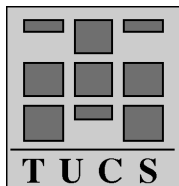


Techniques and Applications of Production Planning in Electronics Manufacturing Systems

Jouni Smed
Mika Johnsson
Tommi Johtela
Olli Nevalainen



Turku Centre for Computer Science
TUCS Technical Report No 320
December 1999
ISBN 952-12-0577-6
ISSN 1239-1891

Abstract

Electronics industry is a major part of modern manufacturing, and electronic systems play an increasingly important role in a majority of today's products. Electronic systems are usually implemented with printed circuit boards (PCBs), and, consequently, PCB assembly has become an important sector of the electronics manufacturing industry overall. However, operating effectively in this industry is becoming more difficult as the companies must compete with high quality standards, rapidly changing technologies, short production cycles, and increasing product variety and complexity. In addition, the capital equipment cost of electronics assembly industry facilities are high in comparison to the usual turnover of a company. As a result, production planning decisions need to be made more and more frequently due to continuous changes in the production conditions.

In this work we discuss production planning in electronics assembly—and, in particular, in PCB assembly. Our intention is to identify the typical problems arising from production planning and to give a survey of the solution methods suggested in the literature. In addition to this theoretical perspective, we will briefly review applications designed for production planning in PCB assembly.

This work is organized as follows: We begin with an introduction to flexible manufacturing systems in general and present a framework for production planning systems in Section 1. Next, we study the fundamentals of PCB assembly process in Section 2 and survey the relevant literature in Section 3. In Section 4 we review existing commercial applications for production planning, and in Section 5 study more closely one of these systems. Finally, in Section 6 we sum up the discussion and outline few important topics for the future research.

Keywords: production planning, printed circuit boards, electronics assembly, flexible manufacturing systems

TUCS Research Group
Algorithmics

1 Introduction

Let us begin our discussion by recalling some basic concepts of industrial production. Five basic types of production operations can be classified according to the degree repetitiveness involved [20]: project, jobbing, batch, flow, and process. The *project form* includes large-scale complex products, and it involves the allocation and coordination of large-scale input to achieve a unique product. *Jobbing* describes a situation where the manufacturing of a whole product is considered as one operation and the work must be completed on each product before starting on the next. In the *batch form*, the volume of products to be manufactured is larger than in jobbing. A regular and consistent demand for a product such that the item can be produced for stock identifies the *flow production*. Lastly, the *process production* requires that the material is involved in a continuous process.

The type of production affects also the plant layout. There are three basic forms of layout [20]: process, product, and group. The *process layout*, in which all the plants associated with a particular type of process are grouped together, is typical in jobbing and batch production. In the *product layout*, which occurs in flow and process forms of production, the plant is laid out according to the sequence of processes required by the product. A *group* (or cellular) *layout* is typical in batch production, and it involves the recognition that many of the products have similarities in their makeup and they can thus be grouped into (product) families.

There are two basic principles for decomposing a manufacturing facility into subsystems: decomposition based on processes and decomposition based on products [42]. *Process based facility decomposition* leads to equipment being arranged into functional machining areas (*work centers*) dedicated to general manufacturing processes (i.e., *job shop*). *Process planning* is the systematic determination of the detailed methods by which parts can be manufactured from raw material to finished products. *Computer-aided process planning* (CAPP) *determines a set of instructions and machining parameters required to manufacture a part and prepares data for production planning and scheduling activities* [85]. As Zijm and van Harten [120] observe, CAPP bridges the gap between *computer aided design* (CAD) and *computer aided manufacturing* (CAM), and thus CAPP represents the “/” in CAD/CAM. *Process control* refers to the automatic monitoring and control of a process by an instrument or system configured or programmed to respond appropriately to process feedback [102].

Product based decomposition utilizes the principle of group technology by dedicating machines to cells in order to produce the associated families of parts (i.e., *cellular manufacturing*). *Production planning* refers to the

process of establishing strategies for producing finished products so that manufacturing resources are used efficiently. When there is a large number of production variables and a long planning horizon, the problem can be approached by breaking it hierarchically into a series of decision levels [109]. *Production control is the systematic planning, coordination and direction of all manufacturing activities to ensure that products (of adequate quality) are made on time and at reasonable cost* [102]. To summarize, in production planning we first *make a plan* anticipating the future events and after that *follow the plan*, whereas in production control we simply *react to the events* as they occur during the production.

In the remainder of this section we discuss flexible manufacturing systems (FMSs) in Section 1.1, and develop a common methodology for building practical production planning systems in FMSs in Section 1.2.

1.1 Flexible Manufacturing Systems

In the late 1950s, several ideas for improving manufacturing began to surface. One of the earliest was the idea of group technology for manufacturers who had to make a variety of different but similar parts. In some types of industry—for example, chemical and oil-refining industries—*automated manufacturing* has a long history, but in the batch-manufacturing industries—like the metalworking industry and the electronics industry—the concept of automated manufacturing was introduced in the early 1970s. By then, a number of technological developments, in particular flexible manufacturing systems, offered solutions to the problems of job shop environments.

Automated manufacturing has a wide variety of potential benefits to offer. One of the most important advantages is the increased ability to respond to changes in demand and changes in the products, which is essential in the today's view of short production cycles. Other advantages include shorter lead times, reduction in the work-in-process levels and improved machine utilization. At the same time it is not an easy task to fully attain these possibilities, because the reality of the shop floor rarely coincides with the theoretical models (as we shall see in Section 1.2.1).

A *flexible manufacturing system* (FMS) aims at achieving a similar level of efficiency for manufacturing several different product types as in the mass production of a single product type. An FMS comprises a *group of programmable production machines integrated with automated material handling equipment which are under the direction of a central controller to produce a variety of parts at non-uniform production rates, batch sizes and quantities* [53]. The machines or work stations are used to perform operations on parts, and each operation requires a number of tools that can be stored in the limited capac-

ity tool magazine of the machines. An automatic tool interchanging device switches the tools during production. Because this interchange is relatively quick, the machine can perform several operations with *virtually no setup time* between the operations, if the required tool is present in the tool magazine.

Electronics assembly (especially printed circuit board assembly) plants are usually FMSs. However, the terminology associated with FMS, which originates from metal industry, can be somewhat confusing when applied to electronics assembly. For example, the concept of ‘tool’ refers to a *feeder*, which contains the electronic component to be mounted, rather than the actual tool (or *nozzle*) which does the printing operation. We discuss the technical aspects of electronics assembly at greater length in Section 2.

The most prevalent analytical approach to real-time FMS control attempts to hierarchically decompose the problem into a number of more easily manageable subproblems, which relate to a variety of decisions concerning *long-term*, *medium-term* or *short-term planning*. One of the main reasons for decomposing the general planning problem is that this problem is too complex to be solved globally, whereas it is easier to solve each subproblem one at a time. The solution to the global problem can then be obtained by solving the subproblems successively. Naturally, this solution is not likely to be globally optimal, even if all subproblems are solved to optimality. Nonetheless, this approach is a productive and popular way to tackle hard problems.

A typical hierarchical classification scheme of FMS [28] discerns

1. *strategic level* or *long-range planning* which concerns the initial deployment and subsequent expansion of the production environments (e.g., the *design and selection of the equipment and of the products* to be manufactured),
2. *tactical level* or *medium-range planning* which determines the allocation patterns of the system production capacity to various products so that external demands are satisfied (e.g., by solving *batching* and *loading problems*), and
3. *operational level* or *short-range planning* which coordinates the shop floor production activities so that the higher level tactical decisions are observed (e.g., by solving *release* and *dispatching problems*).

Maimon and Shtub [82] and Johnsson [54] relate these objectives to electronics assembly: In the strategic level, the planning focuses on determining the best set of production equipment for the operation (e.g., running a simulation on how much money should be invested in new equipments and what kind

of machines should be purchased [35]). These decisions are usually made on economical basis, and they are revised over long operational periods, typically measured in several months [34, 66]. At the tactical level, the decisions concern machine and line configurations, production schedules, batch sizes, and work-in-process levels. Finally, the operational level addresses the day-to-day operation of the equipment (e.g., how to manufacture a product). *The tactical and operational problems have to be solved frequently, and, consequently, the existing production planning systems concentrate on these levels.*

Notwithstanding the similarities, there are also differences between PCB assembly and FMSs. Klegka and Driels [66] study four cases of PCB assembly and conclude that FMS analysis is inappropriate for PCB assembly system analysis: FMS analysis chooses among multiple paths through manufacturing systems, and the best path depends on the state of the system that is changing as the workload changes. In PCB assembly, however, the production cycle is fixed (e.g., receiving inspection, panel preparation, screen paste, paste volume inspection, component placement, solder joint reflow, visual inspection, and X-ray inspection) and that results a single, sequential, and somewhat deterministic manufacturing process. Hence, a linear cost model may be adequate to evaluate and identify the cost elements of a PCB manufacturing system.

Zhou and Leu [119] list three features of PCB assembly distinct from conventional systems for automated assembly of mechanical parts:

1. each PCB requires numerous insertions, and the activities for each board are highly repetitive;
2. there is no strict sequence which has to be followed, and the components can be inserted (almost) in any order (i.e., they have no or just a few precedence constraints); and
3. only a single manipulator can operate at the time on the board even if the machine has multiple manipulators (e.g., see [46]).

Jain *et al.* [52] discuss setup problems in FMSs and PCB assembly. They translate the tool setup problem of FMSs into PCB assembly system and show that the FMS formulation is a relaxed version of PCB component setup problem: In PCB assembly the tooling is more constrained due to the restrictions on the width of the component reels.

1.2 Production Planning in FMSs

Despite the differences mentioned earlier, we will, for the remainder of this section, regard PCB assembly as an FMS. In Section 1.2.1, we begin with a

discussion of the common problems apparent in the theoretic approaches to construct a practical production planning system. After that in Section 1.2.2, we suggest a methodology for overcoming these problems and describe a general framework for modeling production planning problems.

1.2.1 Problems in Constructing a Practical Production Planning System

According to Ammons *et al.* [9] the control of an FMS requires a complex interaction of two components:

1. computers to perform automated control and routing activities, and
2. humans to supervise the automation, to monitor system flows and output, to intervene in the unexpected operation of the system, and to compensate the effect of unanticipated events.

Especially in dynamic production environments (i.e., in FMSs which are subject to limited resources, random machine failures or multiple optimization criteria) the problem of controlling and scheduling the production process is best tackled by a synergy of the computer's scheduling algorithms and the human's effective internal heuristics. In this "interactive scheduling" the production planner remains in control and is able to affect the scheduling process by using his experience and intuition via computer support. In other words, the production planning system should act as a decision support for the production planner.

However, literature references to practical systems where this interaction has been realized are rare, and the models—even if based on reality—tend to be oversimplified. According to Saygin *et al.* [96] the existing software tools are typically (1) too slow and cannot react to changing dynamic shop floor conditions, (2) based on simplistic formulations of the reality that ignore important constraints, (3) based on a single objective function or simplistic trade-offs, and (4) difficult to install and integrate into preexisting commercial shop floor systems. In general, the gap between theory and practice can usually be attributed to the following factors:

- Researchers fail to address the right problems.
- The given solutions are too complex to use.
- Findings are presented in terms that are foreign to the practitioners.
- Researchers focus on certain problems and omit other, often more important issues.

- The realities of the shop floor are ignored.

As Johnsson notes [54], these observations are valid in electronics assembly, where problems are usually tackled by first modeling an existing problem, then finding a solution method to the problem, and after that validating both the solution method and the model by solving some randomly generated artificial test cases. However, this approach does not shed much light on the *practicality* of the method.

In addition to interactivity, real-world scheduling problems usually differ (and often quite radically) from the mathematical models presented in literature. Pinedo lists twelve differences [91]:

1. Theoretical models assume that the scheduling problem is static, whereas in the real world new jobs to be scheduled can emerge at any time and the schedule is constructed without a perfect knowledge of the near future. The *production environment is dynamic* by nature (e.g., jobs may arrive unexpectedly, urgent prototype series may cut in the predefined sequence, machines may break down or have a temporary reduction in the production rate, or the required components may not be available at the present time).
2. Resequencing problem is rarely addressed in literature even though it is present in most of the actual problems. Production planning is based on a *rolling horizon*, which leads to *rescheduling* or *reactive scheduling*, where the schedule is constantly updated and revised to meet events occurring randomly.
3. In the real world, *production environments are more complicated than the models presented in literature*, which often disregard machine, job and time dependent processing restrictions and constraints.
4. The mathematical models assume that the weights or priorities of the jobs are fixed and do not change over time; in practice, the *weight of a job often fluctuates over time* (e.g., a low-priority job may become suddenly a high-priority job).
5. *Preferences* are usually not taken into account in mathematical models. In reality, even if a job can be scheduled on a given machine, there may be a preference to schedule it on another one.
6. Most theoretical models overlook the *machine availability constraints* and assume that machines are available at all times, whereas the real-world production plants have deterministic and random processes which

prevent machines from operating (e.g., shift patterns, preventive maintenance, breakdowns and repairs).

7. Most penalty functions considered in the literature are piecewise linear (e.g., the tardiness of a job or the unit penalty), whereas, in practice, there usually exists a committed *duedate*.
8. Theoretical research tends to focus on models with a single objective. In the real world, there are usually *a number of objectives*, whose weights may vary over time and may even depend on the subjective preferences of the production planner in charge.
9. In practice, whenever the workload appears to be excessive and the due-dates appear to be too tight, the problem can be tackled by assigning *extra shifts* and *scheduling overtime*.
10. The stochastic models studied in the literature usually assume special *processing time distributions* (e.g., exponential distribution). In automated assembly, the processing time is fixed with a very high probability, and with a very low probability there is an additional random time that is exponentially distributed with a very large mean (i.e., if a robot performs a task, the processing time is fixed, and if, by accident, something goes wrong, the processing time immediately becomes significantly larger).
11. *Successive processing times* on the same machine tend to be highly positively correlated in practice, whereas theoretical models usually assume that all processing times are independently drawn from given distributions.
12. In practice, the *processing time distribution* may be subject to change due to learning or deterioration.

In spite of the differences between the real-world and the mathematical models, Pinedo notes that the general consensus is that the theoretical research done in the past has not been in vain, but it has provided valuable insights into many scheduling problems. These insights have proven to be useful in the development of the algorithmic framework for a large number of real-world production planning systems.

1.2.2 Structure of a Production Planning System

The methodology for solving the production planning problems can be divided into four stages:

1. Familiarization with the problem environment.
2. Modeling the problem.
3. Designing and implementing algorithms to solve the modeled problem.
4. Integrating the algorithm to an existing system or including it in a new system.

The far-reaching decisions made in the initial stages influence the overall usability of the system. For example, if the model fails to represent the important aspects of the real-world problem, no algorithm (no matter how cleverly designed and effectively implemented it is) can give results which would satisfy the production planner. In our experience, this modeling process cannot be overlooked nor its importance underestimated: A poor algorithm solving an accurately modeled problem gives better real-world results than an accurate algorithm solving poorly modeled problem.

