Proceedings of the 5th WSEAS Int. Conf. on Signal Processing, Robotics and Automation, Madrid, Spain, February 15-17, 2006 (pp201-206)

Optimized Manufacturing Processes in Multimodel PCB Assembly

Kari Salonen[1] *, Timo Knuutila[1], Mika Johnsson[2], and Olli S. Nevalainen[1]
[1]Department of Information Technology and Turku Centre for Computer Science (TUCS),
University of Turku, FIN-20014 Turku, Finland
[2]Valor, Finland (R&D,Assembly Products) , Ruukinkatu 2, 20540 Turku, Finland

Abstract: Component assembly on bare printed circuit boards (PCBs) is performed by automated placement machines. The main time factors of this process include the actual component placement time and the setup time of the machine when changing the PCB type. An optimization method for minimizing the total manufacturing time for a collection of different types of PCBs is given. The method is a hybridization of the unique and group setup methods. Instead of using coarse low level abstraction for the machine operations and time factors, a state-of-the-art optimizer is applied here.

Key-Words: Printed Circuit Boards, Component assembly, Production control

1 Introduction

The present study concentrates on two short term production problems in the printed circuit board assembly, the selection of the component feeder setup strategy and the optimal machine control for performing the actual component placements. This is why, the manufacturing process with PCB component placement machines, on a very coarse level, can be seen as a sequence of cycles each consisting of a manual *(machine) setup phase* and an automated *(component) placement phase*. The manual setup phase depends on the decisions made in *product scheduling* which specifies the order of processing the PCB jobs with the machine. The time used for the placements depends on the *component-to-feeder* storage mapping and on the *placement sequencing* decisions which specify the order of component pickups and placements [1].

Each PCB job consists of a batch of PCBs of the same type and the types of two different jobs may differ. While the technical details of placement machines vary significantly, they all accept a limited number of component types in their feeders simultaneously. This necessities the manual feeder setup operations between jobs or job groups. The component feeders are normally organized as a linear array on a moving or stationary feeder bank (feeder storage) on the side (or sides) of the machine and one can choose a component-to-feeder storage mapping which most efficiently supports fast component insertions.

Machine setup can be organized in several alternative ways [2]. This work deals with two of these: *Unique setup* optimizes the feeder allocation for each PCB type individually by minimizing the component placement time of each PCB type separately [3]. *Group setup* forms a minimal number of PCB groups in such a way that feeder changes (setup occasions) are needed between PCB groups, only [4, 7].

In this paper we concentrate on the case of a single machine (bottleneck of the production line) and multiple PCB types, and suppose that the batch sizes of the PCB jobs vary significantly from batch to batch. This kind of situation was met by our group in a company which had a number of standard products with high demand as well as other PCB types with low demand or even testing of single prototypes. A natural idea is then to hybridize the grouping technique and the unique setup technique. Given a set of PCB jobs our task is therefore to solve which PCBs should be grouped together and which would benefit from optimized unique settings of the feeders. The objective is to get the minimal production time when considering the time for manual setups and automated component placements.

We suppose that each PCB assembly task is defined in its full detail in a CAD-file. This acts as an input for a production planner system (Valor Trilogy - Line Engineering [http://www.valor.com/]) which performs job grouping and generates an optimized machine control program for one or several PCB

*Corresponding author: kari.salonen@it.utu.fi, FAX: +358-2-333 8600

jobs. This includes the determination of the component-to-feeder storage mapping and the control program for the actual component placements. Unlike most research on setup strategies, we are thus working on a level of full details of a particular chip shooter machine (HSP4791).

The design of an efficient component-to-feeder storage mapping of a PCB group (*i.e.* a set of PCB jobs) is a hard task [2, 6]. The problem is, that the same mapping should support the component placements of all the different PCB types of the group and the component sets of different PCB types are not disjoint. Ohno [5] derived a mathematical formulation and heuristics for the joint problem of grouping and machine control optimization along with the basic time factors of a rotary turret machine.

2 Problem statement

Consider a particular PCB assembly line with a chip shooter as its bottleneck machine. The line includes in addition to that at least a paste/adhesive printing machine and an oven connected by a conveyor belt to the actual placement machine. Various other special or high precision insertion robots and/or other chip shooters may also be situated on the same line.

