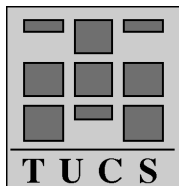


# An Interactive System for Scheduling Jobs in Electronic Assembly

**Jouni Smed**  
**Tommi Johtela**  
**Mika Johnsson**  
**Mikko Puranen**  
**Olli Nevalainen**



**Turku Centre for Computer Science**  
**TUCS Technical Report No 233**  
**January 1999**  
**ISBN 952-12-0364-1**  
**ISSN 1239-1891**

## **Abstract**

In flexible manufacturing systems (FMS) the interaction between the production planner and the scheduling system is essential. This is a typical situation in printed circuit board (PCB) assembly. We discuss the structure and operation of an interactive scheduling system for surface mount component printing involving multiple criteria. The user can compose a schedule by using a heuristic algorithm, but the schedule can be manipulated also directly via a graphical user interface. In addition to system description, we present statistical data of the effect of the system in an actual production environment.

**Keywords:** production planning, printed circuit boards, group technology, flexible manufacturing, multiple criteria, fuzzy scheduling

**TUCS Research Group**  
Algorithmics

# 1 Introduction

A flexible manufacturing system (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. Flexibility in manufacturing provides an opportunity to capitalize on basic strengths of a company. The flexibility of the FMS is characterized by how well it responds to changes in the product design and the production schedules.

The control of the FMS requires a complex interaction of two components [1]:

1. computers to perform automated control and routing activities, and
2. humans to supervise the automation, to monitor the system flow 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 production 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 scheduling system should act as a decision support for the production planner. However, references in literature 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.* [2], the existing software tools are typically (1) too slow and cannot react to the changing shop floor conditions, (2) based on simplistic formulations of 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. As Johnsson notes in [3], these observations are valid in electronic assembly optimization, 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 test cases. However, this approach does not shed much light on the practicality of the method. For this reason, our approach is to build a complete system for daily production

planning in electronic assembly and to validate our results in actual production environments. Accordingly, precise modeling is a more important issue than in cases where the results are not applied to the actual production. Moreover, the usability of a production planning system depends on the interaction between scheduling and optimization algorithms and human production planner: the computer should provide the user with a sufficient support for making the actual decisions (e.g., generate good schedules from which the user chooses—and possibly refines—one for the production).

Our research originates from a *printed circuit board* (PCB) assembly plant<sup>1</sup> where the production line is capable of manufacturing a large selection of different *jobs* (i.e., batches of PCBs); during the last two years the number of different PCB types has been over 350, and their respective annual production volumes vary from one to several thousand boards. Because the batch sizes are relatively small, the production is highly flexible and demands several set-ups daily.

In Group Technology (GT) efficiencies in manufacturing are realized by grouping similar tasks (e.g., according to shape, dimension or process route) and dedicating equipment for performing these tasks [4]. A significant advantage of applying GT principles in scheduling is that the set-up time and, consequently, the set-up costs are reduced. *Job grouping* problem is an example of a scheduling problem where GT is applied. The problem can be stated as follows: A set of jobs (e.g., PCBs) are processed on a machine. During the processing the machine performs one or several operations (e.g., component printing) on the jobs, and each operation requires one or more tools (e.g., components). Tools are stored in a magazine which can hold a limited number of different tools (i.e., it has a certain capacity). We must now find a loading strategy (i.e., a specification of the contents of the tool magazine at the beginning of the processing of each job) with a minimum total set-up time which depends linearly on the number of tool *switching instants*. As a result, the set-up for the whole job group is done on one switching instant and after that all the jobs in the group are processed successively. Tool management in general is addressed in [5]–[9] and set-up strategies for PCB assembly in [10]–[20]. For the theoretical background of the job grouping problem, see Crama *et al.* [8], and for a survey of the relevant literature of GT, see Heragu [21].

The system presented in this paper is based on modeling the multiple criteria of production environment with fuzzy sets. This approach allows us to set weights for the importance of the criteria and possibly compensate the poor satisfaction of some criterion with other criteria (for a theoretical

---

<sup>1</sup>Teleste Corporation, Nousiainen, Finland

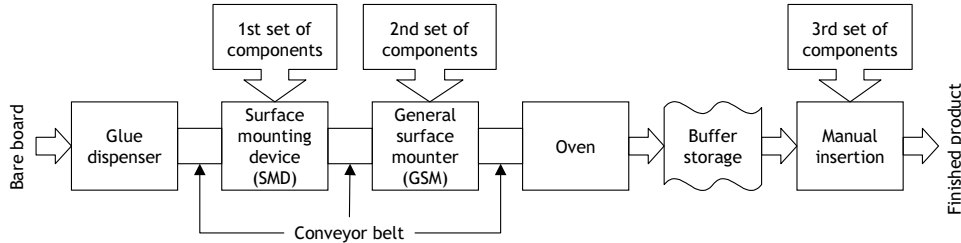


Figure 1: The production phases of surface mount onsertion.