After the initial familiarization stage, the construction of a production planning system begins with building a model which represents the production environment. At the same time one must bear in mind that this model is always an idealization of the actual problem: A coarse model may be easier to understand but it may lack some important aspects, whereas a detailed model may be a more accurate representation but much harder to understand. Because of this duality there are two approaches for using the model: If there is uncertainty about the accuracy of the model, we may want to grant the final decision to a human user, and in this case the model is used to point out the important aspects of the actual problem and possibly for suggesting some solutions. An alternative approach is to solve the problem by using an algorithm which utilizes an objective function based on the model for evaluating the solutions.

Figure 1 illustrates the role of the model in this scheme. A system based on visualization allows the production planner to interact and analyze the schedule, whereas an algorithm driven system solves the given problem efficiently and independent from the user. Although both approaches have their benefits, extremes should be avoided when designing a production planning system: An algorithm is capable of solving a combinatorial problem inexhaustibly, whereas human tends to try only few possible solutions before choosing one. Instead, human usually has some “outside” knowledge about the reality concerning the problem, whereas the algorithm “sees” nothing but the model. Therefore, the usability of a production planning system, in essence, depends on the balance between these two points of view: The computer should provide the user with sufficient support for making the

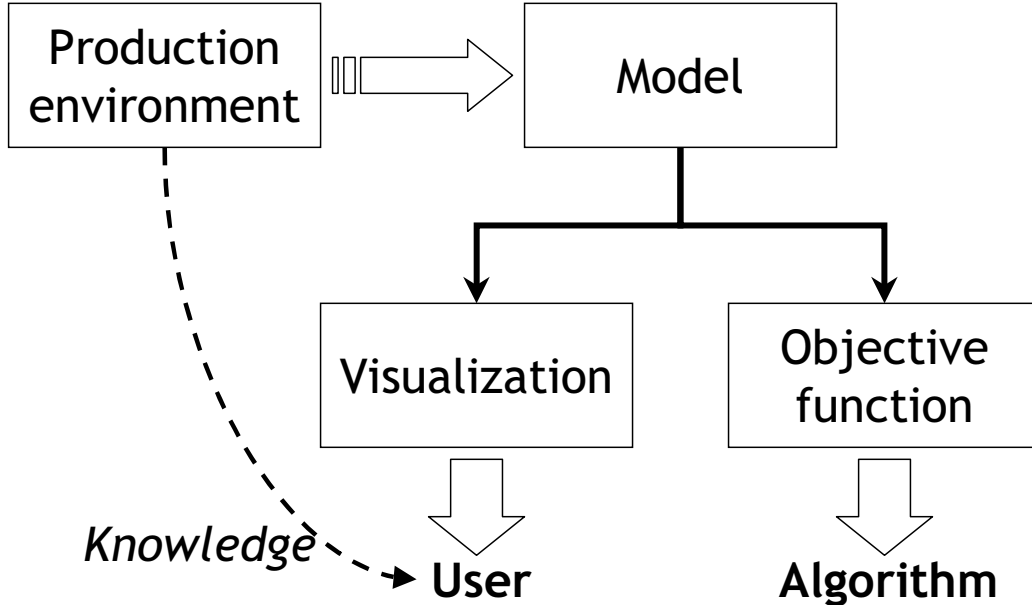


Figure 1: A model of the production environment can be used as a basis for visualization or for calculating an objective function.

actual decision (e.g., generate a number of good schedules from which the user chooses—and possibly refines—one for the production). Saygin *et al.* conclude in [96] that the human scheduler should remain in control and be able to affect the scheduling process, because the model is an abstraction of the reality and, therefore, cannot capture all the characteristics of a problem, and because it is hard to transcribe the entire knowledge of the human scheduler in a computable form. Ammons *et al.* express similar view in [9]: “an ‘optimal’ real-time scheduling system is one that effectively combines computer scheduling algorithms and artificial intelligence methodologies within the context of the versatile capabilities of the human supervisor”. Also Martin-Vega [86] lists the integration of human and technical resources to enhance workforce performance and satisfaction as one of the six grand challenges for future research.

Figure 2 gives a more detailed view of the structure of a general production planning system. There are two ways, which correspond to the division shown in Figure 1, to interact with the system: either directly by altering the production plan or indirectly by controlling the algorithm with the objective function and parameter settings. In the former case, the user makes alterations in the graphical representation of the production plan; the system

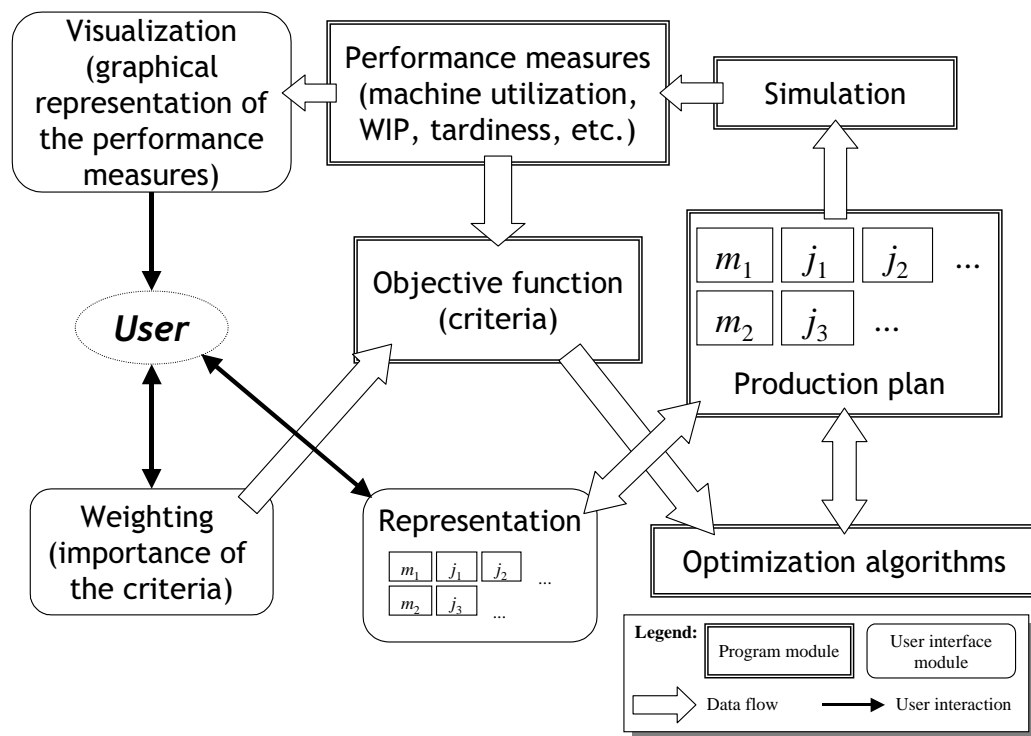


Figure 2: The interaction between the components of a general production planning system

updates the production plan accordingly or informs the user if the suggested change violates some hard constraint of the model. In the latter case, the user adjusts the objective function by setting weights for different criteria. The objective function is then used by an optimization algorithm, which generates a new production plan. After that, the new plan is simulated in order to discern predefined characteristics (e.g., lateness, earliness, workload, line balance, buffer sizes; see [35, 71, 51, 97]), which are used in the next iteration of the objective function and can be visualized to the user.

Essentially, the modules represented in Figure 2 are self-contained and connected by well-defined interfaces. For example, the optimization algorithm alters the production plan, which gives feedback whether the alteration violates any of the hard constraints, and receives an evaluation of the new plan from the objective function. Apart from these definements, the algorithm can be designed and implemented independently of the rest of the system.

Smith and Peters present in [101] a more general framework for production planning and control systems. They observe that the development of

FMS control systems is still implementation-specific and no tools exist to automate this development process. As a result, changes to the systems are often difficult to make, which has led to many instances where FMS installations have failed to live up to expectations due to the inflexibility of the underlying control software. As a solution, Smith and Peters present an FMS control system concept, which comprises three components: a resource model instance (which combines the system configuration and the production requirements), a decision maker module, and an execution module. The *resource model instance* provides structural information required for constructing the other two modules, as well as the operational information required to run the system. The *decision-making module* decomposes the production requirements into specific instructions for the *execution module*, which interacts with the physical equipment and personnel on the shop floor to implement the given tasks. Within this general framework, the model of Figure 2 corresponds to the decision-making module and clarifies the interaction between the system and the user.

2 Printed Circuit Board Assembly

A common characteristic in the printed circuit board (PCB) assembly industry is that customers demand more functions and flexibility each year in the products they buy. In addition, consumers expect reliable and cheap products, and thus a common goal in the PCB assembly industry is to put more functions into a board with the same size and cost [108].

PCB assembly requires complete agility and reliability, which are only achievable with the use of robotics [49]. Manual assembly methods may provide the needed flexibility, but they cannot provide the reliability and speed of robotic automation. When properly tooled, robotic assembly allows quick change from one product to another, handling a higher mix of products with reliability rates well in excess of non-robotic systems. PCB assembly is characterized by designs that range from simple and low-value board assemblies to very complex and high-value board assemblies. Production volumes for different products vary in a very wide range—from millions to less than ten. One assembly system may encounter the assembly of PCBs with frequent design changes in small-batch production, whereas another system may assemble PCBs with a design that is fixed for six months or longer.

A recent development in PCB assembly is the growing role *contract manufacturing*. Many original equipment manufacturers (OEMs) have abandoned the assembly line in favor of outsourcing the manufacturing functions to contract manufacturers (CMs). CMs differ from OEMs that they build a variety

of products for many different customers, whereas OEMs build only their own products. Despite the wider product variety and more dynamic product demand, CMs are expected to operate more efficiently than OEMs. This trend further emphasizes the importance of developing better production methods and systems [84].

This section is organized as follows: In Section 2.1 we present the technical fundamentals and concepts of PCB assembly (for further details, see [43, 70, 102, 110, 118]). Section 2.2 describes most common machine types used in PCB assembly. Finally, in Section 2.3 we concentrate on different plant layouts.

2.1 Fundamentals

A *printed circuit board* (PCB)—or *printed wiring board* (PWB)—is a substrate of a glass fabric impregnated with a resin (an organic polymer which, when mixed with a curing agent, crosslink to form a thermosetting plastics; usually epoxy). A PCB consists of one or more layers of metal conductors and insulating material that allow for electronic components to be electronically interconnected and mechanically supported. A PCB of smaller dimension is commonly referred to as a *card*. A *panel* is an array of (usually identical) separate circuits fabricated on a single substrate.

The simplest form of PCB is the single-layer, single-sided board, which contains metalized conductor on one side of the board only. Greater levels of complexity and component density can be achieved by making double-sided and multi-layered boards: In double-sided assembly the PCB is assembled with components on both sides of the substrate, and multilayering permits tracks to cross over one another, giving the designer more freedom in component layout.

Electronic components are either inserted through holes (e.g., griplet, axial and radial components) in the copper tracks and soldered in position, or are placed directly on to the surface of the board and soldered. These two distinctly different methods of manufacturing PCBs have given rise to different branches of manufacturing technology. The conventional method is known as *through hole plated assembly*, which is still popular for many applications, especially for low volume and manual assembly. Modern *surface mount technology* (SMT) utilizes smaller “flat” components which are well suited to automated assembly process. They are common in small consumer products (e.g., cellular phones), whereas in the larger ones (e.g., televisions and computer monitors), where the competitive product price is a key factor, through-hole components are still widely used.

Components have also other properties that affect the assembly process.

The size of the component defines the recognition camera type, the feeder size and the nozzle (or tool) which must be used when the component is handled. Furthermore, component polarity, orientation in the input tape and different handling speeds (e.g., pickup, recognition, placement and turret) affect the operation of the insertion machines.

2.2 Insertion Machines

Almost all machine types on the market operate in a similar fashion: The substrate is either placed (by the operator) or automatically transported to the staging area. After that, the components are “picked” from assigned pickup bin locations by vacuum and usually realigned either mechanically or optically before they are placed into the appropriate locations on the board. McGinnis *et al.* [87] recognize five fundamental operations common to all machines:

1. positioning for retrieval of a component from feeder,
2. retrieving the component from feeder,
3. transporting the component from the feeder to the circuit board,
4. positioning for placing the component on the circuit board, and
5. placing the component on the circuit board.

Some machine types are flexible in the sense that they can handle a wide range of different substrate sizes as well as a wide range of different component types, whereas others are restricted to a condensed set of components, which they can operate at a much higher speed.

A *feeder* supplies the placement head with components in the proper orientation. Notwithstanding the machine type, the feeder capacity of the machine is usually expressed in the number of 8 mm tape feeders, which is used in almost all currently available machine types. Other feeder types include a stick (or tube) of components, a vibratory slope feeder, and a tray feeder. A *component setup* comprises the required operations to replace one tape feeder to another, and a *machine setup* comprises the required component setups, width adjustments, tooling plate changeovers and printing program updates to change manufacturing from one PCB type to another.

The three most common machine types are:

- *Insertion machines* which have either a fixed head and a moving table or a moving head and a fixed table (Figure 3). The printing head is

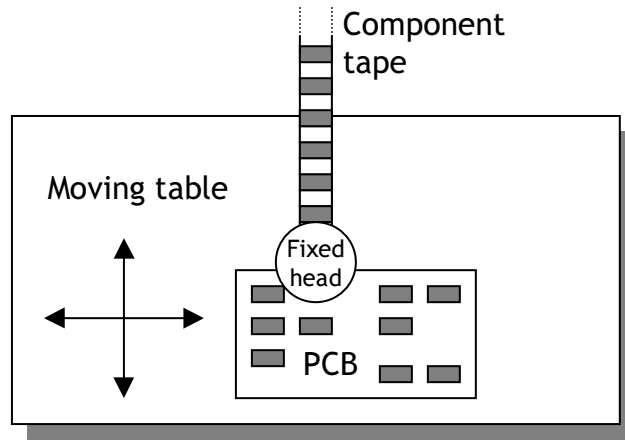


Figure 3: Insertion machine

connected to only one feeder, and a separate machine (which is often called a *sequencer*) produces an appropriate feeder tape, if different component types are needed.

- *Pick-and-place machines* which have a moving printing head, a fixed table and fixed feeders (Figure 4). The head travels to pick a component from the feeder slot, moves it to the component insertion location, prints the component, and finally moves back to the next feeder location.
- *Rotary turret machines* which have moving insertion heads, a moving table and moving feeders (Figure 5).

Currently, the rotary turret machine is the fastest machine type in common use; it can insert one component in less than 0.1 seconds and with a 0.01 mm accuracy. Such a high-speed surface mount component handler and placer is also known as a *chip shooter*.

There are several variations of the basic machine types: Some machines have multiple insertion heads, duplicated feeders and duplicated tables [1, 46, 112]. Moreover, the insertion head may include several different nozzles to make it possible to operate with different component types; however, a nozzle change may require some time and slow down the overall operation speed. Finally, in some machines submachine units are added to increase the overall capacity.

