TUCS

Kari Salonen

Setup Optimization in High-Mix Surface Mount PCB Assembly

Turku Centre *for* Computer Science

TUCS Dissertations
No 109, October 2008

Setup Optimization in High-Mix Surface Mount PCB Assembly

Kari Salonen

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Lecture Hall Lambda, ICT Building on November 22nd, 2008, at 12 noon.

University of Turku
Department of Information Technology
Joukahaisenkatu 3-5, 20520 Turku

2008

Supervisors

Professor Olli S. Nevalainen
Department of Information Technology
University of Turku
Finland

Professor Timo Knuutila
Department of Information Technology
University of Turku
Finland

Reviewers

Professor Hans-Otto Günther
Department of Production Management
University of Berlin
Germany

Professor Matti Linna
Department of Computer Science
University of Vaasa
Finland

Opponent

Professor Jyrki Nummenmaa
Department of Computer Science
University of Tampere
Finland

ISBN 978-952-12-2167-5
ISSN 1239-1883

Abstract

The thesis discusses machine setup problems which arise in high-mix, low-volume production environments of printed circuit board assembly (PCB) industry. We concentrate especially on the hybrid setup problem that is a combination of the job grouping and minimum setup problems. We define an objective function that minimizes a weighted sum of the number of setup occasions and the number of feeder changeovers. The aim of this function is to model real-world production situation. With the help of the weighting parameter one can put more weight on the number of setup occasions or on feeder changes depending on the production planning situations. We describe the problem in an exact way by constructing an integer programming (IP) model. While there are problem instances of practical size which can be solved optimally by current software systems there are still cases where the problem solving with exact methods turns out to be inefficient and time consuming. Unfortunately, the hybrid problem is also so hard combinatorial problem that it can be solved optimally only in small problem instances even with the best exact methods. Therefore, we develop heuristics methods that help us to find near optimal solutions and sometimes even optimal solutions in an easy way. In our publications we present several new algorithms for the problem and compare their efficiency with other methods found in literature. Furthermore, we give a general model that can be applied to construct an efficient hybrid setup heuristic.

Acknowledgements

This dissertation project has taken eight years and done mainly as a part time job. I am very grateful (and may be many others too) that this is over now, finally.

First, I need to thank my supervisor Professor Olli Nevalainen. This dissertation could not have been completed without his regular guidance, help and encouragement.

I wish to extend my thanks to Professor Timo Knuutila for his supervision, Doctor Jouni Smed for his assistance during the first years of the project, and Doctor Mika Johnsson, the Manager of Valor Finland, for his co-operation and advice.

I also need to thank Department of Information Technology of Turku University and Turku Centre for Computer Science (TUCS) for giving me financial support and excellent working conditions.

Lastly, many thanks to all my colleagues with whom I have been working in the same research group.

Lieto, October 2008

List of Original Publications

1. Smed, J., Salonen, K., Johnsson, M., Johtela, T., Nevalainen, O. Grouping PCBs with Minimum Feeder Changes. *The International Journal of Flexible Manufacturing Systems 15*, (2003), 19-35. [51]

2. Salonen, K., Smed, J., Johnsson, M., Nevalainen, O. Grouping and sequencing PCB assembly jobs with minimum feeder setups. *Robotics and Computer-Integrated Manufacturing 22*, 4(2006), 297-305. [47]

3. Salonen, K., Raduly-Baka C., Nevalainen, O. A note on the tool switching problem of a flexible machine. *Computers & Industrial Engineering 50*, 4 (2006), 458-465. [46]

4. Hirvikorpi, M., Salonen, K., Knuutila, T., Nevalainen, O. The general two-level storage management problem: A reconsideration of the KTNS-rule. *European Journal of Operational Research 171*, 1(2006), 189-207. [24]

5. Salonen, K., Knuutila, T., Johnsson, M., Nevalainen, O. Planning and Controlling the Component Assembly for Multimodel PCBs. *WSEAS Transactions on Systems 5*, 4(2006), 855-863. [45]

Contents

Chapter 1

Introduction

In the last decade the volume of electronics industry has increased rapidly, due to new products, new innovations of the products and their low prices. Computers, mobile phones and other electronic devices are much cheaper, more effective and smaller than before. It is general today that middle-class families have many mobile phones, televisions, computers and other entertainment electronics in their private homes. While the selling of electronic products is increasing all the time, the competition between the manufacturers is very hard and tends to become even harder. To reach better positions in the market, electronics industry has struggled to find new ways to cut down production costs and to increase the production volumes. For solving the first problem, to achieve lower production cost, many manufacturers in the field have moved part or all of their production to cheap labour countries. The second problem, to improve the efficiency of the production lines, is included not only in modern automation technology but also in strategies and methods of utilizing these highly automated systems.

The central part of an electronic product, a printed circuit board (PCB), is manufactured in an automated assembly line, which includes one or several stations, where the necessary electronic components are placed onto the boards. Production problems in PCB assembly are closely related to the kind of the production environment. A producer can be specialized into manufacturing large lots of PCBs of the same type whereas some others may produce multiple types of PCBs daily. These kind of high-mix environments are nowadays common. Then, several different products have to be produced in short time to meet the changing demands of the customers. To keep the expensive assembly lines running in a cost effective way, the production must be planned carefully. The vendors of the placement machines offer optimizing software packages to plan the production and to control the operation of the machines. Production engineers can use these production management packages e.g. to maximize the throughput of the production lines or to minimize the setup times when changing the PCB type manufactured on a machine.

In this thesis we study setup optimizing problems which arise in a high-mix and low-volume PCB assembly environment. Most of the problems are hard combinatorial optimization problems, where optimal solutions to practical problems cannot be usually found [11]. Therefore, strategies and solution methods that find near optimal solutions in a reasonable time are needed. Especially, machine setup problems have been widely studied by several researchers because of the strong impact of setup times on the productivity [43] [7] [37] [20] [36]. We concentrate here on general machine setup problems that are met in management of surface mount assembly lines. We present an objective function that simulates the real-world production planning situation in a flexible manner. In particular, we assume that *co-operating with both job grouping and minimum setup strategies in a high-mix and low-volume PCB assembly environment, we can develop methods to the machine setup problems that can improve solutions found with a pure minimum setup or job grouping method.* We will perform practical tests with the new methods using problems taken from a real production environment. In our view, we assume that these methods are easily utilizable as a part of a production management software, because our approach is general avoiding machine specific details.

This thesis is organized as follows. Section 2 introduces generally the main concepts of a surface mount assembly line and the component placement machines as well as a classification of the PCB assembly problems. In Section 3 different strategies for solving the setup problem solving are categorized and presented. In Section 4 mathematical programming is shortly discussed and mathematical models are presented for job grouping, minimum setup and hybrid setup problems. In Section 5 general global optimization methods are introduced to integer programming (IP) problems. Section 6 gives algorithms for job grouping, minimum setup and hybrid setup problems. Section 7 summarizes the contents of the five original publications of the thesis. Conclusions are given in Section 8.

Chapter 2

Surface Mount Assembly Process

In a printed circuit board (PCB) assembly line, a number of electronic components (capacitors, transistors, resistors, integrated circuits, etc.) are placed to predetermined locations on a bare circuit board. PCBs serve the electronic devices in two ways: the necessary components are mounted onto them and the boards provide electrical connections between components. Hundreds or even thousand of components can be placed on one board. The components can be divided according to the assembly technology: through hole and surface mount components.

Through-hole components are inserted through drilled holes in PCB whereas surface mount components are attached onto boards, see Fig. 2.1. With the surface mount assembly method, the manufacturers can cut down production costs and increase productivity, because it has several benefits against the through-hole assembly [32]:

- Components are smaller and lighter
- PCBs are smaller, and require lower power
- Assembly speed is higher and more accurate

Therefore, the newer technology surface mount technology (SMT) has become widely used in the last ten years.

