AN ABSTRACT OF THE THESIS OF

Patcharaporn Neammanee for the degree of Doctor of Philosophy in Industrial Engineering on September 20, 2001.

Title: Integrated Methodology For Board Assignment And Component Allocation In Printed Circuit Board Assembly.

Abstract approved: Redacted for Privacy

Sabah U. Randhawa

The purpose of this research is to develop an approach to minimize makespan for assigning boards to production lines. Because of sequence-dependent setup issue, board assignment and component allocation have to be performed concurrently. An integrated methodology is proposed to obtain a solution of the two problems. The methodology consists of seven phases: PCB grouping, family decomposition, subfamily sequencing, *Keep Tool Needed Soonest* (KTNS), component setup determination, component allocation, and board assignment.

PCB grouping based on component similarity between boards is used to reduce the problem size. Family decomposition is used when total number of feeder slots required by a family exceeds feeder capacity. Subfamily sequencing and *Keep Tool Needed Soonest* are applied to minimize the number of component setups. Classification of setup components into standard, semi-standard, and custom setup components is performed to reduce the complexity of the component allocation problem. A component allocation algorithm is developed to balance workload across machines. Assigning board families to production lines is performed using a

modification of Longest Processing Time (LPT) rule. Assigning entire PCB families to production lines to minimize makespan is difficult to accomplish since the amount of production time for each family is very large compared to that of individual PCB lot. Splitting some subfamilies is allowed as long as this does not increase makespan. The PCB grouping, family decomposition, subfamily sequencing, *Keep Tool Needed Soonest* (KTNS), and component setup determination procedures are derived from published research results. The component allocation and board assignment are developed in this research, as well as an overall methodology to integrate the entire problem.

Data provided by published literature are employed to evaluate performance of the component allocation algorithm and the integrated methodology. To examine the applicability of the methodology, an industrial data is used with the total imbalance due to setup time and placement time of individual PCB and global makespan as the performance measures. Experimentation is conducted with simulated data based on an industry data to investigate impact of threshold value, feeder capacity, and characteristics of data sets on system performance.

Integrated Methodology For Board Assignment And Component Allocation In Printed Circuit Board Assembly

by

Patcharaporn Neammanee

A THESIS

Submitted to

Oregon State University

in partial fulfillment of

the requirements for the

degree of

Doctor of Philosophy

Presented September 20, 2001

Commencement June 2002

Doctor of Philosophy thesis of Patcharaporn Neammanee presented on September 20, 2001

APPROVED:

Redacted for Privacy

Major Professor, representing Industrial Engineering

Redacted for Privacy

Chair of Department of Industrial and Manufacturing Engineering

Redacted for Privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Patcharaporn Neammanee, Author

ACKNOWLEDGMENTS

I would like to take this opportunity to thank the Thai government and the National Institute of Development Administration (NIDA) for their generous financial support.

At this time, I would like to thank Dr. Sabah U. Randhawa, who advised and helped me not only with my thesis but also with some nonacademic matters. It was a great honor to have the opportunity to work with him. I would also like to thank Dr. Charles Brunner, Dr. Kimberly D. Douglas, Dr. David S. Birkes, and Dr. Dean H. Jensen for their help, suggestions, and encouragement with my dissertation.

I would like to express my gratitude to Dr.John Xu and Dr. Yu-an Li on how to apply this work to the real world. Their suggestions were constructive contributed tremendously to the development of the methodology. I was very grateful for having James Yutzie as a fellow gradate student. His advice was valuable to success of this work.

I also would like to thank my sister, Kittiporn Paotrakool for her kindness and assistance in taking care of both my new son and me. A word of thanks to my friends, in particular Sanchai Prayonpokarach, Richard Mattix, and Kannachai Kanlayasiri, for their friendship and kindness.

It suffices to say that nothing would be possible without the sacrifices and support of my parents, Jumnong and Wipha Paotrakool, who taught me the value of a good education and encouraged me to pursue it. I would like to extend a heartfelt thanks to all the member of my family, in particular my

husband, Bunlung Neammanee, for his devotion and unwavering support throughout the years. This thesis is lovingly dedicated both my husband and my parents.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (continued)

LIST OF APPENDICES

LIST OF APPENDIX TABLES

LIST OF APPENDIX TABLES (Continued)

INTEGRATED METHODOLOGY FOR BOARD ASSIGNMENT AND COMPONENT ALLOCATION IN PRINTED CIRCUIT BOARD ASSEMBLY

CHAPTER 1 INTRODUCTION

The importance of electronics can be seen everywhere in our daily life. The Electronics Manufacturing Service (EMS) industry reported that, between 1990 and 1998, this industry grew by 500 percent (Green, 1999). "After reaching $22.5 billion in 1998, the EMS bull market is forecasted to carry through to the next decade. In fact, a recent study predicts that the market will continue to grow 20 to 25 percent annually over the next four years. Furthermore, the North American EMS market is estimated to continue to increase by more than 20 percent per year, reaching $44.8 billion in 2001" (Green, 1999, p25).

Due to global market competition and the resulting impact on product life cycles over the last decade, the United States electronic industry switched from high volume manufacturing to low and medium lot size manufacturing. The global marketplace has forced tremendous competitive pressures on manufacturers to seek electronics products with advanced technology to meet the needs of high productivity levels, customer satisfaction, and cost effectiveness.

Crucial parts of a typical electronics device are Printed Circuit Boards (PCB). Inserting electrical and mechanical devices into PCBs is an important manufacturing process. The assembly of PC boards requires significant capital

investment and high expenditures in labor and overhead costs. The need for expertise in numerous areas including production planning, labor management, quality control, and process planning has been present. In particular, process planning is essential, since managing available resources to meet production demand is critical for continued productivity increases.

Assembly machine manufacturers usually concentrate on making their fast machines run faster and on providing powerful computer programs to drive machines in order to maximize machine performance. Machine speed may not be the controlling variable in determining the overall productivity of an assembly line. For example, in a low volume, high mix production environment, the changeover time becomes a critical issue. The changeover time may occupy more than 35 percent of total machine time (Myers, 1997).

1.1 Problem Statement

Due to continuous miniaturization of electronics products, the design and manufacturing of PCBs has become more complex. The number of components per board has dramatically increased in the last decade. This situation necessitates the need of more effective process planning. Process planning in PCB assembly deals with several issues, including: (1) selection of board types assigned to each line to fulfill production requirements and reduce makespan, (2) allocation of component

types to each machine so as to maximize utilization and reduce setups, and (3) determination of board sequencing to reduce setup times.

Assigning product types to production lines is an important issue in many facets of the electronics industry. This task must consider specific elements of each line, such as machine capacity, so as to match quantity and technical requirement of each board type. Some approaches have been developed to solve the board assignment problem with the purpose of minimizing the total annual cost with the constraints of annual demand and feeder capacity. The total annual cost includes both setup cost and processing cost. In a job order shop with a low volume and high product variety, each board type is produced to order. The expected annual demand is hard to estimate precisely and the production lot size may not be constant. Consequently, minimizing total cost throughout the planning horizon may not be appropriate. Another objective is to maximize the throughput rate for each product type. Each board type is assigned to the line that gives the highest production rate. If one line has a higher production rate than others, the products may have to wait to be produced on that line instead of going to other lines that may be available. As a result, the makespan is increased.

Another goal is to maximize the throughput rate for all products during the planning horizon. Production time reduction is an alternative to achieve this goal. Global makespan represents the completion time of all PC boards and is important in order to serve customer needs as soon as possible. In low volume, high mix production, the main portion of production time is the component setup time.

Minimizing global makespan by reducing setup time and placement time would thus lead to a high throughput rate.

Printed circuit board assembly is usually composed of parallel machine lines with sequential machines in each line. In coupled machines, the estimated production time is derived from solving the component allocation problem, which determines how component types are assigned to machines to balance the workload across machines. The known production time is information needed to allocate boards to production lines. However, production time is sequence-dependent. This means that the specific sequence of PCB types to be produced on each line and the board assignment to production lines both need to be known before production time is estimated. Since component allocation and board assignment depend on each other, the component allocation issue should be incorporated in the board assignment problem.

Consequently, the two research questions addressed in this research are:

(1) How to allocate component types on machines?

(2) How to determine which board type is processed on which line?

1.2 Research Objectives

The complexity of the problem is a major concern to concurrently answering the two research questions stated above. The purpose of this research is

to develop an integrated methodology to solve board assignment and component allocation issues. This methodology includes board grouping, family decomposition, subfamily sequencing, Keep Tool Needed Soonest (KTNS) application, setup component type determination, component allocation, and board family assignment.

To reduce the problem size associated with board assignment, board families are identified using the Group Technology (GT) concept and assigned to production lines. Family decomposition is exploited to divide families into subfamilies until they can be processed with only one machine setup. Component setup reduction between subfamilies would tend to reduce global makespan in the assignment problem; thus subfamily sequencing, incorporated with the use of KTNS application, is performed. In addition, splitting some subfamilies to be produced on another line in the family assignment procedure assists in reducing global makespan. The utilization of setup component types reduces the complexity of the component allocation problem. Then, components for each subfamily are allocated to machines so as to balance the workload across machines. In the assignment problem, each board family is distributed to a production line based on appropriate scheduling rules.

Solution algorithms for component allocation with setup component type determination has not been adequately addressed in prior research in this area. This research has developed an appropriate algorithm to solve this problem. In addition,

the performance of this algorithm and integrated methodology were investigated using published data sets from literature. The applicability of the methodology was then evaluated with both industry data and simulation data.

1.3 System of Study

The type of system studied reflects a make-to-order firm. All units of each order are generally produced in only one lot. Typically, all PCB types are ordered at the start of a planning horizon, with a specific volume and due date for each product.

This study focuses on the surface mount process. The assembly lines in this system are parallel production lines consisting of coupled machines. These machines could be concurrent or sequential. The feeder capacities of these machines are not necessarily identical. All production lines have the same machine configuration. The system of study is delineated in Figure 1.1. Given a set of board types to be produced to meet demand, each board type is assigned to a production line. A set of PCBs assigned to a line are sequenced in order to reduce component setups. To produce each PCB lot, all components are distributed among machines to minimize workload imbalance. In order to increase productivity of individual machines, arrangement of components into feeder slots and sequencing of component placements are utilized.

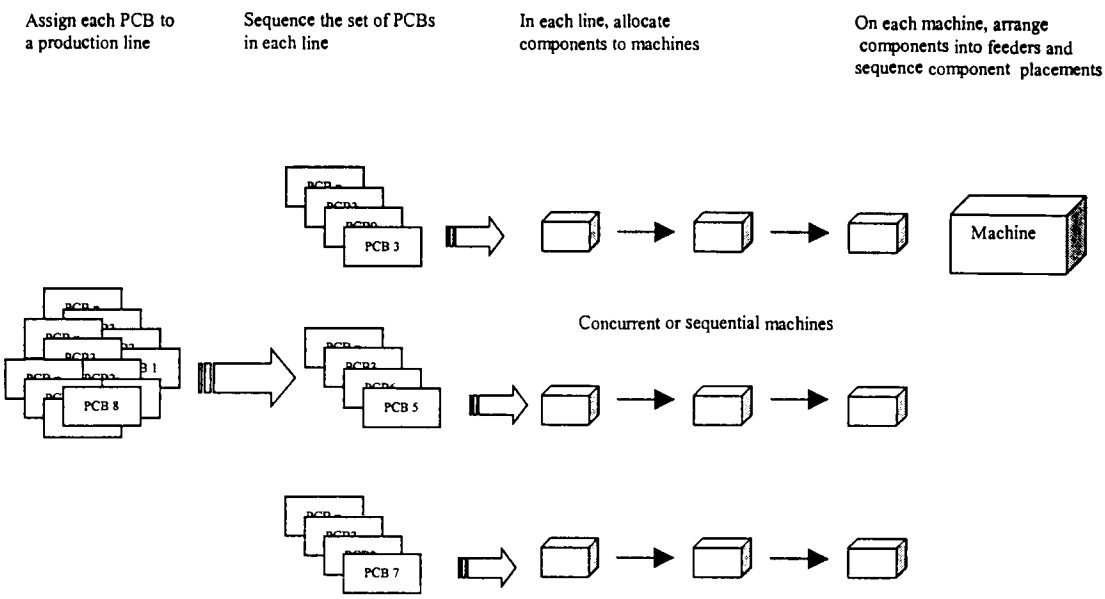


Figure 1.1 System of study

Myers (1997) explained that changeover time for an assembly line is devoted to changing component loading on each machine, loading new programs on each machine, and adjusting the width between rails of the material handling system. The loaded/unloaded component time is a major part of changeover time and can be reduced by efficient process planning.

The component changeover time can be classified into two categories. A "required setup" time is the component loading time when changing from producing one PCB lot to another. An "option setup" time is the time required to rearrange a feeder on a feeder carriage to reduce processing time.

In a low volume, high product variety environment, the setup time is a major proportion of total production time. Reducing setup time would lead to

throughput maximization (Dillon et al., 1998; Myers, 1997). The purpose of integrating board grouping with component similarity is to reduce component setup time by producing the boards that require the same components temporally next to each other. Since PCBs usually demand a large number of component types, the grouping procedure may create huge PCB families, such that all components for processing a family may not possibly be installed on the available machines at the same time. To produce all PCBs in such a family will require more than one machine setup. Practically, for this situation, components are divided into three categories: (1) *Standard* setup components are permanently staged on a machine while producing an entire family, (2) *Semi-standard* setup components can stay on a machine for producing some board types in a family, and (3) *Custom* setup components are loaded/unloaded for individual board types.

1.4 Problem Characteristics

Since several issues are involved in PCB assembly, it may not be possible to include all variations of PCB production in this research. Some assumptions need to be made to keep the problem within manageable limits:

1. Component types for a PCB family may require a number of feeder slots greater than feeder capacity of a production line. However, the number of slots for

loading all components required for every PCB type does not exceed the feeder capacity of a line.

2. A component feeder can occupy only one slot. A component also can be mounted on any slot of any machine.
3. A component type can be loaded multiple times on the same machine in a line. However, it can not be duplicated on particular machines at the same time.
4. A component type for a PCB subfamily is classified as only one type of setup component, that is, standard, semi-standard, or custom.
5. Each PCB is loaded only one time. Allowing PCBs to be loaded on machines more than one time would increase not only the setup time, but also the Work-In-Process (WIP).
6. Each PCB lot is processed on only one line. It is not a good idea to separate a lot, since this will need an incremental board setup and complicate process planning.
7. The estimated processing time for inserting each component type on each machine is constant.
8. Only changeover time for changing components loaded on machines and required setups will be considered.
9. Feeder installation on all machines in a line will be done at the same time. The setup time for each feeder type on a machine is constant.
10. Feeders and components are always available. There are sufficient quantities of components and feeders for producing all PCBs during a planning horizon.

11. The data for processing all PCBs is known before beginning the production.
12. The workload assigned to stations can be less than 100% of full capacity performance.
13. This study will not include due date consideration.

1.5 Research Contributions

Since the production time is sequence-dependent, the board assignment and component allocation problems depend on each other. Solving for optimality is computational intensive and still represents a barrier to effective solutions in today's fast paced information technology environment. This research developed a methodology for addressing the board assignment and component allocation problems. The methodology consists of procedures that aim at reducing the problem size, reducing the global makespan, and balancing the workload across machines. In order to reduce problem size for the assignment problem, board families, rather than individual boards, are assigned to production lines. To reduce global makespan, strategies such as subfamily sequencing, Keep Tool Needed Soonest (KTNS) policy, and limited splitting among subfamilies are used. The allocation of components using a three-tiered classification system, standard, semi-standard, and custom, is used to balance workload across machines.

Using this methodology, scheduling rules for sequence-independent production time are applied in board family assignment procedure while still considering the scheduling of boards with sequence-dependent production time within a family. Furthermore, the application of the integrated methodology is investigated using industry data sets.

CHAPTER 2 LITERATURE REVIEW

With continuous advancements in the development of electronic components and automatic assembly machines, the manufacturing of printed circuit board has become more complex. This has resulted in seeking better process planning approaches to increase the throughput of PCB assembly. This chapter summarizes prior research in this area, with a focus on process planning and setup strategies, board grouping, board sequencing, component allocation, and board assignment.

2.1 Overview of Printed Circuit Board Assembly

Printed circuit board (PCB) assembly is the process by which electronics components are placed on circuit boards. As the demand for PCBs and the need for reliability in smaller packages have increased dramatically, the assembly process has also changed from manual to automatic. Conventional PCBs, namely leaded through-hole boards, are epoxy laminates to which lead components are attached through holes in the boards (Haskard, 1992). Surface Mount Teachnology (SMT) was introduced to address the need to reduce material costs as well as to address issues such as chip packaging developments and higher packing densities. The SMT assembly is easier for automated production and tends to be appropriate for

high throughput or volume product environments. Another the advantage of Surface Mount Technology is that components can be placed on both sides of a board with epoxy glue applied to prevent components falling off during second soldering. However, packaging for some components may not available for SMT assembly. Many manufactures use both through-hole board and surface mount assemblies, called mixed technology (Dillon et al., 1998; Li, 1999). A flow diagram for a typical assembly using mixed technology with double-sided board (both lead and surface mount components on one side and only surface mount components on the other) is shown in Figure 2.1.

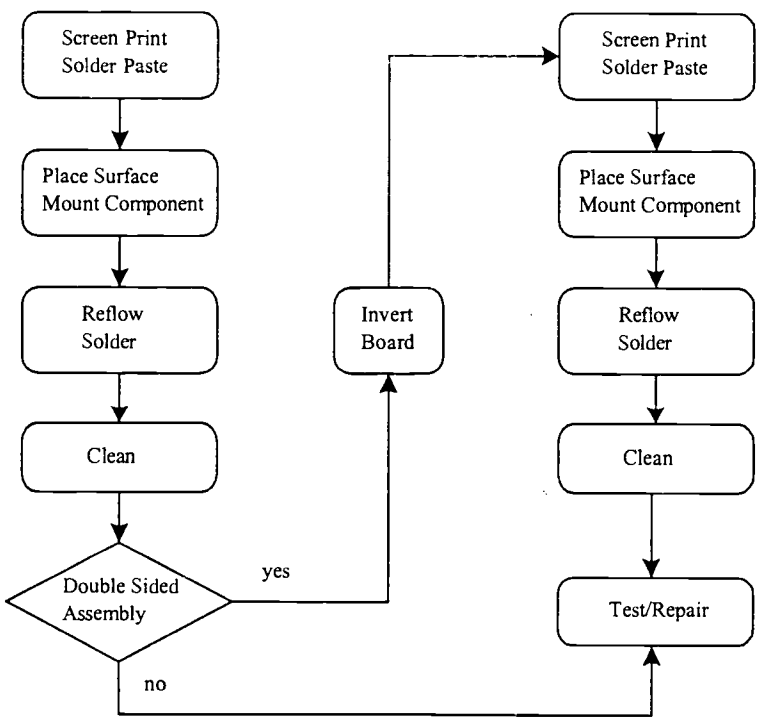


Figure 2.1 Flow diagram for assembling the mixed technology

Surface Mount Technology (SMT) is extensively used in producing PCB assemblies. The surface mount assembly process is a high-speed operation simply involving a series of steps performed by a pick-and-place machine. The primary inputs for the process are PC boards, electronic components, and assembly control programs. There are three standard types of component packaging: plastic tube, tape and reel, and stacked trays. Normally, packaging of small components is in the form of tape and

reel, which is easy to setup on a feeder carriage and come in large quantities to reduce feeder changes.

Many types of SMT machines have been used to place components on printed circuit boards. However, all of them perform the same set of basic operations (McGinnis et al., 1992):

1. Position to retrieve component from feeder
2. Retrieve component from feeder
3. Move component from feeder area to circuit board
4. Position to place component on circuit board
5. Place component on circuit board

A contiguous execution of all five operations is called a machine cycle. Each machine cycle contains only one retrieve and placement motion. SMT machines are classified into two categories based on the sequence of those operations performed.

1. Sequential machine: all five operations are performed in the sequential order. A machine cycle begins with the position to retrieve a single component and end with the placement of that component on the board. A sequential placement machine (Figure 2.2), for example, has the robot, which moves to a feeder on the carriage, retrieves a component from the feeder, move the component to appropriate location of that component, and places it on the board. Therefore, the machine cycle time depends on the placement location of the previous component, the feeder slot location of the current component on the carriage, and the placement location of current component on the board. The machine generally can be used for every type of component packaging.

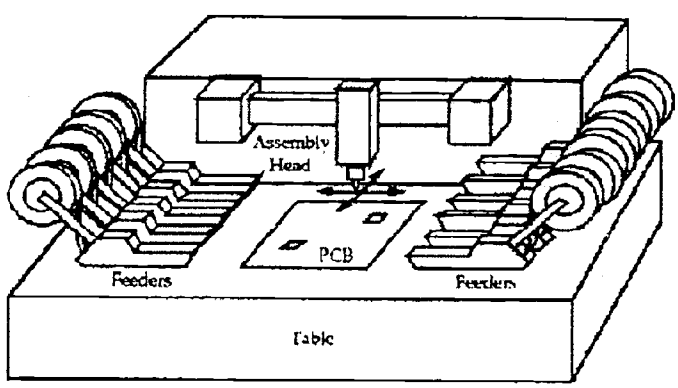


Figure 2.2 Sequential placement machine

2. Concurrent machine: two or more of the five operations may be performed simultaneously. A concurrent machine is a rotating machine (Figure 2.3), which is capable of a high speed of pick-and-place. This machine consists of three main movable parts, namely, (1) rotating pick-and-place heads, (2) an X-Y table where a PCB is positioned on a flat rectangular area and (3) a feeder carriage containing feeders used for holding various surface mount components. The movements of these three parts are independent (Ng, 1998).

The component retrieval and component placement operations are performed concurrently, as well as the feeder carriage movement. The feeder carriage moves until the current component feeder is placed at the retrieval position, while at the same time the X-Y table moves until the current component position on the board is at the placement position.

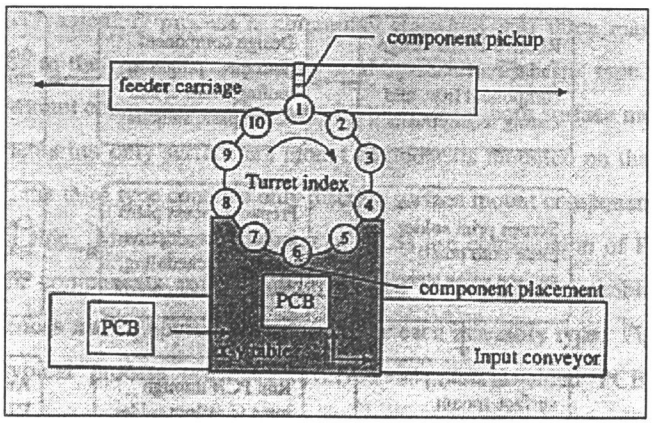


Figure 2.3 Rotating SMT machine (Li, 1999)

Typically the bulk of assembly time is taken by the pick-and-place surface mounting machines. Reducing this processing time will consequently increase the production rate (Dillon et al., 1998; Myers, 1997). Two major approaches to accomplish this objective are minimizing setup time and minimizing operation time (Leipala and Nevalainen, 1989; Ng, 1998; Sadiq, Landers, and Taylor G.D., 1993; Johri, 1990).

2.2 Board Assembly Process Planning

Process planning is particularly important to operations of a PCB assembly system because it provides the instructions necessary to convert product design, Bill Of Material (BOM) and equipment information into assembly machine instructions while seeking to accomplish the system's objectives effectively and efficiently.

In printed circuit board assembly, the functions of process planning, production planning, scheduling, and shop floor control are not purely hierarchical. The relationship between the four functions is depicted in Figure 2.4. When a set of PC boards is ordered with quantities and due dates, process planning uses a static database consisting of product design descriptions, BOM data, and equipment specification and incorporates production planning, scheduling and shop floor status to develop a process plan. This plan consists of setup and processing plans for all machines, component feeders and PCBs.

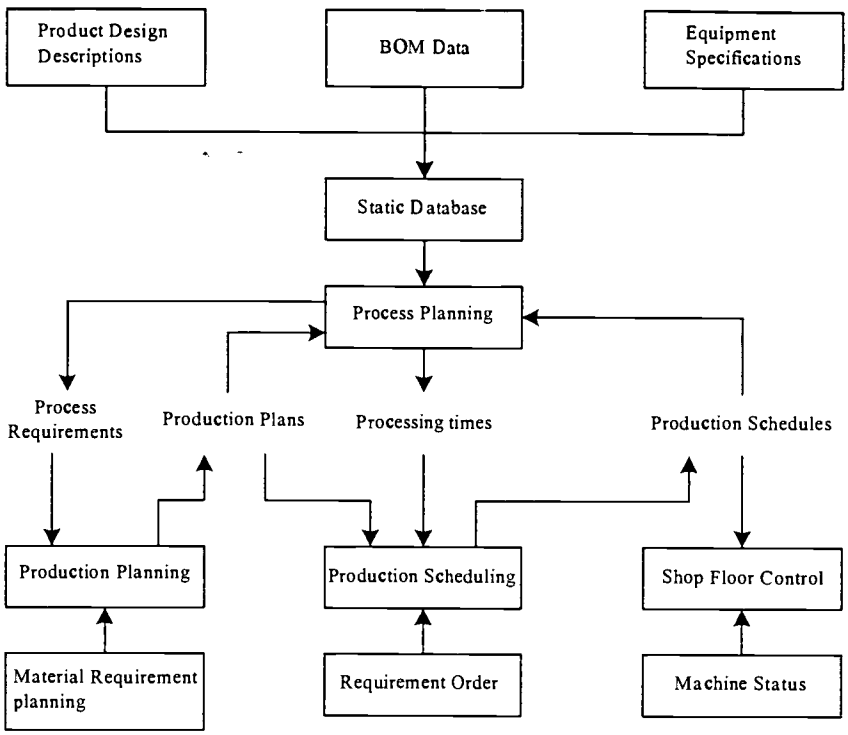


Figure 2.4 Overall Process Planning Process Diagram

Three levels of decisions in process planning that result from the interactions of the functions in Figure 2.4 are grouping, allocation, and arrangement and sequencing, as shown in Figure 2.5, (McGinnis et al., 1992). Grouping is the determination of machine groups and board families as well as assignment of families to the groups. Allocation is the distribution of components to machines within a production line. Arrangement and sequencing are the ordering of component feeders and sequencing of component placements for each machine and each board.

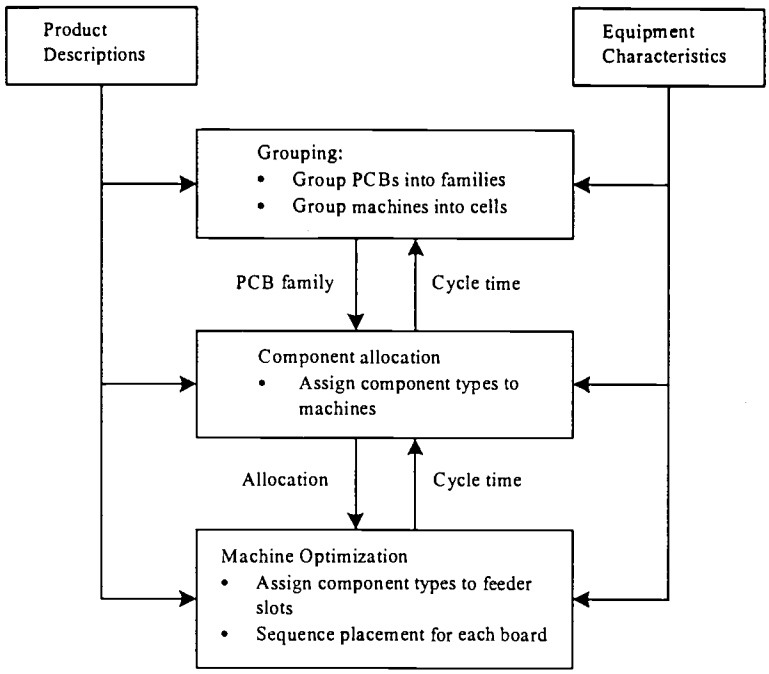


Figure 2.5 Three levels of decisions in process planning

Process planning generally involves two related issues; process optimization and setup management (Ammons, 1997). Process optimization decisions focus on component allocation for a PCB to placement machines, component arrangement on the feeder slots of each machine, and component placement sequence for each circuit board. Setup management decisions focus on determining the setup strategy for an assembly line, assigning board types to lines, grouping board types into families, grouping machines to produce a board family, and sequencing board types. The relationship among these planning decision problems is demonstrated in Figure 2.6 (Ellis, 1996).

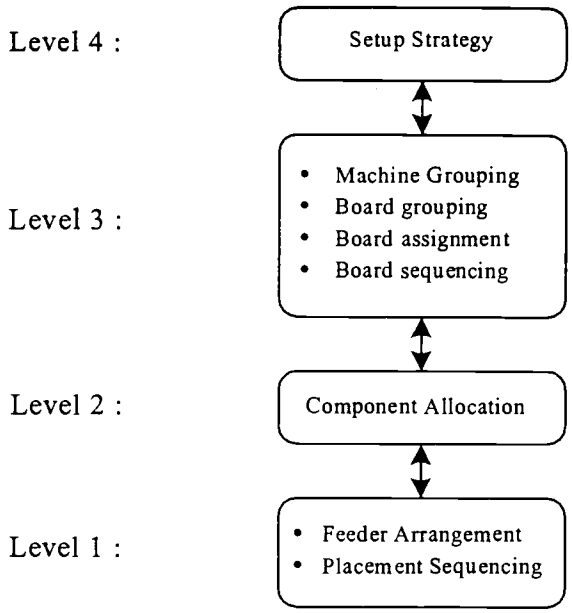


Figure 2.6 Relationship among process planning decision problems

The first level, feeder arrangement and placement sequencing problems refer to machine optimization decisions. The objective of these decisions is to reduce processing time (Bard, Clayton, and Feo, 1994; Crama., Kolen, and Oerlemans, 1990; Khoo and Ng, 1998; Kusiak, 1987; Lee, Park, Lee, Kwon, and Kwon, 1998; Leu, Wong, and Ji, 1993; Moyer and Gupta, 1997; Park and Asada, 1994). The second level, assigning component types to machines in an assembly line configuration, is to minimize unbalance workload across machines (Askin, Dror, and Vakharia, 1994; DePuy, 1995). The purpose of board grouping, board sequencing, machine grouping, and board assignment (level 3) is setup time reduction for an assembly machine or a line. The highest-level decision is setup strategy selection, which focuses on selecting the operating policy for producing

PCBs. The aim of this strategy is to minimize total production time, including setup time and processing time. Generally, the higher-level decisions impact lower level decisions.

The development of an effective process plan would result in setup and processing time reduction. However, these objectives are often in conflict. For example, a process plan developed to minimize setup time may result in a machine configuration that may not necessarily minimize processing time. The effect of this conflicting objective is typically not included in process planning decisions.

Process planning of placement machines considers both machine configuration and PCB routing between machines. There are two main routing categories for a SMT system: coupled and decoupled (DePuy, 1995). Basically, boards in coupled machines visit all the machines in order by traveling on a conveyor between the machines. Boards in decoupled machines do not necessarily visit all machines and do not have to visit the machines in the same order.

An additional consideration with multiple machines is balancing workload across machines; this determines the assignment of component types to machines minimizing imbalanced workloads. The workload consists of assembly time for the placement of components on a board and setup time for producing a board type. Workload balance focuses on minimizing the differences in workload of each board type across machines for coupled machines and minimizing the differences in workload of the all boards in families for decoupled machines. Work-In-Process inventory can be used to relieve the imbalance in decoupled machines.

Consequently, balancing the workload in decoupled machines is easier than that in coupled machines.