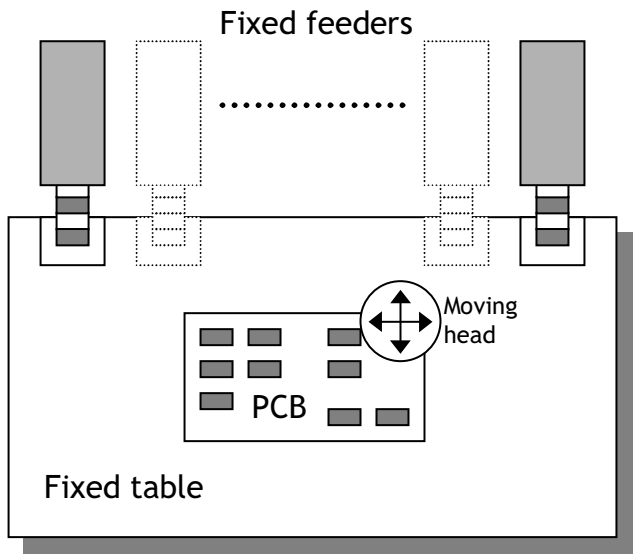


Figure 4: Pick-and-place machine

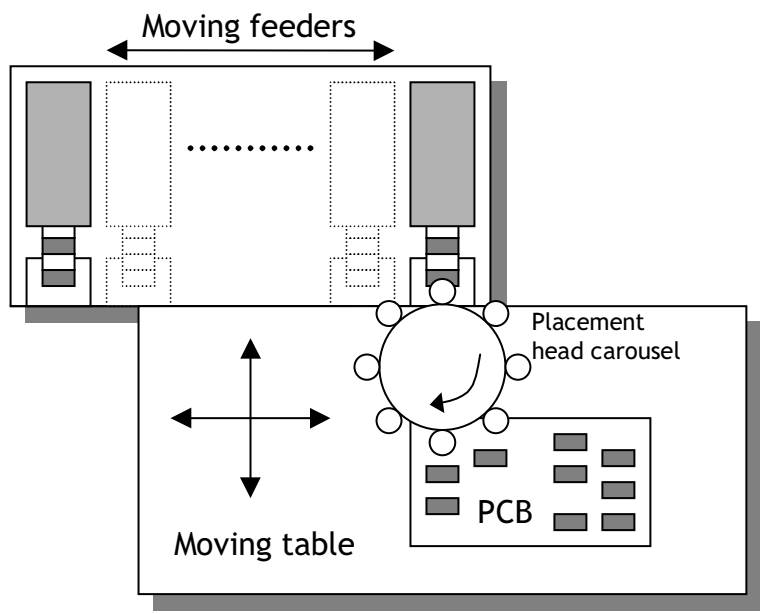


Figure 5: Rotary turret machine

2.3 Plant Layout

Wittrock presents in [115] the *flexible flow line* (FFL) environment which comprises several machine banks or production phases. In this model, the machines in a single machine bank are identical with each other, and a product can be processed in any machine belonging to the machine bank, or it can skip the phase altogether. Each product passes the phases in a predefined order, and the transfer between stages is accomplished with the help of magazines or some other form of transport. The setup time between different products, which is assumed to be negligible, is ignored. Hence, the processing time is a function of the processed product and the phase.

Johnsson *et al.* introduce in [57] a *generalized flexible flow line* (GFFL) environment, which is a generalization of the FFL. The GFFL environment also comprises successive machine banks, but the type of the machines can vary even inside a particular machine bank (unlike in FFL). The machine type defines the speed of the machine, and thus the processing time in GFFL is a function of the product and the machine type. Moreover, setup times are also taken into account (unlike in FFL).

A typical electronic assembly line layout resembles GFFL, because the production is usually organized in successive workphases. Figure 6 gives an example of an existing production plant, where the phases are determined by the component type inserted in the corresponding phase [61] (similar plant layouts are described in [64, 76, 51]). In this case, both through-hole and surface mounted components are assembled, and the phases are organized such that gripsets are printed first on the board. Axial components are inserted in the next phase, followed by radial components. Finally, the surface mounted components are inserted on the board.

The automated line is usually followed by the insertion of odd-shaped component, which is still done, at least partially, manually, because the automated insertion of the most complicated components is hard (or too expensive) to accomplish [55]. In some cases companies have even abandoned automatic insertion and returned back to manual work because of the increased flexibility [92]. In *manual insertion*, the board is set up on a tray, and the operator, following the assembly instructions, obtains components from labeled bins and inserts them manually onto the board. In semi-automatic insertion, the board is set up on a semi-automated component insertion machine, which consecutively opens and moves to near the operator the bin containing the required component, and, at the same time, shines a point of light on the locations of the board where the component is to be inserted [95, 63].

In addition to component insertion, the line may include other phases.

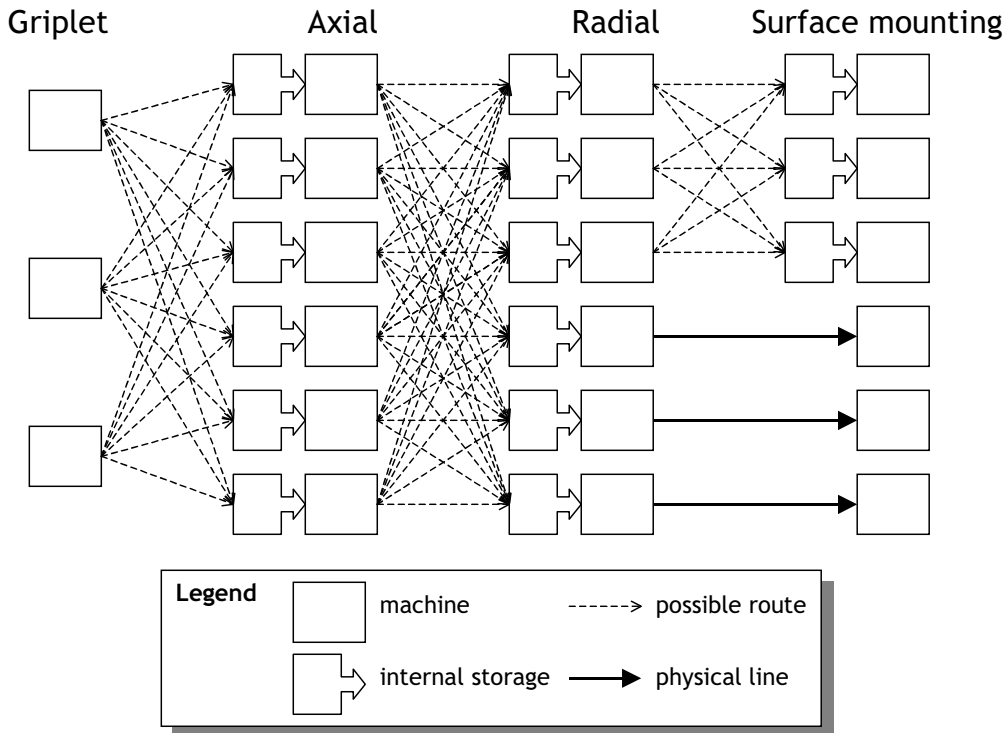


Figure 6: A typical PCB assembly facility including four production phases. The internal storages buffer the products before they are transported to the machine. The dashed lines indicate possible routes between machines. A physical line is usually a conveyor belt which couples two or more machines together. phases

Some machines require that the component tape is preprocessed on another machine [56]. The surface mount onsertion is usually preceded by a glue dispenser and succeeded by an oven which hardens the adhesive and fixates the components, see Figure 7 [99] (similar lines are described also in [29, 44, 117]). Typically, the production also includes inspection and testing phases.

The PCBs are transferred from one phase to another either manually (e.g., in magazines which can hold 10–100 PCBs) or with a conveyor belt. The conveyor-linked machines are usually referred as being *coupled*, and a system with a batch transfer as *uncoupled*. Because the change of board type causes a setup, boards of the same type are collected in a batch, in which they are operated successively. The batch size can vary considerably: A mass-product PCB may remain in active production for several months,

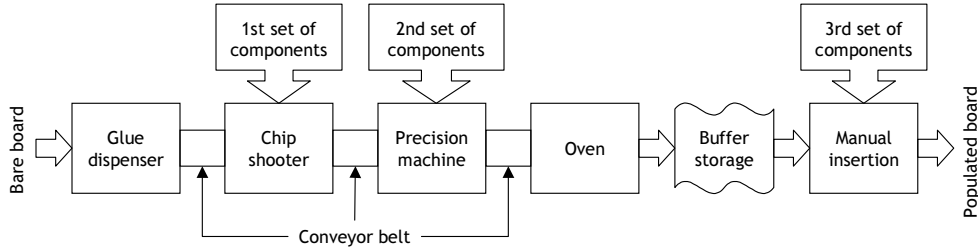


Figure 7: Workphases in an SMT assembly line

whereas a prototype batch usually comprises only few PCBs.

3 Literature Review

Johnsson classifies in [54] the literature on PCB assembly according to the number of different PCB types and machines present in the problem. Accordingly, the four main problem classes are:

One PCB type and one machine (1–1) class comprises *single machine optimization* problems, where the goal is to minimize the printing time of the machine. The class can be further divided into four subclasses [40, 26, 19]:

- feeder arrangement problems (1–1a),
- placement sequencing (or insertion order) problems (1–1b),
- nozzle assignment problems (1–1c), and
- component retrieval problems (1–1d).

Multiple PCB types and one machine (M–1) class comprises *setup strategies for single machine*, where the goal is to minimize the setup time of the machine. The setup strategies can be classified followingly [74]:

- minimum setup strategy (M–1a),
- group setup strategy (M–1b), and
- partial setup strategy (M–1c).

One PCB type and multiple machines (1–M) class concentrates on *component allocation to similar machines*, where the usual objective is balancing the workload of the machines in the same line (usually by eliminating bottlenecks) [87, 37].

Multiple PCB types and multiple machines (M–M) class or *scheduling problems* usually concentrate on

- allocating jobs to lines (M–Ma) which includes routing, lot sizing and workload balancing between lines, and
- line sequencing (M–Mb).

Other classification schemes have been presented by many authors. Crama *et al.* [26] associate a production plan with the following issues:

1. partition the set of board types into *families* which are to be assigned to different lines of placement machines;
2. for each board type, determine a *partition of the set of component locations* on this board (i.e., decide which locations are going to be served by which machine);
3. for each machine, determine a *feeder rack assignment* (i.e., an assignment of feeders to position in the feeder rack);
4. for each pair consisting of a machine and a board type, solve a *component placement sequence* (i.e., an order in which components are placed at the locations on the board by the machine); and
5. for each pair consisting of a machine and a board type, make a *component retrieval plan* (i.e., decide for each component from which feeder it is to be retrieved).

Cases 3, 4 and 5 correspond to problem classes (1–1a), (1–1b) and (1–1d), case 1 is a job allocation problem (M–Ma), and case 2 is a component allocation problem (1–M).

Foulds and Hamacher [40] list the problems associated to PCB assembly:

1. the allocation of component types to machines,
2. the allocation of component types to feeder location at each machine, and
3. the pick and place sequence.

Here, case 1 corresponds to class (1–M), and cases 2 and 3 classes (1–1a) and (1–1b).

Feldman and Feuerstein present in [37] a hierarchical four-level optimization strategy to reduce changeover, setup and processing times based on the hierarchy of the assembly task:

1. *job allocation* and *job clustering* for each assembly line,
2. *line balancing*,
3. optimization of the *feeder configuration*, and
4. optimization of the *insertion sequence*.

In this scheme, cases 3 and 4 correspond to problem classes (1–1a) and (1–1b) respectively, case 2 is equal to problem class (1–M), and case 1 is equal to problem class (M–M).

McGinnis *et al.* [87] recognize three levels of decisions:

1. selection of machine groups and part families and assignment of families to groups,
2. allocation of components to machines when a group has more than one machine, and
3. arrangement of component feeders and sequencing of placement operations for each machine and PCB.

Here the levels correspond to problem classes (M–M), (1–M) and (1–1ab), respectively.

In this work we adopt the classification of setup management strategies (M–1) proposed by Leon and Peters [74]—with the exception that in our scheme the unique setup strategy corresponds to single machine optimization (1–1):

- *Unique setups* consider one board at a time and specify the component–feeder assignment and the placement sequence such that the placement time is minimized. This is a common strategy when dealing with a single product and a single machine in a high-volume production environment.
- *Group setups* form families of similar parts such that setups are incurred only between families.
- *Minimum setup* strategy attempts to sequence boards and determine component–feeder assignments to minimize the total changeover time.
- *Partial setups* are characterized by partial removal of components from the machine when changing over from one product type to the next.

Another classification of setup strategies is given by McGinnis *et al.* [87] and Ammons *et al.* [8]:

1. Single setup strategy: Configure a group of machines to produce a family of board types using a single setup which is sufficient for the entire family. There are two possible options for achieving this:
 - (a) *Unique setup strategy*: A single setup strategy applies to a family which contains only one product type (i.e., mass production).
 - (b) *Family setup strategy*: Several product types are in the same family. There is usually a tradeoff which must be explicitly examined; the family setup eliminates the need for machine setups between board types at the price of potentially increasing the assembly time for each individual board type in the family.
2. Multisetup strategy: Because a limited component staging capacity on the placement machines prohibits the use of the single setup strategy, some additional setups must be performed within a family. In this scenario, the objectives generally include the minimization of the production time consumed by setups along with the minimization of WIP levels. There are two possible multisetup strategies:
 - (a) *Decompose and sequence* (DAS): Break the family into smaller sets of PCB types (possibly of size one), then sequence the subsets so as to minimize the incremental setups between the subsets. This strategy requires some additional WIP at the risk of potentially having to remove a component type to process one subset of PCBs and then having to restage it for a later subset.
 - (b) *Partition and repeat* (PAR): Partition the required components into subset such that the machine group has enough staging capacity for each subset. Populate PCBs with components by configuring the machines with each subset of component types in turn. This approach requires the accumulation of a batch of partially populated PCBs but it includes relatively few setups.

In this scheme, the unique, family and DAS setup strategies correspond classes (1-1), (M-1b) and (M-1a), respectively. The PAR setup strategy (see also [82, 31]) is a special case of (M-1b), where the boards in the group are loaded several times to the same machine but with different component setup. Maimon and Shtub [82] classify four problem types according to how many times (one or many) each component type and job (PCB) is loaded: In

type 1 each PCB and each component is loaded only once, type 2 constraints each component to be loaded only once, type 3 constraints each PCB to be loaded only once, and in type 4 there are no limitations.

In the remainder of this section, we try to give a condensed presentation of the research done so far. We review the literature according to Johnsson’s classification [54]: single machine optimization in Section 3.1, setup strategies for a single machine in Section 3.2, component allocation to similar machines in Section 3.3, and miscellaneous scheduling problems in Section 3.4.

3.1 Single Machine Optimization (1–1)

Although we can recognize four distinct subproblems (feeder arrangement, placement sequencing, nozzle assignment and component retrieval) in single machine optimization, they are strongly intertwined and, therefore, usually not solved altogether independently. For example, an optimal placement sequence does not guarantee optimal printing time if the feeder assignment is neglected (and vice versa). Another aspect of the problem is the wide variety of different machine environments considered in the literature. Most of the work have been based on (variants of) pick-and-place machines, but lately the trend has shifted towards rotary turret machines as they have become more popular in industry. Also, manual and semi-automated operations are considered by some authors.

