

基板生産時における部品装着機の部品配置問題および部品吸着順序問題のモデル化と解法

山田 剛史[†], 宮代 隆平[†], 中森 眞理雄[†],

[†] 東京農工大学大学院工学教育部

基板生産時において部品装着機を用いた工程におけるスケジューリングは基板生産時間に大きな影響を与える。本論文では部品装着機における問題である部品配置問題および部品吸着順序問題に対してヒューリスティックなアルゴリズムを提案する。評価実験において厳密解に到達する結果を得ることができた。

An Algorithm of Feeder Arrangements and Pickup Sequencing of Component Placement Machine of Printed Circuit Board

Tsuyoshi YAMADA[†], Ryuhei MIYASHIRO[†], Mario NAKAMORI[†],

[†] Faculty of Engineering, Tokyo A&T University

It is important that a production schedule considerably affects the production time of a printed circuit board in component placement machines. In this paper, we proposed algorithms that are based on local search heuristic. We show that the proposed algorithm gives satisfactory results.

1 Introduction

As the need of computer and electronic devices grow, printed circuit board (PCB) manufacturing depends more on automated production processes. One of the most important is PCB assembly. In order to get high rates of throughput of PCB assembly machines, we have to produce high quality of instruction sequence. Among many stages of PCB assembly, that of component placement is the most critical.

There are a wide variety of types of component placement machines. In the present paper we consider *Multi-Head Placement Machine* (see Figure 1), which is capable of picking several components from different feeder slots simultaneously.

The multi-head placement machine is characterized by the moving placement arm with several heads (possibly, 10), each of which can perform vacuum pickup/placement of a component by a nozzle. The components to be placed on the circuit boards are stored in tape reels set in feeder slots. The placement heads first pick up components, either simultaneously (gang pick) or by moving along the feeder array, and next move to

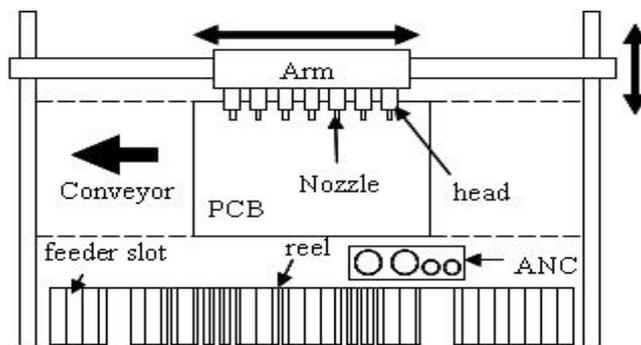


Figure 1: multi-head placement machine

the circuit board to place the components at their appropriate locations.

There are, however, differences of sizes of components, so not all components can be placed by the same type of nozzles, which requires change of nozzles. Nozzle changes are carried out automatically by *Automatic Nozzle Changer (ANC)* throughout the process of component placement. Note that nozzle changes are time consuming and have to be avoided unless it is necessary.

Thus, the process of component placement con-

head	H1	H2	H3	H4
nozzle	A	A	C	B
pick num	2	1	1	2
component	a3	a2	c1	b1

↓ next task

head	H1	H2	H3	H4
nozzle	A	A	C	B
pick num	1	0	0	1
component	a1	-	-	b2

↓ next task (go to ANC)

head	H1	H2	H3	H4
nozzle	D	D	C	E
pick num	5	6	3	4
component	d1	d1	c2	e4

Figure 2: *pickup number*

sists of three operations:

- Placing components onto their locations in the PCB by nozzles.
- Picking up components from reels of feeder slot by nozzles.
- Changing nozzles in the heads at ANC.

These three operations are performed alternately. We call a sequence of a picking operation and the consecutive placing operations a *task*. Considering the production time, we have to minimize the number of tasks. In order to reduce the number of tasks it is necessary to make the number of use of nozzles well-balanced (see Figure 2). *Pickup number*, which is the number of picking up components, is assigned to each node. When pickup number of all nozzles on the arm is zero, the nozzles are exchanged with others in ANC, where the nozzle arrangement also may take place. The optimization of the pickup number is important to reduce the number of tasks and nozzle changes. In the present, however, pickup number is fixed and constant.

When the placement of all components of the ARM in a certain task is finished, the ARM moves to pick the components of the next task from the feeder slots. The circuit board is completed when all the tasks are finished.

