	45
14	90	40	70	66	80
15	51	41	85	67	138
16	61	42	144	68	76
17	62	43	76	69	92
18	26	44	44	70	99
19	76	45	43	71	109
20	76	46	26	72	162
21	67	47	74	73	67
22	45	48	81	74	72
23	59	49	50		
24	53	50	68		
25	56	51	54		
26	61	52	43		

Notes:

Total number of cards in schedule	74
Total number of unique components	2843
Total number of lines	9
Production volume range	1 to 3309

Table C-3. Description of Data Set M3

Card ID	Total Unique Components	Card ID	Total Unique Components	Card ID	Total Unique Components
1	44	32	22	63	61
2	77	33	23	64	45
3	49	34	76	65	52
4	86	35	31	66	34
5	11	36	95	67	58
6	9	37	66	68	64
7	51	38	67	69	43
8	4	39	86	70	44
9	4	40	44	71	61
10	24	41	12	72	64
11	20	42	101	73	40
12	24	43	70	74	45
13	13	44	85	75	80
14	90	45	144	76	138
15	51	46	76	77	76
16	26	47	43	78	92
17	67	48	22	79	92
18	45	49	44	80	161
19	59	50	43	81	170
20	53	51	26	82	99
21	56	52	74	83	112
22	61	53	81	84	114
23	29	54	50	85	109
24	11	55	68	86	162
25	148	56	54	87	67
26	116	57	43	88	72
27	105	58	35		
28	207	59	45		
29	204	60	40		
30	32	61	65		
31	118	62	72		

Notes:

Total number of cards in schedule	88
Total number of unique components	3200
Total number of lines	9
Production volume range	2 to 2394

VITA

Sudeer Bhoja was born on March 7, 1974 in Karnataka, India. In 1991, he began the B.S. degree in Mechanical Engineering at Regional Engineering College, Rourkela, India. Canara Bank, India offered him scholarship from May 1992 - May 1995. He graduated in May 1995 with a first class.

He started work as a full time employee with Rapsri Engineering Industries, Bangalore, India in the month of June 1995. He was responsible for all production, planning and control activities for the die casting section of the factory. He left the company to pursue graduate studies in Industrial Engineering at Virginia Tech in July 1996.

He pursued the manufacturing systems option at Virginia Tech. He was involved in doing research for Siemens AG, Germany as a graduate research assistant under Dr. W. J. Fabrycky. He also worked with Dr. Kimberly Ellis on a project for Lucent Technologies, Columbus, Ohio. In July 1998, he started work as a SAP Consultant for Integration and Networking Consultants Group in Cary, NC, USA.