Ball and Magazine [12] study the problem of determining the component insertion sequence and feeder arrangement for a *pick-and-place machine*. They regard the former problem as a special *traveling salesman problem* (TSP). The authors model TSP as a stacker crane or rural postman problem, and present a heuristic algorithm for solving it. The metrics used in the formulation affects the solution: If the placement head moves only in one orthogonal direction at a time, the Manhattan metric is used for measuring the length of the movements and one can find an optimal solution in polynomial time. If the head can move in the x- and y-directions simultaneously, Chebyshev metric can be applied for the distance calculations but the solutions are then suboptimal. However, the authors prove that in the latter case the maximum error of their solution is bounded. The feeder arrangement problem can be structured as the classical assignment problem when considered separately of the insertion sequencing. Although there are efficient algorithms for solving the problem, in this case the cost of assigning a component to a feeder location cannot be expressed as a simple cost coefficient, and, therefore, they cannot obtain optimal solutions.

Leipälä and Nevalainen [72] recognize the insertion sequence and feeder assignment problems in the single machine problem, and solve them sepa-

rately. The optimal insertion sequence for a fixed feeder setup can be obtained by considering the problem as a three-dimensional asymmetric traveling salesman problem. The optimal assignment of the components to the feeders for a fixed insertion sequence can be formulated as a quadratic assignment problem. The overall problem can then be solved heuristically, which brings suboptimal but, in practice, satisfactory solutions.

Chang and Young [23] introduce a hypothetical machine design for simultaneous mounting. The machine comprises one or more robots or a robotized pick-and-place head, a component supply device (movable or fixed), and an assembly holding device (movable or fixed). To optimize the printing, the algorithm should determine stop positions for a PCB, the components printed at each stop position and the placement of components in the delivery head. The authors present a mixed-integer programming model of the problem and a branch-and-bound based greedy heuristic algorithm for solving it.

Leu and Ji [75] recognize three basic machine types—insertion, pick-and-place, and rotary turret—and discuss methods for solving the component insertion sequence in each of these. In an insertion machine, where a separate machine (a sequencer) produces an input tape for components, the problem is considered to be a TSP. In a pick-and-place machine with a moving head, a fixed table and fixed feeders, the problem can be model as a rural postman, linear assignment or quadratic assignment problem. In a rotary turret machine, the authors model the problem as a TSP and use a *genetic algorithm* (GA) for solving it.

Supinski *et al.* [105] discuss the planning of the printing order and the generation of the robot code on a single machine. These tasks comprise three stages: CAD data importation, PCB assembly planning, and code generation. In the second stage, the authors consider three separate workphases: In adhesive deposition the authors model the problem as a TSP and use the arc-opt or 3-opt method for solving it. In solder deposition, a two-step procedure first clusters the solder pads and then applies a TSP formulation to each individual cluster. In component insertion, the objective is to minimize the number of components to be manually inserted on the board.

Zhou and Leu [119] describe a *Petri net model* for solving feeder assignment, printing order and tool assignment on a single machine. The authors present an ordinary (or non-timed) Petri net model for analyzing the system behavior (e.g., deadlock-freeness, buffer boundedness, reversibility and conflict-freeness), and a temporal Petri net model for evaluating the system performance (e.g., productivity and machine utilization).

Foulds and Hamacher [40] present methods for identifying optimal bin locations, which surround the worktable on all four sides, and for determining a component insertion sequence for the board. The authors model the former

problem as a single-facility location problem, and the latter problem as a TSP.

Zijm and van Harten [120] discuss a *hierarchical decomposition* for a modular component placement system, which comprises several identical successive insertion machines. To minimize the cycle time of a batch of PCBs, the following problems must be solved: Firstly, determine the number of feeders for each component type, and if more than one feeder contains the same component, specify the board locations which are delivered from each feeder (solved with a greedy heuristic). Secondly, allocate feeders to the insertion machines (solved with a local search heuristic). Thirdly, assign the feeders to the feeder positions of a machine (solved with the standard Hungarian method). Finally, specify a placement sequence for a given feeder assignment (solved with the 2-opt heuristic). If a solution on a higher level leads to infeasible or inferior solutions on a lower level, a feedback loop allows the program to change the higher-level solution, and thus avoid the problem.

Wang [112] compares three layout design methods for a dynamic pick-and-place machine with a mobile worktable and magazine, and a single placement head. The layouts include one-magazine-and-one-board (1M1B), one-magazine-and-two-boards (1M2B), and two-magazines-and-one-board (2M1B). Wang *et al.* [113] solve the placement sequence and feeder arrangement for a machine where the board and feeder carriages move on x-direction and the head on y-direction. First, the insertions are sorted according to their x-coordinate on the board, and the sorted list determines the placement sequence. By using this sequence, a matrix of the component exchange frequency (which describes how often a component pair is subsequent in the sequence) is built, and the problem is then modeled as a TSP and solved accordingly.

Sanchez and Priest [95] present a method based on artificial intelligence and an expert system with sequencing decision rules for *semi-automatic assembly*. The approach divides the task into four phases: Firstly, design criteria which simplify operators insertion routines are chosen. Secondly, the insertion rules are prioritized (e.g., insert small components before large ones, insert all component of the same type successively, insert components in a sequence which minimizes the movements of the machine bed). Thirdly, the components are grouped into insertion sequence classes. Lastly, the board movements are modeled as a TSP and solved with the nearest neighbor heuristic. Khoo and Ng [63] discuss a similar problem formulation but apply a genetic algorithm for solving the placement sequence. They compare the results to Sanchez and Priest and observe an improvement in the total distance travelled by the machine bed.

Johnsson *et al.* [55] discuss designing an efficient and easy-to-learn se-

quencing for *manual installation of components*. Favorable sequences can be characterized by several nonstrict rules (e.g., use both hands equally in the insertion, proceed from top to bottom and from the edges to the center, and take the closest next part). The authors present three different sequencing methods: line sweeping, traveling salesman clustering, and weighted 3-matching.

Sadiq *et al.* [94] present a method for arranging feeders in a SMT machine in a low-volume, high-mix environment. First, the slots are assigned by using a heuristic rule, and, after that, the slot rearrangement process tries to locate the components so that they are adjacent in the insertion sequence. The goal is to assign the components inside a contiguous group to make the components required by a PCB adjacent, and thus save feeder carriage movements.

J. Ahmadi and coworkers have studied *dual delivery pick-and-place machine* in several papers: J. Ahmadi *et al.* present in [2] an emulator for studying the effect of different printing process parameters. The overall system is modelled by subproblems: Component allocation problem tries to maximize the number of board completions attainable by a single allocation of components. Partitioning problem minimizes the idle time caused by imbalance in the use of the manipulators and nozzle changes. The authors give a detailed discussion of these problems in [3], where they present mathematical formulations and solve them with a mixed integer programming package. R. Ahmadi and Kouvelis [4] concentrate on the same machine and present methods for solving single product staging problem (SPSP) and multiproduct staging problem (MPSP). SPSP involves the assignment of the required component feeders for the specific board type to the two carriers, while simultaneously assigning the required tools (for dispensing each component type) to the tool magazines by observing capacity constraints. The objective is to minimize the total time to assemble the PCB. The authors give an integer programming formulation as well as present a Lagrangian relaxation based branch-and-bound algorithm for SPSP. In MPSP, the objective is to minimize the total time to assemble a given set of multiple PCB types. Finally, J. Ahmadi *et al.* [1] study the feeder assignment problem by formulating it as a reel positioning problem (RPP), where the goal is to minimize the direction changes and the sum of movements of the feeder carrier. In addition to the mathematical formulation, the authors give a heuristic algorithm based on solving the shortest path in a layered network. A two-manipulator machine is also considered by Hernandez and Leon [46], who discuss interference avoidance of the manipulators working on the same area.

Crama *et al.* [27] and van Laarhoven and Zijm [111] present hierarchical decomposition schemes for a line of *3-headed pick-and-place machines*. In

both papers the goal is to minimize the processing time of the bottleneck machine. Crama *et al.* identify six subproblems: (1) determining how many components each machine must mount and with what equipment, (2) assigning feeder types to machines, (3) determining what components each head must mount, (4) clustering the locations into subsets of three, (5) determining the sequence of pick-and-place operations, and (6) assigning feeders. These subproblems are then solved by using simple heuristic algorithms. Correspondingly, van Laarhoven’s and Zijm’s problem hierarchy has five steps—determining equipment for the heads on each machine, assigning components to the machines, assigning feeders, clustering components for pick-and-place operations, and sequencing the cluster and component within each cluster—which are mainly solved with simulated annealing.

Optimizing feeder arrangement and insertion sequence of a *rotary turret machine* is discussed by several authors. Bard *et al.* [13] model the placement sequence of such a machine as a TSP and solve it with nearest neighbor heuristic. Moreover, they formulate the feeder assignment and component retrieval problems as a quadratic integer program and solve it with Lagrangian relaxation. Yeo *et al.* [117] present a rule-based system, which employs simple heuristics: Feeders are assigned according to a one-pitch incremental heuristic. The placement sequence is modeled as a TSP and solved with the nearest neighbor method. Crama *et al.* [26] use hierarchical decomposition, and solve the feeder assignment, the component placement sequence and the component retrieval plan with local search heuristics. The overall objective is to minimize the sum of makespans on the bottleneck machine on a single production line. Altinkemer *et al.* [7] present an integrated model and heuristic for assigning the feeders and sequencing the placement operations. First, the delivery problem is solved for each component type at every possible feeder location. A feasible solution can be used as the cost of assigning the component type to the feeder location in question. The assignment problem can then be solved by using these costs.

Crama *et al.* [25] discuss the *component retrieval problem* (CRP) in detail. In CRP, the placement sequence of components on the board, and the assignment of component types to (possibly multiple) feeder slots are given. The problem is then to decide from which feeder slot each component should be retrieved. Naturally, this problem emerges only if at least one component type is duplicated in the feeders. The authors describe a PERT/CPM network model of CRP, and present a polynomial algorithm for solving the problem.

Educational and research topics have been considered by Bodner and coworkers and Feldman and coworkers. In [19] and [18] Bodner *et al.* present a virtual machine model, in which three mechanisms—component feeder, board locator, and placement mechanism—interact. The authors describe

two approaches of studying process planning and equipment configuration: *prescriptive models* which emphasize the determination of a near-optimal solution to the given problem, and *descriptive models* which emphasize the performance evaluation of a specified set of solutions to the problem. In this view, a virtual prototype is a descriptive model that allows one to assess performance of a machine or system off-line. Feldman *et al.* [38, 37] describe a similar approach for modeling the whole production line. It allows to demonstrate the production starting from planning and ending to the diagnosis of simple assembly processes.

3.2 Setup Strategy for Single Machine (M-1)

Coble and Bohn [24] recognize two approaches to reduce setup times: (1) reduce the time to set up a feeder, and (2) reduce the number of feeders to be set up. The authors argue that most of the research have concentrated on the latter approach and ignored the former. Consequently, the authors present a two-part approach for reducing the time to set up a feeder: First, processes are re-engineered using SMED (single minute exchange of dies) concepts (which were originally developed for metal fabrication). After that, a factory information system with wireless computers and barcode scanners are used both to reduce the setup time and to increase the setup accuracy. There are two kinds of setup operations: offline setup, which can be done while the machine working, and online setup, which can be done only when the machine is shut down. The SMED approach begins with recognizing internal and external setup tasks and assigning the external tasks to be done offline. Next, as many internal operations as possible are converted to external ones to further reduce online setup. Finally, both internal and external operations are streamlined. The authors also discuss other procedures including hot swapping (fill one feeder carrier while the other is being used), a barcode system for feeders, a computer system for locating parts, adding more operator to the lines, and operator training.

In the remainder of this subsection, we concentrate on the second approach (i.e., reducing the number of feeders to be set up).

Minimum setup strategy (M-1a) Minimum setup strategy attempts to sequence the PCBs and determine feeder assignments to minimize the total component setup time. The idea is to perform only the feeder changes required to assemble the next PCB batch. Here we aim at avoiding additional feeder changes or reorganization, which would reduce the placement time. In general, similar product types are produced in sequence so that little changeover time incurs.

Lofgren and McGinnis [78] point out two key decisions: We must solve in what sequence the PCB batches are to be processed by the machine, and what component types should be staged on the machine for each PCB type. The authors present a sequencing algorithm (SEQ) based on labeling concept, in which the PCBs are first sequenced and after that a setup is determined for each PCB. The heuristic determines myopically the “best” component types to remove and to add whenever the existing setup is not sufficient.

Barnea and Sipper [14] consider a case of one machine and recognize two subproblems: sequence and mix. They present a mathematical model of the problem and use a heuristic approach based on the *keep tool needed soonest* (KTNS) policy introduced by Tang and Denardo in [106]. In each iteration, the algorithm generates a new partial job sequence by using a sequencing algorithm—which decides the next job to be added in the sequence—and mix algorithm—which updates the component mix with KTNS.

Jain *et al.* [52] present a four-stage method for optimizing the setup: Firstly, a greedy heuristic maximizing the component similarity of the jobs is used to determine an initial processing sequence. Secondly, the components are assigned to feeders according to KTNS policy. Thirdly, the jobs are rearranged in the sequence by applying 2-opt heuristic and KTNS. Finally, because the production is a continuous process, at the end of the sequence the frequently used components are preserved for the next production period (i.e., the approach gives heed to the rolling horizon framework).

Günther *et al.* [44] present a typical SMT production line and apply the minimum setup strategy approach, in which the PCBs are sequenced so that each subsequent PCB has the maximum number of components common with its predecessor. The authors discern three different subproblems—job sequencing, component setup and feeder assignment—and solve each of them with heuristic algorithms.

Narendran and coworkers have studied a heuristic approach which starts from an initial setup (a seed) and sequences the PCBs by looking for the similarities between the current setup and the PCBs remaining to be sequenced. In [93], Rajkumar and Narendran form the sequence by considering the overall component requirement of the PCB and the number of extra components required in the setup. In [68], Kumar and Narendran add a third constraint, slack time, and observe better result in comparison to dispatching rules (e.g., earliest due date, shortest processing time and least remaining slack) used normally for single machine scheduling with due dates.

Dillon *et al.* [33] discuss minimizing the setup time by sequencing PCBs on a surface mount technology production line. The authors present four variant of a greedy heuristic which aims at maximizing iteratively the component commonality whenever the PCB type changes. This is realized by using a

component communality matrix from which board pairs with a high number of common components can be identified.

Group setup strategy (M-1b) In the group setup strategy the feeder assignment is determined for a group or a family of similar PCBs. Any board in this group can be produced without changing the component setup, which is only required when switching from one group to another. Because the placement time for a specific board is, in general, larger than in unique setup strategy, some efficiency can be potentially lost. However, this is compensated by less frequent setup operations, which compensates the losses in machine speed especially in high-mix, low-volume production. There are variations of the group setup strategy, where a certain set of common or standard components are left on the machine, while the rest of the components (residual or custom) are added or removed as required for a particular board.

Carmon *et al.* [22] describe a group setup (GSU) method for a high-mix low-volume production environment. The products are divided into groups, each of which is produced in two stages: Set up common components and insert them to the PCBs of the whole group, and set up the residual components and insert them on each PCB separately. In [83], the same authors compare GSU to sequence dependent scheduling (SDS) on three performance measures—line throughput, average WIP level and implementation complexity—and conclude that in general SDS performs better on the last two areas. Practical analysis of these two methods appear also in [29, 69].