There are several factors that strongly affect the time of component placement. Among them, the most important are *Feeder Arrangement Problem (FAP)* and *Pick Up Sequencing Problem (PUSP)*. Both FAP and PUSP are NP-hard, and there have been few literatures on the mathematical formulation of them. However FAP

and PUSP is formulated as 0-1 integer programming problems, the number of variables is 2640 and the number of constraints is 3275 in small PCB (the number of components is 11, the number of types of nozzles is 3, the number of components per type 1). In this paper we propose heuristic algorithms for them with a new construction method of our own for making initial solution. Also, we report the results of our implementation of FAP and PUSP based on VNS and Tabu search.

2 Problem Definition

In the multi-head component placement machines, problems to be solved are as follows:

- (1) *placement sequencing problem*(PSP);
- (2) *feeder arrangement problem*(FAP);
- (3) *pick up sequencing problem*(PUSP).

PSP is to determine the order in which picked components with the heads of each task are placed onto their positions at the PCB. The placement sequence which achieves the minimum placement time should be determined solving the PSP. As suggested in most papers, PSP can be modeled as the *Traveling Salesman Problem*, which is well known to be NP-hard ¹⁾ ²⁾. The cost in the PSP is moving-time of arm on a PCB.

FAP is to determine the allocation of the reels of the feeder slots. PUSP is to determine the order of picking up components from the reels. A feeder arrangement and pick up sequencing that gives the minimum picking time should be calculated in solving FAP and PUSP. The cost in FAP and PUSP is the number of picking up all components. The number of picking up all components will be reduced by *gang-pick* (see Figure 3). In a gang-pick, several components are picked up from neighboring reels simultaneously. If conditions (matching nozzles and component types, the position of arm) are satisfied, it is possible to pick the components simultaneously. In order to satisfy these conditions, nozzle arrangement on arm and *pickup number* of each nozzle are also important. However, they are fixed information before PUSP and FAP. Gang-pick in FAP and PUSP is significant factor in reducing the number of picking all components.

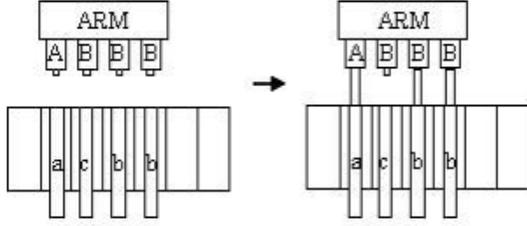


Figure 3: *gang-pick*

In this paper, we propose heuristic algorithms for FAP and PUSP.

3 Algorithm

In this section, we propose algorithms for FAP and PUSP. The algorithm that we propose consists of two phases, constructing an initial solution and local search.

3.1 Initial Solution

Since it is difficult to solve FAP and PUSP at the same time, we solve first FAP and next PUSP.

In order to reduce the number of picking up the components, we have to make simultaneous picking up to be frequent. When the nozzle arrangement on arm is similar to the feeder arrangement, picking up component simultaneously tends to take place. The construction method that we propose for FAP is to construct the feeder arrangement so that it is similar to the nozzle arrangement on arm. For PUSP, we employ the greedy method in Sun⁴⁾. The greedy method that are proposed by them for PUSP scans the reel set from the first feeder slot to the last feeder slot by the arm. At each scanning position i , the maximal set of components that can be simultaneously picked up by the several heads is identified. It calculates such a set in each scanning position i . After the scanning is completed, the largest set is selected in the picking of a certain task. After excluding the heads assigned for picking up the components, repeat the scanning procedure using the remaining heads. Repeat such scanning procedure using the remaining heads until there is no unoccupied head. When there is no unoccupied head in the all tasks, the operation terminates. The

construction of FAP is as follows:

```

Procedure construction of the FAP
(declaration)
number of heads is  $J$ 
pickup number is  $N(j)$  of head  $j$ 
number of component  $i$  is  $C(i)$ 
component is  $F(t)$  in the feeder slot place  $t$ 
nozzle type is  $Nz(j)$  of head  $j$ 
nozzle type is  $Nc(i)$  that adapt  $i$ 
1. (Initialize)
base point is  $bp = 0$ 
2. (Assignments 1)
While ( $N(\exists j) > 0$ ) {
  for( $j=0; j < J; j++$ ) {
    If ( $s$  is a  $max(C(\forall i)) \ \&\& \ Nz(j) == Nc(s)$ )
      Assign  $s$  to  $F(j+bp)$ 
  }

  for( $j=0; j < J; j++$ ) {
     $N(j) = N(j) - C(F(j+bp))$ 
  }
   $bp = bp + J$ 
If ( $\forall i$  are assigned to feeder)
  Output the constructive solution
  break
Else ( $N(\forall j) < 0$ )
  Change  $Nz(\forall j)$  in ANC
}
3. (Assignments 2)
If ( $\forall i$  have not been assigned yet)
  The components assigned to feeder in descending
  order of the number of components