2.3 Setup Strategy Selection

A variety of setup strategies are currently employed by manufacturers. The selection of a setup strategy is significant as the primary strategy selection problem affects the lower level decision problems. Several classifications of setup strategies for a PCB assembly system associated with the number of setup times for each component have been developed. Maimon, Dar-El, and Carmon (1993) described three types of setup strategies. In the traditional setup method all components required by each PC type are setup on the machine before producing that PCB type. In the group setup method PCBs are produced in two stages on the assembly machines. Common components are staged on a machine and placed on all PCB types in the group. The residual components then are loaded on the machine for individual types in the same group. In the sequence dependent setup method the production of the circuit board is sequenced to take advantage of component commonality among the boards in order to reduce setup times.

McGinnis et al. (1992) classified grouping setup strategies as follows:

1. Single setup strategy: Only one machine setup for a PCB family. There are two categories within this strategy. In a unique setup strategy, a family contains only

a single product. This approach would reduce setup between adjacent families. In a family setup strategy, a family contains more than one product. This strategy may reduce the setup time by combining setup for several board types.

2. Multisetup strategy: If a feeder carriage's capacity is not enough for all components required in a board family, more than one machine setup may be required for PCB families. Thus, a board family has to be divided into a subfamily by using two strategies. In the Decompose And Sequence (DAS) strategy, a PCB family is divided into smaller sets of PC boards. The strategy then looks for components common to pairs of subsets, and sequences the subset to minimize the incremental setup between subsets. The DAS strategy requires little work in process but may need many feeder setups. The Partition And Repeat (PAR) strategy divided components in a family into subfamilies so that the number of all component types for each subset does not exceed the feeder capacity.

2.4 Board Grouping

In reality, PC board types usually have some common components. However, it may not be possible to stage many of the components on machines simultaneously. Some components may have to be loaded /unloaded many times to produce PC boards in the same line. The PC boards with similar components need

to be grouped in the same family and sequenced next to each other to reduce the component changeover times, which is an important part of the production time in a low volume/ high product mix environment.

Two interesting issues for board grouping are the criteria for grouping PCBs into a family and the size of a family. Using the Group Technology (GT) concept, several approaches for grouping have been used in Flexible Manufacturing Systems (FMS): Rank Order Cluster (ROC) (Harhalakis, Nagi, and Proth, 1990; King, 1980; King, 1982; Won and Kim, 1997), P-median model (Kusiak, 1987; Li, 1999; Wang, 1998) and Genetic Algorithm (GA) (Hwang and Sun, 1996; Jeon, Broering, Leep, Parsaei, and Wong, 1998). These methods group machines into cells such that all parts can be produced by a single cell of these machines, or group parts into families and each part is then assigned to a family. Some of those approaches have also been applied to group PCBs into a family in order to minimize the component setup time as a goal.

Hashiba and Chang (1991) assumed that the number of components required for each family is less than the feeder capacity. Thus, only one setup per family is needed. They developed a heuristic based on the GT concept with the component commonality among PCB types. Li (1999) proposed component and board grouping for multi-track feeders with the P-median method. The result showed that when the similarity value was high, the number of common components in a group would increase. Thus, by this grouping method, the component changeover time within a PCB family would be minimal.

Due to the high variety of components among PCBs, using traditional similarity measures for PCB grouping may result in a low value of the similarity index. For example, the Jaccard similarity index is defined as the ratio of the number of common components between two boards and the total number of all required components for these boards. When there are many component types among PCBs, the denominators of this index would be high compared to the numerator. Thus it would be difficult for grouping approaches, based on this component similarity measure to distinguish the difference of the index values. In order to generate the family with a large similarity value, Shtub (1992) created a global similarity measure. This measure is the sum of the Jaccard similarity index between a PCB and the others in the same family. This research then demonstrated a framework to form a PCB family with respect to feeder capacity constraints.

There are other concerns for board grouping such as due date, lot size, and the number of required component types for each board. Sanders (1996) focused on short-term production planning in a medium-volume, medium-variety printed circuit board environment, and developed a heuristic to determine a group of PCB lots to be produced in a given day. This heuristic is incorporated with similarity of both due date of PCB lots and component requirements.

Luzzatto and Perona (1993) developed a procedure for PCB grouping giving consideration to production volume and the number of components. The focus of the PCB grouping was setup minimization. The procedure was applied in the

industry case with appropriate production information, such as Bill-Of-Material (BOM), machine capacity, and quantities of PCBs.

Production of all board types in a family usually demands a large number of component types. Thus, the number of slots required for components often exceed the feeder capacity. Processing such board families needs more than one machine setup. Consequently, the feeder slot capacity is a concern in PCB grouping (Carmon, Maimon, and Dar-EL, 1989; Daskin et al., 1997; Maimon and Shtub, 1991; Maimon and Braha, 1998; Shtub, 1992).

Carmon et al. (1989) introduced the Group Set-Up (GSU) method for board grouping based on component commonality within a group to reduce setup times. GSU is performed into two steps. First, common components are staged on a machine once for assembling all PCBs in a family. Second, components are unloaded, the remaining components are loaded on the same machine, and then the assembly continues using these components. Each component is loaded at once, whereas each PCB is loaded more than one time. This method reduces the component setups but it may increase work-in-process. If due date is a consideration, excessive tardiness may result.

Using the GT concept based on component similarities, PCB families are usually large. There have been some attempts to decompose a PCB family into subfamilies such that the feeder capacity is sufficient for producing each subfamily with one machine setup. For this reason, the decision of selecting a setup strategy would impact the classification of subfamilies. With the DAS strategy (McGinnis et

al., 1992), each PCB in a family will be placed on a machine at once, whereas a component may be loaded more than one time per family. On the other hand, for the PAR strategy, each PCB will be placed on a machine more than once, but components are loaded only once. Askin et al. (1994) presented a four-step heuristic approach: Grouping boards into production families, dividing families into subfamilies with similar processing time, allocating component types to placement machines, and then scheduling the subfamilies. However, in low volume, high product variety environment, instead of processing time, the setup time would be a focus of PCB family disintegration.

Maimon and Shtub (1991) applied the Rank Order Clustering (ROC) method to identify PCB groups. Because of the feeder capacity constraint, they divided a PCB group into subgroups based on the group's components such that the number of component types did not exceed the feeder capacity. With the PAR strategy, a nonlinear, mixed-integer program was formulated to minimize the total PCB and component setup times.

Bhaskar and Narendran (1996) defined the cosine similarity coefficient (CSC), as the similarity measure using the cosine of the angle between the pair of vectors representing two PCBs. Based on the Maximum Spanning Tree (MST) approach, they developed a two-stage heuristic to solve the resulting problem. The heuristic consisted of minimizing the number of groups and then splitting a PCB when savings in component setup time is more than the increase in board setup time.

Daskin et al. (1997) proposed component grouping to minimize the total cost of loading components and PCBs. With the feeder capacity constraint, a component and a board for each family could not be loaded only one time. They proposed Mixed Integer Programming (MIP) for both PAR and DAS strategies and developed a branch and bound algorithm for solving this problem.

Johri (1990) introduced a scheduling to sequence PCBs lots with a combination of three objectives including meeting weekly due date, balancing workload among workstations, and minimizing the number of feeder setups. Part of the methodology deals with grouping boards. PCB lots with the same due date are classified into the same groups. These groups are then sequenced with increasing order of due date.

2.5 Board Sequencing

In low volume, high product mixed production, not only is grouping of PCBs into families important, but the sequencing of PCBs needs to be identified so that the number of incremental component setups between the two consecutive PCBs is as small as possible (Garetti, Pozzetti, and Tavecchio, 1995; Gronalt and Zeller, 2000; Li, 1999; Palm, 1996). In a medium-volume, medium-variety printed circuit board environment, the aim of production planning is to maximize throughput (Sanders, 1996). Given the fluctuated product demand levels and due

dates for multiple products, the sequencing model determined the order of products and their quantities to be produced in each day with the assumption that workload balancing problem has already been executed.

Hashiba and Chang (1991) applied the Travelling Salesman Problem (TSP) algorithm to reduce the number of component setups in the product sequencing problem. Using the number of setup changes as the distance measure, the number of setup changes depends on setups for current board type and setups for the previous board type. Thus, the problem is not exactly a TSP.

A scheduling approach suggested by Maimon et al. (1993) is the Sequential Dependent Scheduling (SDS). The SDS allows components required by the consecutive boards to stay on feeders during production of these boards. To reduce the number of setups, this approach applied a common component concept, that is, a board type with maximum number of common components with the board currently being produced is scheduled for production.

Garetti et al. (1995) proposed a production schedule to minimize makespan by minimizating setup time and machine idle time. Part of this study involved determination of a sequence of board families. Two alternatives for sequencing goals were (1) minimization of the operation time and (2) minimization of system setup time. With the operation time objective, the aim is to minimize total production (processing and setup time) on the bottleneck machine for all families. The total production represents the global makespan. The processing time for a machine was the average processing time on all machines for each PCB. With

minimization of the system setup time as an objective, the focus is setup time minimization on a bottleneck machine for all families.

Chen and Dong (1999) applied neural networks to solve PCB scheduling. The Nonlinear Integer Programming model was developed with the objective of minimizing total setup cost. This study concerned sequence-dependent setup time, PCB splitting and due date. An example in this study showed that the neural network method could find the optimal solution to PCB scheduling.

Numerous studies applied existing tools, such as Keep Tool Needed Soonest (KTNS) approach introduced by Tang and Denardo (1988), to minimize the number of setups with respect to feeder capacity constraints. Li (1999) presented component and board grouping for multi-track feeders based on the similarity value and Group Technology (GT) concept. He also developed an approach for intra-group and inter-group sequencing using a similarity measure and the KTNS procedure. The KTNS is useful in reducing component setups. For example, the current PCB does not need the loaded components, but they are the "soonest" components requested by the following PCB. If the present PC board required feeder slots less than the feeder capacity, these components are allowed to stay in the feeder carriage during production of that PC board.

Rajkumar and Narendran (1998) introduced an algorithm using Group Technology concept for the PCB sequencing problem with the purpose of feeder setup minimization. Jaccard's similarity coefficient is used to select the first PCB in the sequence. The following PCBs is determined by PCB index, which are the total

number of component types required by a PCB divided by the number of extra components to be mounted to produce this PCB. In addition, this algorithm also includes the KTNS policy and allows splitting a PCB if it can reduce the total number of setup times.

Barnea and Sipper (1993) developed a heuristic approach to reduce setup time in PC board assembly. The proposed heuristic is composed of two algorithms: a sequence algorithm, and a mix algorithm. In the sequence algorithm PCB types are selected based on similarity, variability, and ratio indices. The similarity index is the number of common components between the current and the next board types. The variability index is the number of different components between the current and the next board types. The ratio index is the ratio of similarity index to the total number of different components of the board types. The decision criteria are to maximize the similarity index, minimize the variability index, and maximize the ratio index. The two basic rules for the mix algorithm are a loading rule and a replacement rule. A new component is loaded on a feeder when the next board type being processed requires it. If the component must be loaded, the components that are kept on feeder slots are those needed soonest, which is the KTNS policy.

Gronalt et al. (1997) proposed a set-up heuristic based on the "Keep Component Needed Soonest" (KCNS) procedure with different feeder width considerations to reduce the number of component setups. This heuristic established the component types needed for a given job that are currently not

mounted, depending on the capacity requirement, and then applied the KCNS procedure to dismount feeders until all additional component types can be set up.

Maimon and Braha (1998) proposed a Genetic Algorithm (GA) with the KTNS for scheduling PCBs on a single machine with the objective of component setup minimization. Since the total number of component switches depends on a PCB sequence and the component types already stayed on the feeder carriage before production of each PCB, this research determined the savings in the total number of component switches. By using the KTNS policy, the number of switches is reduced.

Bean (1994) applied random keys in a Genetic Algorithm for solving sequencing jobs for scheduling problems with the objective of minimizing the total tardiness. Since the traditional crossover process with permutation encoding may create a new invalid offspring, the approach was based on a crossover operation on random keys instead of permutation genes. In the example problem, the GA could perform very well on this sequencing problem and was robust with respect to random seeds.

Rubin and Ragatz (1995) focused on the use of a genetic search to solve a sequencing problem. This study determined a sequence of a set of n jobs on a single machine. With the assumption of job setup times being sequence dependent, the purpose of this teachnique is to minimize the total tardiness for a set of jobs in a single stage process.

Iyengar (1999) attempted to reduce the setup time for the PCB sequencing problem using The GT and GA approaches. Using the GT approach, PC boards are

classified into groups based on component similarity. Production sequences are then developed for groups. Therefore, PCBs are sequenced such that a job requires the same component types, eliminating much of the setup between them. For a large problem size, it is difficult to manipulate PCB- to-component incidence matrices to develop a grouping strategy and board sequencing. Thus, a heuristic approach based on the Genetic Algorithm to find optimal assembly sequence of PCBs within a batch was created. The results show that, with a large problem, the GA approach performs better than GT. Additionally, the GA approach is relatively easily applicable to single and multiple line production environments.

Some of the methods for sequencing products in FMS may be applicable in the electronics industry. Iakovou (1992) focused a part-sequencing problem on a single machine to minimize the total setups (the number of tool switches) and on multiple machines to minimize the total setups and maximize workload balance in FMS. Due to the complexity of the computational result, the problem was decomposed into two separated subproblems, sequencing and loading tools. An optimal part sequence was computed. Determination of loaded tools is accomplished by the use of the KTNS policy so that the number of tool switches is minimal.

Logendran (1990) introduced a part sequencing solution algorithm based on total moves, including total intercell and intracell moves in cellular manufacturing. In the basic idea in cellular manufacturing, the parts that required similar processes are grouped into the same part families, and machines that meet parts' requirement

are grouped into machine cells. Since the intracell move is less important than intercell move, the total moves is the sum of both intercell moves and intracell moves weighted by the importance placed on them.

2.6 Component Allocation

The component allocation problem involves determining the placement of component types on machines for producing a board family. In the case of a single machine, component allocation determines the components to be loaded on the machine to reduce setups between board families for a single setup strategy and within families for multiple setup strategy. Hashiba and Chang (1991) formulated an Integer Programming (IP) formulation with the purpose of component setup minimization for allocating components to a machine. Because of the large problem size, a heuristic was created to generate a component assignment matrix.

In order to avoid one machine becoming a bottleneck and slowing down the entire line, the time for a PCB spent on each machine should be approximately equal in a coupled machine system. On the other hand, in a decoupled machine system, the time for each machine used for producing all PCBs should be almost the same amount. Consequently, the aims of the component allocation in decoupled machines are setup time minimization and workload balance maximization among machines. As previously mentioned, the component setup time for a PCB type can

be reduced if it is produced next to the one that requires the same components. Therefore, PCB setup time should be regarded as a "sequence-dependent setup time." With concern for sequence-dependent setups, maximizing workload balance across machines would lead to maximizing the throughput rate.

Askin et al. (1994) allocated components on various decoupled machines to minimize makespan and reduce mean flow time. An IP model was developed with the objective of minimizing a combination of maximum processing time imbalance within a group and maximum machine workload. They developed three heuristics for component assignment and group formation. The first heuristic grouped boards with similarity of processing time and then used the Longest Processing Time (LPT) rule to order PCBs within a family. The second heuristic sequenced PCBs in increasing order of processing time on all the machines' Short Processing Time (SPT) rule; whenever a machine was available, the first PC board would be processed. Unlike the first heuristic, this approach did not group boards into families. In the third heuristic, all PCBs were grouped into subfamilies based on total board dissimilarities at the component level and then components were assigned to machines with regard to an overall machine workload balancing constraint. The dissimilarity between two boards is the sum across all component types of the absolute difference in mounting times for the two boards divided by the maximum of mounting time between these boards. Thus, boards with identical component requirements have a dissimilarity of zero and boards with no common components have the dissimilarity equal to a number of components. The result

showed that the second approach gave the lowest makespan. The best mean flow time came from the first and the second heuristics. Because the third approach generated board subfamilies and then assigned components to subfamilies, it is possible to create unequal loading across machines. With the assumption of a fixed production time for each PCB lot, minimum makespan would be achieved when all machines carry relatively equal loads. However, this research did not consider the component setup time.

Gunther, Gronalt, and Piller (1996) developed a heuristic to allocate components among identical assembly machines so that the number of assembly stations is minimal. This is analogous to minimize total operation time including setup and processing time. The proposed heuristics applied a combination of typical rules to select available assembly machines and to schedule jobs. The four basic steps of heuristic procedure are: Initialization, where the number of required machines are calculated, job selection, machine selection, and capacity allocation. This study showed that the highest time component ratio of a job, the ratio of processing time required to the number of different component types to be assembled, is the best rule for selecting jobs.

Gronalt and Zeller (2000) proposed an approach to minimize makespan by solving component allocation and combining it with PCB sequencing and feeder assignment problems for two decoupled machines. This research divided the problem into three sub-problems: component allocation, PCB sequencing, and feeder assignment. In the allocation problem, makespan was minimized by

balancing the workload between machines. The workload in this study included setup time and processing time. The PCB sequencing minimized the setup time. Feeder assignment minimized the processing time. To balance workload, the authors assumed that a component required only one slot and each component for producing a PCB could not be loaded on more than one machine. An Integer Programming model was formulated with the objective of minimizing the sum of the workload differences of the PCBs and the number of component setup on both machines. Two heuristic approaches for component allocation, Experimental Component Splitting (CS) and Exact Component Splitting (ECS) were evaluated. The ECS approach allowed component splitting (one component type can be staged on more than one machine), which led to additional setup time. However, it guarantees a decrease in the total throughput time. Although the criterion of component splitting in the CS approach is an optimal splitting factor, it does not guarantee the shortest makespan.

In a coupled machine system, the goal of the component allocation problem is not only to minimize the setup time but also to maximize the workload balance for a board across machines. Ben-Arieh and Dror (1990) addressed component allocation on two-coupled machines in order to maximize the production rate. The objective function of the Integer Programming model was to balance the processing time on two machines. A heuristics based on a Longest Processing Time (LPT) rule for makespan minimization was established by allowing a component to be loaded more than once. The component setup time was not taken into consideration.

Hayrinen et al. (2000) proposed scheduling algorithms for the Generalized Flexible Flow Line (GFFL) problem. In this study environment, each PC board visits in the same order but can skip some machines. The proposed scheduling algorithms can be classified into two major phases: machine allocation and sequencing phases. The algorithm also consists of four steps: (1) initial allocation of batches to machines, (2) improvement of machine allocation, (3) initial sequencing of batches, and (4) improvement of the batch sequence. Initial batch allocation to machines is to determine the initial allocation of board types to the machines. Three algorithms, depending on the criteria of allocation, are allocations by batches, families, and in random order. For allocation by batches, the objectives in the allocation problem is to minimize the imbalance workload including setup and processing time by trying to assign the entire family to the same machine. For allocation by families, a family is assigned to the machine with the lowest workload. This algorithm will not allow disintegrating any family. For random allocation, the workload and family issues are not concerns.

Ammons (1997) and DePuy (1995) presented an integer programming model and two heuristic models of component allocation to balance a combination of the assembly time and the component setup time for each PCB type. The machines in the study are not necessarily identical. Watkins and Cochran (1995) demonstrated a heuristic to balance a line with tradeoffs between savings from a better balanced line and the associated relocation cost. Some components from a bottleneck machine are moved to non-bottleneck machines without creating a new

bottleneck machine. This process continues until savings from reducing cycle times are exceeded by costs to relocate components. In order to achieve an easier workload balance, each component type is allowed to be inserted into more than one machine (Ammons, 1997; Ben-Arieh and Dror, 1990; DePuy, 1995).

In reality, component allocation problems are related to setup strategies. If a PCB family for a multi-setup strategy is large, it may not be possible to allocate all component types on all the machines at the same time. However, using a multiple setup strategy for component allocation may be useful since it allows some common components, used on several different board types in the family, to be staged on the feeder carriage throughout production. For single setup strategy, all components required by a PCB family are located on a machine only once. For a multiple setup strategy, some components have to be loaded/unloaded more than one time because of limited feeder capacity.

Ammons (1997) addressed a mixed integer programming model of the component allocation problem to coupled machines for a unique setup and family setup strategy, as well as a variation of multiple setup strategy. For a multiple setup strategy, setup component types are classified into two types of setup components: common components for all PCBs in the family called *standard* setup components and remaining components temporarily loaded on the feeder for one or a few board types called *custom* setup components. Standard setup components are loaded only one time for each board family, whereas custom setup components may be loaded more than once. Therefore, the model is concerned only with setup time of custom

setup components and component placement time. However, many components are common components for some boards but not for all. Allowing these components to be staged on a machine for producing some board types would reduce setup times.

Like Ammons' (1997) study, Smed, Johnsson, Puranen, Leipala, and Nevalainen, (1999) proposed a system to arrange operations in PCB assembly line with a wide range of different products. The variety of product types usually causes frequent setup operations. Thus, the purpose of this research is to minimize the number of setups. In this research, component feeders are classified into two categories: standard set-up, and custom setup components.

Xu et al. (1999) used an algorithm for feeder bay determination by using benefit-cost analysis. This approach considered benefit of adding an additional component versus the cost of unloaded components from feeder bay because of the feeder's capacity constraint. They also divided the feeder bays into three categories: fixed, semi-fixed, and configurable feeder bays. A component in a fixed feeder bay was used for producing all PCBs. A component in the semi-fixed bay would be installed for the production of a PCB family. Finally, components in the configurable bay, the remaining set of components, varied from one board to the next.

Since a main objective of component allocation is to balance machine workload across machines, accuracy of estimated processing time is significant. In theory, the lowest processing time can be determined by the machine optimization

problem that includes assigning components to feeders and sequencing component placement. Solving the component allocation problem along with the machine optimization problem is very complicated. Previous studies have not included machine optimization considerations in the component allocation model. Furthermore, these studies have used placement time for each component to approximate processing time. For example, Ammons (1997) used estimated placement time for each component type without considering its location on a feeder carriage or placement sequence of that component. Askin et al. (1994) calculated total processing time of a PCB on a machine as being equal to the sum of processing time for all required components placed on that machine.

DePuy (1995) computed the placement time by considering the latency of boards and feeders on concurrent machines instead of machine optimization. Board latency refers to the X-Y table on which a board is positioned. The table moves slower than the head and the feeder. Like board latency, feeder latency means that the feeder movement is slower than that of the head. This approach cannot give an optimal processing time but an estimate of processing time can be improved.

2.7 Board Assignment

As addressed earlier, process planners have to make decisions about board grouping, board sequencing, machine grouping, and board assignments to lines. In

practice, machines are already arranged on the shop floor. When products are ordered with a due dates and quantities, these products are assigned to a line, and then product groupings as well as sequencing routes are executed. Technically, products are assigned to lines according to technological and demand requirements.

Zerangue (1999) created a mathematical model for assigning and scheduling boards on assembly lines. The objective function is to minimize the maximum workload assigned to any machine on any line. Maimon and Shtub (1991) developed a mathematical model to assign a set of boards to a line by using a component similarity measure to deduce changeover times. This approach does not consider other variables such as the quantity of each PCB type, as for example, when two similar boards, one with high volume but the other with low volume, are assigned to the same line. Capps (1997) and Hillier and Brandeau (1998) were concerned with the quantity of each board type and machine capacity. With the objective of minimizing annual setup and manufacturing cost, Capps (1997) presented an Integer Programming (IP) model for assigning boards to lines. Hillier and Brandeau (1998) also introduced an IP model for assigning boards to automatic lines and to a manual line. These studies assume annual demand of each PCB type is known and the lot size produced is constant. Seifoddini and Djassem (1996) introduced a Quality Index (QI), incorporated production volume and processing time. This index is the ratio of intercellular workload to the total plant's workload. Although QI measure could determine the performance of a cellular manufacturing system, it is not concerned with setup time for a machine configuration.

Peters and Subramanian (1996) proposed the strategies to assign PCBs to identical-parallel production lines so that makespan is minimal. With the assumption of a sequence-independent setup time as well as total production of a PCB lot fixed and known in advance, the Longest Processing Time (LPT) rule was applied to create a heuristic for a PCB assignment problem. With assumption of sequence-dependent setup time, a Multi Travelling Salesmen Problem (MTSP) heuristic, which involved assigning PCBs to lines and then sequencing PCBs within a line, was generated.

Since setup time is sequence-dependent, the total production time for a PCB lot depends on the order in which boards are processed. Thus, workload balance is not necessarily aimed at minimizing the global makespan for parallel machines. Garetti et al. (1995) attempted to develop a production schedule to minimize makespan by reducing component setup time and machine idle time. By assuming specific routing of each PC board on only one production line, the methodology can be divided into two steps—generating PCB families, namely No Setup (NS) mixes with unique-family strategy, and sequencing NS mixes. A NS mix is a PCB group created to minimize setups. In order to generate NS mixes, two objectives are considered, an optimal balance and setup minimization.

1. Optimal balance. The load patterns of the NS mixes must be similar to that of the entire planning period. Therefore, each NS mix would be similar.
2. Setup minimization with the similarity between the load pattern of NS mixes. The number of setups for moving from one NS mix to the next will be lower.

As mentioned previously, the two primary objectives of a sequencing problem are minimizing operation time and system setup time. Garetti et al.'s (1995) research investigated the four combinations of the two-step objectives. With setup time minimization of board grouping and sequencing, the system setup time was smallest. On the other hand, with setup time minimization for board grouping and minimization of maximum operation time for board sequencing, the maximum operating time was shortest.

Rajkumar and Narendran (1997) concentrated on loading PCBs to identical parallel machines (one machine per line) with the objective of minimizing the makespan and balancing the number of component setups on the machines. They were considered in a low-volume, high-mix environment. Two concepts employ to reduce global makespan. First, with assigning an equal distribution of PCBs among machines, the setup time on all machines would be approximately the same amount. Since the significant proportion of production is devoted to setup time, the maximum workload balance can be accomplished. Second, minimizing the number of setups would give the smallest makespan. They also indicated that the number of available machines could determine the number of groups. Therefore, their two-step heuristic includes a board assignment and component allocation. A ratio of the product of Jaccard's similarity between a PCB and others with respect to the number of component types for that PCB, namely a seed index, was defined. The use of this index is to allot PC boards to machines. In addition, the KTNS procedure was applied to assign components to a machine.

Prior studies primarily concentrated on parallel machines or sequential machines. Assigning PCBs to the parallel machines, consisting of one machine in each production line, can be considered as a board assignment issue. Allotting components to sequential machines is a component allocation issue. Production situations with parallel machine lines each consisting of sequential machines, typically from several manufacturers, have rarely been considered. Garetti et al. (1995) defined their system as both parallel and sequential machines. However, they assumed that each board was already assigned to a specific line. Thus, there was no consideration of the impact of board assignment. In practice, parallel production lines usually consist of more than one pick and place machine. Numerous components can be processed on those machines. Consequently, considering both issues in the decision making process would enhance the usefulness of the result.

2.8 Integrated Methodology

Because of the complexity of the overall problem, optimality can not be accomplished. Numerous heuristics have been developed to solve such a problem with reasonable CPU time. Developed heuristics for these complex problems are often separated into sub-problems that are hieratically solved.

Crama, et al. (1990) proposed a heuristic approach to optimize throughput rate of a production line of several machines. The approach includes (1) a component assignment to machines, (2) for each machine, assignment of feeders to feeder slots, (3) for each machine, a sequence of pick-and-place rounds, and (4) for each machine and each pick-and-place round, an assignment of nozzles to heads. This study developed the mathematical model for each sub-problem. Since each subproblem is an NP-hard problem, the solution approach was posed.

Xu et al. (1999) presented an integrated methodology for PCB assembly machine configuration. This methodology involves feeder bay distribution among components, component-to-feeder assignment, and machine program generation. This research categorizes feeder into three types-fixed, semi-fixed, and configurable-- by the use of Benefit-Cost analysis. The objective of determining feeder types is to minimize setup time.

Dessouky, Adiga, and Park, (1995) integrated the design of PCB assembly with scheduling methodology to maximize the throughput of the system and minimize work-in-process inventory. The purpose of the design is to control and schedule flow line, which reduce the sequencing problem in scheduling phase.

To obtain a solution of such a problem, these heuristic approaches are applied to solve a sequence of sub-problems. It is important to note that the component allocation problem and board assignment problem are related. In developing an effective solution. Both these problems should be considered at the same time.

CHAPTER 3 RESEARCH METHODOLOGY

This research attempts to answer the two questions: how to allocate component types on machines and how to assign board types to be produced on production lines? As previously mentioned, the component allocation and board assignment problems depend on each other. It is complicated to concurrently solve these two problems for optimality. However, there is a need for a solution approach to obtain a good solution for the aggregate problem.

In this chapter, mathematical models for component allocation and board assignment problems are formulated to structure the problems. The integrated methodology proposed to solve these problems includes seven procedures: PCB grouping, family decomposition, subfamily sequencing, Keep Tool Needed Soonest (KTNS), component setup determination, component allocation, and board assignment procedures. The first five procedures, derived from prior published literatures, are addressed in this chapter. The remaining parts, component allocation and board assignment procedures, are developed in the next chapter.

3.1 Mathematical Model

To understand the overall problem and to develop a structural framework, mathematical models for component allocation and PCB assignment problems were

developed based on the assumptions stated in Chapter one. The production time for a board type includes total setup time and total placement time of all required components. For coupled machine system, total setup time consists of setup time on all machines and idle time on some machines due to imbalance of setup time on machines. For example, on two machines, if the feeder setup time on the first machine is more than on the second one. This means that during loading some component feeders on the first machine, loading component feeders on the second machine is complete and this machine is waiting for the feeder setup on the first machine. This situation is illustrated in Figure 3.1. Like setup time, total placement time is composed of placement time while all machines are busy and the idle time on some machines due to imbalance of placement time on machines. In this example, the second machine is waiting for component placing on the first machine because the placement time on the first machine is greater than on the second machine (see Figure 3.1). Reducing the idle time due to imbalance of setup and placement time among machines would reduce the total production time. Consequently, the purpose of the component allocation problem is to balance the workload across machines for individual PC board types, while the objective of the board assignment problem is to minimize global makespan for all board types.

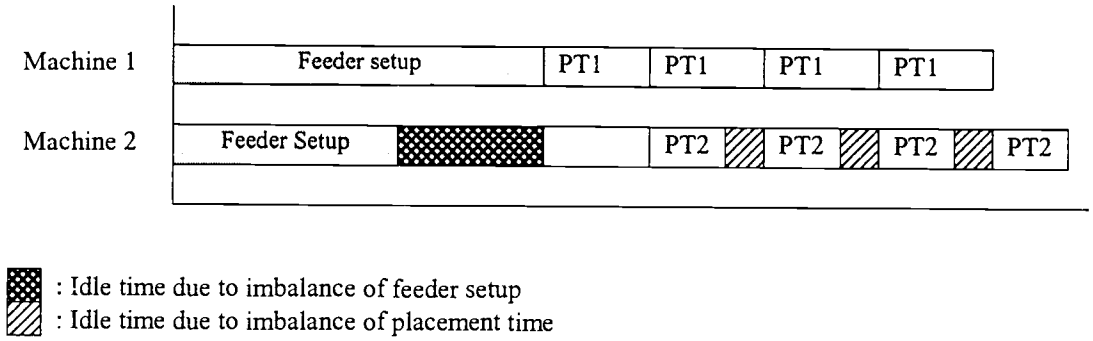


Figure 3.1 Production time on two machines in couple machine system

3.1.1 Component Allocation

Balancing workload across machines within a production line for individual PCB types is equivalent to minimizing the sum of placement time and feeder setup time for individual PCB types.

Variables:

i : index for component types i = 1,2,3,...,N

k : sequence index for the k^{th} PCB type produced

m : index for machines m = 1,2,3,...,R

f_i : the number of slots required by component type i

q_k : the units of the k^{th} PCB type produced

d_{ik} : the number of component types i required by one board of the k^{th} PCB type

N_k : set of components required by the k^{th} PCB

t_{ikm} : processing time for picking and placing one component type i on one board of the k^{th} board type using machine m

s_{im} : estimated feeder setup time for component type i on a feeder on machine m.