Manufacturing of printed circuit boards (PCB) with an automated surface mount assembly (SMA) line consists of the following sequence of operations: solder paste application, component placement, reflow soldering, cleaning, and electrical test, see Fig. 2.2. In solder paste application there is a mixture of solder alloy particles that are applied in a controlled manner

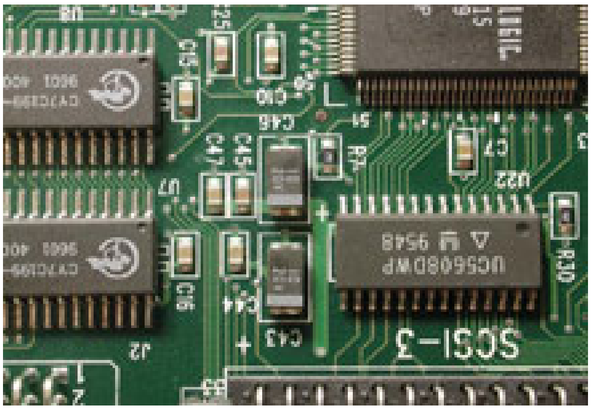


Figure 2.1: Surface mount components on a PCB.

in predetermined amounts at predetermined positions. Solder paste is used to provide the interconnection between a component and the printed circuit board. Usually a screen or stencil is used to apply solder paste. Surface mount components are placed on a printed circuit board after solder paste. The SMA line has one or several computer controlled highspeed machines to perform the actual component placement of operations. The main issue of component placement machines is to select and properly orient components on the circuit board. Reflow soldering is used mainly the soldering of substrates with surface mounted components. In process, surfaces are heated until solder paste flows and then the PCB is cooled to form firm solder joints. Cleaning of the printed boards is required to remove undesirable contaminants that may be left after soldering. To guarantee that the output signals of the PCBs are correct, electrical testing follows the whole assembly process.

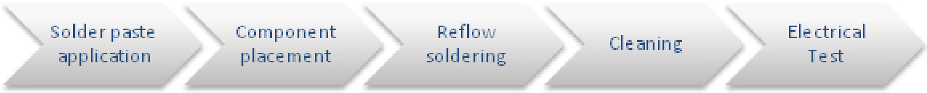


Figure 2.2: Processing steps of SMT line.

2.1 Machine types for SMT component placement

The contemporary trend in electronics industry is towards smaller and more complex products. Hence, the surface mount technology (SMT) production requires machines that are more accurate and more flexible. In addition, the competition between electronic manufacturers drives them to attend also faster machines. Therefore, the SMT machine industry is innovating new better machines continually. Especially, the SMT component placement machines have met many challenges in the past few years. Nowadays the component placement machines must be capable of handling a wide variety of components at high speed (up to 80000 cph) and different board sizes, and in order to remain electronic manufacturer competitive with fewer downtimes and setup times.

Although, the main tasks of the component placement machines are the same: pick components one by one needed for a PCB from a feeding device using a vacuum nozzle and place them correctly on a board, and all the SMT machine type consists of certain main parts: a PCB table, a feeder carrier, a placement head, nozzles and a tool magazine, anyway, their technological and functional characteristics may be significantly different. Ayob et al.[2] gave detailed descriptions of five different component placement machine types, but generally, SMT placement machines can be classified as either a gantry or turret style machine based on the design of the pick and place system.

2.1.1 Gantry-style machines

Gantry-style pick-and-place (or collect-and-place machines) have a placement head with a number of nozzles (e.g. 6 or 12) attached to an X-Y gantry system. The gantry system allows the placement head (or heads) to be moved between the component feeder bank and any component placement location on the board. Usually, the feeder bank and the board do not move during component placement. To increase speed, some placement machines are equipped with two X-Y gantries that both have a placement head, see Fig. 2.3 and Fig. 2.4. The assembly cycle time consists of a fixed setup time, the time to locate the PCB on the worktable, and a variable placement time [17].

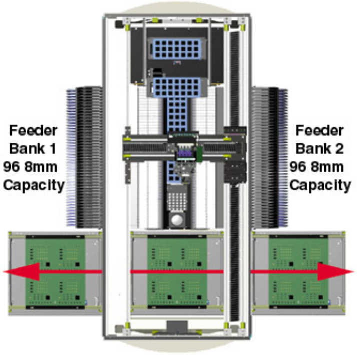


Figure 2.3: Gantry-style machine with one placement head (from: http://www.contactsystems.com).

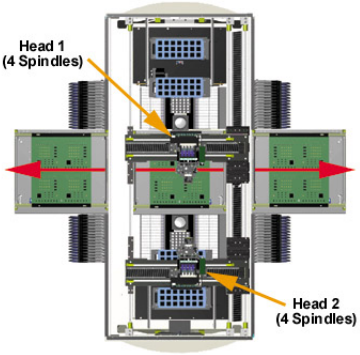


Figure 2.4: Gantry-style machine with dual head (from: http://www.contactsystems.com).

2.1.2 Rotary turret machines

A rotary turret style machine, also called a chip shooter, has a movable feeder carrier, a movable X-Y table carrying a PCB, and a horizontal turret head that rotates on a fixed axis, see Fig. 2.5. The turret consists of multiple placement heads (e.g. 12 or 20) that can handle different nozzles. While the placement head, located at the front of the turret, places a component at a pre-specified location on a board, the opposite placement head picks up a component from the feeders located in the feeder carrier. The feeder carrier holds the component feeders and moves horizontally positioning the next appropriate component feeder and the table simultaneously moves the PCB to the next component placement location. Chip shooter machines are mostly used in a high volume production environment [17].

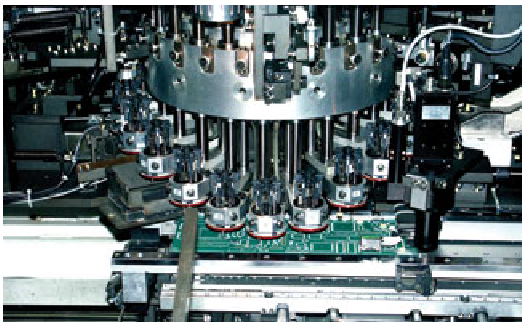


Figure 2.5: Rotary turret machine: Universal 4791 HSP Chipshooter (from: http://www.creonixltd.com/smt.htm).

2.2 Production control problems in SMT assembly processing

The efficiency of PCB assembly production depends strongly on the planning and control of the production process. The present day component placement machines operate automatically as soon as the necessary settings of the component feeders, loading of control programs and conveyor belt modifications have been carried out manually. Depending on the PCB batch sizes and the number of different PCB types, manual or automatic operations should be given the biggest weight while optimizing the production process. The main questions are, do we produce thousands of pieces of one PCB type or a couple of pieces multiple PCB types with one machine or with several machines? According to these questions, the problems in PCB assembly are divided into four major classes [27] [49]:

- One PCB type and one machine class concentrates on single machine optimization problems (e.g. feeder arrangement and component placement sequencing)

- Multiple PCB types and one machine class comprises setup strategies for a single machine

- One PCB type and multiple machines class includes line balancing and component allocation to multiple machine problems

- Multiple PCB types and multiple machines class concentrates on scheduling problems in general

2.3 Recent studies of line or machine optimization in PCB assembly

The main focus of the present thesis is in the setup optimization in the content of PCB assembly. To put our research to a more general perspective we next mention some recent studies which characterize the current trends in the field. The most research deals with the control of individual machines or machine lines by the means of heuristics solution algorithms. Because the volume of the research has been and is very large we content ourselves to mention only few of research results.

Ohno et al. [41] have studied the problem of assembling several types of PCBs on a machine with multiple pick-insertion heads and developed a heuristic that solves the problem using two-optimal local search and evolution strategy. Grunow et al. [17] presented a three-staged heuristic solution for a collect-and-place surface mount placement machine in order to minimize the assembly cycle time for a single PCB. Ayob and Kendall [3] introduced a triple objective function for a pick-and-place machine to minimize the assembly time, the feeder movements and the PCB table movements by improving the feeder setup. Deo et al. [13] formulated an objective function to a pick-and-place machine that can place from both sequenced tape and component feeders with the same setup. They developed also a genetic algorithm for simultaneously optimizing the component placement sequence and feeder assignments. Ellis et al. [16] constructed a conceptual model of a placement time estimator function for turret style machines. It can be used to estimate the placement times for a set of PCBs with a given placement sequence and feeder assignment. Van Hop and Tabucanon [54] studied the feeder assignment and assembly sequence problem for a pick-and-place machine model where also feeder magazine and board move. The objective was to minimize both the magazine and board travel times. Rossetti and Standford [44] examined a PCB assembly system in order to reduce the makespan while maintaining adequate work in process (WIP) and utilization levels. The PCB assembly system consisted of an automated surface mounting line coupled with a manual placement, inspection, and testing line. Williams and Magazine [55] developed four families of heuristics to minimize the total manufacturing time required to process a set of PCBs on a pick-and-place machine. According to their study, there are two main components of the manufacturing time: the machine preparation setup time to process the batch of PCBs and the component placing time for all the boards in the batch. Van Hop and Naganur [53] considered the scheduling problem of n PCBs and m non-identical parallel machines. Their objective was to minimize the total makespan with grouping, sequencing PCBs and component feeder changes. Ho and Ji [22] presented a hybrid genetic algorithm to optimize the sequence of component placements and arrangement of component types to feeders simultaneously for a chip shooter machine. Their objective was to minimize the total assembly time. Yilmaz et al. [56] proposed grouping methods which use machine-specific algorithms for scheduling the machine's assembly operations in order to minimize the global makespan. They tested their group setup approaches for a single-gantry collect-and-place machine equipped with a rotary placement head and an interchangeable feeder trolley.