```

3.2 The Neighborhood

In this subsection, We propose *swap-item* for a local search heuristic for FAP and PUSP. The 2swap-item and the 3swap-item operator attempts to swap two reels or three reels from feeder slot in the feeder arrangement. The new solution will be accepted if the new cost is smaller than the original one. However, the feeder arrangement by such swapping is not consistent with pickup sequence. Therefore, it is necessary to reconstruct the pickup sequencing. In order to calculate the cost, operator reconstruct the pickup sequencing every time reels are swapped from feeder slot in the feeder arrangement.

3.3 Meta-Heuristics

A drawback of the local search heuristic is that it

considers only a small part of the overall problem. Therefore it may get trapped in a local optimum of poor quality. In order to overcome such a poor local optimum we implement a Variable Neighborhood Search(VNS) and Tabu Search heuristic.

The operation of VNS is to change neighborhood adaptively to a local optimal solution³⁾. By changing neighborhood (changing to a larger neighborhood), improved solutions will be found in the neighborhood.

The Tabu Search is known to breakout a local optimal solution. The operation of this meta-heuristic is to make the original solution worse and to start the searching from a peyorated solution. However, the cycling of the solution arise in this condition. In order to overcome the endless cycle of solution, the Tabu Search determines a tabu list to record solutions already found. The Tabu Search that we implemented uses 2swap-item in neighborhood and terminate when the number of moving in neighborhood(iteration count) is M .

4 Experimental Result

In order to evaluate performance of our algorithms, we implemented our algorithms and made experiments on randomly generated data of printed circuit boards.

The details of data are as follows:

$8 \leq$ number of component types ≤ 12

$1 \leq$ number of components per type ≤ 8

number of nozzles per head = 5

$3 \leq$ number of types of nozzles ≤ 5

number of instances of data set = 30 .

M of tabu search in our experiments is 100. The implementation of the algorithms was performed on a Pentium Processer 1GHz and 512KBM.

The experimental result is shown in Table 1. We calculated the optimal solution in ILOG CPLEX7.1. This table shows the avarage of gap between the optimal value and the value in other methods.

Table 1 shows that Tabu Search algorithm is better than other algorithms in all data and gives optimal solution in 5 data.

Table 1: Experimental Result

method	Gap				
	Opt	LS(2)	LS(3)	VNS	TS(2)
Avarage	0.0%	5.0%	4.0%	3.6%	2.8%

5 Conclusion

In this paper, we have discussed the optimization of a multi-head placement machine. In particular, we considered that the feeder arrangement problem and pickup sequencing problem. After formulating the problems, we proposed algorithms that solve these problems by heuristic methods. Algorithms are based on local search techniques in order to achieve fast.

In experiments, Tabu Search algorithm achieved near optimal solution in small data. Also, Tabu Search algorithm achieved solutions better than other algorithms.

Optimizing other factors involved in manufacturing PCBs, especially, placement sequencing, are left for further research.

References

- 1) E.K.Burke, P.I.Cowling, and R.Keuthen. New Models and heuristics for Component Placement in Printed Circuit Board Assembly. *Proc.IEEE Int.Conf. Information, Intelligenece and Sysmes (ICIIS)*, pp.133-140, 1999.
- 2) E.K.Burke, P.I.Cowling, and R.Keuthen. Effective Heuristic and Metaheuristic Approaches to Optimize Component Placement in Printed Circuit Board Assembly. *Proc.Conf. on Evolutionary Computing (CEC2000)*, pp.301-308, 2000.
- 3) N.Mladenovic and P.Hansen. Variable neighborhood decomposition search. *Computers Oper.Res.*, 24, pp.1097-1100,1997.
- 4) Dong-Seok Sun, Tae-Eog Lee, and Kyung-Hoon Kim. Component allocation and feeder arrangement for a dual-gantry multi-head surface mounting placement tool. *International Journal of Production Economics*, 95, 2, pp.245-264,2005.