C_m : the number of feeder slots available on machine m

Υ_k : component setup time for the k^{th} PCB type

λ_k : processing time for processing all boards of the k^{th} PCB type

Decision variables:

X_{ikm} = 1 if component type i is assigned to machine m for the k^{th} PCB produced in a production line,

= 0 otherwise

W_{ikm} = 1 if component type i needs to be loaded on machine m for the k^{th} PCB in a sequence on machine m ,

= 0 otherwise

Objective function:

$$\text{Min } \Sigma_k (\Upsilon_k + \lambda_k)$$

Constraints:

$$\Upsilon_k \geq \Sigma_i s_{ik} W_{ikm} \qquad \forall i \in N_k, \forall j, k \qquad (1)$$

$$\lambda_k \geq \Sigma_i q_k t_{ikm} d_{ik} \qquad \forall i \in N_k, \forall k, m \qquad (2)$$

$$\Sigma_i f_i X_{ikm} \leq C_m \qquad \forall i \in N_k, \forall k, m \qquad (3)$$

$$W_{ikm} \geq X_{ikm} - X_{i(k-1)m} \qquad \forall i \in N_k, \forall k, m \qquad (4)$$

$$\Sigma_k X_{ikm} \geq 1 \qquad \forall i \in N_k, \forall k, m \qquad (5)$$

$$X_{ikm}, W_{ikm} \in \{0, 1\} \qquad \forall i, k, m \qquad (6)$$

The objective of the model is to minimize total production time for PCBs. Constraints (1) and (2) determine setup and processing time for producing each PCB type. Constraint (3) corresponds to the feeder capacity constraint for each machine. Constraint (4) determines the component setups for each PCB and each machine. Constraint (5) ensures that each component has to be placed on a machine at least once. Constraint (6) is the non-negative set of decision variables.

3.1.2 PCB Assignment

PCB assignment aims at minimizing global makespan. As discussed earlier, the production time of a PCB lot is sequence-dependent. In order to solve the assignment problem, it is necessary to assume that the allocation problem is already executed. Consequently, the setup time for all components loaded for the k^{th} PCB (Υ_k) and the processing time for the k^{th} PCB (λ_k) are determined by the allocation problem.

Variables:

j : index for the PCB types, j = 1, 2, 3, ..., M

k : sequence index for the k^{th} PCB produced

l : index for lines, l = 1, 2, 3, …, L

Υ_{kl} : component setup time for the k^{th} PCB produced in line l

λ_{kl} : processing time for processing the k^{th} PCB produced in line l

Decision variable:

Y_{jkl} = 1 if PCB j is assigned to line l in the k^{th} of the sequence ;

= 0 otherwise

Objective function:

Minimize τ

Constraints:

$$\tau \geq \Sigma_j (\Upsilon_{kl} + \lambda_{k\,l}) Y_{jkl} \qquad \forall\, l, k \qquad (1)$$

$$\Sigma_l \Sigma_k Y_{jkl} = 1 \qquad \forall\, j \qquad (2)$$

$$Y_{jkl} \in \{ 0, 1 \} \qquad \forall j, k, l \qquad (3)$$

The objective of the model is to minimize the completion time for all PCBs. Constraint (1) determines the global makespan length. Constraint (2) assures that each PCB type is assigned to only one line at one time. Constraint (3) is the constraint for binary variables.

3.2 Solution Approach

Integer Programming models for the type of problem addressed in this research have been shown to be NP-complete. This implies that solving for optimality may not be feasible in terms of computation time. A heuristic approach was developed to obtain acceptable solutions in a reasonable amount of time.

This study seeks to answer the questions of *which board should be produced on which line and what components should be loaded on which machine for a specified board type.* As mentioned previously, the sequence-dependent setup would impact board production time. Several researchers have been concerned with this factor (Chen and Dong, 1999; Chung, 1991; Peters and Subramanian, 1996; Rubin, 1995). They sequenced boards with the purpose of reducing setup time. Thus, the typical approach (Figure 3.2) to solve board assignment and component allocation problems would consist of (1) board assignment, (2) board sequencing, (3) component allocation, and (4) search to improve the solution with the objective of minimizing makespan. The use of this approach will result in two binary decision variables: the PCB board j assigned to the production line l in the k^{th} sequence of PCBs on line l, and the component i assigned to machine m for PC board j. Consider 12 PCBs with 30 components to be processed on three production lines, each with two machines. By assuming four PC board type for each line, the size of the two variables mentioned above would be $12 \times 3 \times 4 = 144$ and $30 \times 2 \times 12 = 720$, respectively. For a large problem the size of these variables is

extremely large. The computation time to execute an optimal solution for such a problem is infeasible.

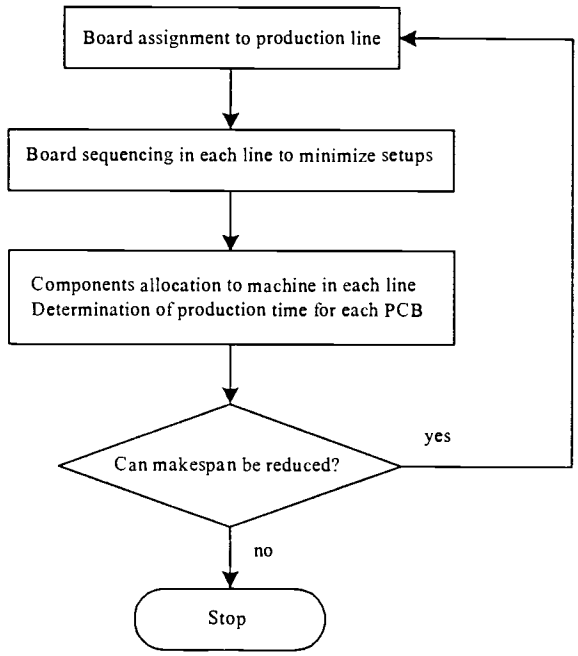


Figure 3.2 Flow diagram of the typical approach

The integrated methodology developed in this research applied the Group Technology (GT) concept to reduce the problem size. Based on component similarity between boards, PC boards are grouped into families. The size of the decision variable for the assignment problem can thus be reduced from the number PCB types to the number of PCB families. As a result of the GT approach, board types within the same family should have more common components than the board types between families. Sequencing PCB types within each family would

impact the number of setups, whereas sequencing families may not be significant for setup reduction. It is reasonable to consider a PCB family to be sequence-independent. Thus, assigning board families to production lines may be accomplished by using typical scheduling rules. For example, the Longest Processing Time (LPT) rule and Shortest Processing Time (SPT) may be applied to distribute families to production lines.

Furthermore, in the grouping procedure, families of "similar" boards are generated without considering the feeder capacity constraint. Relaxation of the capacity constraint in the initial stage of this process is to allow all similar PCBs to be grouped together. However, various components for a family may require more feeder slots than the feeder capacity of a production line. Based on the Decompose And Sequence (DAS) setup strategy, a PCB family may be divided into subfamilies such that the number of required feeders for a subfamily does not exceed the feeder capacity.

In order to minimize makespan in a low-volume, high-mix environment, reduction of setup time within board families should be considered in addition to assigning PCB families to appropriate production lines. Consequently, subfamilies within each family should be appropriately arranged to reduce setups. The other approach for setup time reduction is the use of Keep Tool Needed Soonest (KTNS) policy on available feeder slots of each subfamily. Leaving some components that are not required for the current subfamily but may be needed by a following subfamily tends to decrease the number of setups.

At this point, the sequences of subfamilies and the set of components loaded for each subfamily are determined. By taking advantage of these results, classification of setup components into standard, semi-standard, and custom setup components is performed in order to reduce the complexity of the component allocation problem and to make the shop floor control more manageable.

Assigning entire PCB families to production lines to minimize makespan is difficult to accomplish since the amount of production time for each family is quite large compared to that of each PCB lot. Splitting some subfamilies is allowed as long as this does not increase the makespan. However, the impact of sequence-dependent setup time will lead to recomputing of the production time of split subfamilies.

To summarize, the integrated methodology is composed of seven sub-problems: board grouping, family decomposition, subfamily sequencing, KTNS application, setup component classification, component allocation, and board assignment. The relationship of these sub-problems is shown in Figure 3.3. Board grouping is performed to reduce problem size of board assignment. Family decomposition is employed to determine the subfamilies based on the DAS setup strategy. Component allocation incorporated with setup component type determination can give an estimate of the production time for each family, which is necessary information for the PCB assignment problem. Reducing maximum local makespan and component setups would reduce global makespan. Subfamily grouping based on component similarity, board sequencing and the KTNS are used

to reduce setup time within a family. To decrease maximum local makespan, some subfamilies are allowed to be split and produced on another line.

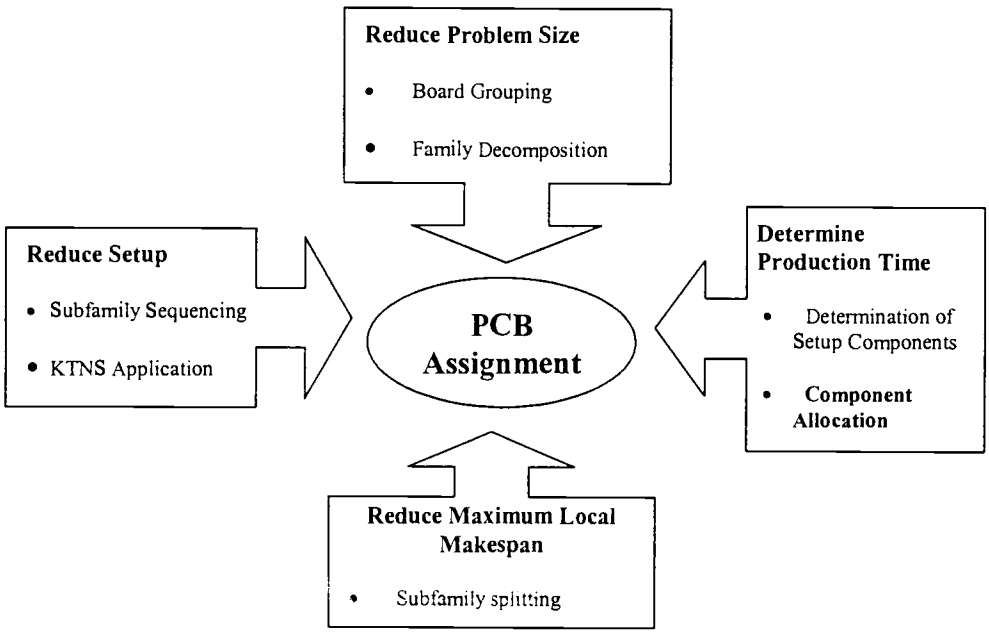


Figure 3.3 Relationships between sub-problems

The integrated methodology in Figure 3.4 builds on published research. The set of heuristics and procedures associated with PCB grouping, family decomposition, subfamily sequence, KTNS, and component setup type determination are taken from the literature. Solution procedures for component allocation and PCB family assignment developed in this research are the subjects of the next chapters. Solution procedures for PCB grouping, family decomposition, subfamily sequence, KTNS, and component setup type determination are discussed in the following sections.

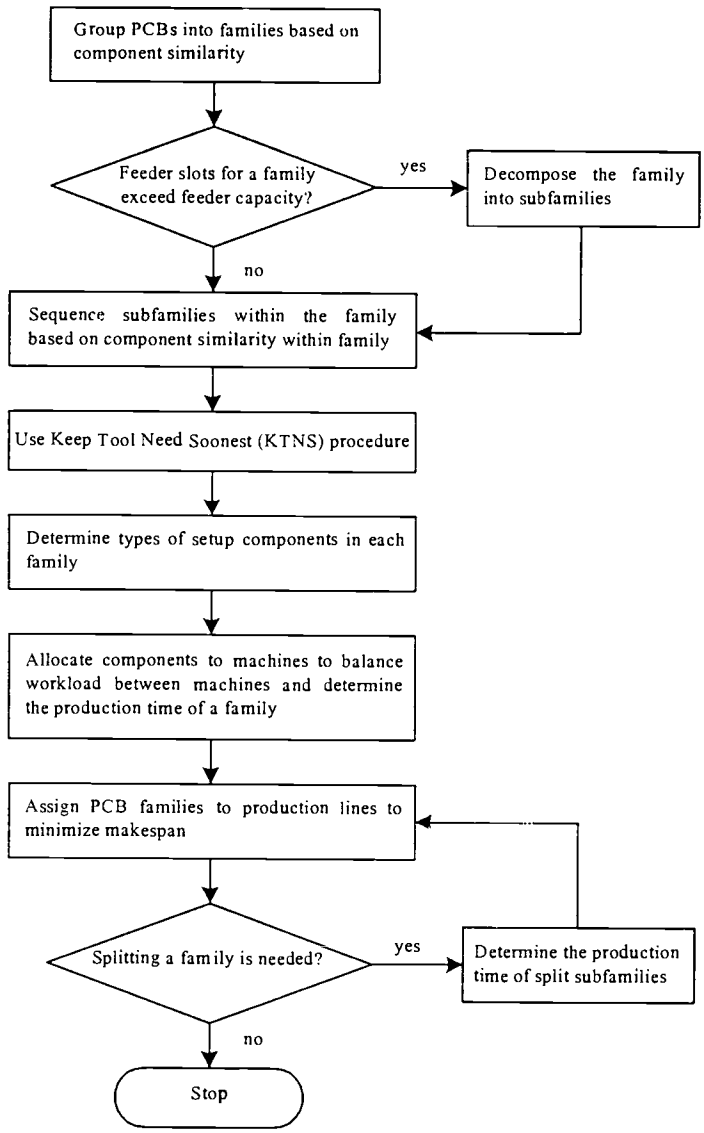


Figure 3.4 Integrated methodology procedures

3.3 Integrated Methodology: Steps 1-5

The first five steps of the integrated methodology, PCB grouping, family decomposition, subfamily sequencing, KTNS, and component setup determination, will be discussed in this section.

3.3.1 PCB Grouping

The primary purpose of PCB grouping is to find a minimum number of groups. By considering common components among PC boards, board grouping will lead to minimizing the number of component setups. The P-median method, a type of Integer Programming model, can generally be used to accomplish this objective (Kusiak, 1987; Li, 1999). The P-median model, however, may not be applicable for large-scale problems. Sule (1992) proposed a tabular approach consisting of two-phase heuristic approaches for PCB grouping. In the first phase, the GT concept is applied to group components by their similarity value. A component-to-component matrix, indicating the number of PC boards demanding both row and column components, is constructed from PCB-to-component incidence matrix. The closeness ratio is defined as the ratio of the relationships between the entering component and current components in the group to the total number of components that are presently assigned to that group. The threshold value, a parameter to measure effectiveness of joining a PCB to a family consisting of other PCBs, determines the minimum of closeness values for joining a PCB to board family. The closeness ratio and threshold are used to make a decision about a component joining a group. In the second phase, each PCB is assigned to the group that has the most required components. In this research, the objective of PCB grouping is to reduce the problem size. Since the next procedure is the sequencing procedure within a family, which aims at feeder setup minimization, grouping PC

boards into families would also help the following procedure to achieve its goal. The grouping approach from Sule (1992) is employed to create PCB families. However, the feeder capacity is not used as a limitation at this step. The advantages of using this grouping procedure are:

1. User-defined group size: Users can specify the maximum group size. The most similar components are assigned to either an existing group or a new group according to the closeness ratio to each group without any capacity constraint.
2. Permission of component duplication in different groups: This grouping procedure allows duplication of a component in different groups, depending on threshold values and the closeness ratio.

Disadvantages from heuristic procedure occur when most PC boards have either very high or low component similarities. With a high similarity value the procedure might create an extremely large board family. Subsequently, a family may have to be decomposed into a number of subfamilies during the family decomposition procedure. The number of subfamilies for such a family will be high. On the other hand, with a low similarity value, the procedure may generate a very small family. At the high end, the procedure may lead to high computation times for subfamily sequencing. At the low end, the methodology may not be used to its potential.

Input for this heuristic procedure consists of a PCB-to-component incidence matrix (Table 3.1) and a threshold value. The PCB grouping procedure is shown in Figure 3.5, an example illustrating the procedure is given in Appendix A1. The result from this step is a PCB-to-component matrix for each family as shown in Table 3.2.

Table 3.1 PCB-to-component incidence matrix

PCBs	Components											
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	...	Cn
1												
2												
3												
4												
...												
M												

Table 3.2 PCB-to-component matrix for each family

Families	PCB	Components											
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	...	Cn
F1	1 2 ... 8												
F2	7 9 ... 20												
... Fm	... ...												

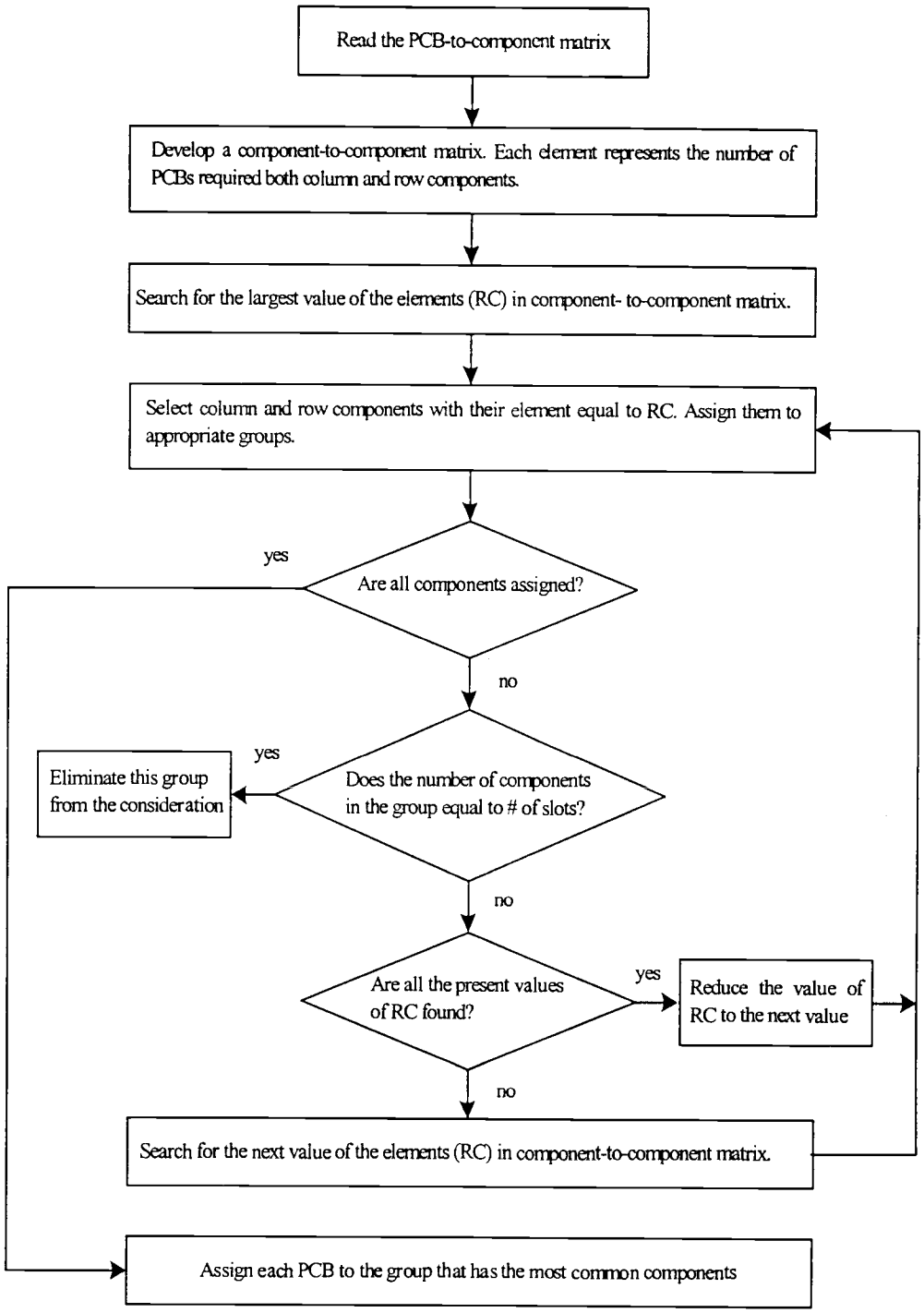


Figure 3.5 Flow of PCB grouping procedure

3.3.2 Family Decomposition Procedure

The number of feeder slots needed by each family may be in excess of the feeder capacity of a production line. Producing such a family requires more than one machine setup. An approach to classify PCBs into subfamilies is thus a requisite. Shtub (1992) described the heuristic to divide a family into subfamilies so that each subfamily could be produced with only one machine setup. Global Jaccard similarity for a PCB is the sum of Jaccard similarities between that PCB and the others in the unassigned PCB group. Global Jaccard similarity is used to select the first PCB into a subfamily. Then Jaccard similarity between the next PCB and current PCBs in the subfamily is applied to choose the entering PCB. The decision of entering that PCB into the subfamily is based on a threshold value. The upper limit for a subfamily size is the number of feeders. An advantage of this procedure is that the first PC board type in each subfamily has the highest similarity since the global similarity measure is used to create subfamilies.

Inputs for this approach are (1) PCB-to-component incidence matrix for a family, a result of PCB grouping process in step 1, (2) Feeder capacity of the machines, (3) Component setup time and PCB setup time used to calculate savings time, and (4) Threshold value. The flow diagram of family decomposition procedure is shown in Figure 3.6. The procedure and example are illustrated in Appendix A2. The result of this procedure is subfamily-to-component matrix and PCBs in each subfamily as shown in Tables 3.3(a) and (b), respectively.

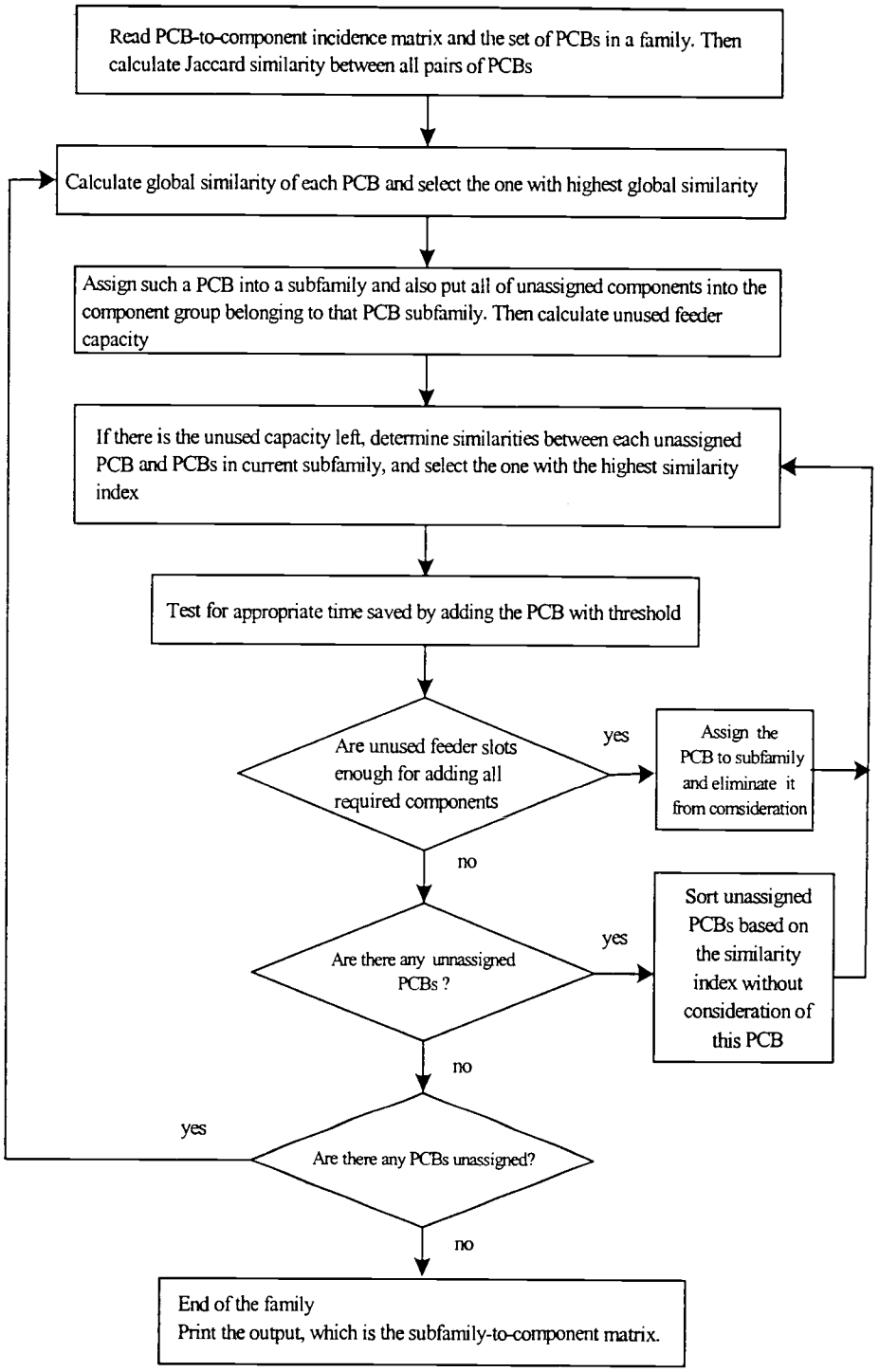


Figure 3.6 Flow of family decomposition procedure

Table 3.3 (a) Subfamily-to-component matrix

Families	Subfamilies	Components								
		C1	C2	C3	C4	C6	C7	C8	...	Cn
F1	s11									
	s12									
	...									
	s1k									
F2	s21									
	s22									
	...									
	s2k									
...	...									
Fm	s k									

Table 3.3 (b) PCBs in each subfamily

Families	Subfamilies	PCBs
F1	S11	
	S12	
	...	
	S1k	
F2	S21	
	S22	
	...	
	S2k	
...	...	
Fm	S k	

3.3.3 Subfamily Sequencing Procedure

The next step after determination of subfamilies is sequencing of subfamilies to reduce component setups within each family. The subfamily that will be next scheduled for production should have the most common components with one currently in production. The number of subfamilies in a family is usually not high. The Best-First (BF) algorithm is a heuristic search methods that uses an *evaluation function,* f(n), as guiding information to move towards the direction of

the goal (Pearl, 1984). The evaluation function is used to decide the order in which nodes should be considered during the search. This search method tends to reach the goal with fewer step comparisons than other approaches. In this research, the BF algorithm is applied to find a PCB subfamily sequence in a family. In this case, the goal of the procedure is feeder setup minimization. Therefore, the heuristic evaluation function, the number of feeder setups, is used to determine the direction of the search. The flow diagram of sequencing procedure is depicted in Figure 3.7. Detail of the approach, using an example is given in Appendix A3.

The largest subfamily is arbitrarily defined as the first subfamily of the sequence. The next subfamily in the sequence is then selected based on minimizing additional setups. Using the BF algorithm, the number of steps to search from the beginning subfamily until the last subfamily will be less than other exhaustive approaches (Pearl, 1984; Tanimoto, 1987). However, if the number of subfamilies is very high, the computation will increase exponentially. Input information for this step is the PCB subfamily-to-component matrix from section 3.2.2. The result of this approach is a subfamily sequence in each family as shown in Table 3.4.

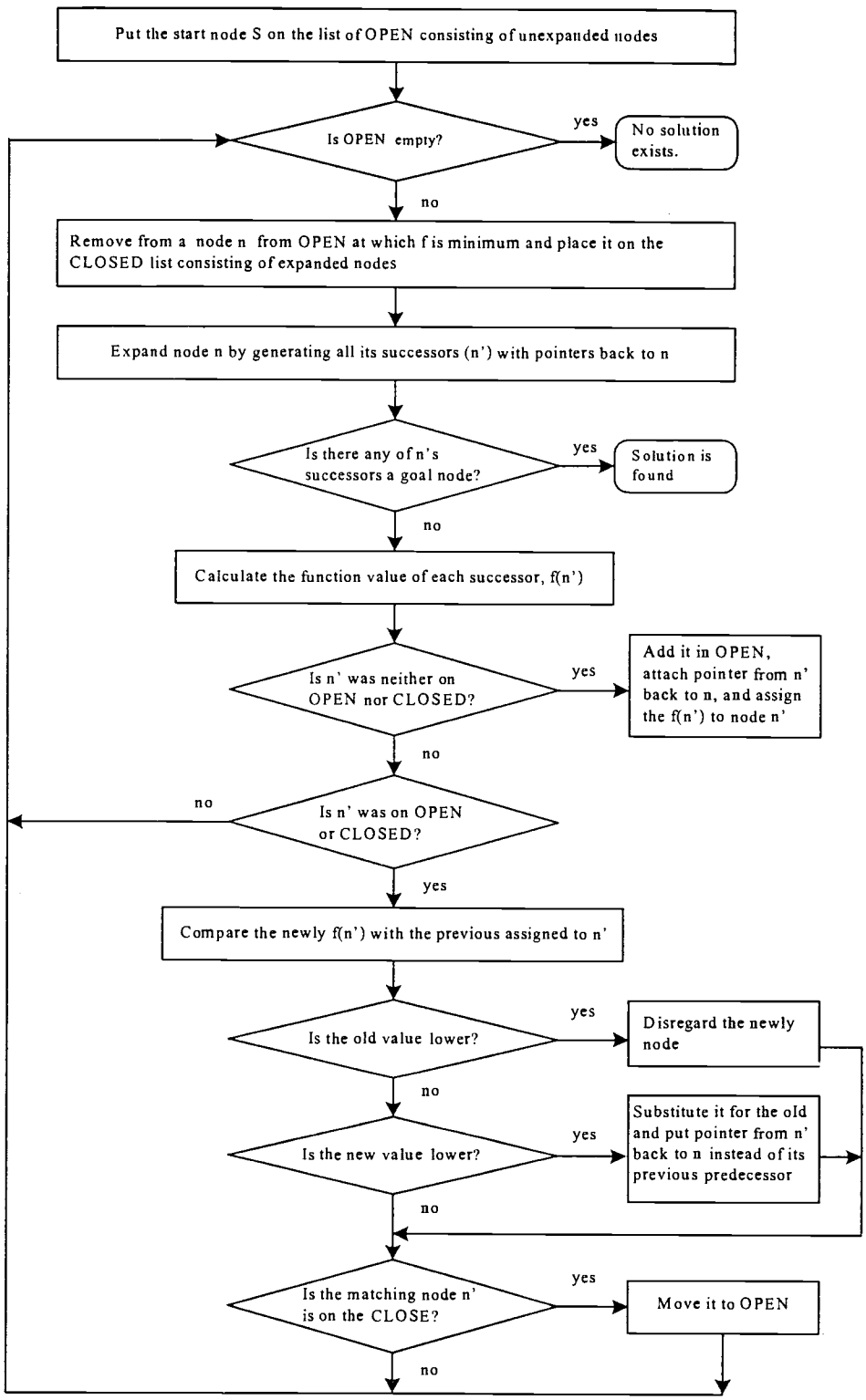


Figure 3.7 Flow diagram of sequencing procedure

Table 3.4 Subfamily sequence for each family

Families	Subfamily Sequence
F1	S13, S15, S14, S16, S12, S11
F2	S22, S24, S21, S23, S25
F3	S31,S32,S33,S35,S34
.	.
.	.
.	.
Fm	Sm2, Sm3, Sm1, Sm4

3.3.4 Keep Tool Needed Soonest (KTNS) Policy

In some cases, a number of required feeders for a subfamily may be less than the feeder capacity. Instead of having free feeders, components that are not demanded by the current subfamily but are needed by a later subfamily should be stored on available feeders. The objective of the KTNS policy is to find the optimal component switches for a specific sequence. The procedure starts with checking the unused feeders used for producing the current subfamily. If there is an unused feeder, a component is selected that has to be unloaded from the feeders but will be required by a later subfamily. However, if the current subfamily is the first subfamily, early installation of the component needed by the following subfamily is acceptable.