Maimon and Shtub [82] present a mixed-integer programming formulation and a heuristic method for grouping a set of PCBs to minimize the total setup time. A user-defined parameter indicates whether multiple loading of PCBs and components is allowed (cf. partition and repeat strategy in [87]). This approach is developed further in [31] by Daskin *et al.* Their goal is to minimize the total component and PCB loading costs subject to a capacity constraint. The authors present a mathematical formulation for the PCB-grouping problem, show that the problem is \mathcal{NP} -complete, and give a branch-and-bound based heuristic algorithm for solving it.

Shtub and Maimon [98] establish that grouping PCBs is an extension of the set-covering problem and present a general heuristic approach based on cluster analysis and similarity measures (e.g., Jaccard's similarity coefficient) which are traditionally found in the literature concerning group technology. Here the goal is to minimize the total production time, but since insertion times are assumed to be constant, the objective reduces to minimizing the total setup time of the groups.

Hashiba and Chang [45] study a single machine case when the objective

is to minimize the number of setups. They decompose the setup problem into three subproblems—grouping PCB types, sequencing the groups, and assigning components for jobs—and apply heuristic algorithms to each of them individually. Furthermore, the authors experiment with a simulated annealing method and observe that it gives better solutions than the heuristic decomposition approach.

Luzzatto and Perona [81] present a heuristic method for grouping PCBs to minimize the setup size. Although the authors consider a production line consisting of several workphases, their model enables the solution for each workphase to be obtained separately from the others.

Bhaskar and Narendran [17] apply graph theory for grouping PCBs. The PCBs are modeled as nodes and their similarities as weighted arcs between the nodes. After that, a maximum spanning tree is constructed for identifying the PCB groups.

Xu *et al.* [116] form PCB groups and divide the feeder slots into three “feeder bays”: fixed, semi-fixed, and configurable (cf. [8, 99]). The fixed feeder bay comprise the most frequently used components and it remains constant throughout the production, whereas the semi-fixed feeder bay is changed whenever the group changes and the configurable feeder bay whenever the board type changes.

Smed *et al.* [99] give an integer programming formulation of the *job grouping problem* (cf. [28]), and compare several heuristic algorithms based on greedy, clustering and repair-based local search methods. Johtela *et al.* [62] expand the problem to account multiple and possibly conflicting grouping criteria—such as different substrate widths, adhesive types, and production priorities—by modeling them as fuzzy sets.

Ohno *et al.* [89] group PCBs to minimize the sum of the assembly and setup times. The authors present a multi-type PCB assembly (MPCBA) optimization problem, which is divided into three subproblems: insertion sequence problem (ISP), reel position problem (RPP), and optimal assembly mode problem (OAMP). ISP is modeled as a TSP for fixed reel positions (i.e., feeder assignment), and solved with a 2-opt heuristic. RPP is viewed as an assignment problem, where the cost is the sum of weighted tour costs of the TSPs for the group, and it is solved with an evolution strategy. Evolution strategy is also applied to OAMP, which is modeled as a set partitioning problem with TSP type constraints.

In addition to placement machines, the group setup strategy has been proposed for *sequencer problems*: Fathi and Taheri [36] present a heuristic algorithm to group products for minimizing the setup of a sequencer for an axial placement machine. Sule [104] applies group technology and develops a heuristic method to minimize the changeover cost and to balance the

workload between sequencers.

Partial setup strategy (M–1c) Partial setup strategy specifies that only a subset of the feeders on a machine are changed when switching from one product to the next. Because the goal is to minimize 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).

In [73], Leon and Peters present a heuristic for composing partial setup and compare its solutions to the corresponding unique, minimum and group setups. Obviously, unique setup dominates when batch sizes are large (such as in mass production). The group setup strategy dominates the minimum setup strategy, because it considers all the PCBs. The partial setup strategy performs well under all scenarios. In [74], the same authors confer similar results from a broader set of experiments. In [90], Peters and Subramanian analyze four partial setup strategies—unique setup, sequence dependent setup, tradeoff dependent setup, and minimum setup—and conclude that no single fixed strategy dominates in all scenarios.

3.3 Component Allocation to Similar Machines (1–M)

Only few papers considering the case of similar (sequential) machines in the same production line have been put forth. Here, the most eminent criterion is workload balancing so that the bottlenecks of the line are eliminated.

Lofgren and McGinnis [79] present a soft configuration decision, which has an impact on two key criteria: workload on each machine (i.e., balancing), and material handling (i.e., machine visits). The soft configuration problem specifies the attributes which are available on each machine and, therefore, it determines the set of operations which could be performed. The authors consider three operating policies: *dynamic* (where tools are added or removed so that the required operation can be done in one machine without routing it to another), *static* (which specifies configuration for a finite production horizon and routes jobs to appropriate machines) and *pseudo-dynamic* (where some attributes are static and the rest dynamic). The authors give heuristic algorithms for static and dynamic operating policy, where the objective is to maximize machine utilization and minimize material flow transactions.

Ben-Arieh and Dror [15] consider assigning components in a case of two insertion machines, so that all boards in a production plan can be produced, and the output rate is maximized. They give a mathematical formulation of the problem and solve it with a heuristic algorithm.

Askin *et al.* [10] discuss a surface mount technology plant with multiple identical machines. They present a four-stage approach for grouping the boards and allocating the components to the machines: First, the boards are grouped into production families. Next, for each family, the component types are allocated to the machines. After that, the families are divided into board groups with similar processing times. Finally, the groups are scheduled. The objective is to minimize the makespan for assembling a batch of boards and to reduce the mean flowtime. The authors present and compare three heuristic methods—component-assignment/workload balancing algorithm (CAWB), workload balancing algorithm with shortest total processing time (WBASPT), and natural board subfamily algorithm (NBSA)—and conclude that CAWB and WBASPT outperform NBSA.

Watkins and Cochran [114] consider a line of similar insertion machines, and propose a heuristic method for rebalancing the workload by moving components from the bottleneck machine to other machines. However, each move is associated with a cost, and the method finishes when the cost of a move outweighs savings.

McGinnis *et al.* [87] give a mathematical model for component allocation for both coupled and uncoupled machines. Ammons *et al.* [8] continue the work by considering component allocation to two or more placement machines, when the objective is to balance a combination of the assembly time and the machine setup time. When machines are coupled, the workload balancing reduces to maximizing the throughput of the bottleneck machine on line. The authors approach the component allocation problem by developing two heuristic methods based on list processing and branch-and-bound technique. Furthermore, they give an integer programming formulation, which tries to minimize the maximum combined PCB assembly and machine setup time for each PCB over all machines.

Brandeau and coworkers consider assigning components to machines in an assembly plant with *different types of workphases* (automatic, semi-automatic and manual). The goal is to minimize the total setup and processing cost for assembling all boards. In [21], Brandeau and Billington present two heuristic algorithms: *stingy component* (which tries to avoid assigning less frequently used components to the automatic workphase) and *greedy board* (which tries to assign a whole board to a single workphase instead of splitting it). After a set of tests, the authors conclude greedy board to be a better method of the two, because the setup cost are high relative to the insertion costs. In [48], Hillier and Brandeau extend the same problem by presenting a new mathematical model and an improved heuristic based on branch-and-bound technique. In [47], the same authors further expand the mathematical model by introducing a workload balancing criterion and introducing CMWB (cost-

minimizing, workload balancing) heuristic.

3.4 Scheduling (M–M)

Production can be analyzed on different levels: On the *long-term level*, the analysis uses general presumptions and a simplified mathematical model, whereas a *short-term analysis*, as a rule, is based on simulation. On the highest level of production planning the problems are general, and they cannot be solved without the support and feedback from the lower levels of planning.

Gershwin *et al.* [41] present a decomposition approach for determining dispatch dates for an FMS, in which the machines are unreliable to satisfy the production requirements. The authors consider, when the parts should be dispatched to satisfy weekly production requirements. The problem is solved in two stages: First, the instantaneous production rates are solved by considering a high level continuous dynamic programming problem. After that, a combinatorial algorithm determines the dispatch times at the lower level.

Johri [58] considers scheduling from a practical point of view, and presents a technique for sequencing PCB batches. The presented method begins with a realistic scenario, in which the production has been active for some time, and new jobs are inserted to existing production sequence. At each insertion, the goal is to minimize the imbalance of the machine workloads and to ensure that the due dates are not violated. In addition, the number of setups is minimized, if possible. The method works as follows: First, it determines desirable production rate for each workstation. After that, the inserted jobs are grouped according to their due dates. Next, the method determines the group with the least slack time, and calculates penalties for each job in that group. Finally, the job with the minimum penalty (other heuristic rules are also possible) is inserted into the sequence. When the situation is updated, iteration continues until all jobs are sequenced.

Johri [59] discusses the design of a production line with respect to the capacity constraints and flowtime minimization. The intention is to determine the amount of different machine types, buffer sizes, lot sizes and the loading sequence, when the products, processing times on each machine type and the machine downtimes are given. The number of machines required in a workphase are obtained from a simple equation, as well as the average flowtime and capacity of the line. The average in-process inventory can be acquired by using Little's law (i.e., it equals the average flowtime multiplied by the production rate), and by minimizing this value the desired flowtime can be attained.

Lofgren *et al.* [80] study the *station routing problem* (SRP), where the sequence of the component assembly operations is determined so that it minimizes the number of workstation visits. The authors use a graph theoretic approach, in which a precedence graph is partitioned according to workstations and used to determine a workstation sequence which has the minimum cardinality. SRP is shown to be \mathcal{NP} -hard, and therefore the authors test and analyze several different heuristics for the problem.

Klegka and Driels [66, 34] put forth a simplified model for measuring costs in long-term production planning. It yields intuitive results when used for comparing different production line layouts. The authors observe that the most economical assembly system is obtained when enough manual assembly personnel are provided to reduce the maximum station time to that of the slowest assembly machine. In batch production, even small lot sizes are economically justified, since the effect of frequent changes is less significant than might be expected.

Ben-Arieh and Maimon [16] consider a case of two different sequential machines which use the same sequence (i.e., a permutation schedule). The objective is to minimize the mean flowtime, and the authors solve the problem by using simulated annealing approach.

R. Ahmadi and Wurgaft [6] consider a mid-variety, mid-volume production environment with synchronized flow manufacturing, where the materials move smoothly and continuously from one operation to the next. The objective is to maximize the throughput rate. Each time there is a major change in the product mix and demand, there are three problems to be solved: (1) how many stations should be created, (2) how many machines each station should have (i.e., machine allocation problem), and (3) which operations are performed by which station (i.e., staging problem). To balance the total workload the maximum workload of a station should be minimized. The authors give a mathematical formulation of the problem as a quadratic integer program.

Dagnino [30] presents a system which combines product design, assembly planning, line balancing and the generation of the shop floor drawings. The main problem is general process planning, which involves deciding how to use the capabilities of the shop floor to manufacture a given product. In PCB assembly, the author recognizes the following stages: (1) identifying the components to be printed on a PCB, (2) determining the required assembly operations, (3) assigning resources to the assembly operations, (4) sequencing the assembly operations, (5) line balancing, and (6) developing shop aids. The major issue is the hierarchy of these decisions and dependencies between the stages: When a decision is made on a higher level, the lower-level characteristics are generalized. However, at some point, the technical details cannot

be omitted, and the system must be able, for example, to calculate accurate operation times for each individual machine.

Khoshnevis *et al.* [64] also recognize the potential of a general scheduling system, and present an integrated system for assembly planning and schedule generation. In assembly planning, the system prepares a detailed sequence of operations to transform a set of disjoint components into a final product. In scheduling, the system assigns manufacturing resources to the operations indicated in the assembly plan in such a way that some relevant criteria (e.g., due dates) are met. The main idea of the solution algorithm is to assign jobs to machines on the basis of the availability of machines (or of a quantifying factor which is calculated with heuristic rules) and the requirements of the unfinished jobs.

Klincewicz and Rajan [67] consider the component grouping problem in an environment of multiple board types and multiple workcells. The problem is to decide how should the components be assigned (i.e., partitioned) among the workcells so that the number of machine visits is minimized and the workload is balanced. The authors give an integer programming formulation, where minimizing the machine visits is modeled as an objective function and the load balancing as a constraint (i.e., each workcell has a maximum allowable deviation from the average insertion volume). GRASP (greedy random adaptive search procedure) method (i.e., a local search replicated many times with different, randomly chosen starting points) is used to solve the problem. The authors present two variants of the initial starting points: Component-based variant assigns a pair of components to the same workcell if a large number of the items to be produced use both components, whereas code-based variant looks at the larger volume board types and attempts to pack the components belonging to a board type on as few new workcells as possible. The initial solutions are then improved by moving components from one workcell to another or by exchanging components between two workcells.

Dessouky *et al.* [32] convert the scheduling problem of flexible assembly lines to a flow line scheduling problem. This approach assumes that each machine processes the product at most once, the setup time depends only on the machine type, processing time variability is negligible, and setup times are small in comparison to batch processing times. The goal is to maximize the throughput while keeping WIP at a minimal level without increasing the resources of the assembly process. The batch sizes and the number of each machine type are given in the problem formulation. The authors present a solution method comprising two phases: In the first phase, machines are grouped into workstations which are visited by each product in the same sequence. Thus, the problem is converted into a flow line configuration problem. Moreover, the task is to determine a permutation schedule,

which maintains the same sequence down the line in each workstation. In the second phase, the products are assigned to the first workstation (since the subsequent workstations will have the same sequence) by using Campbell's heuristic.

Feo *et al.* [39] also present an integrated decision support system, IN-SITES (integrated scheduling and throughput evaluation system), for scheduling, inventory and throughput evaluation. It provides the shop floor managers with the real-time information needed to make critical production planning and control decisions. The user inputs the PCBs, due dates, quantities, WIP, and machine state. After that, the system computes the daily production quantities and generates MIS (management information system) reports. Because planning includes several conflicting objectives (e.g., due dates, workload balancing, throughput maximization and waiting time reduction), the system offers a set of schedules, from which the user can select one for the production. The authors point out that instead of being constant, the load times fluctuate with the job priorities, parts status, processing requirements and customer demands. Variability of the processing capacity defers the work and delays the batches. The system assigns the lot sizes so that they divide the demand evenly, and schedules the jobs according to the slack values.

Kim *et al.* [65] compare four sequencing algorithms, where the objective is to minimize the sum of tardinesses. NEH algorithm sequences the jobs according to a simple objective function (e.g., in increasing order of due dates or in decreasing of total workload). ENS (extensive neighborhood search) method begins with a proper initial sequence and tries to improve it with swap operations. RBO (rolling block optimization) algorithm also begins with an initial job sequence. After that, it resequences a block of sequential jobs (i.e., window) trying to improve the local sequence while the rest of the jobs remain intact. The window gradually moves through the sequence until it reaches the end. The fourth studied approach, *tabu search*, is based on local search methods, where getting back to previously found solutions is prevented with a tabulist.

Johnsson *et al.* [57] introduce a *generalized flexible flow line* (GFFL) scheduling problem, where the products visit successive machine banks, and the processing time is a function of the product and the machine type. The authors present a mathematical model of the problem and describe an interactive production scheduler, which can be used for analyzing schedules created either manually or by a heuristic algorithm.