We suppose that the placement machine is of the rotary turret type with a movable (in x-y directions) fixation table for PCBs, a movable (x-direction) feeder rack (feeder unit, or feeder bank) and a multiheaded pick-and-placement carousel rotating around a fixed axis. The feeder rack is organized as a linear array of feeder slots and components are stored there using component reels (or component feeder) each occupying a certain number of feeder slots (typically 2 to 3) from the capacity Fr_{tot} (typically from 40 to 200 slots). The number of different component types that can be stored in the feeder rack is thus limited while one can consider as unlimited the capacity of a component type which is stored in the feeder rack (due to the possibility of renewing its supply).

A short term planning perspective for production planning and control of a few days is considered in this study. We then have a set of N PCB assembly jobs available at the beginning of the planning period and due dates are not considered. Each job j is formed of a batch of PCBs of the same type and the task is to insert a set of component types $Com(j)$ on each bare PCB. It is supposed that the feeder rack capacity demand Fr_j for $Com(j)$ is less than or equal to the total feeder rack capacity Fr_{tot} but $\sum_{j=1}^{N} Fr_j > Fr_{tot}$, *i.e.* the components of all jobs do not fit the feeder rack simultaneously. In addition to the above data, we suppose that the component-nozzle type assignment, placement speed definitions (*i.e.* is the component ”slow” or ”fast”), precedence constraints for component placements, fiducial marks, component placement (x,y)-coordinates and orientations along with other data, necessary for proper generation of an optimized component-to-feeder assignment and machine control program, are all given as data set $D_{job}(j)$. This extensive set of data is needed in order to make the application of an optimization software system possible. One should recognize here that the feeder rack capacity demands and placement positions form only one part of the rather involved input of the software but we can omit their closer description here.

All jobs are available at the beginning of the planning period and due dates are not considered. Similarly, we omit some other constraints, like conveyor belt width or oven temperature, from the consideration.

As to the costs of manufacturing, the following time components are observed:

The constant time A for performing a machine setup operation. This time includes the manual preparative operations for performing one or more component setups of the feeder rack.

The estimated time B for changing one component reel in the feeder rack.

The manufacturing time $Tpla_j$ for a PCB of type j. This time includes a fixed initialization time and the time for placing all the necessary components of PCB type j. The time depends on the component-to-feeder rack assignment and the machine control program of job j.

The initialization time $Tinit$ for beginning a new job (*i.e.* a new PCB type). The time includes preparative steps, like change of the machine control program.

The number n_j of PCBs in job j.

Let us denote by $nocc_N$ the number of setup occasions for the whole set of N jobs and by nfc_j the number of component tape changes for job j.

The total processing time of the N jobs is then

$$(1) \quad Tot = nocc_N * A + B * \sum_{j=1}^{N} nfc_j + N * Tinit + \sum_{j=1}^{N} Tpla_j * n_j$$

There are three factors in the above expression that one can control. (i) By selecting different ways to partition the N jobs one can have different numbers of job groups. As an extreme, by unique setup, $nocc_N = N$ and as an other extreme, one can minimize its value, like is done in pure job grouping. (ii) The number of different component tapes in the job groups depends on the selection of the jobs to each group i. (iii) The actual placement time $Tpla_j$ of a job (j) depends strongly on the other jobs in the same group to which job j belongs.

The task is to determine a job grouping G^*, component-feeder assignment CFA^* and machine control programs MCP^* for which

$$(2) \quad Tot(G^*, CFA^*, MCP^*) = min!$$

over all feasible selections of G, CFA, MCP.

3 Solution method

A simple greedy heuristic is designed for minimizing the *Tot* time. The algorithm starts from a solution where the unique setup is applied for all jobs, $j = 1, 2, ..., N$. It then considers all feasible pairs of the jobs and calculates for each pair the value of *Tot* (taking into account both the batch size and the number of their common components). A pair of jobs forms a feasible group, if their components fit in the feeder rack. Among these groups that one is accepted which brings the largest saving for *Tot* provided that a decrease of the value is found. The same process continues for the remaining jobs and job groups.

Algorithm *GreedyTot* accepts as its input the machine specifications of a placement machine including among others the feeder rack capacity *FrTot* and the definitions of the N PCB jobs as given by $D_{job}(j)$ ($j = 1$ to N). In addition, the parameters A and B giving the times for setup occasions and component tape changes are given. After that, the solution of the greedy heuristic is given as an input for the simulation program of the Valor Machine Engineering software package that calculates the placement time for each group ("level 2" optimization). The programs of the package perform joint optimization of the component-feeder assignment and control program generation. One should note that the simulation program takes into consideration both the number of PCBs of each type and the joint insertion point (x,y)-closeness of component pairs on PCBs.