The necessary inputs for this heuristic are (1) PCB subfamily-to-component matrix resulting from the family decomposition procedure, (2) PCB subfamily sequence from the subfamily sequencing procedure, and (3) feeder slot capacity. The flow diagram for the KTNS approach is presented in Figure 3.8. Detail of the

heuristic approach and an example are given in Appendix A4. The outcome of this step is subfamily-to-setup component matrix in Table 3.5, which consists of all subfamilies in order and all components stage on feeder for each subfamily.

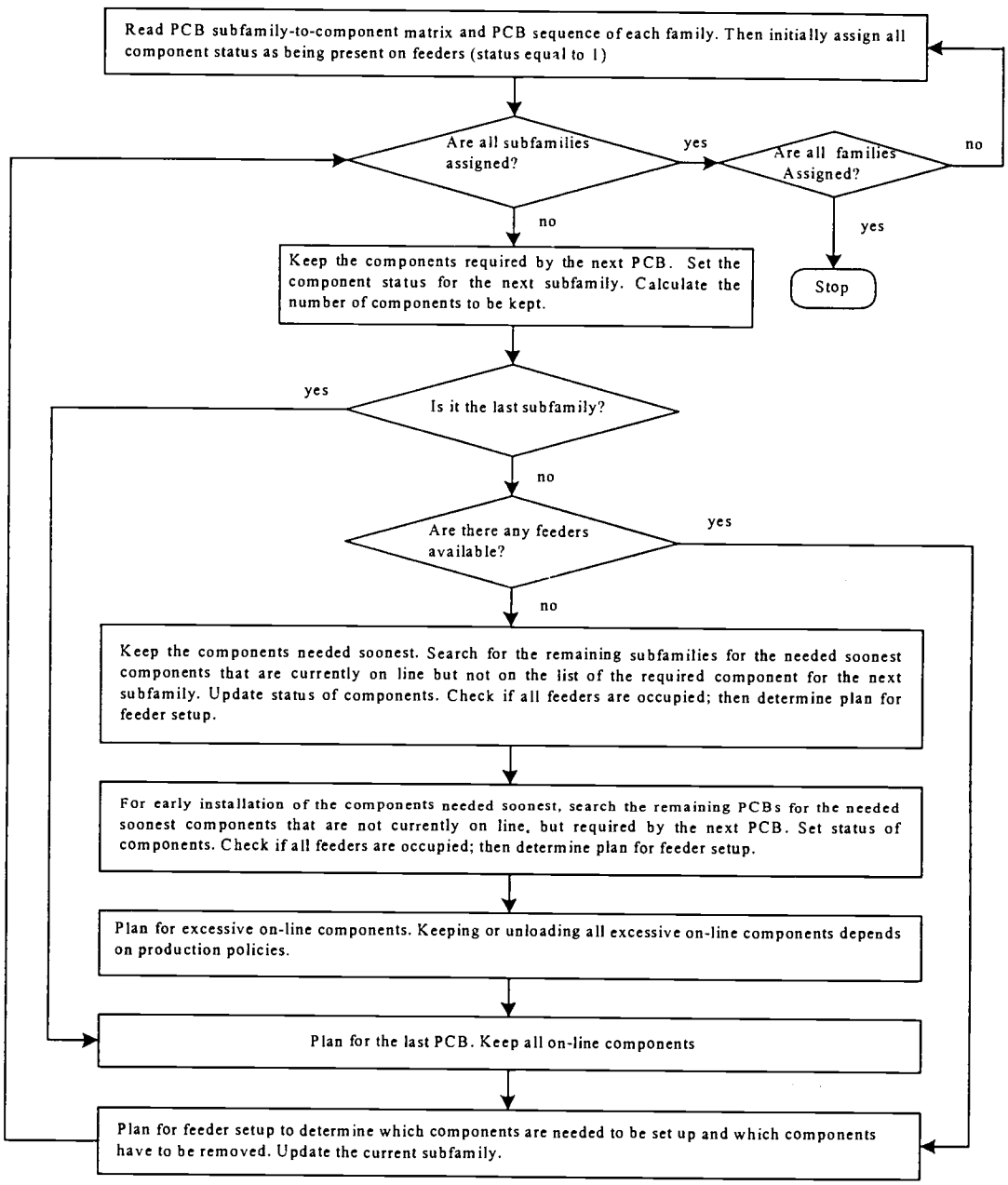


Figure 3.8 Flow diagram of KTNS approach

Table 3.5 Subfamily-to-setup component matrix

Families	Subfamilies (in sequence)	Components								
		C1	C2	C3	C4	C6	C7	C8	...	Cn
F1	S11									
	...									
	S1k									
F2	S21									
	...									
	S2k									
...	...									
Fm	S k									

3.3.5 Setup Component Type Determination

This process is used to categorize components into three types: standard, semi-standard, and custom setup components. This procedure would help to reduce the complexity of the subsequent component allocation problem. In the procedure a component in the current subfamily, if required by all subfamilies, is a standard setup component. If it is present in the next subfamily but not all subfamilies, it is termed a semi-standard; otherwise it is a custom. This procedure needs the subfamily-to-setup component matrix resulting from the KTNS procedure to indicate sets of setup components allowed to stay on machines for each subfamily. The procedure flow diagram is depicted in Figure 3.9. Details for this procedure and an example are given in Appendix A5.

Using these five steps, board types are grouped into families. If a board family needs feeder slots more than feeder capacity, then it is divided into subfamilies. Based on the common components between consecutive board types,

these subfamilies are sequenced. Components required by subfamilies in a family are classified into three categories, standard, semi-standard, and custom setup components. Next, these components will be allocated to machines and the production time for each family will then be calculated. Finally, all board families will be assigned to the lines. These two steps will be discussed in the next chapter.

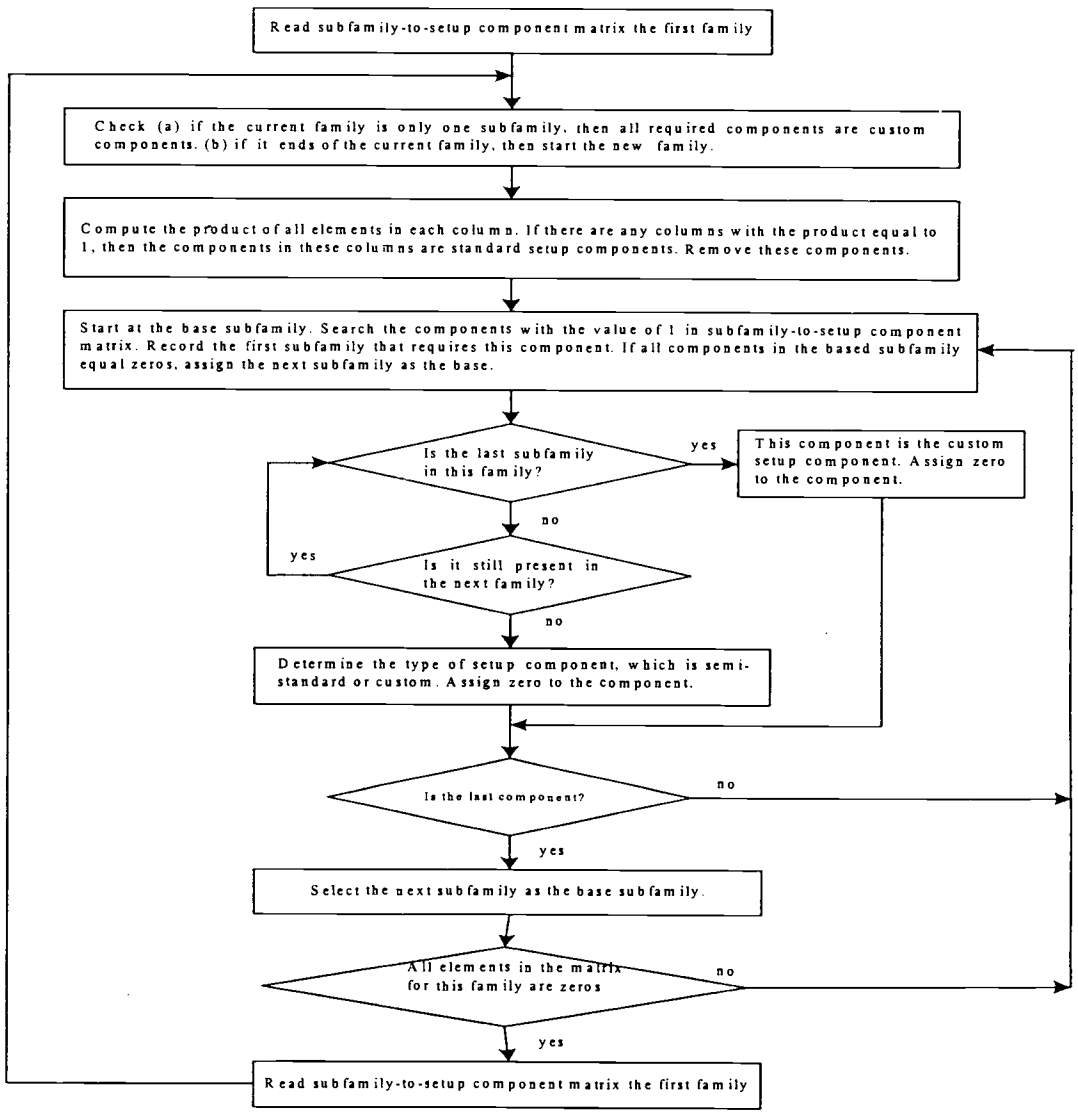


Figure 3.9 Flow diagram for setup component type determination

CHAPTER 4 COMPONENT ALLOCATON AND BOARD ASSIGNMENT PROBLEMS

As stated earlier, the size of the component allocation and board assignment problems are typically large. The approach used in this research is to reduce the problem size by using a grouping procedure. All components are classified into three categories. A standard setup component is permanently installed on a machine during production of an entire family. A semi-standard setup component is allowed to stay on a machine to produce some subfamilies. Finally, a custom setup component is loaded/unloaded for an individual PCB subfamily. For the Decompose And Sequence (DAS) strategy, PCBs in a family are divided into subfamilies until each subfamily can be produced with only one machine setup. Consequently, standard and semi-standard setup components are staged on a machine for more than one machine setup, while the custom setup component is designated for only one machine setup. The use of the setup component type classification will reduce the complexity of component allocation. Instead of assigning all required components at the same time, each component type will be hierarchically assigned to machines. The production time for each PCB and each PCB family can then be calculated. Finally, a board family will be assigned to a production line. In this chapter, procedures to distribute components to machines and to allocate PCB families into production lines are discussed. To measure the applicability and effectiveness of the heuristic algorithm, two data resources,

published literature with the known optimal (or best) solutions and an industry data set, are utilized.

In order to clearly understand how to identify the setup component types, an example of a board family in Table 4.1 is grouped into five subfamilies with eight PCB types and six components. These subfamilies are sequenced in the order (s1, s2, s3, s4, s5).

Table 4.1 Example of a board family

Subfamily Sequence	PCBs	Components					
		c1	c2	c3	c4	c5	c6
s1	P1	1	1	1	0	0	0
	P2	1	0	1	0	0	0
s2	P3	1	1	1	1	0	0
s3	P5	1	1	1	1	1	0
	P7	1	1	1	1	0	0
s4	P4	1	0	0	1	0	1
	P6	1	1	0	1	0	1
s5	P8	1	1	0	0	0	0

1: represents PCB in associated row requires the component in associated column

From Table 4.1, components c1 and c2 are standard setup components. Components c3 and c4 are the semi-standard components for subfamilies s1, s2, and s3, and for subfamilies s2, s3, and s4, respectively. Components c5 and c6 are custom components for subfamilies s3 and s4, respectively.

4.1 Mathematical Model

The general component allocation problem in Chapter 3 is formulated as an Integer Programming (IP) model. The objective is to balance the workload across machines for each PCB type produced. A machine's workload consists of component placement time and feeder setup time for all required components. These components are determined as standard, semi-standard, and custom setup components. The PCBs are grouped into families and each family may then be divided into subfamilies because of feeder capacity constraints. Thus, the objective is to balance the machines' workload among machines for each PCB family produced. Balancing workload is equivalent to minimizing the sum of placement times and feeder setup times for all PCB types in a family.

Variables :

i : index for component types $i = 1, 2, 3, \ldots, N$

j : index for PCB types $j = 1, 2, 3, \ldots, M$

m : index for machines $m = 1, 2, 3, \ldots, R$

g : index for PCB subfamily $g = 1, 2, 3, \ldots, G$

q_j : volume of each PCB type j to be produced.

d_{ij} : the number of components i required by one board of PCB type j.

t_{ijm} : processing time of picking and placement component type i on board type j using machine m.

s_{im} : estimated setup time for component i on a feeder on machine m.

C_m : the number of slots available on machine m

N_g : set of components required by PCB subfamily g

W_{igm}: the number of setup for the semi-standard setup component i for subfamily g on machine m

λ_{jg} : component placement time for each PCB type j in subfamily g

α : feeder setup time for standard setup components

Υ_g : feeder setup time for semi-standard and custom components for subfamily g

Decision variables :

X_{im} = 1 if component i is a standard setup component on machine m ,

= 0 otherwise

Y_{igm} = 1 if component i is a semi-standard setup component for PCB subfamily g on

machine m,

= 0 otherwise

Z_{igm} = 1 if component i is a custom setup component for PCB subfamily g on

machine m,

= 0 otherwise

Objective function :

$$\text{Min} \quad \alpha + \Sigma_g (\Upsilon_g + \Sigma_j \lambda_{jg})$$

Constraints :

$$\lambda_{jg} \geq \Sigma_i q_j t_{ijm} d_{ij} \qquad \forall i \in N_g, \forall j, m \qquad (1)$$

$$\Upsilon_g \geq \Sigma_i s_{im} (W_{igm} + Z_{igm}) \qquad \forall i \in N_g, \forall g, m \qquad (2)$$

$$\alpha \geq \Sigma_i s_{im} X_{im} \qquad \forall i, m \qquad (3)$$

$$Y_{igm} - Y_{ig-1m} \geq W_{igm} \qquad \forall i \in N_g, \forall g, m \qquad (4)$$

$$\Sigma_i (X_{im} + Y_{igm} + Z_{igm}) \leq C_m \qquad \forall i \in N_g, \forall g, m \qquad (5)$$

$$X_{im} + Y_{igm} + Z_{igm} \geq 1 \qquad \forall i \in N_g, \forall g, m \qquad (6)$$

$$X_{im}, Y_{igm}, Z_{igm}, W_{igm} \in \{0, 1\} \qquad \forall i, g, m \qquad (7)$$

The objective function is to minimize the sum of setup and placement time for board types in a family. Since the component setups can be categorized into three types, the setup time for standard setup component is α, which equals to the sum of setup times for all standard components, $\Sigma_i s_{im} X_{im}$, and the setup time for semi-standard and custom setup components for a subfamily is Υ_g, which equal to the sum of setup times for all semi-standard and custom components, $\Sigma_i s_{im} (W_{igm} + Z_{igm})$. The placement time for individual board types (λ_{jg}) is then defined as sum of total placement time per board times the volume of each board type, $\Sigma_i q_j t_{ijm} d_{ij}$.

Constraints (1), (2), and (3) determine processing time for producing a lot of each PCB type and component setup time for a family. Constraint (4) represents the number of component setups for semi-standard components. Constraint (5) corresponds to feeder capacity constraint for each machine. Constraint (6) ensures

that a component on a PCB would be exactly one type of setup component. Constraint (7) is integrality requirements for decision variables.

4.2 Solution Methodology

Solving the IP model to obtain an optimal solution would be the best alternative. However, this is not generally feasible for large problems. A heuristic approach is an alternative to obtain a good solution in a reasonable amount of computation time. The objective function in the IP model consists of setup time and placement time for each type of setup component for a family: the standard setup components based on the family level, the semi-standard setup components based on subfamily level, and the custom component based on the individual PC board level. A machine's workload due to a particular component is the sum of feeder setup time and placement time of that component. Therefore, the heuristic first attempts to allocate standard setup components based on workload requirements of a PCB family. Then the semi-standard components are allocated based on workload requirements of a PCB subfamily, and incorporated into the workload of the standard and semi-standard setup components for that subfamily, which are determined previously. Finally, the custom- based components are allocated based on the workload of individual PCBs including the workload due to standard and semi-standard setup components that have already been assigned. The framework

of the methodology is shown in Figure 4.1 and the heuristic procedure is given in Figure 4.2.

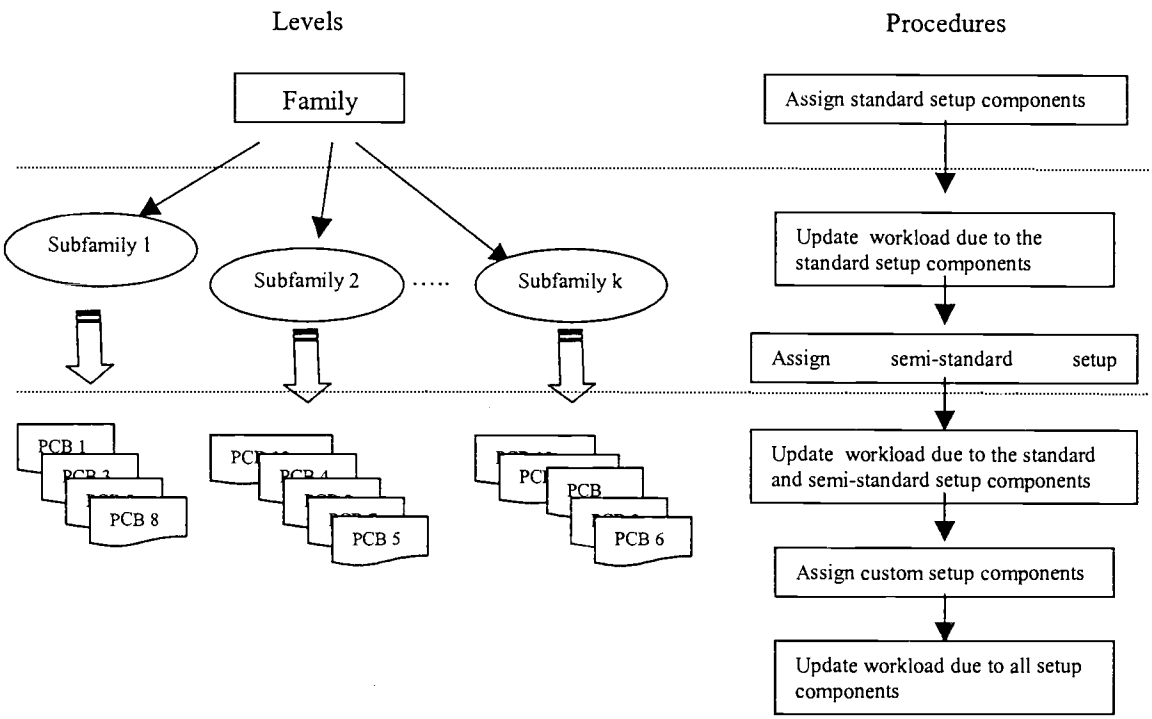


Figure 4.1 Methodology flowchart

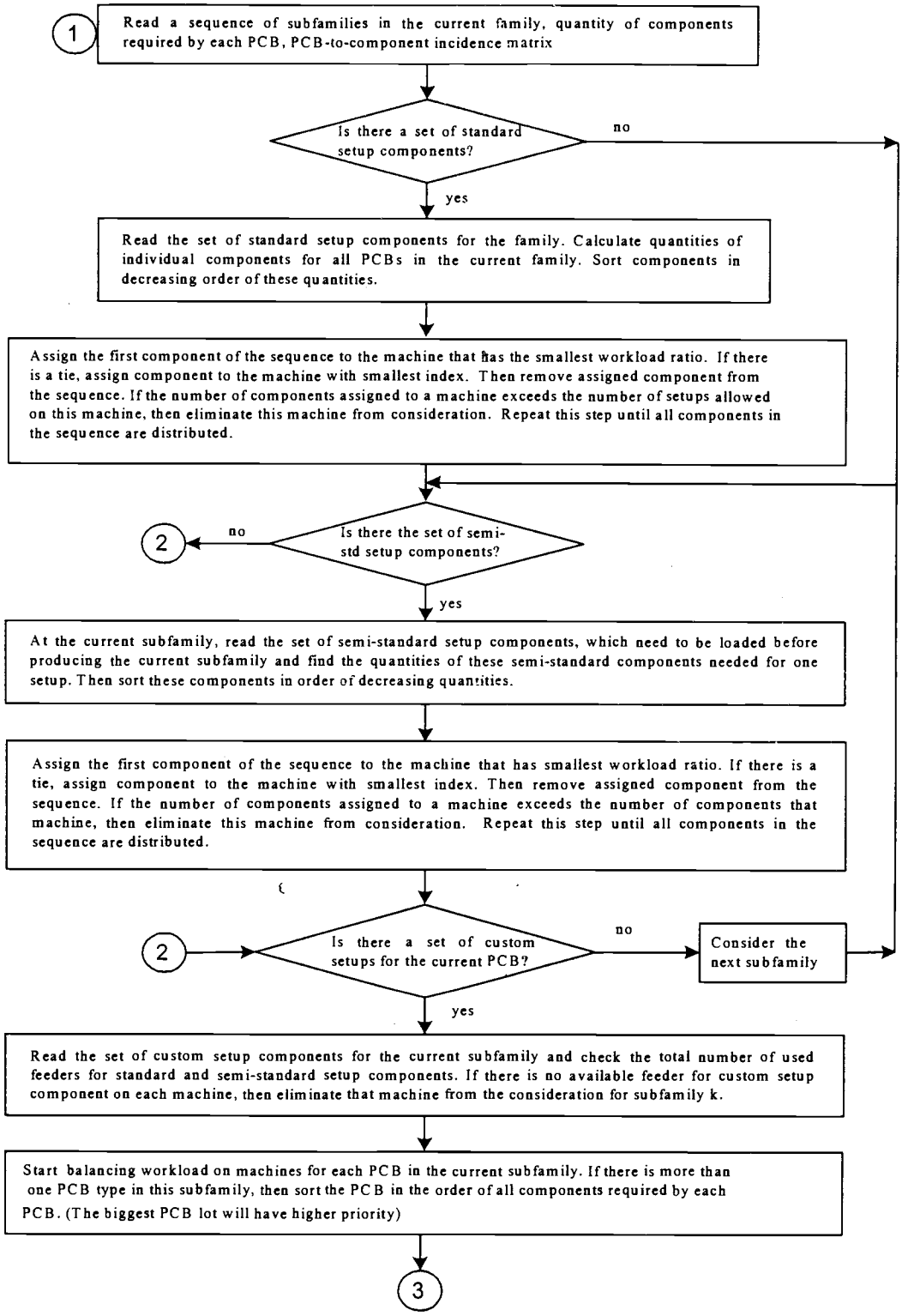


Figure 4.2 Component allocation procedure

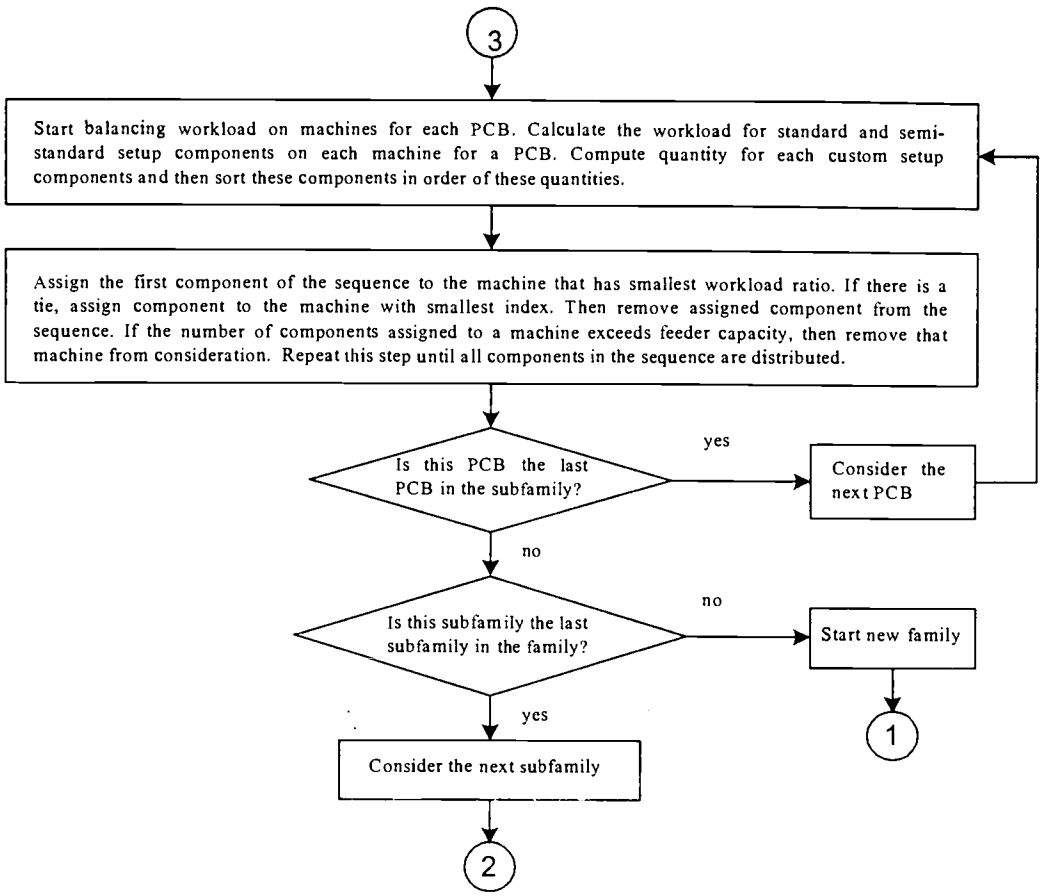


Figure 4.2 Component allocation procedures (cont.)

An example data set with seven PCBs and fifteen components presented in Table 4.2 is used to explain the allocation procedure of Figure 4.2. This table represents the quantity of each component on each PC board. The PCB- to-component matrix after performing the KTNS procedure, presented in Table 4.3(a), is used to determine the three types of setup components. The setup component types and other production information are summarized in Tables 4.3 (b) and 4.3 (c). There are two identical machines, called M1 and M2, each with five feeder

slots. The feeder setup time and placement time for a component are 1.6 and 0.01 units, respectively, for both machines.

Table 4.2 Quantity of components for each PCB

PCB	Components														
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
P1	0	5	0	10	5	5	5	10	5	0	0	5	20	0	0
P2	0	5	0	0	0	0	0	5	20	0	0	0	0	5	0
P3	0	5	0	0	0	0	0	0	5	0	0	0	0	0	5
P4	0	20	5	0	5	0	0	0	15	20	0	5	0	0	5
P5	5	10	5	0	0	5	0	5	0	10	0	0	0	0	0
P6	0	0	5	0	5	10	0	5	0	10	5	5	0	5	0
P7	0	0	5	0	5	15	0	0	0	0	0	0	0	0	0

Table 4.3 (a) PCB-to-component matrix after executing the KTNS procedure

Sequenced Subfamiles	PCB	Components														
		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15
1	P6	0	0	1	0	1	1	0	1	0	1	1	1	0	1	0
2	P4	1	1	1	0	1	1	0	1	1	1	0	1	0	0	1
	P5	1	1	1	0	1	1	0	1	1	1	0	1	0	0	1
3	P2	1	1	1	0	1	1	0	1	1	0	0	1	0	1	1
	P3	1	1	1	0	1	1	0	1	1	0	0	1	0	1	1
	P7	1	1	1	0	1	1	0	1	1	0	0	1	0	1	1
4	P1	1	1	0	1	1	1	1	1	1	0	0	1	1	0	0

1 : component associated with column will be in a feeder slot for producing the PCB associated with row

Table 4.3 (b) Volume of PCBs

PCBs (j)	1	2	3	4	5	6	7
Volume (units)	10	10	10	10	10	10	10

Table 4.3 (c) Production information

Sequenced Subfamiles	Sub-families	PCB	Lot sizes	Feeder setup		
				Standard	Semi-standard	Custom
1	s4	P 6	10	c5, c6,c8,c12	c3, c10	c11,c14
2	s3	P 4	10	c5, c6,c8,c12	c3, c10, c1, c2,	-
		P 5	10		c9, c15	
3	s2	P 2	10	c5, c6,c8,c12	c3, c1,c2,c9,c15	c14
		P 7	10			
		P 3	10			
4	s1	P 1	10	c5, c6,c8,c12	c1, c2, c9	c4, c7, c13

Iteration 1: Read the sequence of subfamilies and the associated production data from Table 4.3(b), and the quantity of components required by each PCB illustrated in Table 4.2. Since there are four standard setup components, (i.e., c5, c6, c8, and c12) these components need to be allocated to machines.

Iteration 2: Calculate the usage of each component for all PCBs in the family (see Table 4.4(a)); then sort them in order of decreasing usages. The component sequence based on decreasing usage is c6-c8-c5-c12.

Table 4.4 (a) Calculation of the standard component usage for the first subfamily

Sequenced Subfamilies	Components			
	c5	c6	c8	c12
1	50	100	50	50
2	50	50	50	50
3	50	150	50	0
4	50	50	100	50
Total	200	350	250	150

Iteration 3: Allocate these components to machines by using the smallest workload basis. Initially, no component is assigned to any machine; therefore, the workload on both machines is zero. The first component in the order (c6) is allocated to machine M1 followed by the second component (c8) to machine M2. The workload of the two machines due to assigning components c6 and c8 are 1*1.6+350*0.001 = 5.10 and 1*1.6+250*0.001 = 4.10, respectively (see Table 4.4(b)). Since the workload on machine M2 after these assignments is smaller, the next component, c5, is assigned to this machine. The process is repeated until all components are assigned. In this example, components c6 and c12 are assigned to machine M1 and components c8 and c5 are assigned to machine M2.

Table 4.4 (b): Allocation of the standard setup component to machines

Components	Number of components	Workload	TW1	TW2
			0	0
c6	350	5.1	5.1	0
c8	250	4.1	5.1	4.1
c5	200	3.6	5.1	7.7
c12	150	3.1	8.2	7.7

Iteration 4: Since components c3 and c10 are semi-standard for the first subfamily in the sequence, these components have to be sorted based on their usage for subfamilies they serve. This usage is shown in Table 4.4(c). This sequence is c10 follow by c3.

Table 4.4 (c) Calculation of the semi-standard usage

Sequenced Subfamilies	Components	
	c3	c10
1	50	100
2	200	300
3	50	0
4	0	0
Total	300	350

Iteration 5: Semi-standard components c10 and c3 are now allocated to machines by considering workload of each machine. Due to allocation of the standard setup components on machines, the workload on each machine has to be updated before assigning the semi-standard setup components. Two feeder setups are required for standard setup components c6 and c12 on machine M1 and for c5 and c8 on machine M2. For sequenced subfamily 1, the usage of components c6 and c12 is 100+50 = 150 and the usage of components c5 and c8 is 50+50 = 100 (Table 4.4(a)). The total workload of the two machines due to standard setup components are 2*1.6+150*0.01 = 4.7 and 2*1.6+100*.01 = 4.2, respectively. The allocation of the semi-standard for the first sequenced subfamily is illustrated in Table 4.4(d). Components c3 and c10 are assigned to machines M1 and M2, respectively. At this point, the components on machines M1 are {c6, c12, c3} and on machine M2 are {c8, c5, c10}.