description and analysis of the mathematical model, see [22]). In addition, the system includes a graphical user interface, different levels of optimizing subsystems and an interactive scheduler. The purpose of this paper is to describe each of these components and to give an overall view of the scheduling system as a whole. The paper is organized as follows: We begin with an introduction to the production environment in section 2. Section 3 gives an in-depth view of the scheduling system. The effects of the system in the production are discussed in section 4. Concluding remarks appear in section 5.

## 2 Environment

The production line considered in this paper comprises several successive work phases (figure 1). An initially bare PCB passes first a *glue dispenser* which inserts glue or adhesive paste for fixating the electric components printed in the next two onsertion phases. The first set of components is printed by a *surface mounting device* (SMD) which is adapted to fast operation and used for the majority of the component onsertions. Components which require specialized tools are onserted by a more flexible but slower machine, a *general surface mouter* (GSM). These two onsertion phases are followed by an *oven* which heats the PCBs in order to harden the glue/paste. After that the PCBs wait in a *buffer storage* and finally pass a *manual insertion* phase in which some large components are inserted and soldered.

Because the set-ups and component printing of the SMD consume most of the production time, it is the bottleneck of the whole production line. The machine gets the surface mount components from carriage modules which have a limited number of feeder slots. Therefore, the size of the component set-up is limited, but because the number of different component types in a PCB is significantly smaller than the capacity of the feeders in the machine, we can choose the appropriate input organization quite freely.

Ammons *et al.* [14] categorize the strategies for set-up management in electronic assembly as follows:

1. *Single set-up strategy*: A group of machines is configured to produce a family of PCBs using a single set-up. This can be done by applying (a) a unique set-up strategy in which the family contains only one product type (i.e., mass production), or (b) a family set-up strategy in which the family comprises several product types.
2. *Multi set-up strategy*: A limited component staging capacity prohibits applying the single set-up strategy. This can be done by applying (a) a decompose-and-sequence technique in which the family is divided into subfamilies which are then sequenced to minimize the incremental set-ups between subfamilies, or (b) a partition-and-repeat technique in which the required components are partitioned into subsets restricted by machine capacity.

Because, in our case, the total number of different component types vastly exceeds the feeder capacity, our approach uses the multi set-up strategy with the decompose-and-sequence technique. Günther *et al.* [19] categorize the component set-up strategies in surface mount printing into *unique set-up strategy*, *minimum set-up strategy*, *group set-up strategy*, and *partial set-up strategy*. In this view, our set-up strategy is a variant of the group set-up strategy: we group the PCB types (or jobs), but instead of exchanging the whole feeder set-up, we have a predetermined subset of components (a standard set-up) which remains in a fixed location for a long period of time (from six to twelve months) and which contains the most frequently used component types.

The number of different jobs (PCB batches) processed on the line is high but the amount of PCBs in a job is usually small (a small-lot, high-variety environment). The daily production program includes typically 4–10 different *products* (PCB types). Normally the due dates are considered the most important restriction, but in this case they are managed by a two-level priority classification: products are either *urgent* or *non-urgent*. There are two different widths for the PCBs, and the change of the conveyor width causes an interrupt in the printing process. Also, 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. The last feature that affects the overall production time is that the oven must be heated or cooled if the type of the adhesive changes.

Because the set-up times form a significant part of the total production time (it can be as much as 50 percent), the main objective is to minimize the