Lin *et al.* [76] present a Bi-Level Scheduling System (BLISS), which integrates both product-level and board-level scheduling on a capacitated multiple flowline and strives to determine economical lot sizes and job sequences. Product level scheduling shifts a complete set of boards for a product so that

no WIP inventory is built up. Because this does not necessarily use the production capacity optimally, the schedule is then improved with a board-level scheduling, which minimizes tardiness but may increase WIP. The authors approach the problem by applying general scheduling methods (e.g., Palmer’s algorithm, Gupta’s algorithm and CDS heuristic). From their experiments the authors conclude that the two-level approach suits well when due dates are very tight or product demand is highly variable.

Nesbit [88] describes general requirements for constructing a computer system for analyzing assembly processes. The possible benefits of such a system include reduced machine setup times, increased quality of the final product, and better throughput.

R. Ahmadi and Kouvelis [5] present a mathematical framework for designing and configuring the layout of electronics assembly lines. When the machines are tightly coupled due to small buffer space, throughput can be increased and WIP levels reduced by decreasing the setup times and balancing the machine workload. Also, configuration can be based on the assumption that around 80 percent of the cumulative product demands are known to follow a regular pattern. The authors analyze three design approaches: In the *mini-line* approach, PCBs are divided into families, and each family is produced in a dedicated line. In the *flexible flow line* (FFL), there is only one line and all the components required for the assembly of the various PCBs are concurrently present on the line. Thus, major line setups can be avoided but the machines are bound to be underutilized. In *hybrid line* approach, several PCB families are produced in the same mini-line. The authors formulate the *electronics assembly line design problem* (EDP), from which they define special cases for each of aforementioned approaches. From the computational experiments, the authors conclude that FFL dominates the other design in terms of throughput performance but with increasing setup times, the mini-line provides higher throughput than the others. Moreover, the authors discuss the suitability of the designs to different production environments.

Balakrishnan and Vanderbeck [11] discuss a high-mix, low-volume production plant with several identical assembly lines. Each line includes a chip-shooter which is the bottleneck the line. The aim is to group products and assign the groups to different lines so that workload is balanced and the setup times are minimized. Because these goals are contradictory, the presented model tries to minimize the setup costs while keeping the workload within prespecified limits. A partial setup (which should not be confused with the partial setup strategy of Leon and Peters [74]) comprises permanent and temporary components (i.e., a standard and custom setup [8]). The authors argue that this partial setup is a compromise between “complete” (i.e.,

unique) and “incremental” (i.e., minimum) setup, because it reduces setup operations by exploiting component commonalities and is easier to handle than incremental setups. The authors give an IP formulation of the problem and reformulate it as a capacitated set covering (CSC) model. CSC has two subproblems: Production selection (PS) decides to which machine a product should be assigned so that setup times are minimized and workload is balanced. Setup optimization (SO) decides which components are permanent and which temporary for a group of products on a single machine so that setup costs are minimized. PS is solved with list heuristics, local search and simulated annealing. SO is solved with a combination of heuristics, lower bounds and enumeration.

Lin and Tardif [77] consider multiple board types in a single line. The line is set up once, and there are no setups between the boards. The problem is to partition the component types to the machines in the line so that all the boards can be processed quickly and with a good workload balance. The board demands and machine breakdown occurrences are predicted with a probability distribution. Moreover, it is assumed that there are no precedence constraints on the order of component placements, the placement time is constant for all the components placed by a machine, and a new batch cannot be processed until the current one is completed. Although heuristic methods (e.g., greedy and two-dimensional vector packing heuristic) often provide satisfactory results, they are not applicable here, because they do not consider demand and capacity uncertainty. Therefore, the authors present a stochastic mixed-integer programming formulation for the problem, where the objective is to minimize the makespan for all the PCBs during a production cycle.

Spedding and Sun [103] describe a cost system which uses an activity based costing (ABC) model. Since much of the significant costs in producing an item are not volume related, ABC should provide a better cost model. The idea is to develop a discrete event simulation (DES) model by observing the actual production times of activities in a plant and then characterizing their statistical variation. Activities are classified into four general categories according to when they are performed: unit level (each time a unit is produced), batch level (each time a batch is produced), product level (needed to support the manufacturing of a product), and facility level (needed to sustain the factory’s performance; e.g., rent, depreciation, and insurance). The authors apply ABC modeling in a simple SMT assembly line.

Simulation-based systems Simulation has been a widely used approach for modeling PCB assembly plants. Almost all papers which compare scheduling algorithms use simulation for testing the methods. However, because the

focus in these papers is on developing actual optimization algorithms, the description of the simulation model is usually terse, and testing and analyzing settle for a modest set of test cases and bypass statistical tests.

Shevell *et al.* [97] present an early simulation system for analyzing a complex PCB manufacturing system. They also discuss the validation of the system and compare the results with real-world data. Taylor [107] also describes a simulation tool for evaluating flexible control strategies.

Johtela *et al.* [61] describe a simulation tool for creating and analyzing schedules in a GFFL environment. The system simulates the production from an initial situation to a given moment in the future. The input defines the product batches, their allocation and sequence for each machine and the due dates. The output includes summaries of the production period including the lateness of the batches and machine workload charts. The user can reconsider the allocation and sequencing of the batches and repeat the simulation and update operations to find a better balancing of the workload. Häyrynen *et al.* [50] describe several scheduling algorithms for GFFL problem, which are implemented and integrated into the system.

Jackson and Johansson [51] present a discrete event simulator to support decentralized decision making. Production control benefits from that the workers have a forecast of the coming workload in real time. The system is directly linked to the information system of the company and it updates the state of the production process every ten minutes. Each machine has a forecast for six hours, which is shown on the screensaver on every PC in the shop floor. The simulation operates on both strategic level (development of a new production system or improvement of a production system) and operational level (estimates of capacity needed for a product mix, consequences of WIP for a particular product mix, and consequences in delivery dates due to a particular plan).

Estremadoyro *et al.* [35] describe a simulation tool, Electronics Manufacturing Simulator (EMS), for modeling and analyzing assembly line layouts. The system uses simple specifications for forming a processing line, generates alternative designs, approximates throughput, simulates the operation with different parameters, and produces a set of output summaries for the given layout. The system comprise a line definer and static analyzer, which produces station processing times (used in estimating the maximum throughput of each station) and line capacity.

Smed *et al.* [100] discuss the structure and operation of an interactive scheduling system for a high-mix low-volume SMT assembly line involving multiple criteria (e.g., different adhesive types, conveyor belt configurations, and priorities). The criteria are modeled with fuzzy sets, and the user can set weights for the importance of the criteria. Thus, the poor satisfaction

of some criterion can be compensated with other criteria. The user can also manipulate the schedule directly via a graphical user interface as well as employ feeder and printing order optimizers selectively.

4 Commercial Software Systems

There are several software tools for production planning available in the market but relatively few of them are specialized to the PCB assembly. Generic tools seldom suit for production planning in PCB manufacturing due to the special requirements associated with it: Firstly, the system must have knowledge about the machinery used in the production environment, its capabilities and limitations. Secondly, a production planning tool should have a tight integration with the existing systems, such as CAD/CAM (*computer aided design/manufacturing*) or MRP (*material resource planning*) systems. Thirdly, in practice the usability of a system reduces drastically, if it does not produce NC-codes for the machines. Since PCB manufacturing facilities are usually highly automatized, the goal of a production planning system is not to impose any further work to the personnel.

In this section we review briefly some of the most popular software systems designed to support PCB assembly. The systems and their key features are listed in Table 1. We concentrate on the optimization features, albeit optimization is rarely the primary task in these systems. The systems are mainly designed for automatizing the PCB design and manufacturing processes, such as joining *bill of materials* (BOM) files and CAD files together and generating NC-codes for machines. We have to point out, however, that our experience with some of these system is limited, and, therefore, we are in some cases forced to resort to information gathered from the vendors' web pages and brochures [60]. The possible errors, omissions and inaccuracies are due to this fact. Moreover, the list of the machine vendors' systems includes only a few entries although almost all machine types in the market have their own proprietary system. However, the current trend is that the machine vendors are starting to co-operate with the general system providers.

In Table 1 the systems are divided into two categories: tools provided by the machine vendors and tools provided by independent companies. A machine vendor's system usually supports only its own equipment and includes a concise interface and some simple code optimization routines. Conversely, since production plants may comprise multiple machines from different vendors, the independent systems aim at providing better support (e.g., a centralized component library, better CAD support, product transportation from one machine to another) for various machine types and makes. They

General Information		Supported Machine Types				Integration		Optimization					
Vendor	Package	Platform(s)	Pick & Place Machines	Rotary Turret Machines	Thru Hole Machines	Input from CAD	I-BOM & CAD Joining	NC-code Generation	Printing Order	Feeders	Setup	Line Balancing	Scheduling
Independent Companies													
Dareba Corporation	cim-SCAN	DOS, Windows, Unix	•	•	•	•	•	•	•	•	•	•	•
Miltron Corporation	CIMBridge '97	Windows	•	•	•	•	•	•	•	•	•	•	•
Tecnomatix-Unicam Software Inc.	UniCam	Windows	•	•	•	•	•	•	•	•	•	•	•
	Factory Advisor	Windows											
Tecnomatix Technologies Ltd.	EXALINE	Windows, Unix	•	•	•	•	•	•	•	•	•	•	•
Unisoft Corporation	EZ-CIM/EZ-CAM	Windows	•	•	•	•	•	•	•	•	•	•	•
FABMaster Inc.	FABMaster	Windows	•	•	•	•	•	•	•	•	•	•	•
AIS Corporation	CircuitCAM	Windows	•	•	•	•	•	•	•	•	•	•	•
Optelco	Optel	Windows	•	•	•	•	•	•	•	•	•	•	•
Harelda Oy.	SMDOPTIMIZER	Windows	•	•	•	•	•	•	•	•	•	•	•
Machine Vendors													
Universal Instruments Corporation	UIC	Windows, OS/2	•	•	•	•	•	•	•	•	•	•	•
Fuji America Corporation	FujiCam	Windows	•	•	•	•	•	•	•	•	•	•	•
	XCME	Windows											
Panasonic Factory Automation	PanaCIM	Unix	•	•	•	•	•	•	•	•	•	•	•
	PanaPRO	Windows	•	•	•	•	•	•	•	•	•	•	•
Sony	TIMMS	OS/2	•	•	•	•	•	•	•	•	•	•	•

Table 1: A summary of the features of commercial PCB assembly support systems

also support both rotary turret and pick-and-place machine types. Some include support for also through hole machines in addition to the surface mount devices. Most of these systems run on a PC under Microsoft Windows. Some systems also contain networking capabilities that allow them to integrate into a local-area network. These systems can obtain on-line data from the machines and use it to control the production.

The optimization—the word ‘optimization’ is used here loosely, since in some systems the term “optimal solution” actually means a feasible solution—features of the systems vary greatly: Some use a hierarchical decomposition strategy to optimize each level of production ranging from the printing order and feeder arrangement optimization of a single machine to the scheduling of the entire production environment. Others offer only a few means to improve the operation of a machine or balance the component allocation between machines. The features listed in Table 1 are printing order and feeder optimization of the supported machines (1–1), setup strategy optimization (M–1), balancing the load between machines in a line (1–M), and scheduling the jobs, which includes their allocation to the lines (M–M). Since in many cases it is difficult to estimate the quality of the solutions produced by the systems, we have not evaluated the utility of the optimization algorithms used in the systems in greater detail.

5 An Example of a Production Planning System

In the previous section we gave a summarized view of the currently available applications for production planning in PCB assembly. To elaborate the topic of practical systems, we will next discuss an integrated production planning system (for further details, see [100]). The system is used for solving three distinct problems (see Figure 8): to group PCBs according their components (i.e., group setup strategy), to assign the components of each group to feeder slots (i.e., feeder optimization), and to minimize the printing time of each individual PCB type on the basis of the feeder setup of group (i.e., printing order optimization).

In the first place, the board data, production demand and due dates are derived from CAD and MRP data. The actual production planning begins with grouping, which can be done algorithmically or manually. After that, the user can sequence the jobs in each group according to his preferences and knowledge of the actual state of the production, which enables him to adapt to sudden changes in the production environment. A screenshot of

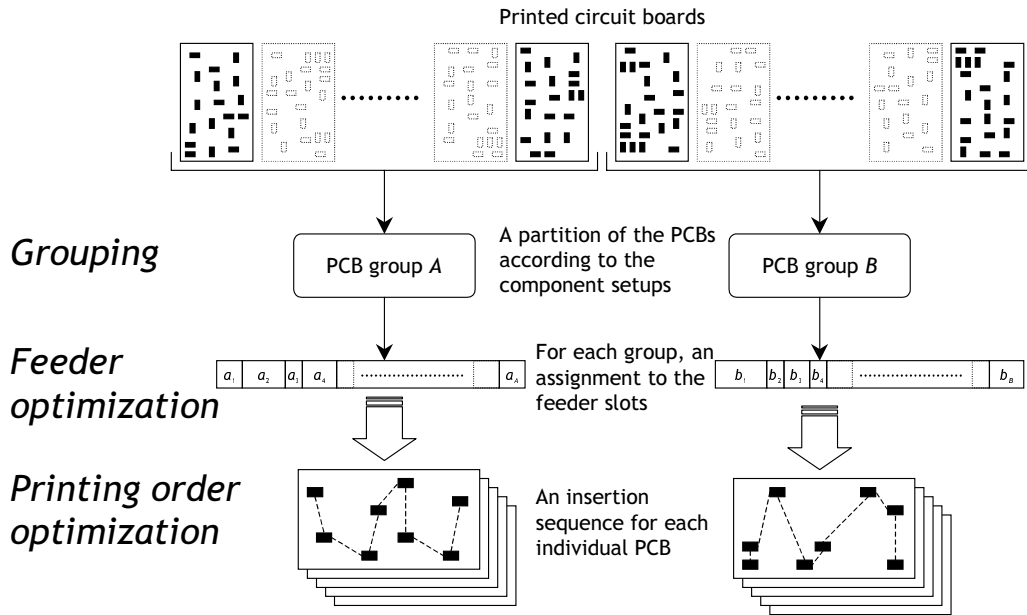


Figure 8: The problems tackled by the integrated production planning system

the system is shown in Figure 9. *Job Repository* window is used to input and edit jobs, and it also acts as a repository for jobs to be scheduled. The jobs are represented by icons whose appearance indicates the attributes of the jobs. The *Machine Schedule* window visualizes the grouping and allows the user to rearrange the jobs; each column corresponds to a job group in a schedule, and each cell in a row represents a particular job. *Job Inspector* and *Selected Jobs/Components* windows give information about the PCBs and their components (e.g., how many feeder slots the jobs require or how long it takes to process the jobs in a machine).

