Scheduling Optimization of Component Mounting in Printed Circuit Board Assembly by Prioritizing Simultaneous Pickup

Toru Tsuchiya, Atsushi Yamashita, Toru Kaneko, Yasuhiro Kaneko, and Hirokatsu Muramatsu

***Abstract*— In this paper, we propose a new method for reducing assembly time in printed circuit board (PCB) assembly by prioritizing efficient simultaneous pickup operation of placement machines. Despite using the same placement machine, the efficiency of the schedule results in major different assembly time. Therefore, it is important to optimize the scheduling of component mounting. There are three major problems of the scheduling: (1) component feeder location (affects efficiency of pickup operation), (2) mounting sequencing (affects total distance of the mounting tour) , and (3) simultaneous pickup (affects efficiency of pickup operation). To solve these problems, this paper proposes the following approaches. We solve (1) and (3) in a heuristic way by using a random multi-start local search. We solve (2) greedily with putting the result of the feeder array to effective use. The effectiveness of the proposed method was shown through simulations.**

I. Introduction

In this paper, we propose a new method for reducing assembly time in printed circuit board (PCB) assembly by prioritizing efficient simultaneous pickup operation of placement machines.

The market in electronics devices is rapidly growing with each passing year, surface-mount technology (SMT) has been advanced. In the SMT, placement machines are used for assembling the PCB. The placement machines mount surface-mounted devices (SMDs) on the PCB which has solder pads where the SMDs are mounted.

By recent advancement of the SMT, PCBs have been able to contain smaller and more components, and sophisticated machines meet the demand of mounting the components quickly and precisely. However, they also cause high complexity of the scheduling of component mounting. Despite using same placement machine, the efficiency of the schedule results in major different assembly time. Therefore, it is important to optimize the scheduling of component mounting.

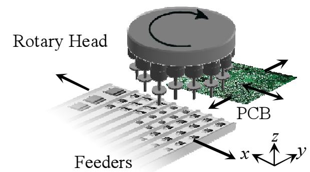
Every placement machine mainly consists of the following elements:

- A component feeder, which contains components,
- A feeder slot, where the feeder is set,
- A nozzle, which picks up the components and mounts them on a PCB,
- A worktable, which holds the PCB.

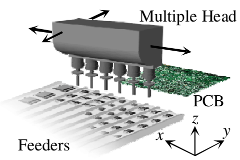
T. Tsuchiya, A. Yamashita, and T. Kaneko are with the Department of Mechanical Engineering, Faculty of Engineering, Shizuoka University, 3–5–1 Johoku, Naka–ku, Hamamatsu–shi, Shizuoka 432–8561, Japan (e-mail: `f0730066@ipc.shizuoka.ac.jp`).

A. Yamashita is now with Department of Mechanical Engineering, California Institute of Technology, Pasadena, CA 91125, USA.

Y. Kaneko and H. Muramatsu are with i-PULSE CO., LTD., 1–9–3 Shinmiyakoda, Kita–ku, Hamamatsu–shi, Shizuoka 431–2103, Japan.



(a) Rotary type.



(b) Non-rotary type.

Fig. 1. An example of the two different types of the placement machines.

Although there are a variety of types of the placement machines, they can be roughly classified into two groups, a rotary type (also known as a revolver type) and a non-rotary type (see also [1] for more detailed grouping) (Fig. 1).

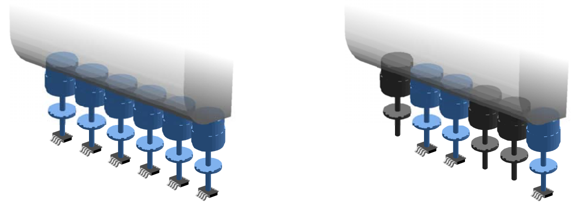
The rotary placement machines have a rotary head to pick up and mount the components simultaneously. While the head is fixed on the placement machine and rotates at the same position, the feeder array and the worktable move so that the head can pick up and mount the components only with its rotation. This type has the advantage of being able to pick up and place the components very speedily, and has been used from the early period on the SMT.

The production planning problems in PCB assembly are surveyed and classified into eight sub problems by [2]. Many of them are NP-hard and it takes high computational cost to solve them. For this reason, conventional works produced a variety of heuristic methods such as [3], [4].

On the other hand, the non-rotary placement machines have a single or several heads and the head(s) can move freely along the X and Y axes depending on a machine model.