Algorithm GreedyTot:
Form N singular job groups $g_1, ..., g_N$ of
  the original jobs j, $j = 1$ to N;
Define an initial grouping $G = \{g_1, g_2, ..., g_N\}$;
$G' \leftarrow G$; Let *changed* ← **true**;
while *changed* **do**
  $G^* \leftarrow G'$; *changed* ← **false**;
  for each pair $g_i, g_j \in G^*$ **do**
    Form group g_e by merging g_i and g_j;
    Let $G'' \leftarrow (G^* \setminus \{g_i, g_j\}) \cup \{g_e\}$
    if $Tot(G'') < Tot(G')$ **and** G'' is feasible **then**
      $G' \leftarrow G''$;
      changed ← **true**;
Let $G \leftarrow G'$.

This algorithm performs the best group merge on each iteration round. The algorithm terminates when no improving merge is found or all merged solutions are infeasible. Observe that the calculation of $Tot(G)$ is a heavy operation involving determination of the component-feeder assignment and control program optimization.

4 Experimental results

In the first test we used it with a sample of 14 PCB jobs from the production program of a manufacturer, see Table 1. The algorithm *GreedyTot* was compared to the unique setup method (*Uniq*) and grouping method (*Group*). The grouping algorithm has been observed to give optimal or near optimal groupings in tests with problems of realistic size. For *Uniq* and *Group* we solved *Tot* in the same manner as *GreedyTot*.

The proper setting of A and B values depends on the particular production case and we therefore tested with several different settings of these; *A*=0 and 300 (sec) and *B*=30, 60, 90,and 120 (sec).

While the component-feeder rack assignments and the construction of the machine control program were determined by the production planner of Valor [http://www.valor.com], the actual placement times $Tpla_j$ were evaluated by the chip shooter simulator of the same system.

Figure 1 compares *GreedyTot*, *Group* and *Uniq* for different settings of *A* and *B*. For *A*=0 the new method has a benefit over the two other methods for all *B*-values tested here. However, for a large *B* (120) the grouping method is practically of the same level. For *A* = 300, the large time for setup occasions forces *GreedyTot* to decrease the number of these occasions and therefore (for large *B*) our new method acts as a minimizer for grouping, see Fig.1b.

Table 2 shows the placement times of the different PCB types (per single PCB) for different production planning methods. The solutions of the three planning methods are demonstrated in Table 3 by giving the result of the grouping for each job. From the two last columns of the table we can see the partial grouping made by *GreedyTot* (*e.g.* group ”7” includes job 7 and 9 and 14 while the group -algorithm adds to the same group also job 8). The effect of grouping is visible in the average assembly time of individual PCB types.

In the second test we used the same set of jobs as in Case 1 but the quantities of the PCBs of each type were dropped to 1. It is natural that grouping is then the best choice(data not shown). *GreedyTot* finds in this case the same grouping solutions as *Group*, but this needs not to be the case due to the more advanced grouping technique applied by *Group*. More important is, however, the observation that also *GreedyTot* strives to group the jobs now.

5 Concluding remarks

Partial grouping and optimization of the component-feeder rack assignment turns out to be an advantageous method in short range production planning and control. Here, the best properties of grouping and unique setup methods are utilized in a flexible manner.

Our study was based on the use of an existing commercial planning system. The system is nowadays widely used in PCB assembly companies. Our tests are therefore realistic. In addition, the results of the optimizer were evaluated by an existing machine simulator which represents the state of the art in the field. In this sense the method should give clear guidelines for production designers. Our discussion still leaves a few points open and one should here consider the production situation carefully. These include:

1. The time penalties *A* and *B* were left as operator selected parameters. When applying the methodology of the present paper, factory-specific values of these should be applied. Some producers use movable feeder rack units which allow parallel setup of component reels. This can be easily modelled by setting *B* to 0 in our model.

2. Auxiliary constraints including aspects like oven temperature, release dates, due dates or conveyor belt widths were bypassed. These might, of course, be included in the main control of *GreedyTot* at the expense of greater complexity of the method.

References

[1] Crama, Y., Oerlemans, A.G., & Spieksma, F.C.R., Production Planning in Automated Manufacturing, *2nd ed. Springer, Berlin*, 1996.

[2] Leon, V.J., & Peters, A., A Comparison of Setup Strategies for Printed Circuit Board Assembly, *Computers & Industrial Engineering*, Vol.34, No.1, 1998, pp.219–234.

[3] Magyar, G., Johnsson, M., & Nevalainen, O., On Solving Single Machine Optimization Problems in Electronics Assembly, *Journal of Electronics Manufacturing*, Vol.9, No.4, 1999, pp.249–267.