Chapter 3

Setup Strategies

This section classifies setup strategies according to different PCB manufacturing environments. In high-mix production environments (case: multiple PCB types and one machine) SMT machines are frequently bottlenecks. It is typical of this kind of production that time consuming setup operations of the machines have to be done repeatedly. Especially, feeder setups occurs often, when a PCB type changes in a SMA line. To reach a higher level of productivity, it is important to minimize the impact of changeover times. Usually, the throughput of the assembly line can be improved significantly by reorganizing the production with a proper setup strategy.

There are different setup strategies to solve production problems which arise in different PCB manufacturing environments. All in all, the production situations can be totally different, therefore, the right choice of the setup strategy is important in order to achieve high productivity in the entire assembly system. Leon and Peters [36] classified the different setup management strategies in four categories: unique, group, minimum, and partial setup. In our studies [51] [47], we have examined also hybrid setup that seems to suit excellently in a high-mix production environment.

3.1 Unique setup

Unique setups consider one board type at a time and specify the component-feeder assignment and the placement sequence so that the placement time for the board is minimized. This is a common strategy when dealing with a single product and a single machine (or the bottleneck machine) in a high-volume production environment [5] [34] [9] [31] [58] [4] [8].

3.2 Group setup

Group setup strategy forms job groups of similar PCBs so that component setups are incurred only between groups. Hence, any board type in the group can be produced without changing the component setup, which is only required when switching from one group to another. A natural objective is to minimize the number of groups. A group setup implies the same feeder setting for all PCB types in the group. As a downside, this setup strategy reduces the possibility of generating efficient control programs for individual PCB types. Therefore, the method is more useful in low-volume production than in mass production and it is very efficient, if batch sizes are small [45]. Numerous researches have studied the group setup strategy [20] [37] [48] [14] [7] [11] [50] [30] [40] [56].

3.3 Minimum setup

Minimum setup strategy attempts to find a sequence for the PCBs and to determine feeder assignments for all the PCBs so that the total component setup time is minimized [52] [26] [18] [15] [29] [28]. The idea is to perform only the feeder changes required to assemble the next PCB with no additional feeder changes or reorganizations that would possibly reduce the placement time. In general, similar products are produced in sequence so that little changeover time occurs between them.

3.4 Partial setup

Partial setup strategy also attempts to sequence the boards as is done in the minimum setup, but it also considers reorganizing the feeders to reduce the placement time [35] [36] [42]. Because the goal is often to minimize the makespan, the partial setup strategy resides between the unique setup strategy (where only the placement time for each individual PCB is minimized) and the minimum setup strategy (where only the changeover time of each PCB is minimized).

3.5 Hybrid setup

Hybrid setup is a hybridization of the group setup strategies and minimum setup strategies. Usually, the machine setup problem is formulated as a job grouping problem or as a minimum setup problem. The objectives of these two strategies are incompatible with each other. However, in a real-world production environment of the PCB assembly it is often needed to consider both problems of the same time, especially in the high-mix production environments. A single component feeder of a placement machine can be changed typically in 1-5 min, but it may take, for instance, 15-25 min to prepare the machine for the component setup operations, because the starting of one or several component changes requires extra manual work by the personnel. Therefore, in our studies [51] [47] we have considered two different setups: *a component setup* comprises the required operations to replace one component feeder with another, and *a setup occasion* takes place every time when the line is interrupted for one or more feeder changes. Therefore, considering a case where one wants to minimize both the number of setup occasions and the number of feeder changes, it can be defined an objective function which is a weighted sum of the number job groups and feeder changes.

Chapter 4

Mathematical Formulations

Problems in PCB assembly are often combinatorial optimization problems such as minimization of some good target (e.g. setup time and placement time) or maximization of something (e.g. throughput and workload balance), under certain constraints (e.g., machine capacity, available production time, and so on) [21]. In this section we state setup optimization problems mathematically by constructing *mathematical models* for them. With mathematical models one can describe the problem settings in an exact way. Mathematical programming approach of problems consist of an objective function (consisting of a certain number of decision variables) which is to be minimized or maximized subject to a certain number of constraints. In the present study we need two modeling approaches; the linear programming (LP) and integer programming (IP) models. In a linear programming model both the objective function and the constrains are all linear functions. The linear programming model can be stated in matrix form as follows:

$$\min c^T x \tag{4.1}$$

subject to

$$Ax = b, \tag{4.2}$$

$$x \geq 0, \tag{4.3}$$

where

$$A \in R^{m*n}, b \in R^m, c \in R^n, \text{and } x \in R^n. \tag{4.4}$$

An integer programming (IP) problem in which all variables are required to be integer is called a pure (IP) problem. If some variables are restricted to be integers and some are not then the problem is a mixed integer programming (MIP) problem.

There are many studies where PCB assembly problems are formulated in mathematical models, see e.g. [3] [41] [37]. For the needs of the present study, the mathematical programming models for job grouping, tool switching and hybridization of these problems are presented here under.

We assume that the assembly of electronic components on PCBs is performed by a single placement machine which has a limited capacity for components of different types. This assumption includes that there is a particular machine which remains as a bottleneck machine of the line for each PCB type. The production program of a planning period includes several different PCB types and the production is organized as batches, each of which contains one or several copies of a particular PCB type.

Let N be the number of PCB batches (jobs) of different PCB types to be processed. The placement machine's component feeder capacity is C feeder slots and the total number of different component types of all PCBs is M. Each PCB type j requires a certain set of components. To simplify, we suppose that all feeder reels demand one feeder slot. We speak here of "feeder reel" or "tape reel" as the actual medium for organizing the supply of copies of certain component type. However, other methods (like sticks or trays) may actually be used in different machines. Although the duplication of components (i.e. component reels containing the same component type) might be advantageous, we omit it here. This means that the feeder reel of a component type appears only ones in the feeder magazine at a time.

Let $A = \{a_{ji}\}$ be a job-component matrix of the size $N \times M$. The element a_{ji} is 1, if at least one copy of the component type i is required for the PCB type j; otherwise it is 0. This means that some of the component types may occur as multiple copies on a board but they do not need any special attention due to the nature of the feeder mechanisms, for example, the component reel feeders with very high capacities.

4.1 Job grouping

The job grouping problem consists of finding a minimum set of feasible groups such that each job is contained in one group [12]. The attribute "feasible" depends on the application domain, but in field of PCB assembly it means that no intervened machine setup operations are needed when processing the jobs of a particular group. While there may be many different kinds of manually performed setup operations when changing the PCB-batch (like changing nozzles, adjusting conveyor belt with etc.), it is common to restrict the discussion on the changes of the component feeders. This stresses the importance of the feeder management in the design of the machine control.

Daskin et al. [14] and Bhaskar et al. [7] have formulated PCB-grouping problem as an integer programming model. According to Crama et al. [12] the PCB-grouping problem can be formulated as follows.

Make the following notations for $i = 1, 2, \ldots, M$, $k = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, N$:

the element x_{kj} is 1, if job (PCB type) j is assigned to group k; otherwise it is 0;

the element y_k is 1, if group k is non-empty; otherwise it is 0;

the element z_{ki} is 1, if component i is used for the production of group k; otherwise it is 0.

A subset (group) of jobs is feasible if these jobs together require at most C components.

$$MIN \sum_{k=1}^{N} y_k \tag{4.5}$$

subject to

$$\sum_{k=1}^{N} x_{kj} = 1 \text{ for } j = 1, 2, \ldots, N, \tag{4.6}$$

$$a_{ji} x_{kj} \leq z_{ki} \text{ for } k, j = 1, 2, \ldots, N; \text{ for } i = 1, 2, \ldots, M, \tag{4.7}$$

$$\sum_{i=1}^{M} z_{ki} \leq C y_k \text{ for } k = 1, 2, \ldots, N. \tag{4.8}$$

The objective function (4.5) minimizes the number of nonempty groups. Constraints (4.6) ensure that each PCB is assigned to some group and constraints (4.7) assure that the necessary components are in the feeders for the production of the group to which the PCB is assigned. Constraints (4.8) take into account that placement machine's feeder capacity is not exceeded for the groups. This basic abstraction of the problem situation makes the simplifying assumption that all component reels are of the same (unit) width.