This type also has almost the same problems as [2]. The machines that have the movable feeder array and worktable are targeted by [5], [6]. However, this type of machines has many movable elements, then having a large size unit is inescapable. Recent years, machines that have fixed feeders and a work table are dominating in the benefit of its space-saving and cheaper cost. Those machines which have a single head are targeted by [7]. Recently, machines which have multiple heads (capable for the simultaneous pickup) are getting popular.

An example of the simultaneous pickup is shown in Fig. 2. When the head approaches a certain position above the feeder, it can pick up not only one component, but also multiple components from the feeder array as many as the other nozzles located above a feeder. This means that the number of pickup times can be minimized to*the quantity of the components divided by the number of heads*.



(a) Picking up of six components simultaneously.

(b) Picking up of three components simultaneously.

Fig. 2. An example of the simultaneous pickup.

For those machines, [8]–[11] proposed optimization methods. However, relatively few studies consider simultaneous pickup which reduces the number of picking up all the components and then reduces the assembly time. Simultaneous pickup execution depends on some conditions, e.g., the positions of heads, the feeder array, and so on. Especially the feeder array is important for simultaneous pickup.

Therefore, we focus on the simultaneous pickup, and propose a method that can optimize a suitable feeder array for simultaneous pickup in order to minimize the assembly time. Because this problem is NP-hard, we solve it in a heuristic way with a random multi-start local search (MLS) which is a simple and robust algorithm. In addition, we consider the case that the feeder pitch and the head pitch are not equal. This case complexifies the problem structure, and no significant studies are provided to solve it.

This paper consists of seven sections as detailed below. In Section II, the placement machine is explained, and the scheduling model of our target is described in Section III. In Section IV, the problem is described and Section V discusses the method of optimization of the placement machine. Section VI shows the computational experiments and results, and Section VII describes conclusions and future works.

II. Placement Machine

Here, we describe more details of the placement machine which has a fixed feeder array, a worktable, and a movable multiple head. The placement machine currently we use in this paper is shown in Fig. 3.

One type of the components is lined up on a tape and wound on a reel. The reel is stored in the feeder so that the components can be served one by one to the nozzle. The feeder has several sizes depending on the tape width. It is installed in a feeder slot, and usually occupies one slot. However, wide feeders occupy more than one feeder slot. Therefore, it is necessary to install the feeders appropriately into the slots without feeder overlapping. The feeder slots are obtained at fixed feeder bank (some placement machines have several feeder banks subjecting to a machine model).

After installation of the feeder, the assembly of the PCB starts. The multiple head that is mounted on X-Y beams has nozzles on each head. The nozzle can pick up and mount the components. The assembly is simply described in the following steps:

1) The head moves to a certain position above the feeder array.
2) The head picks up the components simultaneously as many as possible and iterates from 1) until each nozzle has a component.
3) The head moves to the PCB and places the components one by one on a certain location depending on the component type mounted. After all mounting of the components which are picked up in step 2), the head iterates from 1) until all the components are mounted on the PCB.

In addition, there are some additional concerns. There are several types of nozzles, because each nozzle can pick up its own limited types of components. Appropriate nozzle change is needed to pick up many different types of the components.

III. Scheduling Model

Since there are a variety of the placement machines, it is necessary to define the target model in order to evaluate a substantial assembly time. In this paper, we model the placement machine which has a multiple head (it consists of several heads), a fixed worktable and a single feeder bank as shown in Fig. 4 (this type of placement machine is classified into a multi-head placement machine according to [1]).

To solve the problem, we set the following assumptions:

- The nozzle can pick up any type of components, so that the nozzle changing is unnecessary. (Although,