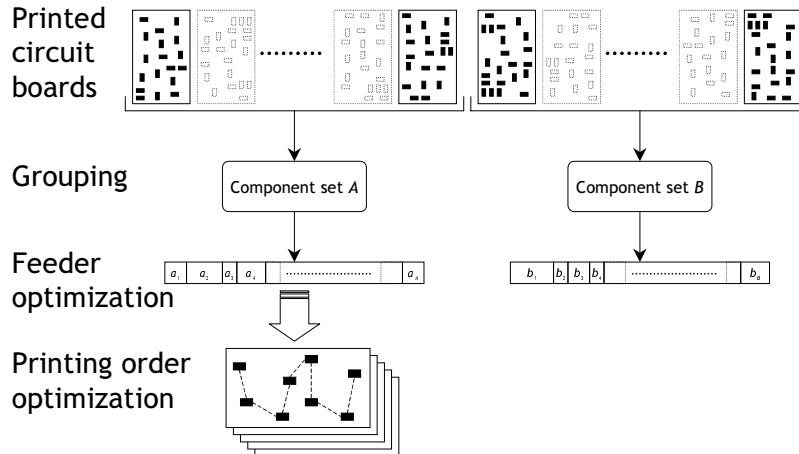


Figure 2: The problems solved by the system

set-up times by forming a minimal number of job groups. In our earlier work [20] we developed several methods (e.g., heuristic algorithms) for solving the grouping, but our solutions lacked a measure which takes into consideration the various “secondary” aspects of the actual production environment. We were able to find a grouping with a minimal number of groups and control somewhat the distribution between the groups, but the aforementioned aspects—urgencies, conveyor widths, oven temperatures, the management of the double-sided PCBs and the size of the set-up—were all ignored. Although the original heuristics improved the actual production, further refinements were still needed. The job grouping problem is extended in [22] by considering also other criteria in addition to the tool set-up. Minimizing the number of set-up occasions is still the primary (or hard) criterion, but we want to find among the feasible solutions the ones which fulfill best the other (soft or relaxable) criteria.

### 3 Description of the Scheduling System

In this section we give a detailed description of the ControlBOARD integrated scheduling system [23]. Figure 2 gives an overall view of the problems solved by the system: PCBs are grouped according to their components (job grouping problem), the components of each group are assigned to feeder slots (feeder optimization), and the printing time of each PCB is minimized separately on the basis of the feeder set-up of the group (printing order optimization).

Figure 3 illustrates the responsibilities of the user and the system. The

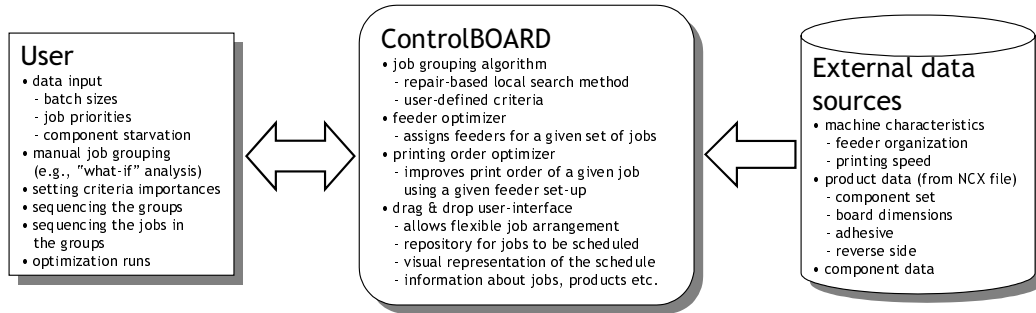


Figure 3: The division of responsibilities

user's responsibilities include data input (e.g., adding new jobs to the schedule), grouping the jobs manually (and possibly locking some groups/jobs in order to prevent the system to make further changes), setting the importance of each criterion for the schedule improvement algorithm, and running feeder and print order optimization for selected jobs/groups. ControlBOARD system provides an algorithm for improving the schedule according to user-defined criteria, feeder optimization for a given set of jobs, printing order optimization for a given set of jobs using a given feeder set-up, and an overall drag & drop user-interface, which allows the user to arrange the jobs in the schedule (and a repository for the unscheduled jobs) and gives a visual representation of the schedule. The system uses external data files for defining the machine characteristics and deriving the required product data. Furthermore, the features of the system include

- a visual presentation of the overall state of the production,
- a possibility to edit the schedule manually (the system checks the capacity constraints automatically) or improve it algorithmically,
- information about products, jobs, components, simulated times etc., and
- a possibility to employ feeder and printing order optimizers selectively.

Figure 4 gives another perspective to the structure of the ControlBOARD system. The system is divided into external parts and internal parts. The external parts include various databases (e.g., NCX files) and external programs for optimization and component library administration. The internal parts include a user interface, a system core, a job grouping optimizer and a fuzzy logic unit. The system core reads data from databases and receives user input from the user interface. The optimizer uses fuzzy evaluation for