4.2 Tool Switching

The tool switching (TSw) problem consists of N jobs which must be processed on a single machine. The machine includes a tool magazine with limited capacity of C slots. Each job requires a set of tools and the total number of tools, M (required to process all the N jobs) is greater than C. We follow Tang and Denardo's [52] mathematical model (translated into PCB environment by Barnea and Sipper [6]), where the objective is to minimize the number of tool switches (component feeder changes): The components required by the PCB type j are given by a vector A_j of length M. Let $e(A_j)^T \leq C$ for each j, where e a M vector of 1's, but $C < M$; that is, each PCB type can be processed with a single setup of components to the feeder magazine but the components of all PCBs do not fit the feeder magazine simultaneously.

To fix the sequence of processing the PCBs, we define a variable x_{jn}, which is 1 if PCB j is n^{th} in the sequence, otherwise 0. Let $W_{[n]}$ be a M-dimensional vector that determines the components placed to the feeders of the machine at the instant n $(1 \leq n \leq N)$. At the beginning of the process there are no components in the machine (i.e., the elements of $W_{[0]}$ are all 0's). Alternatively, $W_{[0]}$ can give the known setup from the end of the previous planning period of production. This could be done with minor modifications. At moment n, the element $w_{[n]i}$ is 1 if the reel for the component i is in some feeder slot of the machine, otherwise it is 0. Let $P_{[n]}$ be an M-dimensional vector ($P_{[n]}$=$(p_{[n]1}, p_{[n]2}, ..., p_{[n]M})$), where the element $p_{[n]i} > 0$, if $w_{[n]i} = 1$ and $w_{[n-1]i} = 0$; otherwise it is 0. Thus, vector $P_{[n]}$ tells us the components to be introduced for the n^{th} product. A 0/1-formulation of the TSw problem is as follows:

$$MIN \sum_{n=1}^{N} (eP_n^T) \tag{4.9}$$

subject to

$$x_{jn} \cdot A_j \leq W_{[n]} \text{ for } n, j = 1, 2, \ldots, N, \tag{4.10}$$

$$eW_{[n]}^T \leq C \text{ for } n = 1, 2, \ldots, N, \tag{4.11}$$

$$\sum_{n=1}^{N} x_{jn} = 1 \text{ for } j = 1, 2, \ldots, N, \tag{4.12}$$

$$\sum_{j=1}^{N} x_{jn} = 1 \text{ for } n = 1, 2, \ldots, N, \tag{4.13}$$

$$P_{[n]} \geq W_{[n]} - W_{[n-1]} \text{ for } n = 1, 2, \ldots, N, \tag{4.14}$$

$$eW_{[0]}^T = 0, \tag{4.15}$$

$$P_{[n]} \geq \overline{0} \text{ for } n = 1, 2, \ldots, N, \tag{4.16}$$

$$w_{[n]i}, p_{[n]i}, y_n \in \{0, 1\} \text{ for } n = 1, 2, \ldots, N; \text{ for } i = 1, 2, \ldots, M. \tag{4.17}$$

The objective function (4.9) counts the number of component feeder changes. In constraints (4.10), (4.14) and (4.16) the operators " $\geq$ " and " $\leq$ " stand for the comparisons between the corresponding vector elements. Constraints (4.10) ensure that if PCB j is the n^{th} PCB to be processed, then all the components required by PCB j must be in the feeders at instant n. Constraints (4.11) assure the number of components does not exceed the machine's feeder capacity at any instant. Constraints (4.12) and (4.13) assign exactly one PCB to exactly one instant. Constraints (4.14) determine the component feeder changes for the n^{th} PCB in the sequence.

4.3 Hybridization of Grouping and TSw problems

As before, consider a high-mix production environment where multiple PCB types are produced on a single machine. The cost function for setup operations can then be taken as a weighted sum of number of setup occasions and number of component changes:

$$F(y, z) = Ry + Sz, \tag{4.18}$$

where R and S are constant weights (i.e., time factors) for the *number of setup occasions* (y) and for the *number of component changes* (z).

The problem of minimizing (18) is interesting in many respects. By setting $R > 0$ and $S = 0$ we have the common job grouping problem. By minimizing the objective function we get a sequence and an assignment of component feeders for each PCB that minimizes the number of setup occasions (groups). By setting $R = 0$ and $S > 0$ we have a tool switching problem where the objective is to minimize the number of tool switches (component feeder changes). However, joining these two objectives (i.e., $R > 0$ and $S > 0$) gives a still more realistic model of the machine setup problem. The number of setup occasions and the total number of feeder changes are both considered in this problem.

We formulate the hybrid machine setup problem as an IP model. In our formulation a decision variable (y_n) takes into account the number of setup occasions (groups). If the value of y_n is greater than zero, a setup occasion occurs and thus the processing of a new PCB group begins. In other words, the decision variable y_n is 1 if at least one of the elements in $P_{[n]}$ is greater than zero; otherwise y_n is 0. According to the above notations we can formulate the hybridization of the *job grouping problem* and the *minimum setup problem* as follows:

$$MIN \sum_{n=1}^{N} (Ry_n + SeP_{[n]}^T) \tag{4.19}$$

subject to

$$P_{[n]} \geq W_{[n]} - W_{[n-1]} \text{ for } n = 1, 2, \ldots, N, \tag{4.20}$$

$$eP_{[n]}^T \leq Cy_n \text{ for } n = 1, 2, \ldots, N, \tag{4.21}$$

$$x_{jn} \cdot A_j \leq W_{[n]} \text{ for } n, j = 1, 2, \ldots, N, \tag{4.22}$$

$$eW_{[n]}^T \leq C \text{ for } n = 1, 2, \ldots, N, \tag{4.23}$$

$$\sum_{n=1}^{N} x_{jn} = 1 \text{ for } j = 1, 2, \ldots, N, \tag{4.24}$$

$$\sum_{j=1}^{N} x_{jn} = 1 \text{ for } n = 1, 2, \ldots, N, \tag{4.25}$$

$$eW_{[0]}^T = 0, \tag{4.26}$$

$$P_{[n]} \geq \overline{0} \text{ for } n = 1, 2, \ldots, N, \tag{4.27}$$

$$w_{[n]i}, p_{[n]i}, y_n \in \{0, 1\} \text{ for } n = 1, 2, \ldots, N; \text{ for } i = 1, 2, \ldots, M. \tag{4.28}$$

The objective function in (4.19) calculates the weighted sum of the number of setup occasions and the number of component feeder changes. Notation $P_{[n]}^T$ stands for the transpose vector of $P_{[n]}$. Constraints (4.20) determine the component feeder changes for the n^{th} PCB in the sequence. Constraints (4.21) ensure that a setup occasion happens every time when there are one or several changes of the feeder reels when moving to a new PCB type. Constraints (4.22) ensure that all the components required for PCB j are on the machine at the instant n. Constraints ensure (4.23) that the number of allocated component feeders does not exceed the existing machine capacity. Constraints (4.24) and (4.25) indicate that each PCB is processed exactly once and at each instant there is exactly one PCB under processing.