Table 4.4(d) Allocation of the semi-standard setup components to machines

Components	Number of components	Workload	TW1	TW2
			4.7	4.2
c10	100	2.6	4.7	6.8
c3	50	2.1	8.9	6.8

Iteration 6: PCB 6 is the only PC board in the first sequenced subfamily and this PCB requires two custom setup components, c11 and c14. The usages of components c11 and c14 are equal. Component c11 with smaller index is allocated to a machine before component c14. These components are scheduled on machines based on the workload distribution in Table 4.4(e). Thus, the sets of components on machines M1 and M2 are {c6, c12, c3, c14} and {c8, c5, c10, c11}, respectively.

Table 4.4(e) Allocation of the custom setup components to machines

Components	Number of components	Workload	TW1	TW2
			8.9	6.8
c11	50	2.1	8.9	8.9
c14	50	2.1	11.0	8.9

Iteration 7: The second sequenced subfamily requires components c1, c2, c9 and c15 as semi-standard setup components. These components can be ordered by their usage shown in Table 4.5 (a). This order would be {c2,c9,c15,c1}.

Table 4.5(a) Calculation of the semi-standard usage for the second subfamily

Sequenced Subfamilies	Components			
	c1	c2	c9	c15
1	0	0	0	0
2	50	300	150	50
3	0	100	250	50
4	0	50	50	0
Total	50	450	450	100

Iteration 8: Since standard setup components are already installed on machines for this subfamily, the total workload on each machine needs to be updated before allotting semi-standard setup components. The revised workloads on machines are shown in Table 4.5(b).

Table 4.5(b) Total machine workload due to the standard setup components

Machines	Components on machines						TW1	TW2
	c3	c6	c12	c5	c8	c10		
1	2.0	0.5	0.5				3.0	
2				0.5	0.8	3.0		4.0

Iteration 9: The semi-standard setup components that are assigned to machines based on smaller machine workloads are shown in Table 4.5(c). Since there is no custom component, sets of components on the two machines for this subfamily are {c6, c12, c3, c2, c1} and {c5, c8, c10, c9, c15}, respectively.

Table 4.5(c) Allocation of the semi-standard setup component to machines

Components	Number of components	Workload	TW1	TW2
			3.0	4.0
c2	300	4.6	7.6	4.0
c9	150	3.1	7.6	7.1
c15	50	2.1	7.6	9.2
c1	50	2.1	9.7	9.2

Iteration 10: For the third sequenced subfamily, there is only one custom setup component (c14). Only c10, a semi-standard component from the previous family, needs to be replaced with c14. Therefore, the sets of components on the two machines for this subfamily are {c6, c12, c3, c2, c1} and {c5, c8, c9, c15, c14}, respectively.

Iteration 11: The fourth subfamily requires three custom setup components, c4, c7 and c13. The usages of these components are 100, 50, and 200, respectively. Thus the order of these components is {c13, c4, c7}.

Iteration 12: Before allocating these three components, the total workload due to standard and semi-standard setup components installed from the previous subfamily is updated as shown Table 4.6(a).

Table 4.6(a) Total machine workload due to the standard and semi-standard setup components for the fourth subfamily

Machines	Components on machines							TW1	TW2
	c6	c12	c1	c2	c8	c12	c9		
1	0.5	0.5	0	0.5				1.5	
2					1.0	0.5	0.5		2.0

Iteration 13: Allocating the custom component is executed as in Table 4.6(b). The resulting sets of components on the two machines are {c6, c12, c2, c1 c13} and {c8, c5, c9, c4, c7}, respectively.

Table 4.6(b) Allocation of the custom setup component to machines

Components	Number of components	Workload	TW1	TW2
			1.5	2.0
c13	200	3.6	5.1	2.0
c4	100	2.6	5.1	4.6
c7	50	2.1	5.1	6.7

For this example, the result of component allocation on each machine is summarized in Table 4.7.

Table 4.7 Result of the component allocation problem

Sequenced	Components														
Subfamilies	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	1	0	2	1	0	2	0	2	2	1	0	1	0
2	1	1	1	0	2	1	0	2	2	2	0	1	0	0	2
3	1	1	1	0	2	1	0	2	2	0	0	1	0	2	2
4	1	1	0	2	2	1	2	2	2	0	0	1	1	0	0

If a standard component is installed on a feeder slot on a machine, it will stay in that position to produce the entire family. In this example, the standard setup component, c5, is staged on machine M2 during the production of all PCBs in this

family. While a semi-standard setup component is assigned to a machine, it will stay on that machine for processing the necessary PCB subfamilies. An example is c1 that is loaded on machine M1 at the beginning of PCBs' production in the second sequenced subfamily and is kept for the next two subfamilies. A custom setup component is loaded/unloaded for individual subfamilies. For example, the custom setup component, c11, will be installed on machine M2 for producing only the first PCB subfamily.

4.3 Evaluation of the Component Allocation Problem Using Literature Data

The purpose of this experiment is to investigate the quality of the heuristic solution by comparing its' solution to a known optimal or best solution. Three literature test problems, listed in Appendix B1, are employed :

(1) Data set from Ben-Arieh and Dror (1990) consists of 10 PCBs and 25 component types. The optimal solution using an enumerative algorithm is known.

(2) Data set from Gronalt and Zeller (2000) consists of 6 PCBs and 12 component types. The heuristic solution and optimal solution are determined.

(3) Data set from Hashiba and Chang (1991) is composed of 9 PCBs and 20 components. The component similarity for this data set is high. The

optimal solution or the best solution for this allocation problem is not presented. Solving the allocation problem of this size by enumerative method was difficult due to computational requirements. Thus, only part of this test problem, using 7 PCBs and 15 components were used to find the optimal solution.

4.3.1 Description of Test Problem

These three test problems came from various sources. It would be a good idea to consider the characteristics of test problems, as these would be helpful to interpret and analyze the results. A summary of data sets is shown in Table 4.8. The density is the total number of 1s in PCB-to-component incident matrix divided by the number of PCBs multiplied by the number of components; that is the sum of 1 in all elements in the matrix divided by all elements in the matrix. The component usage is the total number of PCBs that need a particular component type and this is the sum of the matrix columns. The PCB requirement is the total number of component types demanded for a specific PCB assembly; that is the sum of the matrix rows. The average component quantity is the average amount of all components on all PCB types and equals the sum of all elements in production data matrix (m×n elements) divided by m×n. The average PCB volume is an average size of all PCB lots.

Table 4.8 Data Characteristics for test problems

	Data Sets		
	1	2	3
Number of PCBs (m)	10	6	7
Number of Components (n)	25	12	15
Density (%)	52.8	25	38.0
Minimum Component Usage	2	1	1
Maximum Component Usage	8	4	5
Average Component Usage	5.3	2.1	2.67
Minimum PCB Requirement	7	1	3
Maximum PCB Requirement	21	8	9
Average PCB Requirement	13	5	5.7
Average Component Quantity	5.25	1	7.88
Average PCB Volume	1	1	10
Number of machines	2	2	2
Setup time	-	*	*
Similarity	High	Low	High
Performance measure	Makespan	Workload difference	Makespan

* Considered

Since the best solution to the third data set is not available, the enumerative method is employed to find the optimal solution. Five the experiments with three levels of setup time and two levels of placement time on two machines, each with five feeder slots, are developed, as summarized in Table 4.9. For example, in the first case, the average setup time per feeder is less than the average placement time per board, and the placement time per component is the same on both machines. Similarly, in case 2, the averages of setup time per feeder and placement time per board are equal, and the placement time on both machines are the same. Another three cases consider other possibilities.

Table 4.9 Characteristics of experiment cases

Cases	Setup time	Placement time
1	Y < Z	T(m1) = T(m2)
2	Y = Z	T(m1) = T(m2)
3	Y > Z	T(m1) = T(m2)
4	Y = Z	T(m1) < T(m2)
5	Y = Z	T(m1) >T(m2)

Y : Average setup time per feeder

Z : Average placement time per PCB lot

T(m1), T(m2) : Placement times per component on machine 1 and 2, respectively

4.3.2 Computation and Analysis

The results for the first two data sets of Table 4.8 are summarized in Table 4.10. For the first data set, the optimal makespan is 347 units, heuristic-based literature solution is 349 units, and the solution from the heuristic developed in this research is 348 units. For the second data set, the workload difference based on a published heuristic on two machines is 6.5. The heuristic developed in this research procedure produces an identical result.

Table 4.10 Comparison of results using the first two test problems

	Data sets	
	1	2
Performance Measure	Makespan	Workload difference
Optimal Value	347	-
Heuristic Solution from Literature	349	6.5
Solution developed in this research	348	6.5

The results for the third data set are presented in Table 4.11. The makespan, which is a total production time to produce all these PCBs on the two machines, was recorded as a measure. The results show the effectiveness of the heuristic algorithm in finding a near-optimal solution using five cases of feeder setup time and component placement time. The maximum difference between the optimal and heuristic solutions is 10.76%, and the minimum difference is 5.35 %.

Table 4.11 Comparison of Results with five cases of various production scenarios

Case	Setup Time (min)	Placement time on		Makespan Using		Difference (%)
		Machine 1	Machine 2	Optimal Solution	Heuristic Solution	
1	0.46	0.01	0.01	21.60	24.09	10.76
2	1.60	0.01	0.01	32.10	34.35	7.01
3	2.50	0.01	0.01	40.20	42.35	5.35
4	1.60	0.01	0.02	38.00	41.75	10.01
5	1.60	0.02	0.01	37.20	45.00	8.45

Note: the average placement time is 1.6 min per board

4.4 Evaluation Using Industry Data

Since the component allocation problem is a NP-hard problem, solving a large-scale industry problem for an optimal solution typically requires large computation time. This section will demonstrate the application of the heuristic algorithm to solve real world problems and also examine the impact of some key parameters on the solution. Four data sets from an electronics company located in Portland, Oregon, are used here, as described in Table 4.12. The matrix density is

the sum of 1's in the matrix divided by the number of all elements (m×n) in the matrix. The PCB requirement is the total number of component types for a PCB assembly, and component usage is the total number of PCB types that demand a specific component. Table 4.12 also shows that for the four data sets, on average, a PCB requires 31 components and each component supplies 28 PCB types. Furthermore, on average, the deviations of PCB requirement and component usage are 24 and 52, respectively. The distributions of component usage (Component_USG) and PCB requirement (PCB_REQ) for data set B, are shown in Figure 4.3 (a) and 4.3 (b). These are exponentially distributed with means of 26.29 and 32.66, respectively.

Table 4.12 Characteristics of industry data sets

	Data Sets				Average
	A	B	C	D	
Number of PCB Types	744	620	608	843	704
Number of Component Types	796	770	712	826	776
Density (%)	3.79	4.24	4.27	3.68	4.00
Minimum Component Usage	1	1	1	1	1
Maximum Component Usage	350	315	289	392	337
Average Component Usage	28.23	26.29	25.94	31.06	27.88
Std. Dev. Component Usage	52.71	48.41	46.17	59.04	51.58
Minimum PCB Requirement	1	1	1	1	1
Maximum PCB Requirement	141	140	105	142	132
Average PCB Requirement	30.20	32.66	30.38	30.43	30.92
Std. Dev. PCB Requirement	24.13	23.97	24.42	24.31	24.21
Avg. Component Quantity	10.51	10.54	10.41	10.40	10.47
Avg. PCB Volume	111.70	109.13	109.45	113.04	110.83
Number of Machines	2	2	2	2	2
Number of Production Lines	3	3	3	3	3

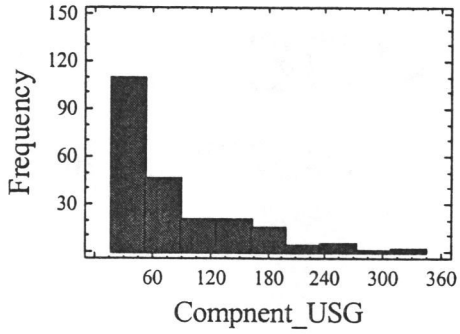


Figure 4.3 (a) Component usage distribution for industry data set B

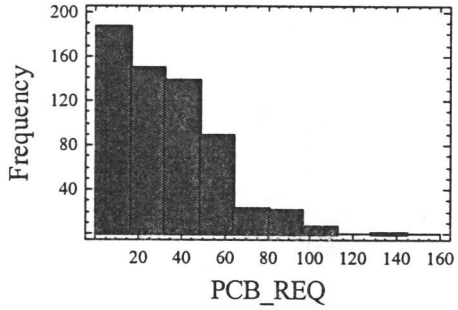


Figure 4.3 (b) PCB requirement distribution for industry data set B

With consideration of only one production line, production time generally consists of component placement time, feeder setup time, and any idle time due to the imbalance of component placement and feeder setup time among machines. The component allocation heuristic attempts to reduce imbalance in both setup and placement time. However, two parameters, threshold and feeder capacity of machines, identified earlier, would influence the allocation problem. Threshold, a key parameter in the grouping procedure, impacts size of a family. Feeder capacity

placcs limitations on subfamily size in the decomposition procedure. The effects of these parameters on the machines' imbalance are examined with 18 trials, six threshold levels, and three feeder capacity levels. The threshold levels vary between 0.05 and 0.30, in increments of 0.05. The three feeder capacity levels are 100, 150, and 300. There appears to be variations of component usage and PCB requirement distributions among data sets; thus, the impact of these variations in the 4 data sets was also considered.

Complete results are shown in Appendix B2 and summarized in Figure 4.4; statistical analysis is summarized in Table 4.13. The feeder capacity (CAPACITY) statistically affects total imbalance at a significance level of 0.05, whereas threshold (THRESHOLD) and difference in data sets (DATA_SET) do not influence the total imbalance. The two-way interactions among these factors do not impact total imbalance (IMBALANCE). The heuristic procedure works efficiently; a trial is performed, on average, in 114 seconds on a 800MHz machine. Table 4.14 shows that THRESHOLD, CAPACITY, DATA_SET, and the interaction between THRESHOLD and DATA_SET affects CPU time (CPU_TIME) at the significance level of 0.05, while the other two-way interactions are not significant.

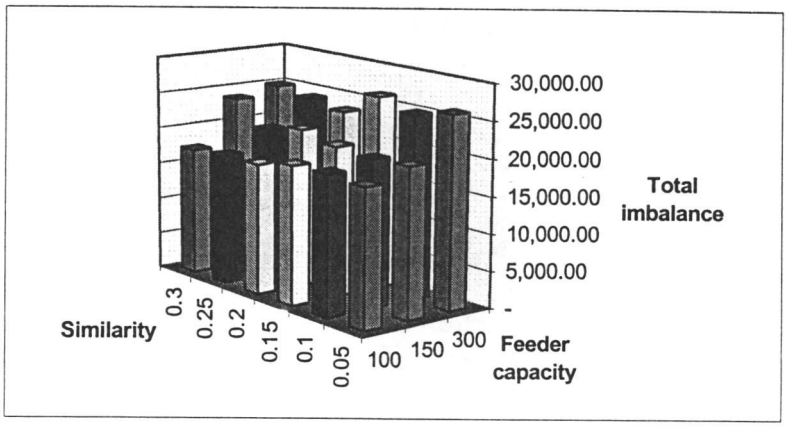


Figure 4.4 Relationship between Total imbalance, similarity, and feeder capacity

Table 4.13 ANOVA on Total Imbalance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	40	3.198E+09	78007119.7	1.16	0.3387
Error	30	2.016E+09	67204671.0		
Corrected Total	71	5.214E+09			

Source	DF	Sum of Squres	Mean Square	F Value	Pr > F
CAPACITY	2	912276039	456138019	6.79	0.0037
THRESHOLD	5	243362789	48672557.8	0.72	0.6106
DATA_SET	3	309655467	103218489	1.54	0.2255
Capacity*THRESHOLD	10	477295972	47729597.2	0.72	0.7080
CAPACITY*DATA_SET	6	379553448	63258908.0	0.94	0.4806
THRESHOLD*DATA_SET	15	876148193	58409879.5	0.87	0.6016

Table 4.14 ANOVA on CPU_TIME

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	40	94208.3056	2297.7636	98.93	<.0001
Error	30	696.8056	23.2269		
Corrected Total	71	94905.1111			

Source	DF	Sum of Squres	Mean Square	F Value	Pr > F
CAPACITY	2	10025.5278	5012.7639	215.82	<.0001
THRESHOLD	5	42313.2778	8462.6556	364.35	<.0001
DATA_SET	3	37767.8889	12589.2963	542.01	<.0001
Capacity*THRESHOLD	10	1210.3056	121.0306	5.21	0.0002
CAPACITY*DATA_SET	6	664.6944	110.7824	4.77	0.0016
THRESHOLD*DATA_SET	15	2226.6111	148.4407	6.39	<.0001

4.5 Board Assignment Procedure

Given PCB families and an assignment rule, each family will be assigned to a production line to achieve the objective of makespan minimization. Splitting subfamilies from their family to be produced on a different line may result in additional component setup time. Consequently, the production time of the split subfamilies should be re-calculated. The flow diagram for this procedure is given in Figure 4.5. In addition, to terminate the procedure, the maximum difference between the maximum and minimum completion time of the lines needs to be identified. The input to the PCB assignment is the production time of each subfamily and each family from the component allocation problem, and a scheduling rule for assigning PCB families. The outcome of the assignment

procedure is a board schedule indicating sets of board families processed on individual lines and the sequences for processing the families and PCBs with in each family. There are several traditional scheduling rules for independent-sequence job scheduling. It has been shown that the Longest Processing Time (LPT) rule works well with parallel production lines where reducing production time imbalance is an objective (Hwang, 1998), since it is easier to adjust the imbalance at the end with the production time of small families. With the objective of reducing makespan, splitting small subfamily that assigned to production line with the maximum local makespan to be produced on another line is also easier than splitting large subfamily. Consequently, a modification of the LPT is used in this research. Instead of sequencing board families in order of decreasing production time, the ratio of production time and the number of subfamilies is used as the measure.

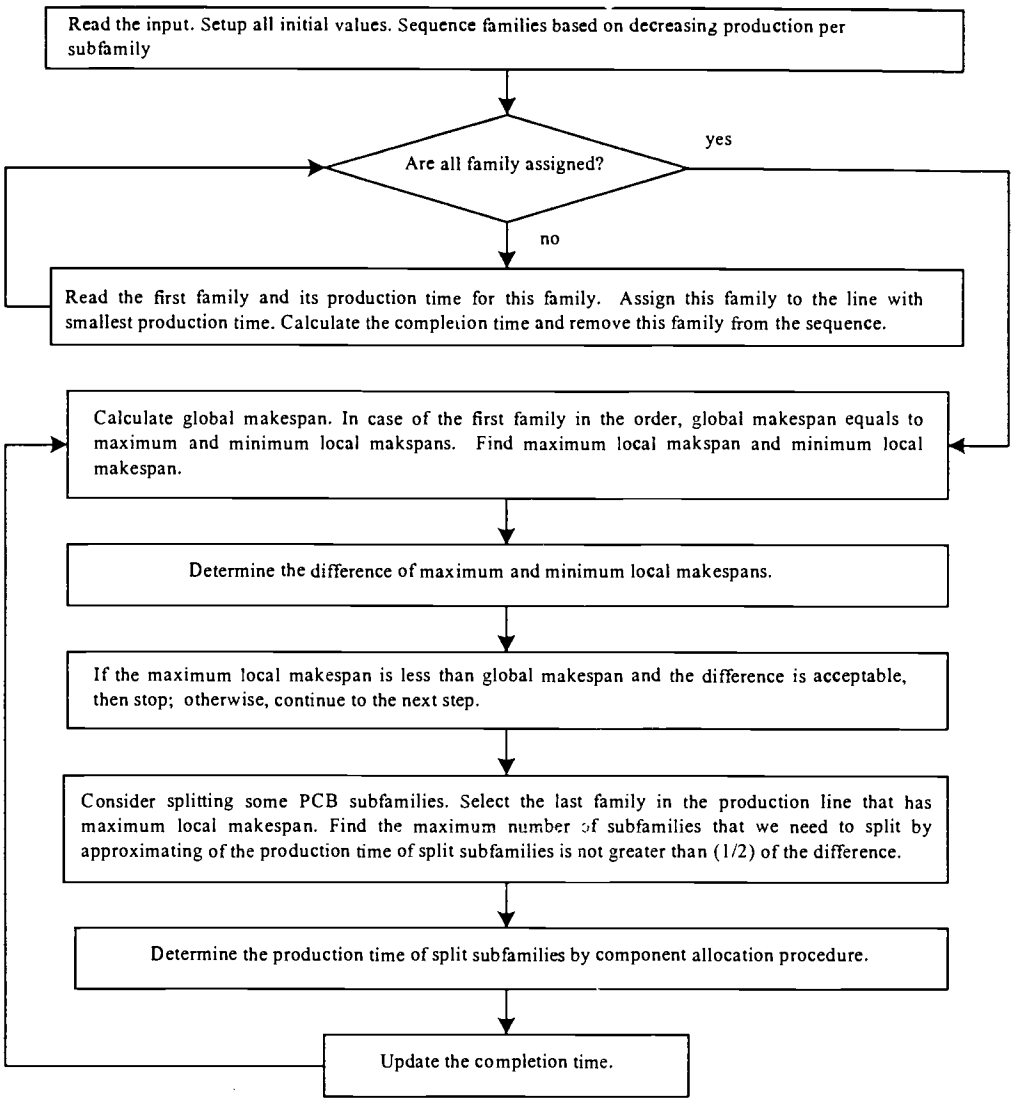


Figure 4.5 Diagram of PCB assignment procedure

An example with five PCB families shown in Table 4.15(a) is used to illustrate the sequencing process.

Iteration 1: Calculate the production time per subfamily ratio for all families and sequence families in order of decreasing value of this ratio. Based on the ratio displayed in Table 4.15(a), the family sequence is {F4, F3, F1, F2, F5}.

Table 4.15 (a) Production time of the example for assignment procedure

Families	Sequenced Subfamilies	Production time	Production time of families	Ratio
F1	s1	5.53		
	s2	2.50		
	s3	1.96		
	s4	2.63	12.62	3.16
F2	s1	4.24		
	s2	1.91		
	s3	2.09	8.24	2.75
F3	s1	4.80		
	s2	2.90		
	s3	3.12		
	s4	1.90	12.72	3.18
F4	s1	4.54		
	s2	2.29	6.83	3.42
F5	s1	3.88		
	s2	2.09		
	s3	1.27	7.24	2.41

Iteration 2: Assign PCB families in that order to the two production lines. The families are scheduled by using the LPT rule, as in Table 4.15 (b).

Table 4.15 (b) Assignment of PCB families

Families	Production Time	Assign family to		Total Production Time	
		Line 1	Line 2	Line 1	Line 2
F4	6.83	✔		6.83	0
F3	12.72		✔	6.83	12.72
F1	12.62	✔		19.45	12.72
F2	8.24		✔	19.45	20.96
F5	7.24	✔		26.69	20.96

Iteration 3: From Table 4.15 (b), the global makespan is 26.69 units of time. The difference of makespan is 5.73 units of time. This may be compared to a maximum acceptable value. For this example, assume this value to be 2.5. Thus, separating

some subfamilies from family F5 to be produced in line 2 needs to be considered. *Iteration 4*: The production time for the last subfamily in F5 is 1.27, which is less than half of the current makespan difference (i.e., 5.73/2 = 2.865). Then the production time for the last two subfamilies in F5 is 3.36, which is greater than 2.865. Therefore, only the last subfamily is rescheduled to be produced on line 2. The production time of this subfamily would then be (3+5*5*0.01+5*0.01) or 3.3. The new schedule makespan is recalculated in Table 4.15(c).

Table 4.15 (c): Assignment of PCB families

Families	Production Time	Assign family to		Total Production Time	
		Line 1	Line 2	Line 1	Line 2
F4	6.83	✔		6.83	0
F3	12.72		✔	6.83	12.72
F1	12.62	✔		19.45	12.72
F2	8.24		✔	19.45	20.96
F5-s1-s2	5.97	✔		25.42	20.96
F5-s3	3.3		✔	25.42	24.26

4.6 Chapter Summary

This chapter presented a methodology for allocating components on machines within a production line. Three categories of setup components, standard, semi-standard, and custom, were used to develop a heuristic procedure. Published data sets were then utilized to evaluate the performance of the heuristics. Four industry data sets were also used to test the applicability of the heuristic and

examine the effect of two critical parameters, similarity and feeder capacity, on the total imbalance of production time between machines. The following conclusions are drawn from these experiments:

(1) The heuristic procedure for the component allocation problem produced good results with the deviations of the heuristic solution from the optimal solution for the tested problem being between 5.4 and 10.8%.

(2) This methodology requires small computation time to arrive at a solution. Using the industry data set, the average computation time is two minutes on a 800 MHz computer (see Appendix B2).

(3) The threshold parameter does not have a statistically significant effect on the total imbalance.

(4) The various distributions of data sets do not influence the total imbalance. Distribution of data sets can be considered as blocking factor in statistical analysis.

(5) The size of the feeder capacity has a significant effect on the total imbalance. With large feeder capacity, a subfamily created by the decomposition procedure would be large. This means that the number of board types in such a subfamily is high. Allocating components to machines for all PCB types in such a subfamily with a single machine setup would balance workload difficult. Therefore, large feeder capacity

tends to have a higher total imbalance (see the results in Appendix B2).

(6) The component allocation heuristic gives priority to components with high usage for a PCB family to be assigned to machines before assigning low usage components. Consequently, such a procedure would make it easy to achieve workload balance.

(7) Assigning components to a subfamily at the custom setup component level would allow large PCB lots that have the high component requirement to be considered before the small ones. Therefore, allocating components required by a small PCB lot would be affected by the result of allocation of large PCB lots. This may create imbalance for small PCB lots. The more the number of smaller PCB lots in a subfamily, the higher the imbalance. However, if a subfamily does not contain many small PCB lots, this strategy would be useful. To understand how the effect of PCB requirement variation on imbalance, the example with 8 PCB lots in a subfamily and 12 component types is shown in Table 4.16. The quantity of each component per board is one. Based on a decreasing order of total number of all required components, PCB lots are sequenced. This means large PCB lots will have more priority to allocate their components than the small ones. Assume that components c1 and c8, which are already installed by previous subfamilies, are standard and semi-standard setup components, while the other components are custom components. The procedure for

assigning all components to two sequential machines starts with allocating components for the first PCB. For P1, components c1, c2, c3 and c4 are assigned to the first machine and components c6, c8, c10 and c12 are distributed to the second machine. Then components required by P2 are assigned. With five feeder slots on each machine component c5 is allocated to the first machine and component c7 is assigned to the second machine. Thus, for this subfamily, components c1, c2, c3, c4 and c5 are on machine one and components c6, c7, c8, c10 and c12 are assigned to machine two. All components required by other PCBs, P3 to P8, are already allocated without considering the workload for these PCBs. This creates high imbalance on these PCB lots. For example, for PCB P8, all components are on machine two.

Table 4.16 Information for 8 PCBs with 12 components

PCB	Components												Volume
In sequence	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	
P1	1	1	1	1		1		1		1		1	40
P2	1	1			1		1	1					40
P3	1	1						1					35
P4				1	1		1	1		1			20
P5						1		1		1		1	15
P6	1	1		1				1					12
P7	1	1	1							1			10
P8						1	1					1	8

In order to assign PC boards to production lines, PC board families are allocated to a production line to minimize global makespan. Splitting some

subfamilies from their parent families to be produced on a different line is acceptable as long as it does not result in an increase of the global makespan.

CHAPTER 5 EVALUATION OF METHODOLOGY

In the previous chapter, procedures were developed to assign all required components for each PCB to machines in a production line. As a result, the production time for each board family can be computed and families can be assigned to production lines. In this chapter, the results of the integrated methodology, which include the Board assignment plan, the Scheduling plan and the Feeder setup plan, are addressed. To investigate the effectiveness of the integrated methodology, published data with the optimal or best solution is utilized so the results can be compared with those from the integrated methodology. Industry data is also employed to examine the performance of the integrated methodology and explore the effects of the two main parameters, threshold value and feeder capacity, on the results. Furthermore, generated data is exploited to examine the impacts of the two primary parameters as well as the variations of PCB requirements and component usage on makespan and CPU time.

5.1 Integrated Methodology Results

The integrated methodology consists of seven procedures—PCB grouping, family decomposition, subfamily sequencing, KTNS, component setup type determination, component allocation, and board family assignment. The input

information of the methodology is composed of (1) the PCB-to-component incidence matrix, (2) production data including quantity of individual component on each PC board type and PCB production volume, and (3) production environment consisting of the feeder setup and placement time, feeder capacity of each machine, and production line configuration. In addition, parameters such as the threshold values need to be specified.

The results from this methodology are separated into three plans: PCB assignment plan, Scheduling plan, and Feeder setup plan.

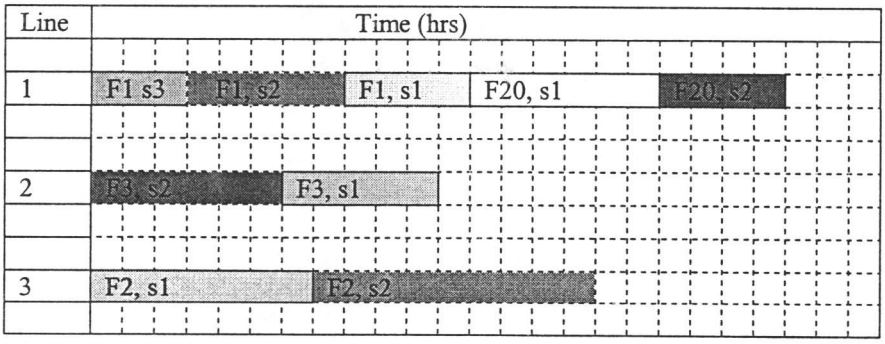
(1) The Board assignment plan provides information for each PC board type to be processed on a specific line. This plan will be used with the scheduling and feeder setup plans to produce PC boards during a planning horizon. The board assignment plan, shown in Table 5.1 consists of PCB types, families, sequenced subfamilies, and board assignment to production lines.

(2) The Scheduling plan determines the processing time for PC boards (in Table 5.2). This plan gives the detail of shop floor production. The PCB assignment plan (Table 5.1) along with the scheduling plan (Table 5.2) gives information about (a) when each PCB is to be produced, (b) the production time of individual PCBs, and (c) the production line to process a particular PCB.

Table 5.1 Board assignment plan

PCB Types	Family	Sequenced	Assign PCB to Production lines		
		Subfamily	Line 1	Line 2	Line 3
P1	F1	3	✔		
P2	F2	1			✔
P3	F2	2			✔
P4	F1	1	✔		
P5	F3	1		✔	
P6	F3	2		✔	
P7	F1	2	✔		
P8	F4	3		✔	
.	.	.			
.	.	.			
.	.	.			
P618	F20	1	✔		
P619	F22	4			✔
P620	F25	2			✔

Table 5.2 Scheduling plan



(3) The Feeder setup plan is used for setting up component feeders on machines. For example, in Table 5.3, when PCBs in the family F1 are produced, components c1, c3 and c8 will be loaded on Machine 1, and components c2, c5 and c13 on Machine 2. Components c1 and c5 are standard setup components for family F1. These components will stay

on machines for producing the entire family. Components c9 and c11 are semi-standard setup components for subfamilies s2 and s1, respectively; thus, they will be loaded at the beginning of producing the PCB subfamily s2 and unloaded at the end of producing subfamily s1.

Table 5.3 Feeder setup plan

Families	Subfamilies	Components												
		1	2	3	4	5	6	7	8	9	10	11	12	13
F1	s3	1	2	1		2			1					2
	s2	1			1	2				2		1	2	
	s1	1				2		1		2		1		2
F2	s1		2		1		1	2			2		1	
	s2		2		1			2	1			2	1	
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
F25	s1	1		1					2			2		1
	s3		2	1		1			2		1	2		
	s2		2	1		1	2		2				1	

5.2 Evaluation Using Literature Data

It is difficult to identify a problem from published literature that spans the entire integrated methodology. Consequently, this research tests part of the methodology that encompasses grouping, decomposition, sequencing and Keep Tool Needed Soonest (KTNS) procedures. The test problem is taken from Hashiba and Chang (1991). The PCB-to-component matrix with 9 PCBs and 20 components is displayed in Table 5.4. With the threshold value of 0.60 for the grouping

procedure, all the PCBs can be classified into the two families as indicated in Table 5.5. All PCBs produced are scheduled on one machine with 14 feeder slots in a production line.