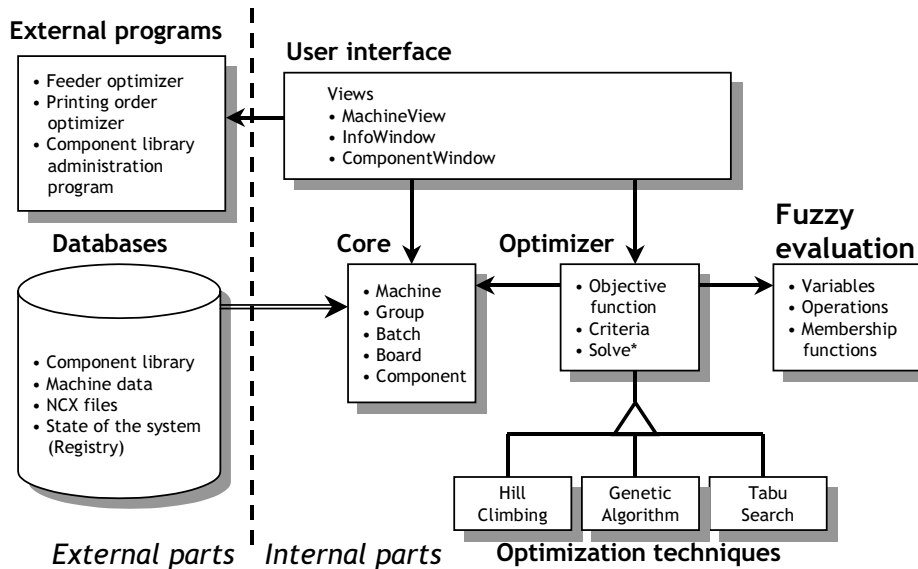


Figure 4: The structure of the system

defining the criteria and the objective function. The actual job grouping optimization method can vary (e.g., it can be based on local search, genetic algorithms or tabu search). The user interface gives the system uniform look-and-feel: it can launch external programs, start the optimizer and interact directly with the schedule stored in the core.

In **ControlBOARD**, the scheduling process begins with the forming of the job groups, which can be done algorithmically or manually. The user can then sequence the jobs in each group according to his preferences and knowledge of the actual situation, which enables him to adapt to sudden changes in the production environment. Furthermore, the user can insert or remove jobs flexibly, and, before the group comes in production, the system generates automatically the machine codes (e.g., NCX) and a “set-up instruction list” for the machine personnel.

Next, we discuss the user interface, optimizer unit and external programs in more detail.

### 3.1 User Interface

The main window of the **ControlBOARD** system is shown in figure 5. The main window is divided to several subwindows which can be moved and resized freely on the screen. *Job Repository* is used to input and edit jobs. This window acts also as a repository for jobs to be scheduled. The jobs are represented by icons whose appearance indicates the attributes (urgency,

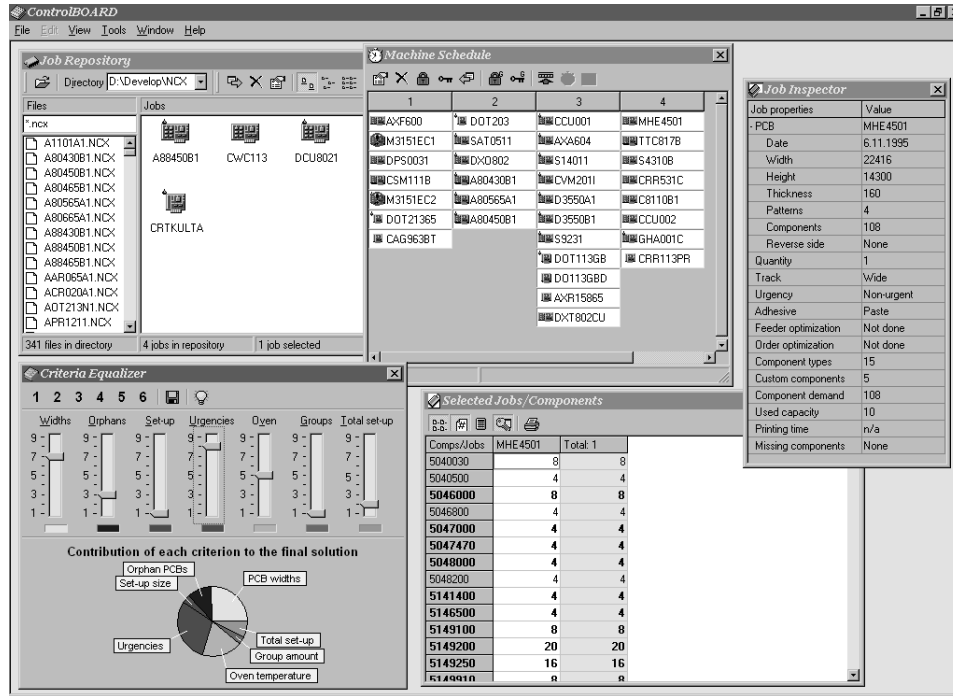


Figure 5: ControlBOARD main window

width, adhesive, reverse side and possible component starvation) of the jobs. *Machine Schedule* shows the grouping of jobs visually. Each column in this window corresponds to a job group in a schedule, and each cell in a row represents a particular job. The user can rearrange jobs by dragging them with a mouse. *Job Inspector* and *Selected Jobs/Components* windows give information about PCBs and their components. The user can select a set of jobs and see, for example, how many feeder slots the jobs require or how long it takes to process the jobs in a machine.