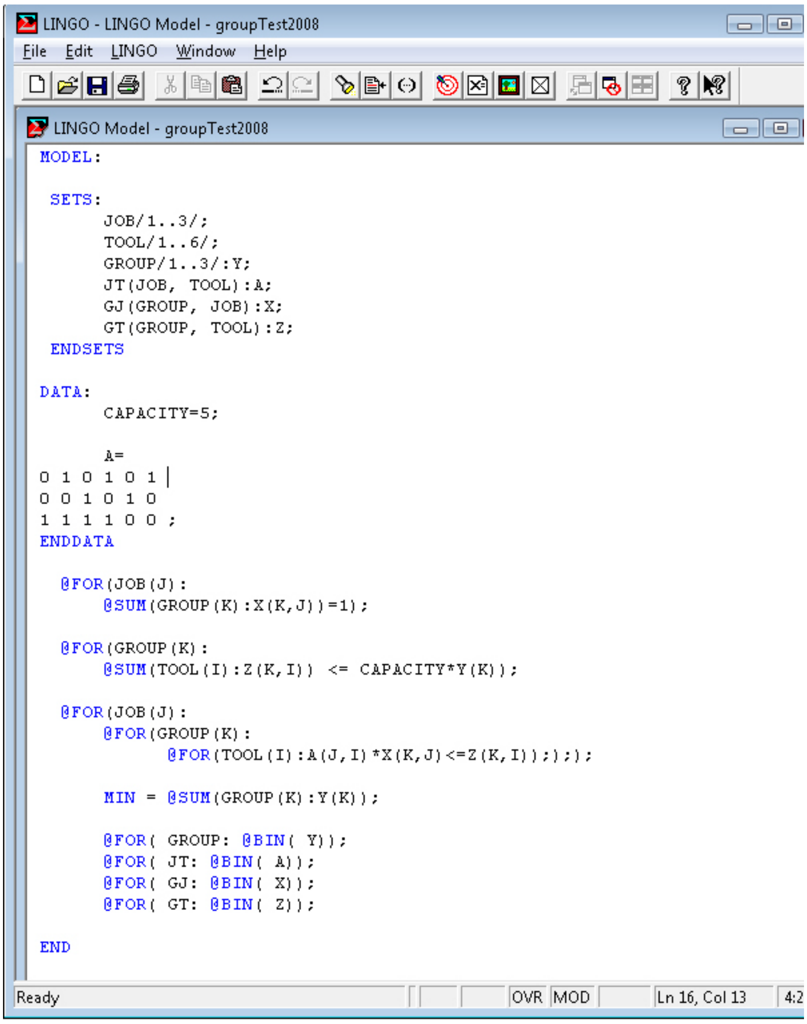
Chapter 5

Exact Methods

This section presents mathematical techniques that will be used to solve the IP problems of the previous section. In optimization the best solution is searched from a set of solutions. A trivial exact algorithm would choose the best solution among all possible solutions to the problem. In many cases searching all possible solutions would take a unbearable amount of time even with the fastest machine(s). Luckily, there are techniques that are able to find optimal solutions without going through the full solution space. These algorithms are effective for solving various search and global optimization problems. Different version of the branch and bound technique are often used to solve IP problems [23].

The branch and bound method [61] [39] is most widely used method for solving IPs. It is a general search method with two separate parts: a branch and a bound. In the branching part the problem is divided into smaller problems and in the bounding part the solutions of these smaller problems are estimated in order to restrict the problem's solution space more. The branch and bound algorithm starts by considering the optimization problem that is restricted to feasible region by explicit mathematical constraints. Lower bounds are searched by a procedure where integrality restrictions are relaxed (LP-relaxation). If the optimal solution to the relaxed problem matches the IP problem, then an optimal solution has been found and the search terminates. Otherwise, the feasible region of the problem is partitioned further to create even smaller subproblems. The algorithm generates recursively a tree of subproblems and searches optimal solutions to subproblems. The solution of the subproblem is not necessary globally optimal, but if it is feasible, it can be used the prune the rest of the tree. The search process continues until all subproblems are solved or subtrees pruned.

The branch and cut [38] method consists of a combination of a cutting plane method with a branch-and-bound algorithm. In the algorithm the lower bound is again provided by the linear-programming (LP) relaxation of the original integer program [61]. After branching, cutting plane methods are used to improve the relaxation by providing iteratively better lower bounds to the formulation. Unless the optimal solution to the LP is integral, a new constraint (a cutting plane) is searched by the algorithm. The new constraint violates to the LP solution, but it does not violate any optimal integer solutions. The new constraint is added to the formulation and the LP is solved again. Cutting planes are added as long as the optimal solution is not integral or the cut can be found.


LINGO - LINGO Model - groupTest2008
File Edit LINGO Window Help
LINGO Model - groupTest2008
Ready
OVR MOD
Ln 16, Col 13
4:2

```
MODEL:

 SETS:
      JOB/1..3/;
      TOOL/1..6/;
      GROUP/1..3/:Y;
      JT(JOB, TOOL):A;
      GJ(GROUP, JOB):X;
      GT(GROUP, TOOL):Z;
 ENDSETS

DATA:
      CAPACITY=5;

      A=
0 1 0 1 0 1
0 0 1 0 1 0
1 1 1 1 0 0 ;
ENDDATA

  @FOR(JOB(J):
      @SUM(GROUP(K):X(K,J))=1);

  @FOR(GROUP(K):
      @SUM(TOOL(I):Z(K,I)) <= CAPACITY*Y(K));

  @FOR(JOB(J):
      @FOR(GROUP(K):
            @FOR(TOOL(I):A(J,I)*X(K,J)<=Z(K,I));););

      MIN = @SUM(GROUP(K):Y(K));

      @FOR( GROUP: @BIN( Y));
      @FOR( JT: @BIN( A));
      @FOR( GJ: @BIN( X));
      @FOR( GT: @BIN( Z));

END
```

Figure 5.1: An simple instance of the job grouping problem modelled with LINGO.

The branch and price algorithm combines the branch and bound algorithm with column generation. The branch and price algorithm is used to solve integer programs when there are too many variables to represent the problem explicitly [61]. Column generation is a technique for solving large scale linear programs. It solves the problem only with a small subset of the decision variables. The branch and price algorithm applies the column generation technique to improve lower bounds at each node of the branch and bound tree until an optimum is reached.

Modeling languages and computer codes for IP algorithms are commonly available in mathematical software packages. For instance, with the LINGO optimization software package IP problems can be modelled and solved easily. Fig. 5.1 shows a simple instance of the job grouping problem written using the LINGO's modeling language.

Chapter 6

Heuristics

It is common that problem solving with exact methods turns out to be inefficient and time consuming. Especially, in many combinatorial problems the solution space may increase exponentially with respect to the problem size and finding an optimal solution in a reasonable time turns out to be often impossible. For example, IP problems arising in PCB assembly are mostly so complex combinatorial problems that they cannot be solved optimally by even the latest algorithms. For instance, the hybrid setup problem is a combination of two NP-hard problems: the group setup [10] and the minimum setup problems [52]. For the NP-hard problems is not known any algorithm that can solve all problem instances in a polynomial time. We have managed to solve (in a practical time) the hybrid setup problem optimally with optimization software for small problem instances [47], only. Therefore, exact methods become impractical for large real-world problems like PCB assembly optimization. In that case, we have to turn to heuristic methods that help us to find near optimal solutions and give sometimes even optimal solutions in an easier way. Good heuristics consume only a fractional part of the run times to in comparison to exact methods. Unfortunately, heuristics find usually local optimum solutions that are sometimes far away from global optimum, because there can be many local optimal solutions, where the heuristic may stuck. Therefore, one of the main quality aspects for a heuristic is how good local optimal solution it finds (how near global optimum it usually reaches). *Genetic algorithms*, *simulated annealing*, and *tabu search* are known heuristics for global optimization problems. A commonality for them is that they iteratively attempt to find better solutions, but their way to search and construct solutions differs quite a lot. A genetic algorithm uses evolutionary principles in the form of selection, crossover and mutation operations to a population of solution candidates. The aim is to iterate these operations to the individuals of the population in order to find better solution candidates. The simulated annealing technique mimics the metal cooling process to construct and improve solutions from previous ones. In the tabu search we are moving from a solution to another avoiding to stuck on a local optimum by accepting every now and then inferior solutions and using a tabulist from previous solutions to prohibit rechecking of previous solutions. For description of heuristics, see [19] [1].

In this section, we give more detailed discussion on group, minimum, and hybrid setup heuristics, because in this thesis we concentrate on a high-mix PCB production environment.

6.1 Group setup heuristic

Numerous researchers have studied the group setup strategy and developed algorithms to it: Hashiba and Chang [20] suggest grouping PCBs as the first step in their three-stage procedure of reducing setup times, and they present heuristics for the grouping problem. Maimon and Shtub [37] present a mixed-integer programming formulation and a heuristic method for grouping a set of PCBs to minimize the total setup time. Shtub and Maimon [48] consider the PCB grouping problem as an extension of the set-covering problem. Daskin et al. [14] present a mathematical formulation for the PCB-grouping problem, show that the problem is NP-complete, and give a branch-and-bound based heuristic algorithm for solving it. Bhaskar and Narendran [7] apply graph theory for solving the grouping problem. Crama et al. [11] analyze the approximation of the job grouping and minimum setup strategies. Smed et al. [50] introduce and compare several heuristics algorithms based on greedy, clustering, and repair-based local search methods. Knuutila et al. [30] compare the results of efficient heuristics to optimal solutions found by 0/1-programming and by constraint logic programming. Zolfaghari and Liang [57] present a genetic algorithm for solving a general machine/part grouping problem in which processing times, lot sizes and machine capacities are considered.