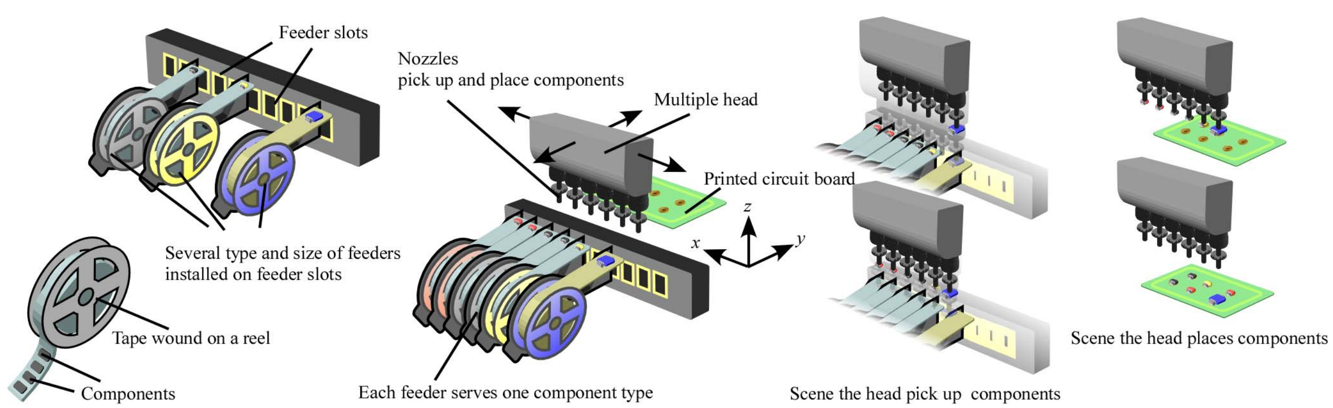


Fig. 3. The structure of the placement machine.

we explained that it is nessesary to change nozzles appropriately in Section II. We simplify this problem with this assumption.)

- The head can move everywhere without any regard for mechanical restrictions of the movable X-Y beams.
- The components have lower height, so the nozzle does not hit the mounted components on the PCB during the assembling.

The following notations are used in this paper:

$p_{\max}$: the total number of the placement points,
t : the type of the components $(t = 1, \ldots, t_{\max})$,
j_t : the number of the components of type t $(j_t = 1, \ldots, j_{\max_t})$,
$C_{t,j_t}(x,y)$: the j_t-th placement coordinate in the components of type t,
f : the number of the feeder $(f = 1, \ldots, f_{\max})$,
s : the number of the feeder slot $(s = 1, \ldots, s_{\max})$,
S_f : the feeder slot number of the feeder f,
T_f : the component type of the feeder f,
Q_f : the component quantity of the feeder f,
r_f : the right half of the width of the feeder f,
l_f : the left half of the width of the feeder f,
D_h : the head pitch,
D_f : the feeder pitch,
h : the number of the head $(h = 1, \ldots, h_{\max})$.

IV. Description of Problem

Many different types of placement machines have common problems which are described by [2]. According to [2], our model is classified into "*single machine, single board type problems*." This problem has mainly three problems related to our model, "*feeder location* (feeder array)", "*placement sequencing* (mounting sequencing)", and "*component retrieval*."

The feeder array requires installing the variety of feeders appropriately. The placement sequencing requires deciding the tour on the PCB so that the head can move effectively. The assembly time varies according to the adequacy of the feeder array and placement sequence. The component retrieval becomes a problem in the case that several component feeders of the same type have been assigned to the feeder bank, and it is necessary to decide from which feeder the head should retrieve the component.

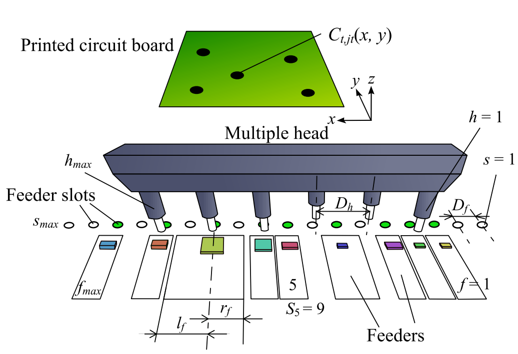


Fig. 4. The structure of our model with the notations ($h_{\max} = 6, f_{\max} = 9, s_{\max} = 18, D_h/D_f = 2$).

Above all, our placement machine model has one more issue that is capable of reducing the assembly time, simultaneous pickup.

V. Optimization of Placement Machine

To solve the problem of the feeder array, we focus on the contribution of the simultaneous pickup to the assembly time. We aim to optimize the feeder array in order to obtain a maximum quantity of the simultaneous pickup. As this problem is highly NP-hard, we approach it in a heuristics way using random multi-start local search (MLS).