The actual optimization of the job grouping is initiated in the *Criteria Equalizer* window, see figure 6. The slider bars in the upper region of the window are used to adjust the relative importance of each criterion. The scale goes from one to nine where nine corresponds to the highest importance and one to the lowest. The pie diagram in the bottom shows how much attention a criterion gets in the final objective function. The purpose of this diagram is to illustrate the *relativeness* of the values using the classification proposed by Saaty [24], where 1 means equal importance, 3 weak importance of one over another, 5 essential or strong importance, 7 very strong or demonstrated importance, and 9 absolute importance, while 2, 4, 6 and 8 are intermediate values between adjacent scale values. When the importance of a criterion is

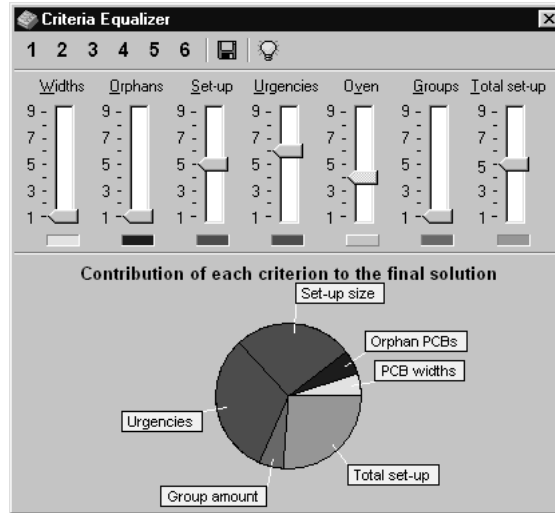


Figure 6: Criteria Equalizer

increased, the relative importance of the rest of the criteria decreases. This is not necessarily apparent to a user who might be tempted to increase the importance of every criterion to the maximum (the diagram shows that this is equal to the case when all importances are set to one).

The following criteria are considered:

- *Widths*: The conveyor track widths of the PCBs in a group should be equal.
- *Orphans*: The opposite sides of a double-sided PCB should be processed in the same group, or the number of orphans (PCBs whose reverse side is in another group) should be minimal.
- *Set-up*: The amount of components needed for the group set-up should be minimal.
- *Urgencies*: Jobs belonging to the same urgency class should be in the same group.
- *Oven*: A group should comprise only glued or pasted boards because of the oven temperature requirements.
- *Groups*: The number of groups should be minimal.
- *Total set-up*: The sum of the set-up sizes of all the groups should be minimal.

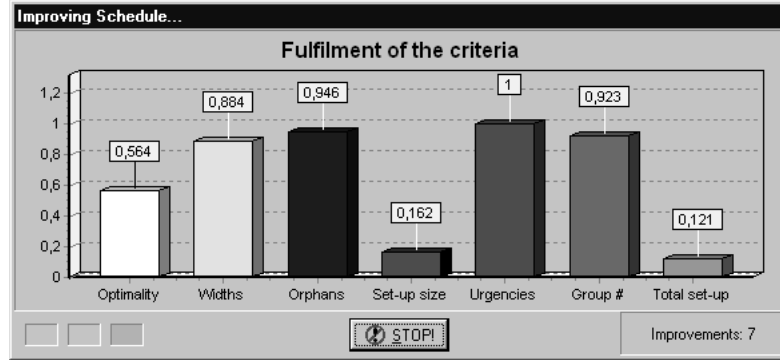


Figure 7: Optimization status window

The mathematical modeling and realization of these criteria is described in detail in [22].

A criterion can be switched off by clicking the colored buttons below the sliders. When a criterion is turned off, its weight becomes zero, meaning that it has no effect on the objective function. This feature is particularly useful because it provides the user with a way to alter the goals of optimization dynamically. The user may also save the slider settings to six memory slots or presets. A saved preset can be restored quickly by pressing one of the numeric buttons in the toolbar.