As an example, consider the solution of the PCB-grouping problem by the use of hierarchical clustering [36]. At the first each PCB forms a singleton cluster and the algorithm calculates Jaccards similarity coefficient for each cluster pair and merges the pair with the highest coefficient value (given that the capacity is not exceeded). If the merge operation cannot be realized, the algorithm chooses the pair with the highest similarity coefficient so that the merge is feasible. After that, the similarity coefficients are updated. The process is iterated until no improvement is possible. A general group setup algorithm (GSA) based on hierarchical clustering is as follows:

```
Input: pcbs(a set of PCBs)
Output: groups(a set of PCB-groups)
function GSA($PCB_{set}$ pcbs)
    $GROUP_{set}$ groups
    for every $pcb_i \in$ pcbs
        change $pcb_i$ to $group_i$
        add $group_i$ to groups
    endfor
    calculate similarity coefficient $s_{ij}$ for every $groupPair_{i,j} \in$ groups
    while (feasible merging is possible)
        merge the pair with the max($s_{ij}$) among the feasible merging groups
        update similarity coefficient $s_{ij}$ for every $groupPair_{i,j} \in$ groups
    endwhile
    return groups
```

The similarity between PCBs can be calculated in many ways e.g.: Shtub and Maimon [48] examined a usage of *Jaccard's similarity coefficient* $S_{ij} = \frac{|E_i \cap E_j|}{|E_i \cup E_j|}$, where the set E_i (E_j) denotes the components of the board $i(j)$. Bhaskar and Narendran [7] define the cosine similarity coefficient of boards i and j as the cosine of the angle between the pair of row vectors $\bar{i}$ and $\bar{j}$ that correspond to the boards $S_{ij} = cos(\Theta_{ij}) = \frac{\bar{i} \cdot \bar{j}}{|\bar{i}| \cdot |\bar{j}|} = \frac{|E_i \cap E_j|}{\sqrt{|E_i|} \cdot \sqrt{|E_j|}}$.

6.2 Minimum setup heuristic

Minimum setup heuristic attempts to sequence a given set of different PCB types so that the number of feeder changes is minimized. Barnea and Sipper [6] translated the mathematical model of the tool switching problem of Tang and Denardo [52] to the PCB environment and presented a heuristic for minimizing setups. Jain et al. [26] developed a mathematical model and a four-stage method for sequencing jobs on a PCB assembly line, where a rolling horizon of production is taken into account. Günther et al. [18] presented and compared three heuristics for solving a minimum setup problem on a typical surface mount technology production line. Dillon et al. [15] presented four variants of a greedy heuristic that aims at maximizing iteratively the component commonality whenever the PCB type changes. Narendran and Rajkumar [29] [28] also applied the minimum setup strategy in their heuristics to minimize the total setup time on a PCB assembly environment.

Hertz et al. [25] presented and compared several algorithms based on efficient traveling salesman problem (TSP) heuristics and new distance definitions for minimizing the number of tool switches in a flexible machine. Al-Fawsan and Al-Sultan [1] proposed a variety of tabu search algorithms for solving the tool switching problem. Tzur and Altman [59] studied the tool switching problem, when each tool may occupy more than one slot in the tool magazine.

The efficient minimum setup algorithm (MSA) by Hertz et al. [25] has the following idea: The algorithm starts by forming a complete weighted graph G where the PCBs are nodes and the weight of the arc between node i and j is calculated from the expression $|E_i \cup E_j| - |E_i \cap E_j|$ (giving the number of non-mutual components of PCB i and j). The algorithm repeats the next four steps N times by taking each node as a starting node:

1. Solve the traveling salesman problem for G with a heuristic by starting from the selected node.

2. Improve the solution with a 2-opt heuristic.

3. Use the *keep tool needed soonest* (KTNS [52]) method to assign components to the feeder magazine when visiting the nodes in the sequence given in the previous steps.

4. Evaluate the solution by using the chosen cost function; if its value is better than the current best, store it.

The TSP can be solved approximately with different heuristics e.g. *nearest insertion* method [33] or *farthest insertion* method [33].

When the PCB type changes one or several component reels may be returned to the secondary storage in order to free feeder space. According to the KTNS-rule those component reels are left to the feeder magazine that are needed soonest in the sequence of processing the different PCB types. In our study [24], we reconstructed the KTNS-rule for the tool loading problem (TLP), where the component reels are of different sizes taking more than one adjacent slot from the feeder magazine. While the job processing order is free in the minimum setup problem, it is assumed to be fixed in TLP.

6.3 Hybrid setup heuristic

In our studies [51][47] we presented a general model that can be applied to construct an efficient hybrid setup heuristic:

The hybrid setup algorithm (HSA) forms PCB groups (or families) like the group setup algorithm (GSA), but each time after merging two groups, it calls the minimum setup algorithm (MSA) and evaluates the cost function. If the found solution is better than the previous one, it is saved. The advantage of this method is that it searches the solution space globally. The algorithm HSA for solving the hybrid setup problem is as follows:

```
Input: pcbs(a set of PCBs)
Output: feeders(sequenced sets of component feeders)
function HSA($PCB_{set}$ pcbs)
    $PCB_{set}$ pcbs2
    $GROUP_{set}$ groups
    $Cost$ cost1=$\infty$
    $Cost$ cost2
    $FeederAssignment_{set}$ fa1
    $FeederAssignment_{set}$ fa2
    for every $pcb_i \in$ pcbs
        change $pcb_i$ to $group_i$
        add $group_i$ to groups
    endfor
    while (feasible merging is possible)
        calculate similarity coefficient $s_{ij}$ for every $groupPair_{i,j} \in$ groups
        merge the pair with the max($s_{ij}$) among the feasible merging groups
        pcbs2=makeSuperPCBs(groups)
        fa2=MSA(pcbs2)
        cost2=costFunction(fa2)
        if(cost2<cost1)
            fa1=fa2
            cost1=cost2
        endif
    endwhile
    return fa1
```

Although HSA is at its best to solve the hybrid setup problem, our study [46] has shown that it often gives better solutions also to the minimum setup problem than the sole application of sequencing heuristics for minimum setup. As a further benefit, the algorithm is fast even for more complex cases of real production data from PCB assembly and it is therefore a good candidate for the practical heuristics of the tool switching problem.

Chapter 7

Summary of Publications

IN THE FIRST PUBLICATION [51], we consider the hybridization of group setup strategies and minimum setup strategies in PCB assembly. Usually, the machine setup problem is formulated as a grouping problem or as a minimum setup problem. The objectives of these two strategies are incompatible with each other. However, if one wants to operate in a real-world production environment of the PCB assembly, it is necessary to consider both problems at the same time. Therefore, we define an objective function which is a weighted sum of the number job groups and feeder changes. We modify three setup algorithms and four minimum setup algorithms proposed previously in the literature to work with this hybrid problem. In addition, we present a new algorithm (GMSA) which combines both setup strategies. The new algorithm uses a hierarchical clustering algorithm the Jaccard's similarity measure for grouping the boards. Each board in the same group can be assembled without any feeder changes. After that, each group is considered as a super-PCB. An efficient minimum setup algorithm (Genius) is used to minimize the amount of work for feeder changes between the super-PCBs. The feeder assignments of each permutation of the super-PCBs are generated by the KTNS (Keep Tool Needed Soonest) rule. According to the KTNS -rule those component feeders are removed from the feeder magazine, which are needed latest. The performance of the algorithms is tested with real-world production data. The GMSA outperformed the other algorithms in these tests, except for the case, where the number of the setup occasions (groups) is omitted.

IN THE SECOND PUBLICATION [47], we study further the hybrid of group setup and minimum setup strategies. The problem is formulated as an Integer Programming (IP) model and small problem instances are solved optimally with the LINGO optimization package. We introduce two more new algorithms (GMSA2, GMSA3) based on efficient group and minimum setup heuristics. The GMSA2 differs from GMSA1 (the GMSA1 is the same as the GMSA in the first paper) with respect to its stopping criterion; GMSA2 finishes the merging of the groups when the similarities between the groups drop below a certain limit. The GMSA3 calls an efficient minimum setup algorithm (MSAGenius) after each time it merges PCB clusters and evaluates the cost function. Only, if the new cluster improves the result, it is saved. The algorithms are tested with PCB data drawn from a high-mix low-volume PCB assembly environment. Practical tests indicated that the new algorithms can improve the solutions of the previous algorithms. The algorithm GMSA3 yielded the overall best results.

IN THE THIRD PUBLICATION [46], we study the power of hybrid algorithms to the tool switching problem (TSwP) which arises in the metalworking industry, where numerically controlled flexible machines are used to manufacture parts. The TSwP is just like the minimum setup problem in the electronic industry, but the objective is to sequence parts instead of PCBs so that the total number of tool switches instead of feeder setups is minimized. We compared our best hybrid algorithm with the best TSwP heuristics proposed in the literature. As the purpose of our study is to find useful methods for real production environments, we consider both the solution quality and the running time. We used two different sets of test problems. The first was a random generated test problems from literature and the second set of problems was generated from PCB data. Out of 10 methods only GENIUS* could beat our method, but the drawback of the GENIUS* is its impractical long running time.