The system is used for planning a typical high-mix low-volume production, where the daily production program includes ten or less different products, and the setup times form a significant part of the total production time. Therefore, the main objective is to minimize the setup times by using the group setup strategy. In addition to the component setup, the system also considers other criteria when forming the groups: Firstly, the widths of the PCBs vary, and the change of the conveyor width causes an interrupt in the printing process. Secondly, some PCBs require component printing on both sides, and in order to avoid unnecessary storing, the other side should be printed as soon as possible after the first side. Thirdly, the surface mounted components are fixated in an oven which must be heated or cooled if the type of the adhesive used in the board changes. Finally, the PCBs are classified

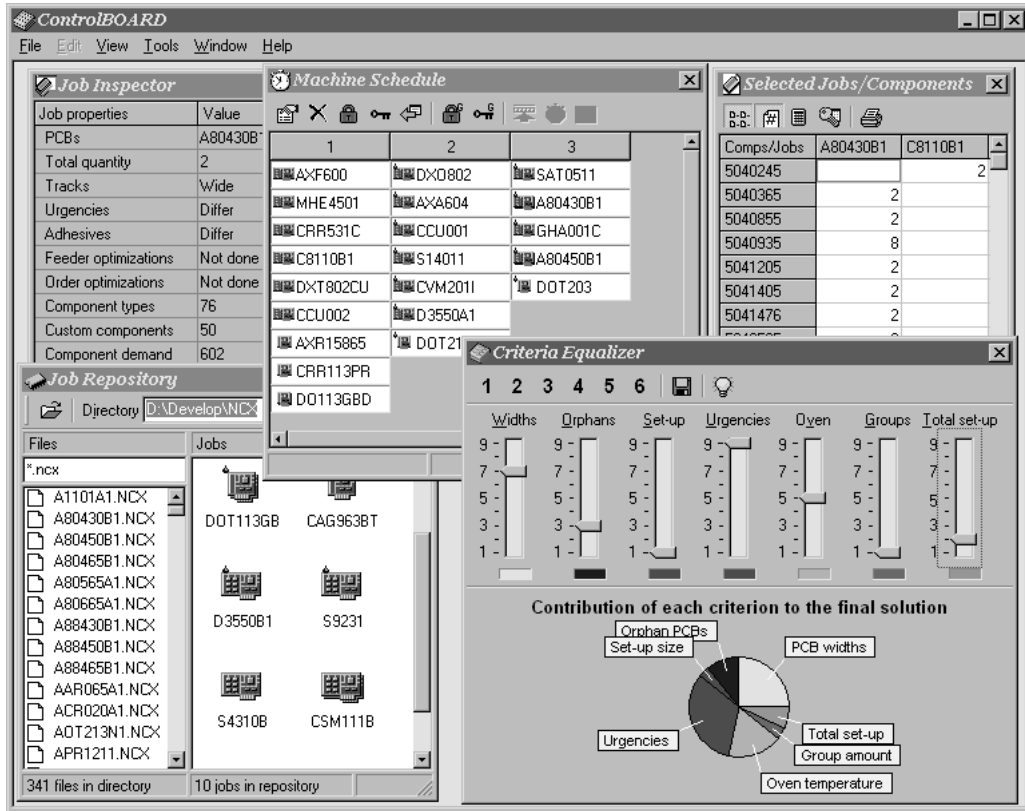


Figure 9: A screenshot of the production planning system

according to their urgency, and the PCBs with similar urgencies should reside in the same group. The slider bars in the *Criteria Equalizer* window are used to adjust the relative importances of the criteria (the scale is from one to nine where nine corresponds to the highest importance and one to the lowest). The pie diagram shows how much attention a criterion gets in the final objective function. The multiple criteria optimization algorithm, which is based on a repair-based local search heuristic, is used to improve the current grouping (for further details, see [62]). The algorithm can be stopped at any time and the currently best solution is available to the user.

When the grouping is satisfactory, the user can define the feeder assignments for the groups. The goal is to arrange the components in the feeder slots so that the total printing time of all the PCBs in the group is minimized. Since this is a difficult task, the system uses heuristic algorithms which take into account the closeness of different components on the PCBs, different component handling speeds (e.g., turret, recognition, placement, pickup and table) and component widths. All this information is stored in the compo-

nent library of the system. The heuristic gives a feeder setup that enables a fast printing of all the PCBs in a group.

Once the feeders have been assigned, the system has to decide the printing order for all PCBs in the group by using an algorithm that is specifically tailored for the machine type in question (i.e., the system uses information about the machine configuration for fine-tuning). The algorithm is based on a local search heuristic which uses discrete event simulation in the cost function. The user can choose from three different optimization algorithms (quick, medium and slow), and the quality of the solution depends on what algorithm is chosen (i.e., the slowest method is presumed to give the best solutions). On average the printing times provided by the optimization algorithm are more than ten percent better than the times obtained by using the machine vendor's optimization system.

The actual machine codes are converted into a generic format, which enables the system to use any kind of numerical control (NC) code: When the machine type changes, the converter can be changed or—if a corresponding converter does not yet exist—it can be implemented in a short period of time. Because all processing is done with the generic code format and the machine configuration files are used to fine-tune the optimization algorithms, the system can be used in different machine configurations. Consequently, the system is general and can be easily tailored to suit new machine types.

6 Final Remarks

We have reviewed in this paper theoretical and practical approaches to production planning in PCB assembly. By regarding PCB assembly as a specialization of a general FMS, we gave a methodology for constructing practical production planning systems and discussed the problems associated with it. We described PCB assembly in different levels: the physical fundamentals, the operation of the insertion machines, and the classification of plant layouts. In the literature review, we classified the problems into four categories—single machine optimization, setup strategy, component allocation to similar machines, and scheduling—and gave a brief summary of the research done so far. Finally, we also looked the problem domain from another perspective: We gave a summary of the commercial applications currently available for PCB assembly, and presented one production planning system in more detail.

To conclude, we would like to point out six key topics for the future research on production planning in PCB assembly:

Rolling horizon Although the production plan is made for a given period of time, the production rarely begins with an empty line, nor does the line remain empty, when the due date of the last job of the current plan expires. Yet, this rolling horizon framework is scarcely considered in the problem formulations of the PCB assembly literature.

Applicability Many of the solution procedures overlook the problems associated with the machine operation and the human workers. For example, partial setup strategy may, in some cases, provide the best theoretical solution for a given setup problem—but it may also result that the human operator, who is required to change few feeders whenever the board type changes, is prone to make more mistakes than if he performs larger feeder changeovers less often. Likewise, the technical considerations (e.g., machine code generation), in the main, are brushed aside in the literature, and thus the suggested solution procedures may have little applicability in actual production environments.

Dynamic production Reality rarely follows a plan; there are machine breakdowns, component shortages and maintenance delays, urgent prototype series surpass normal production, and the production plan itself can be subject to sudden alterations during the production period. Therefore, a practical production planning system must be able to cope with this kind of dynamic production and to give new or refined solutions whenever the integrity of the plan is challenged.

Multiple criteria The bulk of research done in PCB assembly contemplates optimizing one—or rarely a few—criterion (e.g., component setup or due dates). In reality, there are usually several more or less important practical aspects which affect the use of the solution. These aspects either define the space of admissible solutions (e.g., release dates, operation durations, setup times and resource availability) or characterize the quality of scheduling decisions (e.g., due dates, productivity, frequency of tool changes and WIP levels). Some of these multiple criteria must be satisfied for a schedule to be valid, while others may not always be satisfied and might need to be relaxed.

User interaction We discussed the importance of involving the human production planner in the decision making process earlier in this paper. To summarize, as long as the production planning systems are designed for not completely automated manufacturing processes (such as PCB assembly), the production planner must retain the final word on the production plan to be

carried out. This means that the planner must be able to override the algorithmic solutions and effectively take the control if an exceptional situation requires it.

Integration The lack of cooperation with other systems (such as CAD/CAM and inventory management) is a common reason why a new production planning system can be reluctantly accepted by the shop floor personnel. Production data should be interchanged automatically via a network—it must not depend on routinely done manual input. Here the key issue is the seamless integration of the production planning system to the other existing systems.

References

- [1] J. Ahmadi, R. Ahmadi, H. Matsuo, and D. Tirupati. Components fixture positioning/sequencing for printed circuit board assembly with concurrent operations. *Operations Research*, 43(3):444–57, 1995.
- [2] J. Ahmadi, S. Grotzinger, and D. Johnson. Emulation of concurrency in circuit card assembly machines. Technical Report RC 12161, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1986.
- [3] J. Ahmadi, S. Grotzinger, and D. Johnson. Component allocation and partitioning for a dual delivery placement machine. *Operations Research*, 36(2):176–91, 1988.
- [4] R. H. Ahmadi and P. Kouvelis. Staging problem of a dual delivery pick-and-place machine in printed circuit card assembly. *Operations Research*, 42(1):81–91, 1994.
- [5] R. H. Ahmadi and P. Kouvelis. Design of electronic assembly lines: An analytical framework and its application. *European Journal of Operational Research*, 115(1):113–37, 1999.
- [6] R. H. Ahmadi and H. Wurgaft. Design for synchronized flow manufacturing. *Management Science*, 40(11):1469–83, 1994.
- [7] K. Altinkemer, B. Kazaz, M. Köksalan, and H. Moskowitz. The integrated modeling of printed circuit board manufacturing. Forthcoming in *European Journal of Operational Research*, 2000.

- [8] J. C. Ammons, M. Carlyle, L. Cranmer, G. DePuy, K. Ellis, L. F. McGinnis, C. A. Tovey, and H. Xu. Component allocation to balance workload in printed circuit card assembly systems. *IIE Transactions*, 29(4):265–75, 1997.
- [9] J. C. Ammons, T. Govindaraj, and C. M. Mitchell. Decision models for aiding FMS scheduling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(5):744–56, 1988.
- [10] R. G. Askin, M. Dror, and A. J. Vakharia. Printed circuit board family grouping and component allocation for a multimachine, open-shop assembly cell. *Naval Research Logistics*, 41:587–608, 1994.
- [11] A. Balakrishnan and F. Vanderbeck. A tactical planning model for mixed-model electronics assembly operations. *Operations Research*, 47(3):395–409, 1999.
- [12] M. O. Ball and M. J. Magazine. Sequencing of insertions in printed circuit board assembly. *Operations Research*, 36(2):192–201, 1988.
- [13] J. F. Bard, R. W. Clayton, and T. A. Feo. Machine setup and component placement in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems*, 6:5–31, 1994.
- [14] A. Barnea and D. Sipper. Set-up reduction in PCB automated assembly. *Computer Integrated Manufacturing Systems*, 6(1):18–26, 1993.
- [15] D. Ben-Arieh and M. Dror. Part assignment to electronic insertion machines: Two machine case. *International Journal of Production Research*, 28(7):1317–27, 1990.
- [16] D. Ben-Arieh and O. Maimon. Annealing method for PCB assembly scheduling on two sequential machines. *International Journal of Computer Integrated Manufacturing*, 5(6):361–7, 1992.
- [17] G. Bhaskar and T. T. Narendran. Grouping PCBs for set-up reduction: A maximum spanning tree approach. *International Journal of Production Research*, 34(3):621–32, 1996.
- [18] D. A. Bodner, M. Damrau, P. M. Griffin, L. F. McGinnis, and A. McLaughlin. Virtual prototyping of electronics assembly systems. In *Proceedings of the 1998 Industrial Engineering Research Conference*, 1998.

- [19] D. A. Bodner, M. Damrau, P. M. Griffin, L. F. McGinnis, A. McLaughlin, M. L. Spearman, and C. Zhou. Virtual machine models in electronic assembly. In *Proceedings of the 1997 Deneb International Conference and Technology Showcase*, Troy, MI, 1997.
- [20] W. Bolton. *Production Planning & Control*. Longman Scientific & Technical, Essex, UK, 1994.
- [21] M. L. Brandeau and C. A. Billington. Design of manufacturing cells: Operation assignment in printed circuit board manufacturing. *Journal of Intelligent Manufacturing*, 2:95–106, 1991.
- [22] T. F. Carmon, O. Z. Maimon, and E. M. Dar-El. Group set-up for printed circuit board assembly. *International Journal of Production Research*, 27(10):1795–810, 1989.
- [23] C.-M. Chang and L. Young. A simultaneous-mounting process for automated printed circuit board assembly. *International Journal of Production Research*, 28(11):2051–64, 1990.
- [24] S. B. Coble and R. E. Bohn. Setup time reduction for electronics assembly: Combining simple (SMED) and sophisticated methods. Technical Report 97-03, University of California, San Diego, CA, 1997.
- [25] Y. Crama, O. E. Flippo, J. van de Klundert, and F. C. R. Spieksma. The component retrieval problem in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems*, 8:287–312, 1996.
- [26] Y. Crama, O. E. Flippo, J. van de Klundert, and F. C. R. Spieksma. The assembly of printed circuit boards: A case with multiple machines and multiple board types. *European Journal of Operational Research*, 98(3):457–72, 1997.
- [27] Y. Crama, A. W. J. Kolen, A. G. Oerlemans, and F. C. R. Spieksma. Throughput rate optimization in the automated assembly of printed circuit boards. *Annals of Operations Research*, 26:455–80, 1990.
- [28] Y. Crama, A. Oerlemans, and F. Spieksma. *Production Planning in Automated Manufacturing*, volume 414 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 1994.
- [29] B. Cyr, S. Lambert, G. Abdul-Nour, and R. Rochette. Manufacturing flexibility: SMT factors study. *Computers & Industrial Engineering*, 33(1–2):361–4, 1997.

- [30] A. Dagnino. Integrated architecture for assembly planning in an electronics manufacturing environment. *Integrated Manufacturing Systems*, 5(4/5):77–86, 1994.
- [31] M. S. Daskin, O. Maimon, A. Shtub, and D. Braha. 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(6):1617–38, 1997.
- [32] M. M. Dessouky, S. Adiga, and K. Park. Design and scheduling of flexible assembly lines for printed circuit boards. *International Journal of Production Research*, 33(3):757–75, 1995.
- [33] S. Dillon, R. Jones, C. J. Hinde, and I. Hunt. PCB assembly line setup optimization using component commonality matrices. *Journal of Electronics Manufacturing*, 8(2):77–87, 1998.
- [34] M. Driels and J. S. Klegka. An analysis of contemporary printed wiring board manufacturing environment in the USA. *International Journal of Advanced Manufacturing Technology*, 7:29–37, 1992.
- [35] D. N. Estremadoyro, P. A. Farrington, B. J. Schroer, and J. J. Swain. Simulation of memory chip line using an electronics manufacturing simulator. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 1330–7, Atlanta, GA, Dec. 1997.
- [36] Y. Fathi and J. Taheri. A mathematical model for loading the sequencers in a printed circuit pack manufacturing environment. *International Journal of Production Research*, 27(8):1305–16, 1989.
- [37] K. Feldmann and R. Feuerstein. Developments and challenges in automatic assemblies of electronic products. Technical report, Institute for Manufacturing Automation and Production Systems, 1998.
- [38] K. Feldmann, J. Franke, and B. Zöllner. Optimization of SMT-systems by computer-aided planning, simulation and monitoring. In *IEEE/ISHM '90 IEMT Symposium*, pages 102–113, Italy, 1990.
- [39] T. A. Feo, J. F. Bard, and S. D. Holland. Facility-wide planning and scheduling of printed wiring board assembly. *Operations Research*, 43(2):219–30, 1995.