The multiple criteria optimization algorithm tries to improve the grouping shown in the Machine Schedule window. When the optimization is in progress, a window which displays the fulfillment of each selected criterion is shown (figure 7). The user may stop the optimization at any time. The algorithm will terminate by itself only when all the criteria are completely fulfilled, which happens very seldomly. After termination, the best solution found so far is shown in the Machine Schedule.

An initial schedule for jobs in the Repository can also be computed automatically. This operation uses a heuristic algorithm which tries only to minimize the number of groups in the schedule [20]. This provides often a good basis for the improvement algorithm which is guided by the fuzzy criteria. An alternative way to minimize the number groups is to turn off all other criteria except *Groups* and *Total set-up* and use the improvement algorithm. This demonstrates the versatility of the multiple criteria optimizer compared to traditional heuristics with static objective functions.

## 3.2 Optimizer

ControlBOARD uses a repair-based local search heuristic. Repair-based in this case means that the hard constraints can be violated occasionally to broaden the scope of the search after which the repair operations are used to bring the search back to the set of feasible solutions. The algorithm can be stopped at any time and the currently best solution is available to the user. For a more detailed description of this method and other heuristic algorithms, see [20].

Among the multiple criteria, minimizing the number of switching instants is considered a hard criterion that cannot be violated, but, in addition, we want to find among the feasible solutions the ones which best fulfill the other criteria. Fuzzy techniques are used for modeling the soft criteria and for evaluating the solutions. All the criteria can be taken into account by representing each of them as a fuzzy set and aggregating them together to give an overall optimality measure of the solution. The task is to search for a grouping which has the maximum degree of satisfaction of the specified goals and constraints, both of which may be subject to imprecision. The fuzzy multiple criteria model is discussed and analyzed in detail in [22].

## 3.3 Feeder and Printing Order Optimization

When current grouping is satisfactory, the user can begin to define the feeder set-ups for the groups. The goal is to assign the components in the feeder slots in such a way that the total printing time of all the PCBs is minimized (the same type of arrangement is used in [15]). Since this is a hard task (e.g., see [25] where a similar type of problem is modeled as a quadratic assignment problem), we have developed 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 component library of our system. The heuristic gives a feeder set-up that enables a fast printing of all the PCBs (see figure 8).

The standard feeder set-up can also be defined by using the feeder optimization. However, in this case we have to choose first the components that belong to the standard set-up, and after that we can apply a procedure similar to that we use for defining the feeder set-up for a PCB group. Figure 9 shows a dialogue which can be used when deciding the characteristics of the standard components. After that, the components are arranged according to the frequency of their usage, but the user can still alter the result suggested by the system.

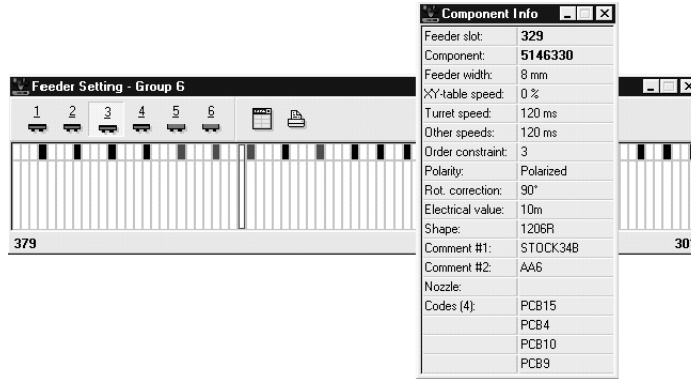


Figure 8: A feeder set-up for a group

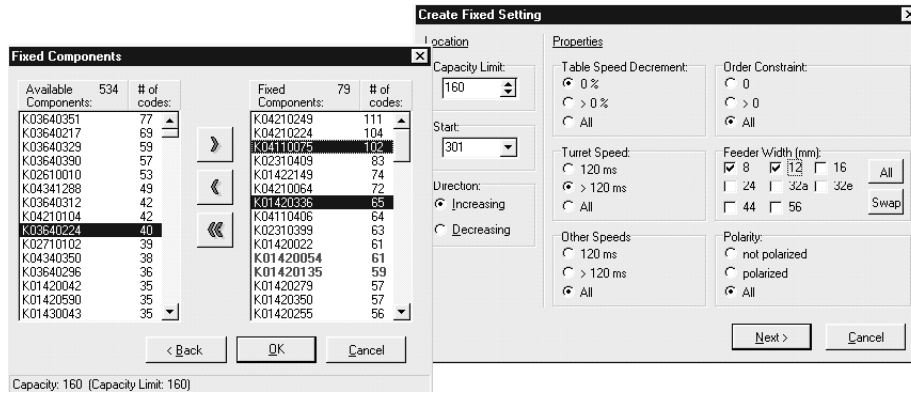


Figure 9: Standard set-up dialogues

Once the feeder set-up has been defined, we have 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 a local search heuristic which uses discrete event simulation (DES) [26] 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 gives the best solutions). The printing order optimization is very effective and the printing times are on average more than ten percent better than the times obtained by using the machine vendor’s optimization system.