IN THE FOURTH PUBLICATION [24], we concentrate on the tool loading problem (TLP), which is subproblem of the TSwP. In the TLP it is assumed that the job processing order is fixed. The objective of the TLP in a PCB context is to minimize the component reel switching costs. Before the components can be placed from the feeders to the board, the component reels have to be loaded from the secondary storage to the feeder magazine (or rack). If the feeder magazine of the component placement machine is fully loaded, one or several component reels are returned to the secondary storage in order to free feeder space. In the case where every component reel takes only one feeder slot of the linear feeder magazine space the TLP can be solved optimally by the Keep Tool Needed Soonest (KTNS) -rule. The TLP becomes more complex, if the component reels are of different sizes taking more than one adjacent slot from the feeder rack. Therefore, when changing the PCB type, feeder reels have to be reorganized in the feeder rack because of the fragmentation of free slots. First, we give a formulation of the general two-level tool management problem (GSM-1), where component reels are of the same size. A mathematical programming solution is then given for the GSM-1. Then we generalize the GSM-1 by extending it to concern different component reel sizes (GSMM). We develop heuristic solution algorithms for both problems (GSM-1 and GSMM) and compare their efficiency against naive, random, and lower bound methods. The results of the new algorithms are not far from optimal solutions and the solutions are found in short running times for practical problem sizes.

IN THE FIFTH PUBLICATION [45], we consider production planning and machine control strategies for a component placement machine in different product volume situations. It was known before hand that unique setup methods are beneficial for big batch sizes and group setup methods should be favoured if batch sizes are small. We propose a new method, GreedyTot, that is an intermediate between job grouping and unique setup techniques taking into consideration the benefits of both techniques. The GreedyTot method starts from a solution, where the unique setup strategy is applied for all PCBs and it continues by forming feasible pairs of PCBs. If the best PCB pair decreases the production time they are combined to form a group. The grouping steps are repeated as long as an improving group can be found. The algorithm takes into account several factors when counting production times: PCB batch sizes, the number of different components and their quantities, the number of setup occasions, and the number of feeder changes. We compare the new technique to group setup and unique setup techniques by varying the production batch sizes from low-volume to large-volume. With the help of a machine simulator of an existing commercial planning tool, we get realistic production times to the different techniques. Based on our tests, the GreedyTot method yields clearly better results than the unique or group setup methods for moderate batch sizes.

Chapter 8

Conclusions

Due to the competition between PCB manufacturers, they are forced to specialize their assembly manufacturing to either a few product types in large volumes or a large variety of products in small volumes [11]. In this work, we have discussed setup optimization in SMT assembly in the case, where a wide variety of PCBs are produced. Nowadays, the above mentioned situation is common in many factories.

To increase the productivity in a high mix production environment, it is more important to minimize setup times instead of minimizing component placement operation times, because setup times for PCB types are usually large in comparison to component placement time on a board. Therefore, we have omitted certain machine designs in our studies and concentrated on minimizing the setup times. The machine downtime consumed by different setup operations reduce on production output significantly. On the other hand, if batch sizes of the PCBs increase significantly, more machine specific examinations are needed to improve production throughput.

In our studies, minimizing the total setup time was considered as the main objective. We presented an objective function included both the number of setup occasions and the number of feeder changes. The idea of this hybridization model was that it simulated the real-world production planning situation in a flexible way. By selecting suitable parameters the model can put more weight on the number of setup occasions or on feeder changes.

The setup problems were also presented more concisely using mathematical formulations. We developed an integer programming model to the hybrid setup problem. Although, there are mathematical techniques (exact methods) that are capable of finding optimal solutions for optimization problems without going through the whole solution space, this hybrid setup problem can be solved optimally only for small instances with exact methods.

Thus, efficient heuristics are needed for problems of realistic size. Different heuristics from earlier studies were collected and studied in the setup optimization problems. One of our main task has been to develop for the setup problems efficient heuristic methods that attempt to find solutions globally. We have constructed a specific heuristic for the hybrid setup problem that takes into account the situation changes in the problem and considers the tradeoff between manual operation times and machine times. We indicated by considering both the solution quality and the running time that our hybrid algorithms were useful also to the tool switching problem (TSwP) which arises in the metalworking industry.

Bibliography

[1] Al-Fawzan M. A., and Al-Sultan K. S. A tabu search based algorithm for minimizing the number of tool switches on a flexible machine. *Computers & Industrial Engineering*, (2002), 35-47.

[2] Ayob M., Cowling P., and Kendall G. Optimisation for Surface Mount Placement Machines. *Proc. of the IEEE ICIT'02*, (2002), 498-503.

[3] Ayob M., and Kendall G. A trible objective function with a Chebychev dynamic pick-and-place point specification approach to optimise the surface mount placement machine. *European Journal of Operational Research*, 164(2005), 609-626.

[4] Bard J. F., Clayton R. W., and Feo T. A. Machine Setup and Component Placement in Printed Circuit Board Assembly. *International Journal of Flexible Manufacturing Systems*, 6(1994), 5-31.

[5] Ball M. O., and Magazine M. J. Sequencing of Insertions in Printed Circuit Board Assembly. *Operations Research 36*, 2(1988), 192-201.

[6] Barnea A., and Sipper D. Set-up Reduction in PCB Automated Assembly, *Computer Integrated Manufacturing Systems*, 6(1993), 18-26.

[7] Bhaskar G., and Narendran T. T. Grouping PCBs for Set-up Reduction: A Maximum Spanning Tree Approach, *International Journal of Production Research*, 34(1996), 621-632.

[8] Crama Y., Flippo O. E., van de Klundert J., and Spieksma F. C. R. The Assembly of Printed Circuit Boards: A Case with Multiple Machines and Multiple Board Types. *European Journal of Operational Research*, 98(1997), 457-472.

[9] Crama Y., Kolen A. W. J., Oerlemans A. G., and Spieksma F. C. R. Throughput Rate Optimization in the Automated Assembly of Printed Circuit Boards. *Annals of Operations Research*, 26(1990), 455-480.

[10] Crama Y., Kolen A. W. J., Oerlemans A. G., and Spieksma F. C. R. Minimizing the Number of Tool Switches on a Flexible Machine. *International Journal of Flexible Manufacturing Systems*, 6(1994), 33-54.

[11] Crama Y., Kolen A. W. J., and Spieksma F. C. R. Production plannig problems in printed circuit board assembly. *Discrete Applied Mathematics*, 123(2002), 339-361.

[12] Crama Y., Oerlemans A. G., and Spieksma F. C. R. Production Planning in Automated Manufacturing. Springer-Verlag, 1994.

[13] Deo S., Javadpour R., and Knapp G. M. Multiple setup PCB assembly planning using genetic algorithms. *Computers & Industrial Engineering*, 42(2002), 1-16.

[14] Daskin M. S., Maimon O., Shtub A., and Braha D. Grouping Components in Printed Circuit Board Assembly with Limited Components Staging Capacity and Single Card Setup: Problem Characteristics and Solution Procedures. *International Journal of Production Research*, 35(1997), 1617-1638.

[15] Dillon S., Jones R., Hinde C. J., and Hunt I. PCB Assembly Line Setup Optimization Using Component Commonality Matrices, *Journal of Electronics Manufacturing*, Vol.8, No.2, 1998, pp.77–87.

[16] Ellis K. P., Kobza J. E., and Vittes F. J. Development of a placement time estimator function for a turret style surface mount placement machine. *Robotics and Computer-Integrated Manufacturing*, 18(2002), 241-254.

[17] Grunow M., Günther H., Schleusener M., and Yilmaz O. Operations planning for collect-and-place machines in PCB assembly. *Computers & Industrial Engineering*, 47(2004), 409-429.

[18] Günther, H. O. and Gronalt, M. and Zeller, R. Job Sequencing and Component Set-up on a Surface Mount Placement Machine. *Production Planning & Control*, 9(1998), 201-211.

[19] Haataja J. Optimointitehtvien ratkaiseminen. CSC-Tieteellinen laskenta Oy, 2004, 3. uudistettu painos.

[20] Hashiba S., and Chang T. C. Heuristic and Simulated Annealing Approaches to PCB Assembly Setup Reduction, In G. j. Olling and F.

Kimura (Eds.), Human Aspects in Computer Integrated Manufacturing, IFIP Transactions b3 (pp. 769-777), Amsterdam, The Netherlands: North Holland, Elsevier Science Publishers, 1992.

[21] Ho W., and Ji P. Optimal Production Planning for PCB Assembly. Springer London, 2007.