Before deciding the tour, we make pickup patterns capable of the simultaneous pickup regardless of the same type of feeders installed in the feeder bank. As we get pickup patterns first, there is no need to solve the component retrieval which is to decide from which feeder the head should pick up the component in the case that several component feeders of the same type have been assigned to the feeder bank. Considering component distributing to each component feeder of the same type is important. To solve this, we assume that components are distributed equally to the feeders of the same type.

To solve the placement sequence, we choose the shortest way greedily on the basis of the combinations of the pickup patterns.

A. Multi-start Local Search

The MLS is one of the meta heuristics method that is an effective approximate for combinatorial optimization problems.

In the local search (LS), an initial solution x is generated at first. Next, a better solution x' adjacent to x is searched changing x slightly. If x' is obtained, adopt x' as x and iterate searching x' until no more better x' can be obtained. The MLS is the method that generates the initial solutions a lot, and it outputs the best solution among the converged solutions.

B. Feeder Array

Step 1: Initialization. We define a condition of a feeder array installed to feeder slots as the initial solution. The initial solution is generated as follows:

1) Choose $f_{\max}$ number of feeder slots in a random manner.
2) Initialize all the feeder slots as 0.
3) Allot $f(f = 1, \ldots, f_{\max})$ to each feeder slot chosen by 1), in a random manner.

Step 2: Neighborhood Search. We propose a simple swapping method that swaps two feeder slots chosen in a random manner.

Step 3: Evaluation. In order to obtain an ideal feeder array, it is important to evaluate the solution properly by treating the structure of the problem. Therefore, we designed three evaluation functions, i.e. "feasibility", "head efficiency", and "simultaneous pickup availability." Those evaluation functions output values of V_1, V_2, and V_3 respectively. In order to evaluate the solution, we use the following value V,

$$V = \alpha V_1 + \beta V_2 + \gamma V_3, \tag{1}$$

where α, β, and γ are weighting factors for the evaluation functions. The search is processed by our MLS in a direction of minimizing V.

Step 3–a: Feasibility. The feeder array generated in a random manner may have a case that it is not feasible to install them on the feeder slots because of a feeder overlapping (refer to Fig. 5 for an example). We actually allow the infeasible feeder array for diversity of the earlier solutions. As a final feeder array has to be feasible, we evaluate the feasibility by (2) setting α weighty rather than β and γ,

$$V_1 = \sum_{f=1}^{f_{\max}} A(f), \tag{2}$$

where $A(f)$ is a quantity of the overlap of a feeder f.

Step 3–b: Head Efficiency. When the head picks up components, it should move effectively to go to and from the PCB without any loss of time. To solve this problem, it needs to gather feeders together near the mounting point on the PCB. Since generating initial solution and neighborhood search tend to have feeders installed discretely on the feeder slots, this function also operates in order not to have scattered feeders. We evaluate the head efficiency by (3),

$$V_2 = \sum_{f=1}^{f_{\max}} Q_f \left| S_{G_{t_f}} - S_f \right|, \tag{3}$$

where $S_{G_{t_f}}$ is the nearest feeder slot from G_{T_f} which is the arithmetic average of T_f type of components, and G_{T_f} is calculated as follows:

$$G_{T_f} = \sum_{n=1}^{j_{\max T_f}} C_{T_f,n}(x,y). \tag{4}$$

Step 3–c: Simultaneous Pickup Availability. The simultaneous pickup has $h_{\max}$ types which are subject to components number that the multiple head picks up simultaneously. We describe simultaneous pickup which picks up $h_{\max}$ components as "$h_{\max}$-simultaneous pickup."

In order to minimize the pickup times, it is desireble to have the $h_{\max}$-simultaneous pickups as many as possible. In case of having the other simultaneous pickups less than the $h_{\max}$-simultaneous pickup, it is also important to have larger simultaneous pickups than smaller ones.

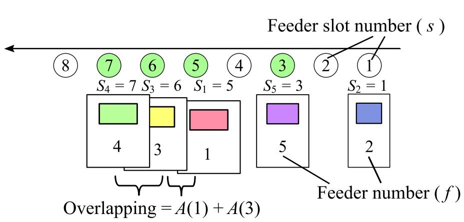


Fig. 5. An example of the overlapping.

We propose the head template referring to Fig. 6 to calculate the number of each simultaneous pickup available from the feeder array. The head template is created by D_h/D_f, where D_h is the head pitch and D_f is the feeder pitch. The head template (Fig. 6(a)) shows the case of $D_h/D_f = 2$, and Fig. 6(b) shows component quantity of each feeder by a symbol "□". When the template lies on as Fig. 6(c), 5-simultaneous pickup can be detected(Fig. 6(d)).