Table 5.4 PCB-to-component matrix

PCBs	Components																			
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1
2	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1	0	0	0	1
3	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0
4	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
5	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1	0	0	1	1	0
6	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1
7	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0
8	0	0	1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	0	1	0
9	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 5.5 PCB families

Families	PCBs						
F1	2	3	4	5	7	8	9
F2	1	6					

Because of the feeder slot capacity constraint, only 14 component types can be installed on machines at one time. The family F1 requires more than 14 component types; thus, this family needs to be decomposed into subfamilies by using the family decomposition procedure. With family F2 the number of component types required is less than the feeder capacity; consequently, the decomposition was not used for this family. Four subfamilies resulted from the decomposing family F1 (Table 5.6).

Table 5.6 Family decomposition for family F1

Subfamilies	PCBs	Number of required feeders	Number of available feeders
s1	5, 7	10	4
s2	2	12	2
s3	3, 4, 9	12	2
s4	8	10	4

In order to reduce the number of feeder setups, the subfamilies need to be ordered using the subfamily sequencing procedure. The resulting sequence of these subfamilies is s4-s1-s3-s2 with 42 feeder setups. Table 5.6 shows the number of available feeders. The KTNS process can be applied to reduce unnecessarily loaded/unloaded feeders. The result of KTNS in Table 5.7 exhibits that the total number of loaded/unloaded feeder setups is 30 and the total number of loaded feeder setup is 22. This result exactly equals to the result given in Hashiba and Chang (1991). This indicates that parts of the integrated methodology tested here perform very favorably.

Table 5.7 Result of KTNS

Compo-nents	Family 1				Family 2	Number of loaded/unloaded Setups	Number of loaded Setups
	s4	s1	s3	s2			
1	0	1	**1**	**1**	1	1	1
2	0	1	1	1	**1**	1	1
3	1	1	1	0	0	2	1
4	0	0	0	1	1	1	1
5	0	1	1	1	1	1	1
6	1	1	1	1	**1**	1	1
7	1	1	1	1	1	1	1
8	0	0	0	1	1	1	1
9	1	1	1	1	**1**	1	1
10	0	1	1	1	**1**	1	1
11	1	1	0	0	0	2	1
12	1	0	0	0	1	3	2
13	0	1	1	1	1	1	1
14	1	1	**1**	1	0	1	1
15	1	1	0	0	0	2	1
16	0	0	0	1	1	1	1
17	1	0	1	0	0	4	2
18	0	1	1	0	0	2	1
19	1	1	1	**1**	1	1	1
20	0	0	1	1	1	1	1
Total setups	10	14	14	14	14	30	22

Note: the Number of setups is the number of loaded and unloaded feeder slots.

5.3 Evaluation Using Industry data

The board assignment objective is to minimize global makespan. Global makespan is the completion time of all PC boards produced during the planning horizon or the maximum of total production time for all board types. CPU time also reflects the efficiency of the methodology. Consequently, the performance measures of the integrated methodology are makespan and CPU time.

The effects of the threshold parameter and feeder capacity on global makespan (MAKESPAN) and CPU time (CPU_TIME) were also investigated with the four industry data sets used in Chapter 4 (shown in Table 4.12). Six levels of the threshold and three levels of the feeder capacity are utilized for all industry data sets. The ANOVA table for MAKESPAN and CPU_TIME are presented in Tables 5.8 and 5.9, respectively. Tables 5.8 and 5.9 illustrate that only DATA_SET has an influence on both MAKESPAN and CPU_TIME at the 0.05 significance level. CAPACITY influences only MAKESPAN, while THRESHOLD affects CPU_TIME. The two-way interactions among these factors do not impact MAKESPAN, but interaction between THRESHOLD and DATA_SET does impact on CPU_TIME.

The statistical significance of DATA_SET on MAKESPAN and CPU_TIME indicates that the variation of industry data sets would have an impact on the two measures. Tables 5.10 and 5.11 report effects of the two factors on MAKESPAN and CPU_TIME for individual data set. These mean that the impacts of THRESHOLD and CAPACITY vary among data sets. In addition, the sum of squares for DATA_SET in Table 5.8 is very high compared to that of THRESHOLD and CAPACITY. This implies that the model may be missing other influential variables.

Table 5.8 Result of ANOVA on MAKESPAN

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	41	1.775E+09	43288298.9	20.9	<.0001
Error	30	62145468.4	2071515.61		
Corrected Total	71	1.837E+09			

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
CAPACITY	2	24375597.6	12187798.8	5.88	0.0070
THRESHOLD	5	10080743.4	2016148.67	0.97	0.4499
DATA_SET	3	1.686E+09	561952461	271.28	<.0001
CAPACITY*THRESHOLD	10	21562379.5	2156237.95	1.04	0.4350
CAPACITY*DATA_SET	6	1927771.37	321295.229	0.16	0.9865
THRESHOLD*DATA_SET	15	31016379.7	2067758.65	1.00	0.4816

Table 5.9 Result of ANOVA on CPU_TIME

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	41	881085.009	21489.8783	7.96	<.0001
Error	30	81020.5580	2700.6853		
Corrected Total	71	962105.567			

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
CAPACITY	2	114.6298	57.3149	0.02	0.9790
THRESHOLD	5	93187.0198	18637.4040	6.90	0.0002
DATA_SET	3	513425.410	171141.803	63.37	<.0001
CAPACITY*THRESHOLD	10	43476.4178	4347.6418	1.61	0.1516
CAPACITY*DATA_SET	6	18422.9957	3070.4993	1.14	0.3654
THRESHOLD*DATA_SET	15	212458.535	14163.9024	5.24	<.0001

Table 5.10 Effects of CAPACITY and THRESHOLD on MAKESPAN for individual data set at significance level of 0.05

Data Set	CAPACITY effect	THRESHOLD effect	Interaction*
A	Yes	No	Yes
B	No	No	Yes
C	No	No	No
D	Yes	No	No

* Using Tukey Test (Miller, 1986)

Table 5.11 Effects of CAPACITY and THRESHOLD on CPU_TIME for individual data set at significance level of 0.05

Data Set	CAPACITY effect	THRESHOLD effect	Interaction
A	No	Yes	No
B	No	No	No
C	Yes	Yes	No
D	No	Yes	Yes

5.4 Experiment and Analysis

The conclusion in the previous section suggests that the difference in data sets affects makespan and CPU time. Variations of the PCB requirement and component usage are two major characteristics of data sets (Li, 1999). As addressed in Chapter 4, the PCB requirement is the total number of component types for a PCB assembly, and component usage is the total number of PCB types that demand

a specific component. The distributions of Component usage (Comp_USG) and the PCB requirement (PCB_REQ) for data set B, as an example, duplicated from Figure 4.3(a) and (b), are shown in Figure 5.1 (a) and (b), respectively. Component usage and PCB requirement are exponential distributed with means of 26.29 and 32.66, respectively.

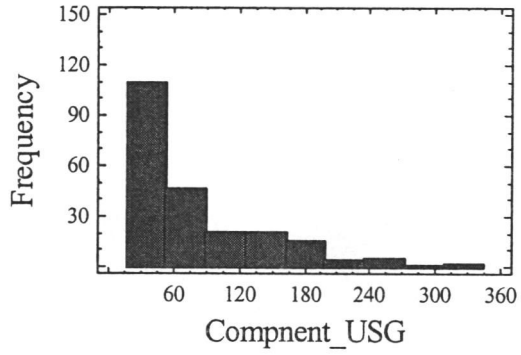


Figure 5.1 (a) Component usage distribution for industry data set B

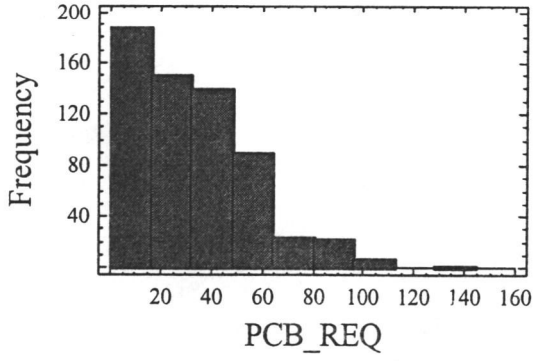


Figure 5.1 (b) PCB requirement distribution for industry data set B

To understand the performance of the integrated methodology in different production environments, simulated data that would represent a range of industry situations is generated. The PCB requirement and component usage variations should be considered in the experiment.

5.4.1 Experimental Design on Generated Data

In order to generate the simulated data, values of two primary parameters, the component usage variation and the PCB requirement variation are controlled. The combination of these two parameters is used to develop the four cases of generated data. With two levels of each parameter, the four cases are: (1) HH, high component usage and high PCB requirement variation, (2) HL, high component usage but low PCB requirement variation, (3) LH, low component usage but high PCB requirement variation, and (4) LL, low component usage and low PCB requirement variation. Other parameters that are controlled are:

(1) Number of PCBs
(2) Number of components
(3) Matrix density
(4) Maximum, minimum and mean of component usage
(5) Maximum, minimum and mean of PCB requirement
(6) Average component quantity
(7) Average PCB volume
(8) Number of machines and production lines

To preclude the impact of the above eight parameters, the experimented data are generated with specific values for these parameters. The industry data set B, selected randomly from the four industry data sets, is used in the analysis. Thus, the size of PCB-to-component incidence matrix, matrix density, average component quantity, average PCB volume, and the number of machines and production lines are set at the same value as in data set B.

Two replicates of the four cases of component usage and PCB requirement deviations are generated with different random number seeds as shown in Table 5.12. The maximum component usage can not be more than the number of the PCBs (620 in this case). The maximum number of PCB requirement also can not exceed the number of components (770 in this case). Both component usage and the PCB requirement means are set at the same value as in data set B. The high levels of component usage (Comp USG) and PCB requirement (PCB REQ) standard deviations are set to 150% of these standard deviations. The low level of Comp USG and PCB REQ standard deviations are controlled at 50% of the deviations.

Table 5.12 Characteristic of simulation data

	Experiments							
	HH1	HL1	LH1	LL1	HH2	HL2	LH2	LL2
# PCBs	620	620	620	620	620	620	620	620
# Components	770	770	770	770	770	770	770	770
Desity (%)	4.24	4.24	4.24	4.24	4.24	4.24	4.24	4.24
Min Comp USG	1	1	1	1	1	1	1	1
Max Comp USG	227	227	128	118	252	255	135	135
Avg. Comp USG	26.29	26.29	26.29	26.29	26.29	26.29	26.29	26.29
Std. Dev. Comp USG	36.32	36.31	20.63	19.3	36.31	39.41	20.10	20.18
Min PCB REQ	1	1	1	2	1	1	1	1
Max PCB REQ	132	68	145	73	172	70	164	70
Avg. PCB REQ	32.65	32.65	32.65	32.65	32.65	32.65	32.65	32.65
Std. Dev. PCB REQ	26.33	11.65	26.89	12.15	27.43	11.65	28.25	11.81
Avg. Comp Quantity	10.46	10.46	10.46	10.46	10.46	10.46	10.46	10.46
Avg. PCB Volume	106.0	111.9	112.7	110.9	111.4	112.3	114.6	110.1

Analysis in section 5.3 indicated that the threshold and feeder capacity per line would have an impact on the response variables. Therefore, the experiment of a 2^4 full factorial design with two levels of four factors—Component usage deviation (Comp USG_DEV), PCB requirement deviation (REQ_DEV), threshold value (THRESHOLD), and feeder capacity (CAPACITY), is developed (Table 5.13). As mentioned earlier, the main performance measures for the integrated methodology are global makespan and CPU time. Consequently, the following section will evaluate the effects of four factors on these two performance measures.

Table 5.13 Experimentation of 2^4 full factorial design

No. Experiments	Component usage Variation	PCB requirement variation	Threshold	Feeder Capacity
1	High (1)	High (1)	Low(-1)	Low(-1)
2	High (1)	High (1)	Low(-1)	High (1)
3	High (1)	High (1)	High (1)	Low(-1)
4	High (1)	High (1)	High (1)	High (1)
5	High (1)	Low(-1)	Low(-1)	Low(-1)
6	High (1)	Low(-1)	Low(-1)	High (1)
7	High (1)	Low(-1)	High (1)	Low(-1)
8	High (1)	Low(-1)	High (1)	High (1)
9	Low(-1)	High (1)	Low(-1)	Low(-1)
10	Low(-1)	High (1)	Low(-1)	High (1)
11	Low(-1)	High (1)	High (1)	Low(-1)
12	Low(-1)	High (1)	High (1)	High (1)
13	Low(-1)	Low(-1)	Low(-1)	Low(-1)
14	Low(-1)	Low(-1)	Low(-1)	High (1)
15	Low(-1)	Low(-1)	High (1)	Low(-1)
16	Low(-1)	Low(-1)	High (1)	High (1)

5.4.2 Experimental Results

By using generated data with the 2^4 full factorial design, the result of 32 experiments on the integrated methodology procedures is presented in Appendix

C2. Regression analysis is applied to analyze the significance of factors and predict the impact of these factors on response variables. Because the plot of residual versus the global makespan and the CPU time showed violation of the normality assumption, the response variables were transformed into the log scale.

5.4.2.1 Analysis of Four Effects on Global Makespan

The result of the ANOVA and the regression analysis for log (MAKESPAN) is displayed in Tables 5.14 and 5.15, respectively. Table 5.14 illustrates that three main factors, REQ_DEV, USG_DEV, and THRESHOLD, as well as the two- and three-way interactions between these three significant factors have p-values less than 0.05, whereas CAPACITY and the interactions between CAPACITY and the other three factors have p-value greater than 0.05. Thus, it is concluded that these factors, which are REQ_DEV, USG_DEV, and THRESHOLD, and their interactions influence log (MAKESPAN) at the same significance level. In Table 5.15, the R-square statistic denotes that the fitted model can explain 87.7% of total variability of log (MAKESPAN). The final fitted model on log (MAKESPAN) is:

Log (MAKSPAN) = 10.4904-0.0585 REQ_DEV+0.0496 USG_DEV
- 0.0246 THRESHOLD - 0.0582USG_DEV*REQ_DEV
- 0.0246 USG_DEV*THRESHOLD
+ 0.0243 REQ_DEV*THRESHOLD
+ 0.0243 USG_DEV*REQ_DEV*THRESHOLD

Table 5.14 ANOVA of full model on log (MAKESPAN)

Analysis of Variance for LOG_MAKESPAN

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:REQ_DEV	.109647	1	.109647	35.05	.0000
B:USG_DEV	.0786174	1	.0786174	25.13	.0001
C:CAPACITY	.0011715	1	.0011715	.37	.5492
D:THRESHOLD	.0194326	1	.0194326	6.21	.0240
E:USG_DEV*REQ_DEV	.108332	1	.108332	34.63	.0000
F:USG_DEV*THRESHOL	.0194213	1	.0194213	6.21	.0241
G:USG_DEV*CAPACITY	.0000160671	1	.0000160671	.01	.9438
H:REQ_DEV*THRESHOL	.0188764	1	.0188764	6.03	.0258
I:REQ_DEV*CAPACITY	.0000595607	1	.0000595607	.02	.8920
J:THRESHOLD*CAPACITY	.000143556	1	.000143556	.05	.8331
K:USG_DEV*REQ_DEV*THR	.0188876	1	.0188876	6.04	.0258
L:USG_DEV*REQ_DEV*CAP	.000506308	1	.000506308	.16	.6928
M:USG_DEV*THRESHOL*CAP	.000142582	1	.000142582	.05	.8336
N:REQ_DEV*THRESHOL*CAP	.000211069	1	.000211069	.07	.7984
O:USG_DEV*REQ_DEV*CA	.000212253	1	.000212253	.07	.7978
RESIDUAL	.0500514	16	.00312821		
TOTAL (CORRECTED)	.425728	31			

Table 5.15 Final fitted model on log (MAKESPAN)

Multiple Regression Analysis

Dependent variable: LOG_MAKESPAN

Parameter	Estimate	Standard Error	T Statistic	P-Value
CONSTANT	10.4904	0.0082691	1268.63	0.0000
REQ_DEV	-0.058536	0.0082691	-7.07888	0.0000
USG_DEV	0.0495661	0.0082691	5.99413	0.0000
THRESHOLD	-0.0246428	0.0082691	-2.98011	0.0065
USG_DEV*REQ_DEV	-0.0581839	0.0082691	-7.03631	0.0000
USG_DEV*THRESHOLD	-0.0246356	0.0082691	-2.97924	0.0065
REQ_DEV*THRESHOLD	0.0242876	0.0082691	2.93715	0.0072
USG_DEV*REQ_DEV*T	0.0242948	0.0082691	2.93802	0.0072

Analysis of Variance

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	0.373214	7	0 0533162	24.37	0.0000
Residual	0.0525143	24	0.00218809		
Total (Corr.)	0.425728	31			

R-squared = 87.6648 percent
R-squared (adjusted for d.f.) = 84.0671 percent

Figure 5.2 shows that with a low PCB requirement and high usage variations, the high threshold value can reduce 17.8% of the makespan with the low value. Consequently, in the case of a low PCB requirement, but high component usage variations, using a high threshold value in the integrated methodology procedures would result in a global makespan reduction compared to using a low value.

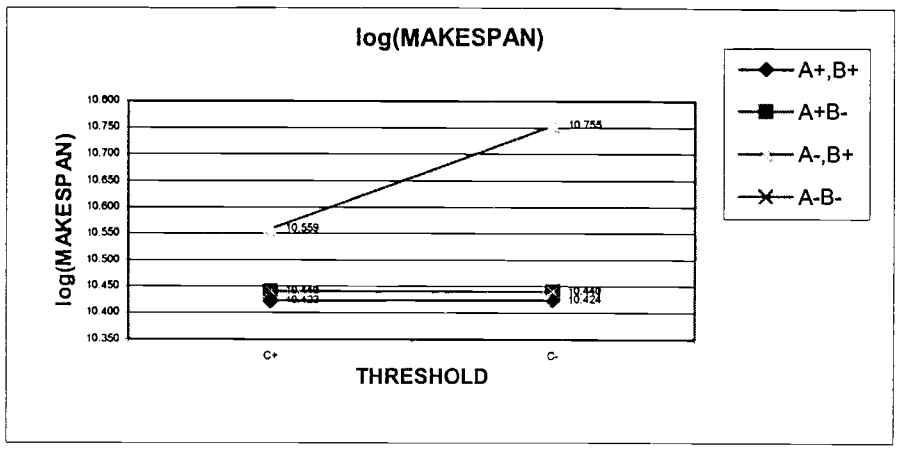


Figure 5.2 Three-way interactions among REQ_DEV(A), USG_DEV(B) and THRESHOLD(c)

5.4.2.2 Analysis of Four Effects on CPU Time

The result of the ANOVA and regression analysis for log (CPU_TIME) is illustrated in Tables 5.16 and 5.17. In Table 5.16, only one main factor, REQ_DEV, and the two-way interaction between REQ_DEV and USG_DEV have p-values less than 0.05. Even though USG_DEV does not influence log (CPU_TIME), the interaction between REQ_DEV and USG_DEV is significant. Thus, it is meaningful to include the USG_DEV factor in the model. In Table 5.17, the R-square statistic indicates that the fitted model can explain 67.80% of total variability in log (CPU_TIME). The final fitted model on log (CPU_TIME) is:

$$\text{Log(CPU\_TIME)} = 4.983 - 0.0238\,\text{USG\_DEV} + 0.155\,\text{REQ\_DEV} - 0.102\,\text{USG\_DEV}*\text{REQ\_DEV}$$

Figure 5.3 displays the impact of interaction between REQ_DEV and USG_DEV on CPU_TIME. At low USG_DEV, the difference of REQ_DEV levels has the most impact on CPU_TIME (40.2%).

Table 5.16 ANOVA of full model on log (CPU_TIME)

Analysis of Variance for LOG_CPU_TIME

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:REQ_DEV	1.46127	1	1.46127	12.80	.0025
B:USG_DEV	.21516	1	.21516	1.88	.1888
C:CAPACITY	.122958	1	.122958	1.08	.3149
D:THRESHOLD	.013762	1	.013762	.12	.7330
E:USG_DEV*REQ_DEV	.823624	1	.823624	7.21	.0162
F:USG_DEV*THRESHOL	.0119828	1	.0119828	.10	.7502
G:USG_DEV*CAPACITY	.204822	1	.204822	1.79	.1992
H:REQ_DEV*THRESHOL	.00106164	1	.00106164	.01	.9244
I:REQ_DEV*CAPACITY	.61704	1	.61704	5.40	.3336
J:THRESHOLD*CAPACITY	.00617304	1	.00617304	.05	.8191
K:USG_DEV*REQ_DEV*THR	.0028128	1	.0028128	.02	.8773
L:USG_DEV*REQ_DEV*CAP	.110294	1	.110294	.97	.3404
M:USG_DEV*THRESHOL*CAP	.0116406	1	.0116406	.10	.7537
N:REQ_DEV*THRESHOL*CAP	.00215123	1	.00215123	.02	.8925
O:USG_DEV*REQ_DEV*CA	.000319876	1	.000319876	.00	.9584
RESIDUAL	1.82727	16	.114204		
TOTAL (CORRECTED)	5.43234	31			

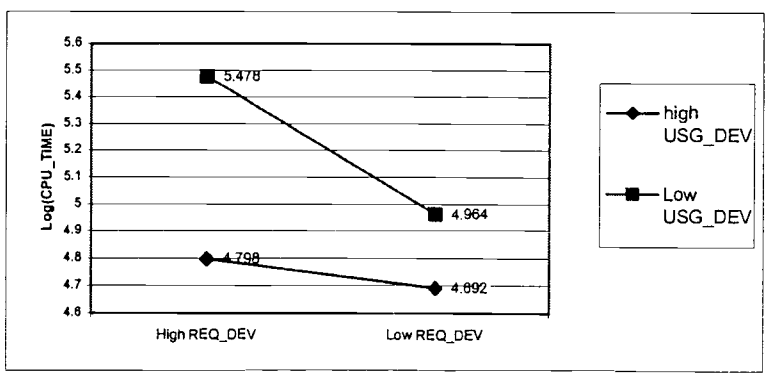


Figure 5.3 Interaction between REQ_DEV and USG_DEV on log (CPU_TIME)

Table 5.17 Final fitted model for log (CPU_TIME)

Multiple Regression Analysis

Dependent variable: LOG_CPU_TIME

Parameter	Estimate	Standard Error	T Statistic	P-Value
CONSTANT	4.98343	0.0244298	203.99	0.0000
USG_DEV	-0.0237861	0.0244298	-0.973653	0.3386
REQ_DEV	0.155481	0.0244298	6.36439	0.0000
REQ_DEV*USG_DEV	-0.102219	0.0244298	-4.1842	0.0003

Analysis of Variance

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	1.12604	3	0.375346	19.65	0.0000
Residual	0.534744	28	0.019098		
Total (Corr.)	1.66078	31			

R-squared = 67.8017 percent
R-squared (adjusted for d.f.) = 64.3519 percent

5.5 Chapter Summary

The integrated methodology consisting of seven sequential procedures generates three plans: PCB assignment plan, Scheduling plan, and Feeder setup plan. These plans are useful for process planners in the electronics industry. Based on the performance evaluations conducted using the industry data sets in the chapter, the following conclusions are drawn:

(1) For individual family, all required components have to be installed. Thus, the number of feeder setups is independent from the prior family

being produced. In reality the number of setups for a family can be reduced if it shares common components with the previous family, therefore, the makespan here is the "upper bound makespan".

(2) Feeder capacity per production line impacts the makespan. If the feeder capacity is increased, the size of a subfamily would increase as well, whereas the number of subfamilies would decrease. This would lead to increased unbalance, which is a part of makespan.

(3) There is an impact of the threshold value for the grouping procedure on CPU time, but no effect on the global makespan.

(4) The industry data influences the two performance measures, global makespan and CPU time. This implies that characteristics of data sets may impact output measures. It would be a good idea to consider some characteristics of industry data. Furthermore, high variability due to the difference in data sets on global makespan suggests that other factors may impact dependent variables.

Four cases of the combinations between component usage and PCB requirement variations were examined. A 2^4 factorial experiment with one replicate was developed to analyze the integrated methodology. The stepwise regression method is employed to analyze the effect of the main factors and their interactions on the global makespan and CPU time.

The following conclusions can be drawn from this experiment:

(1) The variations of component usage and the PCB requirement affect the global makespan. High component usage deviation tends to increase the makespan. On the other hand, high PCB requirement usage would lead to reducing the makespan.

(2) The use of a high threshold value (~0.25) would decrease the makespan. Using a high threshold value in the grouping procedure would create small families. Assigning small families to production lines to minimize global makespan is easier to accomplish than with larger families.

(3) The combinations among threshold value, board requirement variation, and component usage variation also impact global makespan. In low board requirement and high component usage variation environments, the use of a high threshold can reduce global makespan. A low requirement variation reflects a small difference in a number of component types required by each PCB type. A high component usage variation indicates a large difference in a number of PCB types demanding individual component types. A high component usage means that many component types appear in most PCBs. A low component usage means that many component types are placed on a few PCB types. Consequently, with high component usage variation, many components can be identified as the standard setup components. The component

allocation procedure would result in significant reductions in setup times with a large number of standard setup components.

(4) Feeder capacities on machines do not have a significant impact on the global makespan and CPU time. When feeder capacity is large, a subfamily might contain many PCB types. Allocating components to machines within a production line for such a subfamily would be difficult to balance workload due to placement time for each PC board type. In particular for the last board type in a subfamily, assigning its components to machines has to be considered the other components that already installed. This situation would lead to an increase in total workload imbalance. However, for the larger capacity, the size of a subfamily is bigger and then the number of setups would be smaller. The total workload imbalance due to placement time and feeder setup time is a part of makespan. Consequently, feeder capacity cannot have an effect on the global makespan.

(5) Component usage and PCB requirement variations influence CPU time, as does the combination of the two. The high usage variation would reduce CPU time, while the high requirement variation would increase CPU time.

CHAPTER 6 SUMMARY AND CONCLUSIONS

The focus of this research was to develop a methodology to solve the PCB assignment and component allocation problems. The following sections describe the conclusions of this study and suggest areas for future research.

6.1 Summary

This research seeks to answer two questions: How to allocate component types to machines and how to assign PCB to production lines. The complexity of the problem is a primary consideration. Solving for optimality will obviously give the best results; however, this is not usually feasible in practice. The purpose of this research is to develop a methodology to provide answers to the two questions in a relatively effective and efficient manner. The integrated methodology consists of seven interconnected procedures—PCB grouping, family decomposition, subfamily sequencing, Keep Tool Needed Soonest (KTNS), component setup type determination, component allocation, and PCB family assignment.

(1) PCB grouping: Based on the Group Technology (GT) concept, grouping algorithm from literature is used to reduce the problem size. Component similarity between PCBs is used as the criterion for making families.

(2) Family decomposition: Because of feeder capacity constraints, the maximum number of feeder slots that can be assigned to a PCB family can not exceed feeder capacity of a production line. The family decomposition procedure divides a family into subfamilies so that each subfamily requirement for feeder slots does not exceed the feeder capacity. Jaccard similarity and threshold values are used in the criterion to select the next PCB to join a subfamily.

(3) Subfamily sequencing: The purpose of this procedure is to sequence subfamilies within a family so that the number of feeder setups is minimal. The Best First Search algorithm is applied to find an optimal sequence with the goal of feeder setup minimization.

(4) KTNS: In order to minimize feeder setups, the KTNS procedure is employed to identify components that should be left on unused feeder slots for producing the current board type.

(5) Setup component type determination: This procedure classifies the setup component types into three categories: standard, semi-standard, and custom setup components. The result of this procedure is useful for the component allocation procedure.

(6) Component allocation: The purpose of this procedure is to minimize workload imbalance. A heuristic algorithm for the component allocation problem is developed.

(7) Family assignment: By assuming that the production time of each family is sequence independent, the result of the component allocation procedure is used to find the production time of each family and subfamily. Modification of the Longest Processing Time (LPT) scheduling rule based on production time per subfamily is applied.

To evaluate the performance of the component allocation algorithm, two data sets from the literature with know optimal solutions were used. The results indicate that the solutions from the algorithm matched the results from the optimal solution and were better than the result reported in Ben-Arieh and Dror (1990) using a different heuristic procedure. For larger-scale problems, an enumerative method was utilized to obtain the optimal solutions to relatively large data sets in various production environments, (e.g., different component placement time and setup time on each machine). The results of the allocation procedure differ from the optimal solution within 10.8 %. In addition, application of the component allocation procedure to industry data was also analyzed with the total imbalance due to setup time and placement time of individual PCB as the performance measure.

Assessing the performance of the entire methodology approach may not be fully possible since the literature reports no data sets in literature was identified that would fit all dimensions of the problem. Consequently, only the first four parts of

the methodology were investigated with the literature data. The results indicate that the performance of these parts of the methodology to be quite good.

The results for component allocation and integrated methodology using the industry data suggest that the variation of the industry data characteristics impacts the performance measures. Simulation data was generated based on an industry data set. The simulated data was used to evaluate the relative importance of the PCB characteristics, threshold value, and feeder capacity per production line on the makespan and computation time. Statistical analyses of this experiment reveal that when PC boards in a planning period have a low PCB requirement variation and a high component usage variation, a high threshold value can significantly reduce makespan. However, the feeder capacity would not impact makespan. In addition, only PCB characteristics, PCB requirement, and component usage variations affect the computation time. At low component usage variations, the difference in PCB requirement variation levels is important in reducing CPU time.

6.2 Conclusions

The integrated methodology presented in this dissertation is the first approach to concurrently consider both component allocation to machines within a production line and PCB assignment to the production lines for a large problem. The solution methodology developed can obtain the upper bound global makespan

in a reasonable amount of computation time. The component allocation heuristic procedure proposed in this research also gave a good solution to total workload imbalance for a problem with industry dimensions. The result of the methodology, including PCB assignment, scheduling and feeder setup plans, would be useful for process planners in industry. Additionally, the classification of feeder setup components applied in this methodology would be applicable in reality to structure the feeder setup plan.

From this study, three main conclusions can be drawn:

1. Feeder capacity has an impact on total workload imbalance but not on global makespan. With larger feeder capacity, allocating components to machines within a production line for each subfamily would increase the workload imbalance for individual PCB types, while the number of feeder setups would be smaller. Both total workload imbalances due to placement time and feeder setup time are part of makespan. Consequently, feeder capacity does not influence global makespan. This result suggests that increasing feeder capacity in a production line would not help to reduce makespan.
2. Threshold value has a significant effect on the global makespan. A high threshold value can decrease global makespan. The use of a high threshold in family grouping procedure will create small board families. The production time of these families would then be small. Assigning each of

these small board families to a production line with the purpose of global makespan minimization can be achieved easier relative to assigning the large board families.

3. The combinations among threshold value PCB requirement variation and component usage variation also affect global makespan. In a low board requirement and high component usage variation environment, the use of a high threshold can reduce global makespan. For other levels of this combination, the impact of high threshold value on the makespan is quite small. It can be concluded that the use of a high threshold would be beneficial for makespan minimization.

As discussed in Chapter 1, it is not feasible to consider all possible PCB production environments in this research. Some of the assumptions being made in the study place limits on the usefulness of the results. Significant among these are:

1. A component feeder can occupy only one feeder slot on any given machine. In the family decomposition procedure, a PCB family is divided into subfamilies so that the number of required component types for each subfamily is less than the total number of feeder slots on a production line. Each component type can take only one feeder slot and can be installed on any machine. Practically, the size of components varies depending on their function and packaging. A component type may take more than one feeder slot and also can be installed only on a specific machine.