[22] Ho W., and Ji P. Component scheduling for chip shooter machines: a hybrid genetic algorithm approach *Computers & Operations Research*, 30(2003), 2175-2189.

[23] Hillier F. S., and Lieberman G. L. Introduction to Mathematical Programming. Singapore: McGraw-Hill International editions, 1995.

[24] Hirvikorpi M., Salonen K., Knuutila T., and Nevalainen O. The general two-level storage management problem: A reconsideration of the KTNS-rule. *European Journal of Operational Research 171*, 1(2006), 189-207.

[25] Hertz A., Laporte G., Mittaz M., and Stecke K. E. Heuristics for Minimizing Tool Switches When Scheduling Part Types on a Flexible Machine, *IIE Transactions*, 30(1998), 689-694.

[26] Jain S., Johnson M. E., and Safai F. Impelementing setup optimization on the shop floor. *Operations Research*, 43(1996), 843-851.

[27] Johnsson M., and Smed J. Observations on PCB Assembly Optimization. *Electronic Packaging & Production 41*, 5(2001), 38-42.?

[28] Kumar K. R., and Narendran T. T. A Heuristic for Sequencing PCBs with Due-dates. *International Journal of Operations & Production Management*, 17(1997),446-467.

[29] Kumar K. R., and Narendran T. T. A Heuristic for Sequencing PCB Assembly to Minimize Set-up Times. *Production Planning & Control*, 9(1996):465-476.

[30] Knuutila T., Puranen M., Johnsson M., and Nevalainen O. Three Perspectives for Solving the Job Grouping Problem. *International Journal of Production Research*, 39(2001), 4261-4280.

[31] van Laarhoven P. J. M., and Zijm W. H. M. Production Preparation and Numerical Control in PCB Assembly. *International Journal of Flexible Manufacturing Systems*, 5(1993), 187-207.

[32] Landers T. L., Brown W. D., Fant E. W., Malmstrom E. M., and Schmitt N. M. *Electronics Manufacturing Processes.* Prentice-Hall, Englewood Cliffs, NJ, 1994.

[33] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., and Shmoys, D. B. The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley & Sons, New York, NY, 1985.

[34] Leipälä T., and Nevalainen O. Optimization of the Movements of a Component Placement Machine. *European Journal of Operational Research*, 38(1989), 167-177.

[35] Leon V. J., and Peters B. A. Replanning and Analysis of Partial Setup Strategies in Printed Circuit Board Assembly Systems. *International Journal of Flexible Manufacturing Systems*, 8(1996), 389-412.

[36] Leon J. V., and Peters B. A. A Comparison of Setup Strategies for Printed Circuit Board Assembly. *Computers & Industrial Engineering*, 34(1998), 219-234.

[37] Maimon O., and Shtub A., Grouping Methods for Printed Circuit Boards. *International Journal of Production Research*, 29(1991), 1379-1390.

[38] Mitchell J., and Lee E. Branch-and-cut algorithms for integer programming. Encyclopedia of Optimization, Volume II, pages 519-525, Kluwer Academic Publishers, 2001.

[39] Mitchell J., and Lee E. Branch-and-bound methods for integer programming. Encyclopedia of Optimization, Volume II, pages 509-519, Kluwer Academic Publishers, 2001.

[40] Narayanaswami R., and Iyengar V. Setup reduction in printed circuit board assembly by efficient sequencing. *International Journal of Advanced Manufacturing Technology*, 26(2005), 276-284.

[41] Ohno K., Jin Z, and Elmaghraby E. S. An optimal assembly mode of multi-type printed circuit boards. *Computers & Industrial Engineering*, 36(1999), 451-471.

[42] Peters B. A., and Subramanian G. S. Analysis of Partial Setup Strategies for Solving the Operational Planning Problem in Parallel Machine Electronic Assembly Systems. *International Journal of Production Research*, 34(1996), 999-1201.

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

[44] Rossetti M. D., and Stanford K. J. A. Group sequencing a PCB assembly system via an expected sequence dependent setup heuristic. *Computers & Industrial Engineering*, 45(2003), 231-254.

[45] Salonen K., Knuutila T., Johnsson M., and Nevalainen O. Planning and Controlling the Component Assembly for Multimodel PCBs. *WSEAS Transactions on Systems* , 4(2006), 855-863.

[46] Salonen K., Raduly-Baka C., and Nevalainen O. A note on the tool switching problem of a flexible machine. *Computers & Industrial Engineering* , 4(2006), 458-465.

[47] Salonen K., Smed J., Johnsson M., and Nevalainen O. Grouping and sequencing PCB assembly jobs with minimum feeder setups. *Robotics and Computer-Integrated Manufacturing* , 4(2006), 297-305.

[48] Shtub A., and Maimon O., Role of Similarity in PCB Grouping Procedures, *International Journal of Production Research*, 30(1992), 973-983.

[49] Smed J. Production Planning in Printed Circuit Board Assembly. Ph.D. Thesis, university of Turku, Turku, Finland (2002).

[50] Smed J., Johnsson M., and Puranen M. Leipälä T., Nevalainen O. Job Grouping in Surface Mounted Component Printing, *Robotics and Computer-Integrated Manufacturing*, 15(1999), 39-49.

[51] Smed J., Salonen K., Johnsson M., Johtela T., and Nevalainen O. Grouping PCBs with Minimum Feeder Changes. *The International Journal of Flexible Manufacturing Systems*, 15(2003), 19-35.

[52] Tang C. S., and Denardo E. V. Models Arising from a Flexible Manufacturing Machine, *Operations Research*, 36(1988), 767-784.

[53] Van Hop N., and Naganur N. N. The scheduling problem of PCBs for multiple non-identical parallel machines. *European Journal of Operational Research*, 158(2004), 577-594.

[54] Van Hop N., and Tabucanon M. T. Multiple criteria approach for solving feeder assignment and assembly sequence problem in PCB assembly. *Production Planning & Control*, 12(2001), 736-744.

[55] Williams S. K., and Magazine M. J. Heuristic approaches for batching jobs in printed circuit board assembly. *Computers and Operations Research* , 34(2007), 1943-1962.

[56] Yilmaz I. O. , Grunow M., Günther, H. O. and Yapan, C. Development of group setup strategies for makespan minimisation in PCB assembly. *International Journal of Production Research*, 45(2007), 871 - 897.

[57] Zolfaghari S., and Liang M. A new genetic algorithm for the machine/part grouping problem involving processing times and lot sizes. *Computers & Industrial Engineering*, 45(2003), 713-731.

[58] Zijm W. H. M., and van Harten A. Process Planning for a Modular Component Placement System. *International Journal of Production Economics*, 30-31(1993), 123-135.

[59] Tzur M., and Altman A. Minimization of tool switches for a flexible manufacturing machine with slot assignment of different tool sizes. *IIE Transactions*, 36(2004), 95-110.

[60] http://www.tutorialsweb.com/smt/smt.htm (15.1.2008)

[61] http://www.cs.sandia.gov/opt/survey/mip.html (15.1.2008)

Publication Reprints

Publication I

Smed, J., Salonen, K., Johnsson, M., Johtela, T., Nevalainen, O. Grouping PCBs with Minimum Feeder Changes. *The International Journal of Flexible Manufacturing Systems 15*, (2003), 19-35.

Publication II

Salonen, K., Smed, J., Johnsson, M., Nevalainen, O. Grouping and sequencing PCB assembly jobs with minimum feeder setups. *Robotics and Computer-Integrated Manufacturing 22*, 4(2006), 297-305.

Publication III

Salonen, K., Raduly-Baka C., Nevalainen, O. A note on the tool switching problem of a flexible machine. *Computers & Industrial Engineering 50*, 4 (2006), 458-465.

Publication IV

Hirvikorpi, M., Salonen, K., Knuutila, T., Nevalainen, O. The general two-level storage management problem: A reconsideration of the KTNS-rule. *European Journal of Operational Research 171*, 1(2006), 189-207.

Publication V

Salonen, K., Knuutila, T., Johnsson, M., Nevalainen, O. Planning and Controlling the Component Assembly for Multimodel PCBs. *WSEAS Transactions on Systems 5*, 4(2006), 855-863.

Turku Centre for Computer Science
TUCS Dissertations

76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory
86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly

Turku Centre *for* Computer Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Information Technologies

Turku School of Economics
- Institute of Information Systems Sciences

ISBN 978-952-12-2167-5
ISSN 1239-1883

Kari Salonen Setup Optimization in High-Mix Surface Mount PCB Assembly

Kari Salonen Setup Optimization in High-Mix Surface Mount PCB Assembly