The number of each simultaneous pickup is calculated as follows:

1) Choose a feeder slot that has the symbol □ from right side. If there is no feeder slot which has symbol □, stop.
2) Set the head template so that head number "$h = 1$" is located in the feeder slot chosen by 1).
3) Calculate the number of execution of simultaneous pickup.
4) Iterate 3) until the all symbol □ of the feeder slot is taken up.
5) Iterate from a) with transforming "right side" into "left side."

We evaluate the simultaneous pickup availability by (5),

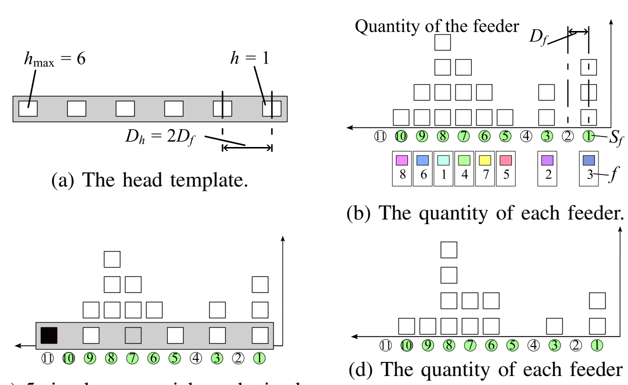
$$V_3 = \sum_{n=1}^{h_{\max}} (h_{\max} - n) A(n), \tag{5}$$

where:

$$A(n) = \begin{cases} (p_{\max} - C_n) & \text{if } n = h_{\max} \\ C_n & \text{otherwise,} \end{cases} \tag{6}$$

C_n is the number of n-simultaneous pickups calculated in the above method 1)–5).

Step 4: Updating. This step compares two evaluated value V of solutions of before and after the swapping neighborhood search. If the swapped solution is better than the other, the MLS iterates from Step 2 with updating the solution by the swapped one. Otherwise it iterates from Step 2 until the neighborhood is all searched. In this case, it saves the converged solution and iterates from Step 1 with generating initial solution until it iterates as many as the specified iteration count. The MLS ends with outputting the best solutions from the converged solutions.



(a) The head template.

(b) The quantity of each feeder.

(c) 5-simultaneous pickup obtained.

(d) The quantity of each feeder after (c).

Fig. 6. An example of calculating the number of the simultaneous pickups ($h_{\max} = 6, f_{\max} = 8, s_{\max} = 11, D_h/Df = 2$).

C. Pickup Combination

In order to obtain the set of n-simultaneous pickup patterns (FIg. 7 (a)) from the optimized feeder array, we reapply the method used in the "simultaneous pickup availability."

Except the patterns of the $h_{\max}$-simultaneous pickup, the other patterns of simultaneous pickup have to be combined into a group of $h_{\max}$ components (Fig. 7(b)).

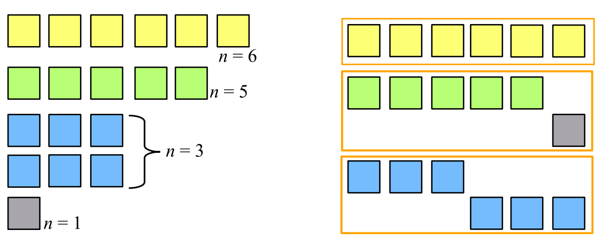
Since this combination is also NP-hard, we combine them using our combination list referring to Table I. Rank 1 means to make a group of 6-simultaneous pickup. Rank 2 means to make a combination of 5-simultaneous pickup and single pickup, which takes 2 times of a pickup action. In the pickup combination, we make the combinations of each rank from the optimized feeder array as many as possible, prioritizing a higher rank over a lower one. In order to prioritize a higher rank of the list, we divide some simultaneous pickup patterns into fewer multiple patterns.

The list is made in priority to the following three criteria,

1) The combination which has the $h_{\max}$ components in all,
2) The combination which enables us to pick up $h_{\max}$ components with fewer pickup times,
3) The combination which contains larger number of simultaneous pickups as many as possible.

D. Pickup and Placement Mounting

Lastly, we solve the mounting sequencing by using the combined pickup patterns.



(a) n-simultaneous pickup patterns obtained through the MLS.