The machine codes (e.g., NCX) are converted into a generic format, which enables us to use any kind of NC-code in our system: When the machine type changes, we change the converter or—if a corresponding converter does not yet exist—implement it. 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. Currently these external optimization programs support more than ten different machine types from Fuji, Panasonic, Sanyo and Universal.

## 4 Observations

Table 1 contains numerical data of the production. All the values represent averages of the data gathered from 6–10-week test periods. In addition, in each row the first value is scaled to 100 and the rest are proportioned to it. The first column represents the situation before the first version of the scheduling system was introduced, and the second column after its installation (these effects are analyzed in detail in [20]). The third column corresponds to the situation six months after the introduction of the first version. The last column represents the situation after the introduction of the current system.

The first row indicates the net amount of components printed. The initial increase eroded during the next six months almost back to the original level because the type of production was somewhat altered: the batch sizes decreased (i.e., there are now more jobs to be scheduled) and component starvations narrowed the usability of the grouping. The new system, however, copes better with the present type of production and the net amount is again increased. The second row illustrates the effect of the print order

	Before the scheduling system (spring 1997)	First version of the scheduling system (summer 1997)	Situation six months later	ControlBOARD system (autumn 1998)
Components/hour	100.00	157.60	111.77	134.11
Components/hour in printing time	100.00	115.83	116.31	108.95
Finished jobs/week	100.00	127.27	140.91	231.82
Printing time/job	100.00	111.93	70.64	71.56
Set-up time/job	100.00	64.47	78.95	55.26

Table 1: Production data

optimization, which was introduced already in the first version of the system. After that the value of the components per hour in printing time has remained on the same level. The reason for the decrease in the last column is that several nozzles of the SMD machine were temporarily out of order during the last test period, which affects all the values.

The third row represents the number of jobs completed during a week. The effect of decreased batch sizes is easily observed in the fourth row, which shows the average printing time for a job. Because batches are currently smaller, the completion time in the third and fourth columns is about 63 percent of that of the second column. The real strength of the new system is illustrated in the fifth row which shows the average set-up time required by a job. After the introduction of the first version, the set-up time increased (while the number of completed jobs did not increase in the same proportion) because the system could not adapt to the unexpected changes in the production. The new system, however, managed to restore—and even reduce—the set-up time, which is essential in the current situation. Furthermore, the number of jobs completed weekly has increased 65 percent while the average batch size has remained on the same level.

The system benefits the production planner as well as the workforce assigned to operate the machine. The job grouping approach has increased the accuracy of the production because there are now less set-up operations (e.g., the risk of misplacing a component feeder diminishes). It has also allowed to produce smaller batches efficiently and to reduce the size of the work-in-process storage. Printing order optimization has enhanced the component placement speed and, consequently, increased the productivity. The interactive scheduling system has enabled better reactivity to changes in the production and provides an easy-to-use tool for the production planner.



## 5 Concluding Remarks

We presented a scheduling system for electronic assembly. The system allows interaction between the user and optimization algorithms, which is essential in real-world FMS applications. The system comprises several subsystems (e.g., for different levels of optimization) and provides an integrated platform for the production planning. We have gained promising results from the real-world production, and the system has been accepted by the production planning personnel.

Further research on line-balancing and optimization methods is still needed. We are currently developing algorithms for balancing the load between SMD and GSM machines. Furthermore, we are currently utilizing the experience gained in highly dynamic electronic assembly environments also in other fields of production (e.g., in steel industry).

### Acknowledgments

The authors wish to thank Mr. Jouni Lehtinen from Teleste Corporation for his time and cooperation.

## References

- [1] J. C. Ammons, T. Govindaraj, and C. M. Mitchell, "Decision models for aiding FMS scheduling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 5, pp. 744–56, 1988.
- [2] C. Saygin, S. E. Kiliç, T. Tóth, and F. Erdélyi, "On scheduling approaches of flexible manufacturing systems: Gap between theory and practice," in *Intelligent Manufacturing Systems 1995 (IMS '95): A Proceedings Volume from the 3rd IFAC/IFIP/IFORS Workshop* (T. Borangiu and I. Dumitrache, eds.), pp. 61–6, Oxford, UK: Pergamon Press, 1997.
- [3] M. Johnsson, "Operational and tactical level optimization in printed circuit board assembly." Manuscript, 1999.
- [4] A. Kusiak, "Group technology in flexible manufacturing systems," in *Handbook of Flexible Manufacturing Systems* (N. K. Jha, ed.), pp. 147–94, San Diego, California: Academic Press, 1991.