2. In order to produce a PCB type, each required component can be installed on only one machine. Even if the required component is a high usage component, it cannot be duplicated on another machine. As discussed above, high component usage variation influences the global makespan. Allowing a high usage component to be loaded on more than one machine would reduce the workload imbalance due to placement time among machines. This would lead to makespan reduction. However, the number of feeder setups would increase and the tradeoffs would have to be explored.
3. This research assumes that the production lines are identical in terms of feeder capacity and time required for a feeder setup and component placement. However, production line configurations generally are not the same. Each line might have different types of machines for producing a specific PC board. In order to apply the integrated methodology in such a production environment, the PCB types need to be classified based on their production specification. The integrated methodology can then be performed for each PCB group.

6.3 Extensions for Further Research

There are several potential areas of future research resulting from the work in this dissertation:

1. Relaxing some of the key assumptions: The three major assumptions are discussed in the previous section. Relaxing these assumptions may increase the generality of the solution methodology. For example, it has been assumed that each component type occupies only one feeder slot. Technically, the size of some component types may be so large that they cannot be installed on the machines with only one feeder slot. Alternate procedures would have to be developed to accommodate this enhancement.
2. Combination of the traditional scheduling research and the family assignment problem: As previously stated, PCBs in the same family have high component similarity, while PCBs in different families have low component similarity. The number of common components between the family being removed and the family being loaded may be negligible. Assigning PCB families to production lines can be considered as sequence-independent. Consequently, application of typical scheduling rules such as Short Processing Time and due date may be investigated.
3. Integration of the integrated methodology with optimization for lower levels of process planning: In this research the placement time of a component is fixed. Actually, placement time can be determined by solving the sequencing of component placement and feeder arrangement. Additionally, feeder and board latencies can be taken into

account to get a better estimate of placement time, which is an important parameter for component allocation and PCB assignment problems.

4. Integration with information technology: Process planning needs to use information from process requirement plans, production plans, production schedules, and appropriate databases for practical applications. The developed methodology needs to be integrated with these information sources. Integration of the methodology and information technology would create a powerful tool for the electronics industry.

REFERENCES

Ammons, J. C., Carlyle, M., Cranmer L., DePuy G., Ellis K., McGinnis L.F., Tovey C.A., and Xu H.,. (1997). Component allocation to balance workload in printed circuit card assembly systems. *IIE Transcations, 29*(1-6), 265-275.

Askin, R., Dror, M., and Vakharia, A. J. (1994). Printed circuit board family grouping and component allocation for a multimachine, open-shop assembly cell. *Naval Research Logistics, 41*(5), 587-608.

Bard, J. F., Clayton, R. W., and Feo, T. A. (1994). Machine setup and component placement in printed circuit board assembly. *The International Journal of Flexible Manufacturing Systems, 6*, 5-31.

Barnea, A., and Sipper, D. (1993). Set-up reduction in PCB automated assembly. *Computer-integrated Manufacturing Systems, 6*(1), 18-26.

Bean, J. C. (1994). Genetic Algorithms and Random Key for Sequencing and Optimization. *Operation Research Society of America, 6*(2), 453-459.

Ben-Arieh, and Dror, M. D. (1990). Part assignment to electronic insertion machines: two machines case. *International Journal of Production Research, 28*(7), 1317-1327.

Bhaskar, G., and Narendran, T. T. (1996). Grouping PCBs for set-up reduction: a maximum spanning tree approach. *International Journal of Production Research, 34*(3), 621-632.

Capps, C. H. (1997). *Setup Reduction in PCB Assembly: A group Technology Application Using Genetic Algorithms.* Unpublished Master thesis in Industrial Engineering, Oregon State University, Corvallis, OR.

Carmon, T. F., Maimon, O. Z., and Dar-EL, E. M. (1989). Group Set-Up for printed circuit board assembly. *International Production Research, 27*(8), 1795-1810.

Chen, M., and Dong, Y. (1999). Applications of neural networks to solving SMT scheduling problems-a case study. *International Journal of Production Research, 37*(17), 4007-4020.

Chung, J. H. (1991). *Lot size scheduling problem with two level setup cost/time structure.* Unpublished Doctoral dissertation, Georgia Institute of Technology, GA.

Crama., Y., Kolen, A. W. J., and Oerlemans, A. G. (1990). Throughput rate optimization in the automated assembly of printed circuit boards. *Annals of Operations Research, 26*, 455-480.

Daskin, M. S., Maimon, O., Shtub, A., and Braha, D. (1997). Grouping components in printed circuit board assembly with limited component staging capacity and single card setup: problem characteristics and solution procedure. *International Journal of Production Research, 35*(6), 1617-1638.

DePuy, G. W. (1995). *Component allocation to balance workload in printed circuit board assembly systems.* Doctoral dissertation, Georgia Institute of Technology, GA.

Dessouky, M. M., Adiga, S., and Park, K. (1995). Design and scheduling of flexible assembly lines for printed circuit boards. *International Journal of Production Researches, 33*(3), 757-775.

Diilon, S., Jones, R., Hinde, C. J., and Hunt, I. (1998). PCB assembly line setup optimization using component commonality matrices. *Journal of Electronics Manufacturing, 8*(2), 77-87.

Ellis, P. K. (1996). *Analysis of setup management strategies in electronic assembly systems.* , Georgia Institute of Technology.

Garetti, Pozzetti, and Tavecchio. (1995). Production scheduling in SMT eletronic boards assembly. P*roduction Planning & Control, 7*(2), 197-204.

Green, D. (1999,). EMS Industry 2000. *circuits Assembly, Oct'99,* 25-28.

Gronalt, M., Grunow, M., Gunther, H. O., and Zeller, R. (1997). A heuristic for component switching on SMT placement machines. *International Journal of Production Economics, 53*, 181-190.

Gronalt, M., and Zeller, R. (2000). Component allocation and job sequencing for two series-connected SMD placement machines. *International Journal of Production Research, 38*(2), 409-427.

Gunther, H. O., Gronalt, M., and Piller, F. (1996). Component kitting in semi-sutomated printed circuit board assembly. *Interantional Journal Production Economics, 43*(2-3), 213-226.

Harhalakis, G., Nagi, R., and Proth, J. M. (1990). An efficient heuristic in manufacturing cell formation for group technology applications. *International Journal of Production Research, 28*(1), 185-198.

Hashiba, S., and Chang, T. (1991). PCB assembly setup reduction using Group Technology. *Computers and Industrial Engineering, 21*(1-4), 453-457.

Haskard, M. R. (1992). *Electronic Circuit Cards and Surface Mount Teachnology.* New York: Prentice Hall.

Hayrinen, T., Johnsson, M., Johtla, T., Smed, J., and Nevalainen, O. (2000). Scheduling algorithms for computer-aided line balancing in printed circuit board assembly. *Production Planning & Control, 11*(5), 497-510.

Hillier, M. S., and Brandeau, M. L. (1998). Optimal component assignment and board grouping in printed circuit board manufacturing. *Operation Research, 46*(5), 675-689.

Hwang, H., and Sun, J. (1996). A genetic-algorithm-based heuristic for the GT cell formation problem. *Computer Industrial Engineering, 30*(4), 941-955.

Iakovou, E. (1992). An hierarchical approach to machine batching, loading, and tool allocation problems. *Doctoral dissertation, Cornell University.*

Iyengar, K. V. (1999). *Setup reduction in printed circuit board assembly.* Master Thesis, Iowa State University.

Jeon, G., Broering, M., Leep, H. R., Parsaei, H. R., and Wong, J. P. (1998). Part family formation based on alternative routes during machine failure. *Computer Industrial Engineering, 35*(1-2), 73-76.

Johri, P. K. (1990). A heuristic algorithm for loading new work on circuit pack assembly lines. *Interational Journal of Production Researches, 28*(10), 1871-1883.

Khoo, L. P., and Ng, T. K. (1998). A genetic algorithm-based planning system for PCB component placement. *International Journal of Production Economics, 54(3)*, 321-332.

King, J. R. (1980). Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research, 18*(2), 213-232.

King, J. R., and Nakornchai, V. (1982). Machine-component group formation in Group Technology: review and extension. *International Journal of Production Research, 20*(2), 117-133.

Kusiak, A. (1987). The generalized group technology concept. *International Journal of Production Research, 25*(4), 561-569.

Lee, S. H., Park, T. H., Lee, B. H., Kwon, W. H., and Kwon, W. (1998). *A dynamic programming approach to a reel assignment problem of a surface mounting machine in printed circuited board assembly.* Proceedings of the 1998 IEEE he International Conference on Robotic & Automation.

Leipala, T., and Nevalainen, O. (1989). Optimizaion of the movements of a component placement machine. *European Journal of Operational Research, 38*, 167-177.

Leu, M. c., Wong, H., and Ji, Z. (1993). Planning of component placement/insertion sequence and feeder setup in PCB assembly using Genetic Algorithm. *Transactions of the ASME, 115*, 424-432.

Li, Y. (1999). *Component to multi-track-feeder assignment and board sequencing for printed circuit board assembly.* Doctoral dissertation, Oregon State University.

Logendran, R. (1990). A workload based model for minimizing total intercell and intracell moves in cellular manufacturing. *International Journal of Production Research, 28*(5), 913-925.

Luzzatto, D., and Perona, M. (1993). Cell formation in PCB assembly based on production quantitative data. *European Journal of Operational Research, 69(3)*, 312-329.

Maimon, O., and Shtub, A. (1991). Grouping methods for printed circuit board assembly. *International Journal of Production Research, 29*(7), 1379-1390.

Maimon, O. Z., and Braha, D. (1998). A genetic algorithm to scheduling PCBs on a single machine. *International Journal of Production Research, 36*(3), 761-784.

Maimon, O. Z., Dar-El E.M., and Carmon, T. F. (1993). Set-up saving schemes for printed circuit board assembly. *European Journal of Operational Research, 70(2)*, 177-190.

McGinnis, L. F., Ammons, J. C., Carlyle, M., Cranmer, L., Depuy G.W., Ellis K.P., C.A., T., and H., X. (1992). Automated process planning for printed circuit card assembly. *IIE transactions, 24*(4), 18-46.

Miller, R.G. (1986). Beyond ANOVA, Basics of Applied Statistics, Willey, NY, USA

Moyer, L. K., and Gupta, S. M. (1997). An efficient assembly sequencing heuristic for printed circuit board configurations. *Journal of Electronic Manufacturing, 7*(2), 143-160.

Myers, R. L. (1997). Increasing the output of automated PCB assembly. *Circuit Assembly, 8,* 66-68.

Ng, M. (1998). Heuristics approach to printed circuit board insertion problem. *Journal of the Operational Research Society, 49(10)*, 1051-1059.

Palm, R. C. (1996). Reducing setup time for printed circuit assembly. *Hewlett-Packard Journal, 47*(4), 84-90.

Park, J., and Asada, H. (1994). *Sequencing optimization for high speed robotic assembly using simulated annealing.* Paper presented at the IEEE International conference on Robotics and Automation.

Pearl, J. (1984). *Heuristics : Intelligent search strategy for computer problem solving*. Massachusetts: Addison-Wesley Publishing Company.

Peters, B. A., and Subramanian, G. S. (1996). Analysis of partial setup strategies for solving the operational planning problem in parallel machine electronic assembly systems. *International Journal of Production Research, 34*(4), 999-1021.

Rajkumar, K., and Narendran, T. T. (1997). A bi-criteria model for loading on PCB assembly machines. *Production Planning & Control, 8*(8), 743-752.

Rajkumar, K., and Narendran, T. T. (1998). A heuristic for sequencing PCB assembly to minimize set-up times. *Production Planning & Control, 9*(5), 465-476.

Rubin, P. A., and Ragatz, G.L. (1995). Scheduling in a sequencing dependent setup environment with genetic search. *Computers Operations Researches, 22*(1), 85-99.

Sadiq, M., Landers, T. L., and Taylor G.D. (1993). A heuristics algorithm for minimizing total production time for a sequence of job on a surface mount placement machine. *International Journal of Production Research, 31*(6), 1327-1341.

Sanders, R. C. (1996). *Operational planning for electronic assembly on a two-machine mixed-model assembly line.* Unpublished Ph.D Dissertation, Texas A&M University, TA.

Seifoddini, H., and Djassem, M. (1996). A new grouping measure for evaluation of machine-component matrices. *International Journal of Production Research, 34*(5), 1179-1193.

Shtub, A., and Maimon O. (1992). Role of similarity measures in PCB grouping procedures. *International Journal of Production Research, 30*(5), 973-983.

Smed, J., Johnsson, M., Puranen, M., Leipala, T., and Nevalainen, O. (1999). Job grouping in surface mounted component printing. *Robotics and Computer-integrated Manufacturing, 15*(1), 39-49.

Sule, D. R. (1992). A heuristic procedure for component scheduling in printed circuit pack sequencers. *International Journal of Production Research, 30*(5), 1191-1208.

Tang, C. S., and Denardo, E. V. (1988). Models arising from a flexible manufacturing machine, part I: Minimization of the number of tool switches. *Operations Research, 36*(5), 767-777.

Tanimoto, S. L. (1987). *The elements of artificial intellignece.* Seattle, Washington, USA: Computer Science Press.

Wang, J. (1998). A linear assignment algorithm for formulation of machine cell and part families in cellular manufacturing. *Computer Industrial Engineering, 35*(1-2), 81-84.

Watkins, R. E., and Cochran, J. K. (1995). A Line balancing heuristic case study for existing automated surface mount assembly line setup. *Computer Industrial Engineering, 29*(1-4), 618-685.

Won, Y., and Kim, S. (1997). Multiple criteria clustering algorithm for solving the group technology problem with multiple process routings. *Computer Industrial Engineering, 32*(1), 207-220.

Xu, Z., Carlson, K., Kurschner, R., Li, Y., and Randhawa, S. (1999). An integrated methodology for surface mount PCB configuration. *Journal of Electronic Manufacturing, 8*(3-4), 225-234.

Zerangue, N. F. (1999). *Board scheduling for circuit board assembly: Computational testing of integer programming approach.* Unpublished undergraduate research fellow program, Texas A&M University.

APPENDICES

A1. PCB Grouping Procedure

Define :

Q : Component-to-component matrix

P : PCB-to-component incidence matrix

RC : Relationship Counter

CR : Closeness Ratio

MCR : Maximum Closeness Ratio

MTV : Minimum Threshold Value

Step 1: Develop a component-to-component matrix (Q), which describes the number of PCBs that require both column and row components.

Step 2: Select the largest element in the matrix Q and assign it as the present value of Relationship Counter (RC).

Step 3: Define a parameter U, between 0 and 1 ($0<U<1$), which is a measure of effectiveness of joining a tool to a group consisting of other tools. This parameter states the closeness all the existing tools within a group in order for the entering tool to join that group. Based on the different values of U chosen, the number of group may be different.

Step 4: Starting with the first row, search each row for a value that equals RC.

Step 5: (a) If none of the associated components in the row and column are already in a group, then form a group consisting of these two components and go to step 8.

(b) If both components are already assigned to the same group, then go to step 8.

(c) If one of the components in the pair is in a group and the other has not been assigned yet, go to step 6. (d) If both of the components are assigned to different groups, then go to step 7.

Step 6: Calculate the "Closeness Ratio" (CR) of the entering component with each group that has already been formed. A closeness ratio is defined as the ratio of the total of all relationships the entering component has with the components that are currently in the group to the total number of components that are presently assigned to that group.

The entering component is placed in a group that has the "maximum closeness ratio" (MCR), as long as this maximum is greater than or equal to "minimum threshold value" (MTV), calculated as U multiplied by the present value of RC. If the value of MCR is less than MTV, then a new group is formed consisting of two components having the relationship value that equals to the present value of RC. Go to step 8.

Step 7: Duplicate of one or more component is suggested. There are two possible alternatives, and they are checked sequentially in order of importance. The first alternative is to duplicate one additional component of either type and place it in the appropriate cell, and the second alternative is to duplicated both components, one of each type, and form a new group or place the appropriate one in each of the existing groups. The following rule are suggested:

(1) Calculate the effect of duplicating one component. Check component A as the entering component for the groups where component B exists and

B as the entering component for the groups where A exists. Determine the maximum closeness ratio, MCR, from all the groups that are checked and note the associated group and entering component

(2) If MCR>RC×U, the noted component is duplicated and assigned to the associated group. Go to step 8.

(3) If the maximum closeness ratio in the previous calculation was less than RC×U, a check must be made to see if both components should be duplicated. From the previous calculation determine the maximum closeness ratio for the groups where A is the entering component (MCRA) and the maximum closeness ratio for the groups where B is the entering component (MCRB). Calculate the index value as maximum RC×U/2. If both MCRA and MCRB are greater than the index value and $|MCRA-MCRB| < P \times RC/2$, duplicate both components and place each in an appropriate group. If either MCRA or MCRB is greater than the index value, regardless of the value of $|MCRA-MCRB|$, form a new group consisting of component A and B. go to step 8.

(4) If none of the above conditions exists, ignore this observation and go to step 8 since the contribution of any duplicating component in improving the efficiency of grouping is very limited.

Step 8: Check to see if all components are assigned to groups. If the number of components in any group is equal to the number of slot on the sequencer head,

"fathom" the associated group. Fathoming a group means not allowing the group to be part of any further consideration that would add a new component to the group

Continue the check of component-to-component matrix with the present value of RC proceeding sequentially in rows. If an element is found that is equal to the present value of RC, go to step 5. If no such element is found, go to step 9.

Step9: Reduce the value of RC to the next value in decreasing order of magnitude and return to step 5.

Step 10: Assign each PCB to an appropriate group. This is accomplished by investigating each PCB and assigning it to a group that has the most component it needs.

To illustrate the procedure, an example with ten PCBs and fifteen components was generated. The PCB-to-component incidence matrix (P) and component-to-component matrix (Q) are in Table A1.1 and A1.2 respectively.

Table A1.1: PCB-to-component incidence matrix

PCBs	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
3	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0
4	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0
5	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1
7	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0
10	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

Table A1.2: Component-to-component matrix (Q)

Com-	Components														
ponents	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-	3	2	1	2	0	0	1	1	0	2	0	0	0	0
2		-	3	1	3	0	0	1	1	0	1	1	0	0	0
3			-	1	2	0	0	0	1	0	0	1	0	0	0
4				-	2	2	1	1	2	1	0	0	1	0	0
5					-	0	0	1	0	0	2	1	0	0	0
6						-	1	1	1	1	1	1	1	0	0
7							-	2	3	2	0	0	1	0	0
8								-	2	2	1	0	0	0	0
9									-	2	0	0	1	0	0
10										-	1	0	1	1	1
11											-	1	1	1	1
12												-	0	1	0
13													-	1	1
14														-	1
15															-

Iteration 1: From the component-to-component matrix, the maximum value is 3. Thus, RC= 3. The first value of 3 is associated with components 1 and 2. Since both of them are not assigned, these components form a new group, G1 (step 5).

Iteration 2: Let U = 0.5. The second pair components with element value of 3 in the matrix are component 2 and 3. The component 2 is already assigned to G1. Then, calculate CR of the entering component with group G1 (step 5 and 6). Since the maximum closeness ratio (MCR=2.5) is greater than minimum threshold value (MTV= 1.5), the component 3 can be formed in G1. Corresponding calculations are shown in Table A1.3. Since the number of components in G1 is 3 (<6, the maximum number of feeders), the group G1 is not fathom (Step 8).

Table A1.3: Checking for entering component 3

RC	Com. Enter	Existing groups G1	Rel	G2	Rel	
3	3	1	2			MTV = U*RC = 1.5
		2	3			MCR = 2.5
	Total	2	5			
	CR		2.5			

Iteration 3: From component-to-component matrix, the components 2 and 5 have 3-relationship. Component 2 is also in G1 but component 5 is not assigned yet. Like the component 3 in iteration 2, the maximum closeness ratio (MCR=2.3) is greater than minimum threshold value (MTV= 1.5), the component 5 can be formed in G1 (see in Table A1.4).

Table A1.4: Checking for entering component 5

RC	Com. Enter	Existing groups G1	Rel	G2	Rel	
3	5	1	2			MTV = U*RC = 1.5
		2	3			MCR = 2.3
		3	2			
	Total	2	7			
	CR		2.3			

Iteration 4: The other pair of components with 3-relationship is component 7 and 9. Since both of these components are not assigned to any groups, these components can form a new group G2. Since the number of components in G2 is 2 (<6, the maximum number of feeders), the group G2 does not fathom (Step 8).

Iteration 5: The maximum value among the elements in component-to-component matrix is 2. Thus, RC = 2 (Step2).

Iteration 6: Components 1 & 3, 1 & 5, and 3&5 have 2-relationship. They are in the same group (step 5).

Iteration 7: Components 1 and 11 has 2-relationship and only component 1 is group G1. The component 11 can be the entering component in G1 because of MCR>MTV (Table A1.5).

Table A1.5: Checking for entering component 11

RC	Com.	Existing groups				
	Enter	G1	Rel	G2	Rel	
2	11	1	2	7	0	MTV = U*RC = 1.0
		2	1	9	0	MCR = 1.3
		3	0			
		5	2			
	Total	2	5	2	0	
	CR		1.3		0	

Iteration 8: Components 4 and 5 has 2-relationship, and component 5 is in G1. Check for entering component 4 in G1. However, CR of G2 is greater than CR of G1 and MCR>MTV. The component 4 is entered in G2 (step 6). The associated calculation is presented in Table A1.6.

Iteration 9: Like iteration 8, components 4 and 6 have 2-relationship and component 4 is in G2. Because of MCR>MTV, component 6 is assigned to G2

Table A1.6: Check for entering component 4

RC	Com. Enter	Existing groups G1	Rel	G2	Rel	
2	4	1	1	7	1	MTV = U*RC = 1.0
		2	1	9	2	MCR = 1.5
		3	1			
		5	2			
		11	2			
	Total	5	7	2	3	
	CR		1.4		1.5	

Table A1.7: Checking for entering component 4

RC	Com. Enter	Existing groups G1	Rel	G2	Rel	
2	6	1	0	7	1	MTV = U*RC = 1.0
		2	0	9	1	MCR = 1.33
		3	0	4	2	
		5	0			
		11	1			
	Total	5	1	3	4	
	CR		0.2		1.33	

Iteration 10: With 2-relationship, components 4, 9 and 11 are in the same group and the number of component in the group still less than feeder capacity. This group does not fathom (Step 6 and 8).

Iteration 11: With 2-relationship of components 4 and12, check for entering component 12 whereas component 4 is in G2. Since MCR<MTV, then both of these components form a new group (Step 6).

Table A1.8: Check for entering component 12

RC	Com.	Existing groups				
	Enter	G1	Rel	G2	Rel	
2	12	1	0	7	0	MTV = U*RC = 1.0
		2	1	9	0	MCR = 0.75
		3	1	4	2	
		5	1	6	1	
		11	1			
	Total	5	4	4	3	
	CR		0.8		0.75	

Iteration 12: With 2-relationship between components 7 and 8, check to enter component 8 when component 7 is in G2. Because MCR>MTV, the entering component 8 form in G2.

Table A1.9 Checking for entering component 8

RC	Com.	Existing groups						
	Enter	G1	Rel	G2	Rel	G3	Rel	
2	8	1	1	7	2	4	1	MTV = U*RC = 1.0
		2	1	9	2	12	0	MCR = 1.5
		3	0	4	1			
		5	1	6	1			
		11	1					
	Total	5	4	4	6	2	1	
	CR		0.8		1.5		0.5	

Iteration 13: Consider components 7 and 10 with 2-relationship. Component 10 can enter in G2 with MCR = 1.5. There are 6 components in G1; thus this group fathoms.

Iteration 14: Now the maximum relationship in matrix is 1. Thus RC = 1. Consider component 4 and 13 with 1-relationship. Since component 4 is in G3, and MCR for entering this component is greater than MTV, component 13 is put in G3.

Iteration 15: Like component 13, components 14 and 15 have 1-relationship with component 11, which is in G1 and G3. The MCR for entering component 14 is 0.66 and for component 15 is 0.75 on group G3. Thus, both of these components can be members of G3.

All components are already assigned. Consequently, we can assign PCB to component group that have the most common components for that PCB. Table A1.10 is depicted the PCB groups

Table A1.10 PCB groups

Group	Common components	PCBs	Total components
G1	1,2,3,5,11	1,2,4,7,8	9
G2	7,9,4,6,8,10	3,5,10	11
G3	4,12,13,14,15	6,9	10

A2. Family Decomposition Procedure.

Assumptions : (1) The number of components required by all PCBs are greater than the number of feeder slots.

(2) The number of components required by each PCB type is not more than the number of feeder slots.

(3) Each component type occupies only one feeder slot.

Define:

i, j : Index of PCBs

J : Set of all PCBs

θ : A subset of PCB types not assigned with all their components to the groups

s : Component setup time (s)

S_j : setup time for PCB j to calculate saving time

Q_k : Set of PCBs for subfamily k

R_k : Set of components corresponding to PCBs in Q_k

Step1: Initially, $\theta = J$. Calculate Jaccard similarity index (SI) for all pairs of PCBs (i and j). SI is defined as the number of components that are required for both PCB i and j divided by the number of components that are required for at least one of the PCBs i and j. Develop similarity matrix of PCBs (S). Each element in S is defined as s_{ij}.

Step2: For each $j \in J$, compute the "global similarity measure" (SM_j) for each PCB j

$$SM_j = \Sigma SI_{i,j} \text{ for all } i \in J \text{ and } i \neq j$$

Step 3: Let k = k+1. Start a new group Q_k with PCB l so that

$SM_l = MAX \{ SM_j \}$ for $j \in J$

Let the first n components of PCB l be members of R_k (if these components are not present in R_k, let all of them be members of R_k), and PCB i is in Q_k. Set $s_{lj} = 0$ for l =i. Remove PCB i from θ.

Step 4: Calculate the unused capacity of the machine (UC). UC is the difference between C and the number of components in Q_k.

Step5: For each PCB_j that is not assigned to any subfamilies, Calculate the similarity index SI_j^* between that PCB and all PCB members of Q_k:

$$SI_j^* = | n_j \cap \{\cup_{i \in Qk} n_i\} | / | n_j \cup \{\cup_{i \in Qk} n_i\} |$$

The sorted subset (T) is a set of PCBs sorted by the value of SI_j^*. A tie can be broken arbitrarily.

Step6: Test the appropriate time saved by adding the first PCB in T to Q_k by calculating the value of P_j^*.

$$P_j^* = [s\{ | n_j^* | - | n_j^* \cap \{\cup_{i \in Qk} n_i\} | \}] / S_j^*$$

If P_j^* is no longer than a predetermined threshold (U) then delete j* from T and go to step 10.

Step 7: If the unused is enough for additional components, then (1) add the PCB j and its components that are not members of R_k, into Q_k and R_k respectively, and (2) remove PCB j from θ and T. If the unused capacity is not enough but $\theta \neq \phi$, then go to step 2.

Step 8: Let all elements of s_{ij} associated with PCB_j* equal zeros. If any components were added to R_k in the last iteration then go to step 5.

Step 9: If not all the entries in $S(s_{ij})$ are zero, then go to step 11.

Step 10: Stop and print the solution.

Step11: If T is empty, then go to step1, otherwise go to step 6.

An example with five PCBs and nine components was generated. The PCB-to-component incidence matrix (P) is in Table A2.1.

Table A2.1 PCB-to-component incidence matrix

PCB	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
4	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0
7	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0

Iteration 1: Set $\theta = J = \{1,2,4,7,8\}$, $R_i = \phi$, $Q_i = \phi$. Calculate Jaccard similarity index for PCBs i and j. For example, PCBs 1 and 2 share three common components.

The total number of components required to produce these two PCBs is 6. Thus, $s_{12} = 3/6 = 0.5$. The similarity matrix (S) is presented in Table A2.2.

Table A2.2 Similarity matrix

PCBs	PCBs				
	1	2	4	7	8
1	-	0.5	0.5	0.33	0.6
2		-	0.25	0.29	0.29
4			-	0.5	0.29
7				-	0.14
8					-

Iteration 2: Calculate " global similarity measure" (SM_i) for PCB i (Step 2).

$SM_1 = s_{12}+s_{14}+s_{17}+s_{18} = 0.5+0.5+0.33+0.6 = 1.93$

$SM_2 = 1.32$, $SM_4 = 1.54$, $SM_7 = 1.26$, and $SM_8 = 1.32$.

Iteration 3: Let k=1. Based on the maximum value of SM_i, start group Q_1 with PCB 1. Add components of PCB 1 in R_1, R_1= {1, 2, 3, 5}, and add PCB 1 in Q_1, Q_1 = {1} (Step 3). Assume feeder capacity (C) = 6, the unused capacity (UC) = 6-4 = 2 (Step 4).

Iteration 4: For each PCB in θ, θ ={2,4,7,8}. Calculate similarity index (SI_j*) between PCB j in θ and PCB 1 in Q_1.

SI_2* = 0.50, SI_4* = 0.50, SI_7* = 0.33, and SI_8* = 0.60.

Based on SI_j*, the sorted subset T = {8,2,4,7} (Step 5).

Iteration 5: Test for appropriate time saved by adding PCB 8 to Q_1. Let the time for adjust a machine to the new PCB is 2 units and time to setup a component on a feeder is 1 unit. Assume threshold value (U) equals 0.5 (Step6).

$$P8^* = 1(4-2)/2 = 1$$

Iteration 6: Since P8* is greater than threshold, PCB 8 is added in Q_1 (Step7). $Q_1 = \{1, 8\}$, $R_1 = \{1,2,3,5,9\}$ and UC = 6-5 = 1. Remove PCB 8 from θ, and set $s_{i8} = 0$ for all i. Since component 9 can be added in R_1, then consider adding anther PCB in Q_1 (Step 8).

Iteration 7: For each PCB in θ, $\theta = \{2,4,7\}$. Calculate similarity index between that PCB and all PCBs in Q_1. The PCB-to-component incidence matrix is shown in Table A2.3.

Table A2.3 PCB-to-component incidence matrix

PCB	Components								
	1	2	3	4	5	8	9	11	12
1&8	1	1	1	0	1	0	1	0	0
2	0	1	1	1	1	0	0	0	1
4	1	1	0	0	1	1	0	1	0
7	1	0	0	1	1	0	0	1	0

$SI_2^* = 3/7 = 0.43$, $SI_4^* = 0.43$, and $SI_7^* = 0.33$. Based on SI_j^*, the sorted subset T = {2,4,7} (Step 5).

Iteration 8: Test for appropriate time saved by adding the PCB 2 to Q_1. Assume threshold (U) = 0.5 (Step 6).

$$P2^* = 1(5-3)/2 = 1$$

Iteration 9: Although P2* is greater than threshold (0.50), the unused capacity of Q_1 is not enough to add components for producing PCB 2. Since $\theta \neq \phi$, then go to form a new group (Step7).

Iteration 10: When $\theta = \{2,4,7\}$, the updated similarity matrix (Table A2.4), calculate (SM_i) for PCB i (Step 2).

Table A2.4 Similarity matrix

PCB	PCB		
	2	4	7
2	-	0.25	0.29
4		-	0.5
7			-

$SM_2 = s_{24}+s_{27} = 0.25+0.29 = 0.54$, $SM_4 = 0.75$, and $SM_7 = 0.79$.

Iteration 11: Let k = 2. With the value of SM_7 (0.79), start group Q_2 with the PCB 7. Add components of the PCB 7 in R_2, $R_2 = \{1, 4, 5, 11\}$, and PCB 7 in Q_2. Then remove the PCB 7 from θ (Step 3). The unused capacity (UC) = 6-4 = 2 (Step 4).

Iteration 12: For each PCB in θ, $\theta = \{2,4\}$. Calculate similarity index between that PCB and the PCB 7 in Q_2. $SI_2^* = 0.29$ and $SI_4^* = 0.50$. Based on SI_j^*, the sorted subset T = {4, 2} (Step 5).

Iteration 13: Test for adding the PCB 4 to Q_2 (Step 6).