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

[5] Ohno, K., Jin, Z., & Elmaghraby S.E., An optimal assembly mode of multi-type printed circuit boards, *Computers & Industrial Engineering*, Vol.36, 1999, pp.451–471.

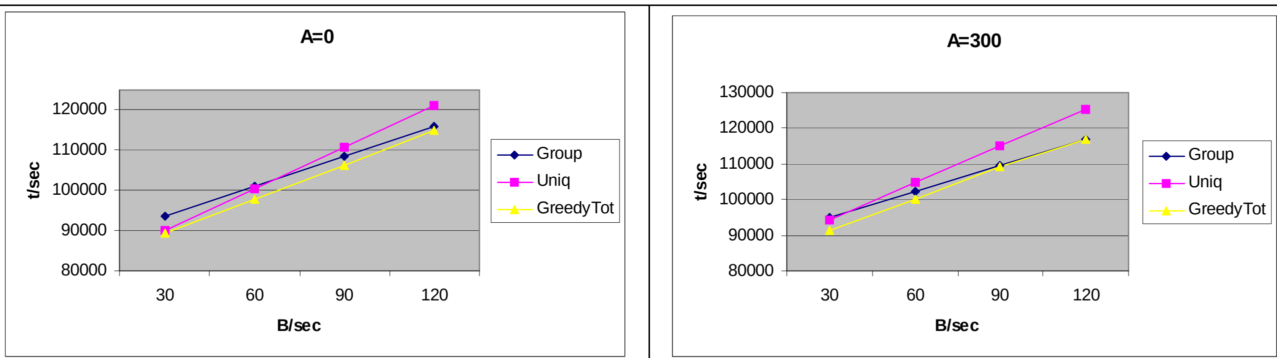
[6] Klomp, C., Klundert, J., Spieksma, F.C.R., & Voogt, S., The feeder rack assignment problem in PCB assembly: A case study, *International Journal of Production Economics*, Vol.64, 2000, pp.399–407.

[7] Knuutila, T., Puranen, M., Johnsson, M., & Nevalainen, O., Three Perspectives for Solving the Job Grouping Problem, *International Journal of Production Research*, Vol.39, No.18, 2001, pp.4261–4280.

[8] Smed, J., Johnsson, M., Puranen, M., Leipälä, T, & Nevalainen, O., Job grouping in surface mounted component printing, *Robotics and Computer-Integrated Manufacturing*, Vol.15, 2005, pp.39–49.

Table 1: Test case 1 with 14 PCB jobs. The table shows for each job the batch size (quantity), the number of different components and the total number of components on a PCB.

Job	Quantity	Numb. of diff. comp.	Total numb. of comp.
1	100	4	84
2	5	11	194
3	7	5	166
4	350	38	83
5	1000	24	38
6	2	23	35
7	1	33	316
8	500	12	97
9	100	27	181
10	10	54	199
11	1000	4	38
12	200	55	705
13	4	14	486
14	25	38	537



a) A=0 b) A=300

Figure 1: Comparison of the (*Tot*) total production times of the jobs in Table 1 for different values of *B*. Moderate production volumes.

Table 2: Placement times (sec) for different jobs (case 1) where $A = 0$ and $B = 30$. The times have been estimated by the chip shooter simulator of Valor for a SMT machine of the type HSP4791.

Job	Uniq	Group	GreedyTot
1	22,02	21,79	22,02
2	38,44	41,98	40,70
3	32,34	33,39	32,41
4	21,80	25,96	21,80
5	15,49	19,32	16,10
6	15,21	19,46	16,30
7	54,07	59,96	64,05
8	23,34	26,04	23,34
9	33,78	36,42	35,01
10	36,94	37,06	38,73
11	15,19	16,45	15,19
12	105,47	107,98	105,47
13	72,83	74,81	72,83
14	84,44	89,58	87,66

Table 3: Solutions (case 1) where $A = 0$ and $B = 30$. The total number of feeders of each group is given on the first row where the group number occurs.

	Uniq		Group		GreedyTot	
Job	*Group no*	*Numb. of feeders*	*Group no*	*Numb. of feeders*	*Group no*	*Numb. of feeders*
1	1	4	1	57	1	4
2	2	11	2	59	2	63
3	3	5	2		2	
4	4	38	2		4	38
5	5	24	2		5	28
6	6	23	2		5	
7	7	33	3	69	7	68
8	8	12	3		8	12
9	9	27	3		7	
10	10	54	1		2	
11	11	4	2		11	4
12	12	55	4	61	12	55
13	13	14	4		13	14
14	14	38	3		7	
Total	**14**	**342**	**4**	**246**	**9**	**286**