- [40] L. R. Foulds and H. W. Hamacher. Optimal bin location and sequencing in printed circuit board assembly. *European Journal of Operational Research*, 66:279–90, 1993.
- [41] S. B. Gershwin, R. Akella, and Y. F. Choong. Short-term production scheduling of an automated manufacturing facility. *IBM Journal of Research & Development*, 29(4):392–400, 1985.
- [42] N. N. Z. Gindy and S. M. Ratchev. Integrated framework for selection of machining equipment in CIM. *International Journal of Computer Integrated Manufacturing*, 11(4):311–25, 1998.
- [43] D. Golding. PCB assembly. *Assembly Automation*, 15(2):10–3, 1995.
- [44] H. O. Günther, M. Gronalt, and R. Zeller. Job sequencing and component set-up on a surface mount placement machine. *Production Planning & Control*, 9(2):201–11, 1998.
- [45] S. Hashiba and T. C. Chang. Heuristic and simulated annealing approaches to PCB assembly setup reduction. In G. J. Olling and F. Kimura, editors, *Human Aspects in Computer Integrated Manufacturing. IFIP Transactions B-3*, pages 769–77. North Holland, Elsevier Science Publishers, Amsterdam, The Netherlands, 1992.
- [46] W. Hernandez and V. J. Leon. An overview of the operations analysis of a two-head flexible assembly machine with interference avoidance. *Computers & Industrial Engineering*, 33(1–2):405–8, 1997.
- [47] M. S. Hillier and M. L. Brandeau. Cost minimization and workload balancing in printed circuit board assembly. Manuscript, 1997.
- [48] M. S. Hillier and M. L. Brandeau. Optimal component assignment and board grouping in printed circuit board manufacturing. *Operations Research*, 46(5):675–89, 1998.
- [49] G. W. Holcomb. Justifying flexible automation for PCB assembly. *Assembly Automation*, 15(2):14–6, 1995.
- [50] T. Häyrynen, M. Johnsson, T. Johtela, J. Smed, and O. Nevalainen. Scheduling algorithms for computer-aided line balancing in printed circuit board assembly. Forthcoming in *Production Planning & Control*, 2000.

- [51] M. Jackson and C. Johansson. Real time discrete event simulation of a PCB production system for operational support. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 832–7, Atlanta, GA, Dec. 1997.
- [52] S. Jain, M. E. Johnson, and F. Safai. Implementing setup optimization on the shop floor. *Operations Research*, 43(6):843–51, 1996.
- [53] N. K. Jha, editor. *Handbook of Flexible Manufacturing Systems*. Academic Press, San Diego, CA, 1991.
- [54] M. Johnsson. *Operational and Tactical Level Optimization in Printed Circuit Board Assembly*. PhD thesis, University of Turku, 1999. TUCS Dissertations 16.
- [55] M. Johnsson, T. Leipälä, and O. Nevalainen. Determining the manual setting order of components on PC-boards. *Journal of Manufacturing Systems*, 15(3), 1996.
- [56] M. Johnsson, T. Leipälä, T. Pulliainen, and O. Nevalainen. Integrated program generation system for a PC-board assembly line. Technical Report 52, Turku Centre for Computer Science, 1996.
- [57] M. Johnsson, S. Peltonen, T. Leipälä, and O. Nevalainen. Work load balancing of a generalized flexible flow line in printed circuit board production. In R. V. Mayorga, editor, *Proceedings of the Fifth IASTED International Conference on Robotics and Manufacturing*, pages 382–9, Cancun, Mexico, May 1997. IASTED, IASTED/ACTA Press.
- [58] P. K. Johri. A heuristic algorithm for loading new work on circuit pack assembly lines. *International Journal of Production Research*, 28(10):1871–83, 1990.
- [59] P. K. Johri. Engineering a circuit board assembly line for a desired capacity and flowtime. *Journal of Manufacturing Systems*, 10(6), 1991.
- [60] T. Johtela, M. Johnsson, J. Smed, and O. Nevalainen. Links to commercial software systems for PCB assembly, 1999. <http://www.cs.utu.fi/scheduling/pcbssystems/>.
- [61] T. Johtela, J. Smed, M. Johnsson, R. Lehtinen, and O. Nevalainen. Supporting production planning by production process simulation. *Computer Integrated Manufacturing Systems*, 10(3):193–203, 1997.

- [62] T. Johtela, J. Smed, M. Johnsson, and O. Nevalainen. Fuzzy approach for modeling multiple criteria in the job grouping problem. Technical Report 227, Turku Centre for Computer Science, Dec. 1998.
- [63] L. P. Khoo and T. K. Ng. A genetic algorithm-based planning system for PCB component placement. *International Journal of Production Economics*, 54(3):321–32, 1998.
- [64] B. Khoshnevis, G. Bottlik, and A. R. Azmandian. Simultaneous generation of assembly plans and schedules in electronics assembly operations. *Integrated Manufacturing Systems*, 5(4/5):30–40, 1994.
- [65] Y.-D. Kim, H.-G. Lim, and M.-W. Park. Search heuristics for a flow-shop scheduling problem in a printed circuit board assembly process. *European Journal of Operational Research*, 91:124–43, 1996.
- [66] J. S. Klegka and M. Driels. Case studies in printed wiring assembly. *Manufacturing Review*, 4(4):286–92, 1991.
- [67] J. G. Klinecicz and A. Rajan. Using GRASP to solve the component grouping problem. *Naval Research Logistics*, 41:893–912, 1994.
- [68] K. R. Kumar and T. T. Narendran. A heuristic for sequencing PCBs with due-dates. *International Journal of Operations & Production Management*, 17(5):446–67, 1997.
- [69] S. Lambert, B. Cyr, G. Abdul-Nour, and J. Drolet. Comparison study of scheduling rules and set-up policies for a SMT production line. *Computers & Industrial Engineering*, 33(1–2):369–72, 1997.
- [70] T. L. Landers, W. D. Brown, E. W. Fant, E. M. Malstrom, and N. M. Schmitt. *Electronics Manufacturing Processes*. Prentice-Hall, Englewood Cliffs, NJ, 1994.
- [71] A. M. Law and M. G. McComas. Simulation of manufacturing systems. In S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, editors, *Proceedings of the 1997 Winter Simulation Conference*, pages 86–9, Atlanta, GA, Dec. 1997.
- [72] T. Leipälä and O. Nevalainen. Optimization of the movements of a component placement machine. *European Journal of Operational Research*, 38:167–77, 1989.

- [73] V. J. Leon and B. A. Peters. Replanning and analysis of partial setup strategies in printed circuit board assembly systems. *International Journal of Flexible Manufacturing Systems*, 8(4):389–412, 1996.
- [74] V. J. Leon and B. A. Peters. A comparison of setup strategies for printed circuit board assembly. *Computers & Industrial Engineering*, 34(1):219–34, 1998.
- [75] M. C. Leu and Z. Ji. Computer-aided process planning for printed circuit board assembly. In *Proceedings of NEPCON East Conference*, pages 289–97, Boston, MA, June 1991.
- [76] F.-R. Lin, M. J. Shaw, and A. Locascio. Scheduling printed circuit board production systems using the two-level scheduling approach. *Journal of Manufacturing Systems*, 16(2):129–49, 1997.
- [77] W.-L. Lin and V. Tardif. Component partitioning under demand and capacity uncertainty in printed circuit board assembly. *International Journal of Flexible Manufacturing Systems*, 11(2):159–76, 1999.
- [78] C. B. Lofgren and L. F. McGinnis. Dynamic scheduling for flexible printed circuit card assembly. In *Proceedings IEEE Systems, Man, and Cybernetics*, pages 1294–7, Atlanta, GA, 1986.
- [79] C. B. Lofgren and L. F. McGinnis. Soft configuration in automated insertion. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pages 138–42, San Francisco, CA, 1986.
- [80] C. B. Lofgren, L. F. McGinnis, and C. A. Tovey. Routing printed circuit cards through an assembly cell. *Operations Research*, 39(6):992–1004, 1991.
- [81] D. Luzzatto and M. Perona. Cell formation in PCB assembly based on production quantitative data. *European Journal of Operational Research*, 69(3):312–29, 1993.
- [82] O. Maimon and A. Shtub. Grouping methods for printed circuit boards. *International Journal of Production Research*, 29(7):1370–90, 1991.
- [83] O. Z. Maimon, E. M. Dar-El, and T. F. Carmon. Set-up saving schemes for printed circuit boards assembly. *European Journal of Operational Research*, 70(2):177–90, 1993.
- [84] C.-H. Mangin. Line efficiency and productivity measures. *Surface Mount Technology*, pages 114–6, Oct. 1999.

- [85] H. B. Marri, A. Gunasekaran, and R. J. Grieve. Computer-aided process planning: A state of art. *International Journal of Advanced Manufacturing Technology*, 14:261–8, 1998.
- [86] L. A. Martin-Vega. DMII: Past, present and future perspectives. Keynote presentation in the 25th International Conference on Computers & Industrial Engineering, New Orleans, LA, Mar. 1999.
- [87] L. F. McGinnis, J. C. Ammons, M. Carlyle, L. Cranmer, G. W. DePuy, K. P. Ellis, C. A. Tovey, and H. Xu. Automated process planning for printed circuit card assembly. *IIE Transactions*, 24(4):18–30, 1992.
- [88] A. Nesbit. Maximising throughput of the component assembly process. *Soldering & Surface Mount Technology*, 10(1):6–9, 1998.
- [89] K. Ohno, Z. Jin, and S. E. Elmaghraby. An optimal assembly mode of multi-type printed circuit boards. *Computers & Industrial Engineering*, 36(2):451–71, 1999.
- [90] B. A. Peters and G. S. Subramanian. 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, 1996.
- [91] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Prentice-Hall, Englewoods Cliffs, NJ, 1995.
- [92] B. Powell. The ‘spider’ and the yen. *Newsweek*, page 46, Aug. 7, 1995.
- [93] K. Rajkumar and T. T. Narendran. A heuristic for sequencing PCB assembly to minimize set-up times. *Production Planning & Control*, 9(5):465–76, 1998.
- [94] M. Sadiq, T. L. Landers, and G. D. Taylor. A heuristic algorithm for minimizing total production time for a sequence of jobs on a surface mount placement machine. *International Journal of Production Research*, 31(6):1327–41, 1993.
- [95] J. M. Sanchez and J. W. Priest. Optimal component-insertion sequence-planning methodology for semi-automatic assembly of printed circuit boards. *Journal of Intelligent Manufacturing*, 2:177–88, 1991.
- [96] C. Saygin, S. E. Kiliç, T. Tóth, and F. Erdélyi. On scheduling approaches of flexible manufacturing systems: Gap between theory and

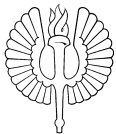
- practice. In T. Borangiu and I. Dumitrache, editors, *Intelligent Manufacturing Systems 1995 (IMS '95): A Proceedings Volume from the 3rd IFAC/IFIP/IFORS Workshop*, pages 61–6. Pergamon Press, Oxford, UK, 1997.
- [97] S. F. Shevell, J. A. Buzacott, and M. J. Magazine. Simulation and analysis of a circuit board manufacturing facility. In J. Wilson, J. Henriksen, and S. Roberts, editors, *Proceedings of the 1986 Winter Simulation Conference*, pages 686–93, 1986.
- [98] A. Shtub and O. Maimon. Role of similarity in PCB grouping procedures. *International Journal of Production Research*, 30(5):973–83, 1992.
- [99] J. Smed, M. Johnsson, M. Puranen, T. Leipälä, and O. Nevalainen. Job grouping in surface mounted component printing. *Robotics and Computer-Integrated Manufacturing*, 15(1):39–49, 1999.
- [100] J. Smed, T. Johtela, M. Johnsson, M. Puranen, and O. Nevalainen. An interactive system for scheduling jobs in electronic assembly. Forthcoming in *International Journal of Advanced Manufacturing Technology*, 2000.
- [101] J. S. Smith and B. A. Peters. Simulation as a decision-making tool for real-time control of flexible manufacturing systems. In *IEEE International Conference on Robotics and Automation*, pages 586–90, Leuven, Belgium, May 1998.
- [102] SMTnet. Glossary, 1999. <http://www.smtnet.com/>.
- [103] T. A. Spedding and G. Q. Sun. Application of discrete event simulation to the activity based costing of manufacturing systems. *International Journal of Production Economics*, 58:289–301, 1999.
- [104] D. R. Sule. A heuristic procedure for component scheduling in printed circuit pack sequencers. *International Journal of Production Research*, 30(5):1191–208, 1992.
- [105] M. R. Supinski, P. J. Egbelu, and E.-A. Lehtihet. Automatic plan and robot code generation for a PCB assembly. *Manufacturing Review*, 4(3):214–24, 1991.
- [106] C. S. Tang and E. V. Denardo. Models arising from a flexible manufacturing machine. *Operations Research*, 36(5):767–84, 1988.

- [107] G. D. Taylor, Jr. Simulation of memory chip line using an electronics manufacturing simulator. In O. Balci, R. P. Sadowski, and R. E. Nance, editors, *Proceedings of the 1990 Winter Simulation Conference*, pages 567–9, New Orleans, LA, Dec. 1990.
- [108] S.-H. G. Teng and S. S. Garimella. Manufacturing cost modeling in printed wiring board assembly. *Journal of Manufacturing Systems*, 17(2):87–96, 1998.
- [109] S. D. Thompson and W. J. Davis. An integrated approach for modeling uncertainty in aggregate production planning. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1000–12, 1990.
- [110] Universal Instruments Corporation, 1999. <http://www.uic.com/>.
- [111] P. J. M. van Laarhoven and W. H. M. Zijm. Production preparation and numerical control in PCB assembly. *International Journal of Flexible Manufacturing Systems*, 5:187–207, 1993.
- [112] C. Wang. Layout designs for robotic PCB assembly. *Soldering & Surface Mount Technology*, 10(2):36–48, 1998.
- [113] C. Wang, L.-S. Ho, and D. J. Cannon. Heuristics for assembly sequencing and relative magazine assignment for robotic assembly. *Computers & Industrial Engineering*, 34(2):423–31, 1998.
- [114] R. E. Watkins and J. K. Cochran. A line balancing heuristic case study for existing automated surface mount assembly line setups. *Computers & Industrial Engineering*, 29(1–4):681–5, 1995.
- [115] R. J. Wittrock. An adaptable scheduling algorithm for flexible flow lines. *Operations Research*, 36(3):445–53, 1988.
- [116] Z. Xu, K. Carlson, R. Kurschner, and S. Randhawa. An integrated methodology for surface mount PCB configuration. *Computers & Industrial Engineering*, 35(1–2):53–6, 1998.
- [117] S. H. Yeo, C. W. Low, and K. H. Yong. A rule-based frame system for concurrent assembly machines. *International Journal of Advanced Manufacturing Technology*, 12(5):370–6, 1996.
- [118] P. Zarrow. Automatic components placement systems. *Circuits Assembly Magazine*, Oct. 1996.

- [119] M. Zhou and M. C. Leu. Petri net modeling of a flexible assembly station for printed circuit boards. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 2530–5, Sacramento, CA, Apr. 1991.
- [120] W. H. M. Zijm and A. van Harten. Process planning for a modular component placement system. *International Journal of Production Economics*, 30–31:123–35, 1993.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

<http://www.tucs.fi>



University of Turku
• **Department of Mathematical Sciences**



Åbo Akademi University
• **Department of Computer Science**
• **Institute for Advanced Management Systems Research**



Turku School of Economics and Business Administration
• **Institute of Information Systems Science**