$P4^* = 1(5-3)/2 = 1$

Iteration 14: Since P4* is greater than threshold, PCB 4 is added in Q_2. Thus, Q_2 = {4, 7} and R_2 = {1,2,4,5,8,11}. There is no available feeder. Thus, the group G2 will be terminated (Step7).

Iteration 15: Since PCB 2 is the only one PCB left in θ, PCB 2 can be put in G3.

A3. Subfamily Sequencing Procedure

Assumptions: (1) The unloaded component setup time is not a concern.

(2) The number of requisite components for a subfamily does not exceed the feeder capacity.

(3) At the beginning, all required components are not placed on any machines.

Define:

s : Starting subfamily to be produced in each family

OPEN : Set of unexpanded subfamilies

CLOSED : Set of expanded subfamilies

n : Subfamily which is going to be expanded

n' : Successor of the subfamily which is going to be expanded

f(n) : Total number of setups to produce subfamily n and all its ancestors

f(n') : Total number of setups to produce subfamily n' and all its ancestors

Step 1 : Put the start node s on the list called OPEN of unexpanded nodes. In this case, s node is the subfamily with the highest number of setups.

Step 2: If OPEN is empty, exit with failure; no solution exists.

Step 3: Remove from OPEN a node n at which f is minimum. Any tie can be broken arbitrarily. Then place the node n on a list called CLOSED to be used for expanded node

Step 4 : Expand node n, generating all its successors with pointers back to node n

Step 5 : If any of n's successors is a goal node, exit successfully with the solution obtained by tracing the path along the pointers from the goal back to s

Step 6 : For every successor n' of n :

- Calculate f(n')
- If n' was neither on OPEN nor on CLOSED, add it to OPEN. Attach a pointer from n' back to n. Assign the newly computed f(n') to node n'.
- If n' already resided on OPEN or CLOSE, compare the newly computed f(n') with the value previously assigned to n'. If the old value is lower, discard the newly generated node. If the new value is lower, then substitute it for the old(n' now points back to n instead of to its previous predecessor). If the matching node n' resided on CLOSED, move it back to OPEN.

Step 7 : Go to step 2.

In order to describe the use of the sequencing procedure, the problem with 4 subfamilies and 20 components. The subfamily-to-component matrix is shown in Table A3.1.

Table A3.1 Subfamily-to-component matrix

Subfamily	Components																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s1	1	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	1	1
s2	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1	0	0	0	1
s3	0	1	1	0	1	1	1	0	1	1	0	0	1	0	0	0	1	1	1	1
S4	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	0	1	1	0

Iteration 1 : Since the subfamily s4 has the highest number of setups (14 setups), node s4 is put in the list, called "OPEN" of unexpanded nodes, OPEN = {s4} (step1).

Iteration 2 : Because set of OPEN is not empty. Then go to step 3 (step2).

Iteration 3 : Remove s4 from the OPEN and place it in CLOSED list. OPEN = { }, CLOSED = {s4} (step3).

Iteration 4 : Expand node s4 by generating all s4's successors with the pointer back to s4. The successors are s1, s2 and s3 (step 4).

Iteration 5 : Since s1, s2 and s3 are not the goal nodes. Go to step 6 (step5).

Iteration 6 : Calculate f(n') for each successors (s1, s2 and s3), f(s1) = 14+5 = 19, f(s2) = 14+4 = 18 and f(s3) = 14+2 = 16. Since all s1, s2 and s3 are not in OPEN or CLOSED, add these nodes in OPEN. Assign pointer from s1, s2 and s3 to s4 defined as (s1, s4), (s2, s4), and (s3, s4). Thus, OPEN ={s1, s2, s3} with f(s1) = 19 from (s1, s4), f(s2)= 18 from (s2, s4), and f(s3) = 16 from (s3, s4). Go to step 2 (step 6).

Iteration 7 : Since OPEN ≠ {} and the function of s3 node is minimum, placing s3 in CLOSE and expand node s3 (step 2 and 3). CLOSED = {s4, s3}, OPEN = {s1, s2}.

Iteration 8 : Expand node S3 by generating all successors, which are s1 and s2 with the pointer from s1 to s3, (s1, s3), and s2 to s3 defined as (s1, s3), respectively (step 4).

Iteration 9 : Both s1 and s2 are not the goal nodes. Calculate function f(s1) = 16+5 = 21 and f(s2) = 16+4 = 20. Since nodes s1 and s2 are in OPEN. However, the new f(s1) = 21 greater than previous function value f(s1) = 19. The new f(s2) = 20 also greater than the old f(s2) = 18. Thus, new values are disregarded. Thus, OPEN = {s1, s2} with f(s1) = 19 and f(s2) = 18, and the pointer (s1, s4), and (s2, s4), respectively

Iteration 10 : Because OPEN ≠ {} and the function of s2 node is minimum, placing s2 with pointer (s2, s4) in CLOSED and expand node s2 (step 2 and 3). CLOSED = {s4, s3, s2}, OPEN = {s1}

Iteration 11 : Expand node s2 by generating the successors s1 and s3 with the pointer (s1, s2) and (s3, s2) (step 4).

Iteration 12 : Both s1 and s3 with pointer (s1, s2) and (s3, s2) are not the goal nodes. Calculate function f(s1) = 18+3 = 21 and f(s2) = 18+4 = 22. Node s1 in OPEN. However, the new f(s1) = 21 greater than previous function value f(s1) = 19. The new f(s2) = 22 equals to the old f(s2) = 22. Thus, new values are disregarded and move node s3 to OPEN. Thus, OPEN = {s1, s3} with f(s1) = 19

and f(s3) = 22, and the pointer (s1, s2), and (s3, s2), respectively. CLOSED = {s2, s4} (step 5 and 6).

Iteration 13 : Because OPEN ≠ {} and the function of node s1 is minimum, placing s1 with pointer (s1, s2) in CLOSED and expand node s1 (step 2 and 3). CLOSED = {s4, s3, s1}, OPEN = {s2}

Iteration 14 : Expand node s1 by generating the successors s2 and s3 with the pointer (s2, s1) and (s3, s1) (step 4).

Iteration 15 : Both s2 and s3 with pointer (s2, s1) and (s3, s1) are not the goal nodes. Calculate function f(s2) = 19+5 = 24 and f(s3) = 22+7 = 29. The node s3 is in CLOSED but the node s2 is in OPEN. The new f(s2) = 24 greater than previous function value f(s1) = 20 and new f(s3) = 29 greater than the old f(s3) = 22. Thus, new values are disregarded and move node s2 to OPEN. Thus, OPEN = {s2, s3} with f(s2) = 20 and f(s3) = 22, and the pointer (s2, s1), and (s3, s1), respectively. CLOSED = {s4, s1} (step 5 and 6).

Iteration 16 : Because OPEN ≠ {} and the function of node s2 is minimum, placing s2 with pointer (s2, s1) in CLOSED and expand node s2 (step 2 and 3). CLOSED = {s4, s1, s2}, and OPEN = {s3}.

Iteration 17 : Expand node s2 by generating the successor s1 with the pointer (s1, s2) (step 4).

Iteration 18 : Because s1 is only one node in OPEN and it is a goal node. The solution can be found. The pointers indicate from successors to mother are (s1, s2), (s2, s3) and (s3, s4) with the objective function 14+2+4+3 = 23.

A4. Keep Tool Needed Soonest (KTNS) Application Procedure.

Assumptions: (1) Each component feeder required only one slot.

(2) Before starting the assembly, assume that all components are installed on the machines initially without concerning feeder slot arrangement

Define:

i : Sequence index.

B_i : Board i^{th} of the sequence

KN : Number of components to be kept

RN : Number of component required by the next board in the sequence

SN : Number of available slot spaces.

$E_1(k)$: Current status of component k

$E_1(k) = 1$ when component k stages on the machine and is used by the remaining sequenced boards "on-line".

$E_1(k) = 0$ when component k does not stage on the machine "off-line".

$E_1(k) = -1$ when component k stages on the machine but is not used by the remaining sequenced boards.

$E_2(k)$: Component status for the next sequenced PCB

$E_1(\bullet)$: Current status of all components

$E_2(\bullet)$: Component status for the next sequenced PCB

Step 1: Initialization: i = 1. Assuming that all components are installed on the machines initially without concerning feeder slot arrangement, $E_1(\bullet)$ for all k are initially set to 1.

Step2: Keep the components required by the next PCB: If there is no PCB left then the procedure is completed. The components that are required by the first sequenced board (B_1) are kept at this stage, thus $E_1(\bullet)$ is set as the same board incidence vector $a_{B1\bullet}$ from the incidence matrix. Consequently, the number of components to be kept (KN) = $SN - RN_i$

If B_i is the last PCB, then go to step 6. If B_i is not the last PCB and the next PCB needs all slot space (KN= 0), then go to Step 7 for feeder setup plan. If the next PCB does not use all slot spaces (KN>0), then go to Step 3 and search the components needed soonest.

Step 3: Keep the components need soonest. If B_i is not the last PCB, then starting from B_{i+1} those components are kept which are currently on-line but not on the list of next setup based on the urgency of need. Every time a component is kept denoted as KN, is decreased by one. The search is repeated until either KN = 0 (go to Step 7) or the last PCB is encountered (go to Step 4).

Step 4: Early installation of the components needed soonest: Occasionally, especially near the end of production, some components that are currently on-line are no longer used by the remaining boards; thus there is no reason to keep them. On the other hand, some other components are not currently on-line may be required by subsequent boards but not the next board. If KN is not zero after step 3,

it means that there are empty slots where the components can be installed for future use. This early installation of components will not increase the number of setup because the components will be needed eventually. The search is again repeated until either KN = 0 (go to step 7) or the last PCB is encountered (go to Step 4).

Step5: Plan for excessive components: A more extreme case for KN being not zero is that the current setup can be used to process the remaining boards, Depending on production policies, these components can be either left on the machine or unloaded from the machine. This step ensures that KN is zero. Then go to step 7.

Step 6: Plan for the last PCB. No extra action is needed to setup for the last PCB as long as the required components are on-line. For component purposes, these on-line components are kept whether they are used or not.

Step 7: Plan for feeder setup and reset the current component status. The component status $E_1(\bullet)$ and $E_2(\bullet)$ is compared. A feeder setup is required if $E_1(k) = 0$ and $E_2(k) = 1$, and component k is removed from the machine if $E_1(k) = 1$ and $E_2(k) = 0$. $E_2(\bullet)$ is assigned to $E_1(\bullet)$. Set $i = i+1$ for next sequenced PCB and repeat step 1.

Due to each subfamily consisting of more than one PCB type need only one machine setup, KTNS should be applied to determine kept components on feeder slot for each subfamily. To clearly understand the procedure, an example with four subfamilies and fifteen components was generated. Assuming that four subfamilies, which are s1, s2, s3, and s4 are sequenced from the subfamily sequencing

procedure, the subfamily-to-component incidence matrix (P) is in Table A4.1. The feeder capacity is six slots and each component occupies only one feeder slot.

Table A4.1 Subfamily-to-component incidence matrix

Sub-family	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0
S2	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
S3	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
S4	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0

Iteration 1: Initially, set to $E_1(\bullet) = 1$. The next subfamily produced is subfamily S1, which required 5 components as shown in Table 4.2. KN = RN – SN = 6-5 = 1, only one component can be kept from E_1. Since KN>0, consider to keep the needed soonest component (Step 1).

Iteration 2: Starting from the component with status –1, keep this component based on the urgency of need. The component 4 is the component that is not used for the current subfamily (S1), but it will be needed in the next subfamily (S2). The component 4 is kept as presented in Table A4.2. KN = 1-1 = 0 (Step 3).

Table A4.2 Illustration of KTNS for iteration 2

Sub-family	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
E_2	1	1	1	1	1				1						
S1	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0
S2	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
S3	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
S4	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0

Iteration 3: Update status of component 4. $E_1(4) = 0$ but $E_2(4) = 1$; thus the feeder setup for subfamily S1 is {1,2,3,4,5,9}. When k = 6,7,8,10,11,12,13,14,and 15, $E_2(k) = 0$. Remove these k components. Assign value of $E_2(\bullet)$ to that of $E_1(\bullet)$ and add i = 2 (Step 7).

Iteration 4: Subfamily S2, the next subfamily produced, required 5 components, {2,3,4,5 and 12} as shown in Table A4.3. KN = 6-5 = 1. The current components on feeder are 1,2,3,4,5 and 9. Component 1 is the soonest component to be kept. KN = 1-1 = 0 (Step 3).

Table A4.3 Illustration of KTNS for iteration 4

Sub-family	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E_1	1	1	1	1	1				1						
E_2	1	1	1	1	1							1			
S2	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0
S3	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
S4	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0

Iteration 5: Update status of component 1. $E_1(1) = 0$ but $E_2(1) = 1$, thus the feeder setup for subfamily S2 is {1,2,3,4,5,12}. When k = 12, $E_2(12) = 0$. Remove the component 12 and assign $E_2(\bullet)$ to $E_1(\bullet)$ and add i = 2 (Step 7).

Iteration 6: Subfamily S3, the next subfamily produced, required 5 components, {1,2,5,8 and 11} as shown in Table A4.4. KN = 6-5 = 1. The current components on feeder are 1,2,3,4,5 and 12. Since component 1 will be used in the following

subfamily (s4), this component is the soonest component to be kept. KN = 1-1 = 0 (Step 3).

Table A4.4 Illustration of KTNS for iteration 6

Sub-family	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E_1	1	1	1	1	1							1			
E_2	1	1		1	1			1			1				
S3	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0
S4	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0

Iteration 7: Update status of component 4. $E_1(4) = 0$ but $E_2(4) = 1$. When k = 3, $E_2(k) = 0$. Remove the component 3. At k = 8 and 11, $E_2(k) = 0$. Setup the component 8 and 11. Thus the feeder setup for subfamily S3 is {1,2,4,5,8,11}. (Step 7).

Iteration 8: Subfamily S4, the next subfamily produced, required 6 components, {1,4,5,11 and 4} as shown in Table 4.5. KN = 6-5 = 1. The current components on feeder are 1,2,4,5,8 and 11(Step 3). Since S4 is the last subfamily, the early installation of the component needed soonest (Step4).

Iteration 9: Since KN= 1 from step 3 means that there is one empty slot, where a component can be installed for future use. The early installation will not increase the number of setup, thus component 2 should be allow to stay on feeder as the early installed component as depicted in Table A4.5 (Step4).

Table A4.5 Illustration of KTNS for iteration 8

Sub-family	Components														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E_1	1	1		1	1			1			1				
E_2	1	1		1	1						1			1	
S4	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0

A5. Setup Components Type Determination Procedure

Assumptions: (1) The number of components required by a subfamily does not exceed the feeder capacity.

(2) The subfamilies in this matrix are in the sequence from sequencing procedure.

Define:

j : Sequence index of components

s, p : index of subfamily

E : subfamily-to-component matrix which has elements e_{ij} =1 when the component j present in feeder setup for subfamily i ; e_{ij} = 0 otherwise.

E(p) : Row vector of subfamilies

L : Maximum number of components in E.

M : Maximum number of subfamilies in Q.

$C_j(s)$: Component j in subfamily s

x_j : Component j being a standard setup components, $x_j \in X$

$y_j(p, k)$: Component j being a semi-standard setup component for subfamily k to p, $y_j(p, k) \in Y$

$z_j(s)$: Component j being a custom setup component for subfamily s, $z_j(s) \in Z$

X, Y, Z : Sets of standard, semi-standard, and custom setup components

Step 1: Read subfamily-to-component matrix (Q) for a family. Set Q = E. Starting with s = 1 and j = 1.

Step 2: Check if the current family has only one subfamily. If s = M then all components are custom setup components. Check if it ends of a family. If s > M, which is end of the current family, then go to step 1 to start a new family. If s ≠ M then p = s. For all j, If product of all elements in column j equal 1, then component j is standard setup component, which is x_j =1. To eliminate the component j from the consideration, set values of e_{sj} for component j and all s equal to zeros.

Step 3: Start checking component in subfamily s. Let p = s. select the j^{th} component in E(s), a row vector s in E. If $e_{sj} = 0$, then j = j+1 repeat until $e_{sj} = 1$ or j >L. Assign $C_j(s) = e_{sj}$ and $e_{sj} = 0$.

Step 4: When all components in the current subfamily were assigned, consider the component in the next subfamily. If j > L and s ≠ M, then s = s+1, j = 1 and go to step 3; otherwise, stop and print the set of X, Y, and Z.

Step 5: Let p = p+1. If p > M, then go to step 8; otherwise and search for the $C_j(s)$ in the set of components E(p). If $C_j(s) = e_{pj} = 1$, then set $e_{pj} = 0$ and go to step 6; otherwise go to step 7.

Step6: Check and record a semi-standard setup component. T represents the current status of a component. T = 1 when the components is a member in both of the current subfamily and the comparing subfamily. T = 0 when the component is a member in the current subfamily but not in the comparing subfamily.

If T = 0 and p is the last subfamily in this family, then the component $C_j(s)$ is a semi-standard setup component, $y_j(p, p-1) = 1$ and go to step 8. If T = 0 but p is not the last subfamily in this family, then assign T = 1, k = s and check if component $C_j(s)$ presents in the next subfamily (Go to step 5). If T = 1 and p is the last subfamily then component j is a semi-standard setup component, $y_j(p-1, k) = 1$, T = 0 and go to step 8. If T = 1 and p is not the last subfamily then go to step 5.

Step 7: Check if it is a semi-standard or custom setup component. If T = 0, then the component $C_j(s)$ is custom setup component, which is $z_j(s) = 1$ and go to step 8. If T = 1, then this component is semi-standard, which is $y_j(p-1, k) = 1$, set T = 0, and go to step 5.

Step 8: Check if all elements of e_{sj} in E are considered. Let j = j + 1. When j >L, if s ≠M, then s=s+1, j = 1 and go to step 3.; otherwise stop the procedure and print the set of X, Y, and Z.

To illustrate this procedure, an example consisting of five components and four sequenced subfamilies was generated. The subfamily-to-component incidence matrix is presented in Table A5.1. Using this procedure, we can determine the types of setup components.

Table A5.1 subfamily-to-component incidence matrix

Subfamilies	Components				
	1	2	3	4	5
1	1	1	0	1	0
2	1	1	1	0	0
3	1	0	1	1	1

Iteration 1: Read subfamily-to-component matrix (Q) presented in Table A5.1 and assigned to matrix E. In this example, the maximum number of components (M) is 3. Set p = s =1, and j = 1.

Iteration 2: Check for standard setup components. For j = 1, the product of all elements in column 1 equal 1. Thus, the component 1 is a standard setup component. Assign x_1 =1 and assign $e_{s1} = 0$ for all s (Step 2).

Iteration 3: Let p = s =1. Select component 2 from subfamily 1, $C_2(1) = e_{12} = 1$. Set $e_{12} = 0$ (Step 3).

Iteration 4: Check if component 2 presents in subfamily 2, p = 2. Searching in subfamily 2 found that $C_2(1) = e_{22} = 1$, then set $e_{22} = 0$ (Step 5).

Iteration 5: Check if component 2 was assigned in the previous subfamily. Since T= 0, and subfamily 2 is not the last subfamily, set T = 1 and k = 1. (Step 6).

Iteration 6: Check if component 2 presents in subfamily 3. Searching in subfamily 3 found $C_2(1) \neq e_{32}$ (Step 5).

Iteration 7: Check if it is a semi-standard or custom setup component. Because of T = 1, the component 2 is a semi-standard setup component. $y_2(p-1, k) = y_2(2, 1) = 1$, T = 0 (Step 7).

Iteration 8: j = 3. Some components in this subfamily still were not compared with the other subfamilies. The updated matrix E is presented in Table A5.2. We need to continue the comparison (Step 8).

Table A5.2 The updated matrix E for iteration 18

Subfamilies	Components				
	1	2	3	4	5
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1

Iteration 9: Set p = s = 1. Select component 4 in subfamily 1, $C_4(1) = e_{14} = 1$ and set $e_{14} = 0$ (Step 3).

Iteration 10: Searching in subfamily 2, we found that $C_2(1) \neq e_{24}$ (Step 5).

Iteration 11: Check if it is a semi-standard or custom setup component. Because of T = 0, the component 4 is a custom setup component. $z_4(1) = 1$ (Step 7).

Iteration 12: When j = 5 and p =1, $e_{15} = 0$. Component 5 is also the last component in subfamily. Thus, start to consider the component in subfamily 2. The current matrix E is shown in Table A5.3 (Step 8).

Table A5.3 The updated matrix E for iteration 12

Subfamilies	Components				
	1	2	3	4	5
1	0	0	0	0	0
2	0	0	1	0	0
3	0	0	1	1	1

Iteration 13: Let p = s = 2. Select component 3 in subfamily 2, $e_{23} = 1$ and assign $C_3(2) = e_{23}$. Set $e_{23} = 0$ (Step 3).

Iteration 14: Searching in subfamily 3 found that $C_3(2) = e_{33} = 1$, then set $e_{33} = 0$ (Step 5).

Iteration 15: Check if this component is a semi-standard setup component. Since T= 0, and subfamily 3 is the last subfamily, then component 3 is semi-standard for subfamily 2 and 3, $y_3(3, 2) = 1$ (Step 6). The current matrix E is displayed in Table A5.4.

Table A5.4 The updated matrix E for iteration 15

Subfamilies	Components				
	1	2	3	4	5
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	1	1

Iteration 16: Check the next component in E. Set p = s = 3. Select component 4 in subfamily 3, $e_{34} = 1$ and assign $C_4(3) = e_{34}$. Set $e_{34} = 0$ (Step 3). Since subfamily 3 is the last one and T = 0. Consequently, component 4 is the custom for subfamily 3, $z_4(3) = 1$ (Step 7).

Iteration 17: Like iteration 16, p = s = 3. The component 5 in subfamily 3 is the custom setup component. $z_5(3) = 1$ (Step 7). In step 8, s =M = 3. Thus the procedure is terminated (Step 8).

B1. Three Literature Test Problems

(1) Test Data from Ben-Arieh (1990)

PCBs	Components																								Vol
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20	c21	c22	c23	c24	c25	
1	0	0	0	0	0	2	13	13	1	5	2	0	5	0	0	0	0	0	4	12	1	5	0	20	0	1
2	3	6	4	4	0	0	0	0	4	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0	1
3	1	4	2	6	0	0	0	0	17	0	0	0	1	1	2	3	13	3	0	1	0	0	0	0	0	1
4	1	0	0	0	0	6	1	8	1	0	0	5	0	0	0	7	0	0	0	2	0	0	0	0	0	1
5	2	5	3	9	1	2	13	14	1	5	3	0	5	0	2	1	11	7	3	8	1	5	0	19	0	1
6	2	7	3	8	1	13	0	1	0	0	2	0	0	0	2	1	11	10	0	0	0	0	0	0	0	1
7	1	1	0	0	14	1	8	5	2	0	6	4	1	1	0	1	1	2	1	9	1	0	18	21	1	1
8	1	1	0	0	13	1	9	11	2	0	2	30	22	1	0	1	1	2	0	2	4	0	19	2	3	1
9	2	1	0	0	15	0	0	0	0	0	0	14	3	1	0	1	0	1	0	0	0	0	0	0	0	1
10	0	6	1	15	0	2	3	2	0	1	5	0	0	5	5	0	0	0	1	0	2	0	0	3	3	1

(2) Test Data from Gronalt and Zeller (2000)

PCBs	Components												Vol
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	
1	4	0	3	1	0	0	0	0	0	0	0	0	5
2	5	0	7	0	0	0	0	0	4	0	0	8	5
3	9	0	0	0	0	0	0	0	4	0	0	0	5
4	0	0	0	5	0	0	0	0	4	0	0	6	5
5	0	3	0	0	10	6	0	0	0	0	4	0	5
6	0	0	0	0	0	0	0	0	6	0	0	6	5

(3) Test Data from Hashiba and Chang (1991)

PCBs	Components																			
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	c16	c17	c18	c19	c20
1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	1
2	0	1	0	1	1	1	1	1	1	1	0	0	1	1	0	1	0	0	0	1
3	0	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0
4	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1
5	0	1	1	0	1	1	0	0	0	1	1	0	1	1	1	0	0	1	1	0
6	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1
7	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0
8	0	0	1	0	0	1	1	0	1	0	1	1	0	1	1	0	1	0	1	0
9	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1

(4) Reduce Hashiba and Chang's data

PCBs	Components															Vol
	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	c15	
1	0	5	0	10	5	5	5	10	5	0	0	5	20	0	0	10
2	0	5	0	0	0	0	0	5	20	0	0	0	0	5	0	10
3	0	5	0	0	0	0	0	0	5	0	0	0	0	0	5	10
4	0	20	5	0	5	0	0	0	15	20	0	5	0	0	5	10
5	5	10	15	0	0	5	0	5	0	10	0	0	0	0	0	10
6	0	0	5	0	5	10	0	5	0	5	5	5	0	5	0	10
7	0	0	5	0	5	15	0	0	0	0	0	0	0	0	0	10

B2. Result from Component Allocation Problem on Industry Data

Data Set	Threshold	Feeder Capacity	Imbalance (Minutes)	CPU Time (Seconds)
A	0.05	100	11,170	92
	0.05	150	16,755	92
	0.05	300	33,510	104
	0.1	100	11,170	120
	0.1	150	16,755	125
	0.1	300	33,510	136
	0.15	100	11,170	73
	0.15	150	16,755	91
	0.15	300	33,510	92
	0.2	100	11,170	105
	0.2	150	16,755	122
	0.2	300	33,510	150
	0.25	100	11,170	91
	0.25	150	16,755	115
	0.25	300	33,510	134
	0.3	100	11,170	144
	0.3	150	16,755	149
	0.3	300	33,510	176
B	0.05	100	10,913	74
	0.05	150	16,370	72
	0.05	300	32,739	80
	0.1	100	16,370	86
	0.1	150	32,739	93
	0.1	300	32,739	108
	0.15	100	10,913	61
	0.15	150	16,370	70
	0.15	300	32,739	83
	0.2	100	16,370	102
	0.2	150	32,739	104
	0.2	300	10,913	120
	0.25	100	10,913	75
	0.25	150	16,370	92
	0.25	300	32,739	103
	0.3	100	10,913	116
	0.3	150	16,370	123
	0.3	300	32,739	153

B2. Result from Component Allocation Problem on Industry Data (cont.)

Data Set	Threshold	Feeder Capacity	Imbalance (Minutes)	CPU Time (Seconds)
C	0.05	100	10,945	63
	0.05	150	16,418	67
	0.05	300	32,835	75
	0.1	100	10,945	85
	0.1	150	16,418	91
	0.1	300	32,835	106
	0.15	100	10,945	57
	0.15	150	16,418	60
	0.15	300	32,835	82
	0.2	100	10,945	99
	0.2	150	16,418	115
	0.2	300	32,835	126
	0.25	100	10,945	74
	0.25	150	16,418	87
	0.25	300	32,835	103
	0.3	100	10,945	115
	0.3	150	16,418	120
	0.3	300	32,835	149
D	0.05	100	11,304	109
	0.05	150	16,956	128
	0.05	300	33,912	132
	0.1	100	11,304	131
	0.1	150	16,956	167
	0.1	300	33,912	170
	0.15	100	11,304	86
	0.15	150	16,956	109
	0.15	300	33,912	126
	0.2	100	11,304	141
	0.2	150	16,956	161
	0.2	300	33,912	191
	0.25	100	11,304	116
	0.25	150	16,956	141
	0.25	300	33,912	168
	0.3	100	11,304	194
	0.3	150	16,956	198
	0.3	300	33,912	232
Average			20,622	114

C1. Result from Integrated Methodology on Industry Data

Data Set	Threshold	Feeder Capacity	Makespan (Minutes)	CPU Time (Seconds)
A	0.05	100	38,974	304
	0.05	150	39,534	277
	0.05	300	42,485	278
	0.1	100	39,919	297
	0.1	150	38,585	284
	0.1	300	39,533	313
	0.15	100	38,531	256
	0.15	150	38,698	276
	0.15	300	39,789	265
	0.2	100	38,346	277
	0.2	150	39,023	280
	0.2	300	39,440	327
	0.25	100	38,352	269
	0.25	150	38,949	294
	0.25	300	39,676	294
	0.3	100	38,464	314
	0.3	150	39,640	301
	0.3	300	39,812	347
B	0.05	100	36,885	354
	0.05	150	33,512	226
	0.05	300	37,606	235
	0.1	100	34,298	239
	0.1	150	34,458	237
	0.1	300	34,922	261
	0.15	100	34,246	226
	0.15	150	34,416	224
	0.15	300	34,916	235
	0.2	100	34,000	252
	0.2	150	34,721	246
	0.2	300	35,096	272
	0.25	100	34,303	227
	0.25	150	34,726	242
	0.25	300	35,105	252
	0.3	100	34,468	262
	0.3	150	34,760	261
	0.3	300	35,081	303

C1. Result from Integrated Methodology on Industry Data

Data Set	Threshold	Feeder Capacity	Makespan (Minutes)	CPU Time (Seconds)
C	0.05	100	31,776	244
	0.05	150	31,735	213
	0.05	300	32,316	231
	0.1	100	31,794	218
	0.1	150	31,783	211
	0.1	300	32,231	232
	0.15	100	31,434	200
	0.15	150	31,691	191
	0.15	300	28,813	225
	0.2	100	31,324	229
	0.2	150	31,690	234
	0.2	300	42,107	250
	0.25	100	31,505	216
	0.25	150	32,245	216
	0.25	300	32,329	246
	0.3	100	31,563	242
	0.3	150	32,090	234
	0.3	300	32,397	271
D	0.05	100	44,435	754
	0.05	150	44,717	960
	0.05	300	45,684	470
	0.1	100	44,519	364
	0.1	150	44,760	376
	0.1	300	46,160	387
	0.15	100	44,602	362
	0.15	150	44,529	379
	0.15	300	45,631	379
	0.2	100	44,337	361
	0.2	150	44,597	369
	0.2	300	45,739	401
	0.25	100	44,450	387
	0.25	150	44,817	405
	0.25	300	45,926	413
	0.3	100	44,512	410
	0.3	150	45,311	398
	0.3	300	45,641	438

C2. Result of Experiments on The Integrated Methodology Procedures

No. Experiments	Component usage Variation	PCB requirement variation	Threshold Value	Feeder Capacity	Makespan (unit)	CPU Time (minute)
1	1	1	-1	-1	32,365	153.3
2	1	1	-1	1	32,699	140.3
3	1	1	1	-1	32,284	157.4
4	1	1	1	1	32,544	140.4
5	1	-1	-1	-1	49,753	130.5
6	1	-1	-1	1	53,627	153.5
7	1	-1	1	-1	38,335	110.2
8	1	-1	1	1	38,335	137.2
9	-1	1	-1	-1	33,5665	664.4
10	-1	1	-1	1	34,512	156.4
11	-1	1	1	-1	33,566	654.5
12	-1	1	1	1	34,512	155.5
13	-1	-1	-1	-1	34,385	107.8
14	-1	-1	-1	1	34,385	121.8
15	-1	-1	1	-1	34,385	105
16	-1	-1	1	1	34,385	123
17	1	1	-1	-1	34,772	199.1
18	1	1	-1	1	34,865	142.1
19	1	1	1	-1	34,637	136.4
20	1	1	1	1	35,062	142.4
21	1	-1	-1	-1	42,480	129.2
22	1	-1	-1	1	42,480	149.2
23	1	-1	1	-1	38,697	121.3
24	1	-1	1	1	38,697	157.3
25	-1	1	-1	-1	34,143	172.1
26	-1	1	-1	1	34,659	188.1
27	-1	1	1	-1	34,143	184.7
28	-1	1	1	1	34,656	199.7
29	-1	-1	-1	-1	34,098	112.2
30	-1	-1	-1	1	34,098	129.2
31	-1	-1	1	-1	34,098	112.4
32	-1	-1	1	1	34,098	115

High level = 1 and Low level = -1