- [5] C. S. Tang and E. V. Denardo, “Models arising from a flexible manufacturing machine,” *Operations Research*, vol. 36, no. 5, pp. 767–84, 1988.
- [6] O. Maimon and A. Shtub, “Grouping methods for printed circuit boards,” *International Journal of Production Research*, vol. 29, no. 7, pp. 1370–90, 1991.
- [7] R. G. Askin, M. Dror, and A. J. Vakharia, “Printed circuit board family grouping and component allocation for a multimachine, open-shop assembly cell,” *Naval Research Logistics*, vol. 41, pp. 587–608, 1994.
- [8] Y. Crama, A. Oerlemans, and F. Spieksma, *Production Planning in Automated Manufacturing*, vol. 414 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, 1994.
- [9] U. R. Kulkarni and M. Y. Kiang, “Dynamic grouping of parts in flexible manufacturing systems—a self-organizing neural networks approach,” *European Journal of Operational Research*, vol. 84, pp. 192–212, 1995.
- [10] T. F. Carmon, O. Z. Maimon, and E. M. Dar-el, “Group set-up for printed circuit board assembly,” *International Journal of Production Research*, vol. 27, no. 10, pp. 1795–810, 1989.
- [11] M. L. Brandeau and C. A. Billington, “Design of manufacturing cells: Operation assignment in printed circuit board manufacturing,” *Journal of Intelligent Manufacturing*, vol. 2, pp. 95–106, 1991.
- [12] 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*, vol. 6, pp. 5–31, 1994.
- [13] 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*, vol. 8, no. 4, pp. 389–412, 1996.
- [14] 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*, vol. 29, no. 4, pp. 265–75, 1997.
- [15] 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

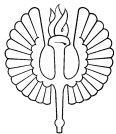
- and multiple board types,” *European Journal of Operational Research*, vol. 98, no. 3, pp. 457–72, 1997.
- [16] T. Johtela, J. Smed, M. Johnsson, R. Lehtinen, and O. Nevalainen, “Supporting production planning by production process simulation,” *Computer Integrated Manufacturing Systems*, vol. 10, no. 3, pp. 193–203, 1997.
- [17] K. R. Kumar and T. T. Narendran, “A heuristic for sequencing PCBs with due-dates,” *International Journal of Operations & Production Management*, vol. 17, no. 5, pp. 446–67, 1997.
- [18] M. S. Hillier and M. L. Brandeau, “Optimal component assignment and product grouping in printed circuit board manufacturing,” *Operations Research*, vol. 46, pp. 675–89, 1998.
- [19] H. O. Günther, M. Gronalt, and R. Zeller, “Job sequencing and component set-up on a surface mount placement machine,” *Production Planning & Control*, vol. 9, no. 2, pp. 201–11, 1998.
- [20] J. Smed, M. Johnsson, M. Puranen, T. Leipälä, and O. Nevalainen, “Job grouping in surface mounted component printing.” Forthcoming in *Robotics and Computer Integrated Manufacturing*, 1999.
- [21] S. S. Heragu, “Group technology and cellular manufacturing,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 2, pp. 203–15, 1994.
- [22] T. Johtela, J. Smed, M. Johnsson, and O. Nevalainen, “Fuzzy approach for modeling multiple criteria in the job grouping problem,” Tech. Rep. 227, Turku Centre for Computer Science, Dec. 1998.
- [23] T. Johtela, J. Smed, and M. Johnsson, “ControlBOARD user manual,” Tech. Rep. M-98-2, University of Turku, Computer Science, 1998. In Finnish.
- [24] T. L. Saaty, *Analytic Hierarchy Process*. McGraw-Hill, 1980.
- [25] T. Leipälä and O. Nevalainen, “Optimization of the movements of a component placement machine,” *European Journal of Operational Research*, vol. 38, pp. 167–77, 1989.
- [26] A. Maria, “Introduction to modeling and simulation,” in *Proceedings of the 1997 Winter Simulation Conference* (S. Andradóttir, K. J. Healy,

D. H. Withers, and B. L. Nelson, eds.), (Atlanta, Georgia), pp. 7–13,  
Dec. 1997.



**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**