(b) Combined pickup patterns with (a).

Fig. 7. An example of pickup patterns before combination and after combination ($h_{\max} = 6$).

TABLE I

THE COMBINATION LIST ($h_{\max} = 6$)

Rank	Number of pickup times 1	2	3	4	5	6
1	6					
2	5	1				
3	4	2				
4	3	3				
5	4	1	1			
6	3	2	1			
7	2	2	2			
8	3	1	1	1		
9	2	2	1	1		
10	2	1	1	1	1	
11	1	1	1	1	1	1

We calculate $\overline{G}_n$ which is the arithmetic average of the n-th combined pickup patterns.

Fig.8 shows an overview of the method, and detailed steps are shown as follows:

1) Input the initial location of the head C_{init}.
2) Search the nearest $\overline{G}_n$ to C_{init}, and choose n-th combination pickup pattern.
3) Choose components which are used by n-th combination pickup pattern from the one near the $\overline{G}_n$.
4) Make the shortest tour among the placement points.
5) Return to 1) redefining the C_{init} as the last location of the placement point until all placements are done.

VI. SIMULATION

In our experiments, we used the parameters as follows:

- Randomly generated 498 placement points ($p_{\max} = 498$),
- 20 feeders ($f_{\max} = 20$) referring to Table II,
- 60 feeder slots ($s_{\max} = 60$),
- The head pitch is 30 mm ($D_h = 30$) and the feeder pitch is 15 mm ($D_f = 15$).

Table II shows a component type (T_f), quantity (Q_f), left half of width (l_f), and right half of width (r_f) of each feeder number (f).

In the MLS method, we used 1000 times of the generating initial solutions as end condition. We decided the weighting factor of $\alpha = 1.5 \times 10^2$, $\beta = 5 \times 10^{-4}$, and $\gamma = 7.5 \times 10^{-1}$ by trial and error.

Fig. 9 shows the availability of n-simultaneous pickup using the best results of each iteration of the initial solution, and each result is the average of five times of the MLS with changing the random number sequence we used. In the beginning of the iterations, the 6-simultaneous pickups are relatively rare rather than the other ones. As the iterations progressed, our method found better solutions for the 6-simultaneous pickups, and finally it output the best solution which had the 6-simultaneous pickups accounted for 40 % of total pickups. The experimental results in Fig.10 shows an assembly time of five runs on the assembly data set (AD) we generated as follows:

AD1: Data by our proposed method,

AD2: Data by changing simultaneous pickup into single

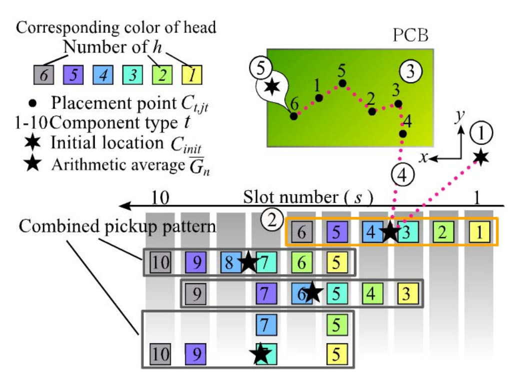


Fig. 8. The overview of the pickup and placement sequencing.

TABLE II

PARAMETERS OF THE FEEDER

Feeder number (f)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Component type (T_f)	1	1	1	2	2	2	3	3	4	4	5	6	7	8	9	10	11	12	13	14
Quantity of feeder (Q_f)	30	30	30	40	40	40	29	29	19	19	22	13	32	44	17	7	36	6	10	5
Left half of width (r_f)	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	11.0	11.0	11.0	13.0	13.0	8.5	8.5	16.0	16.0	22.8	23.5	36.3
Right half of width (l_f)	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	12.0	12.0	12.0	14.0	14.0	10.8	10.8	12.2	12.2	18.0	19.9	32.8

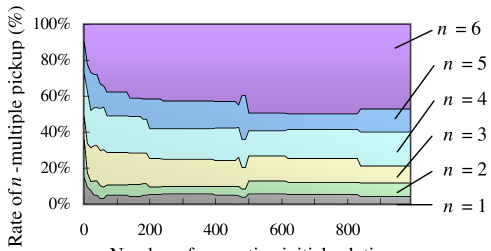


Fig. 9. Rate of n-simultaneous pickup.

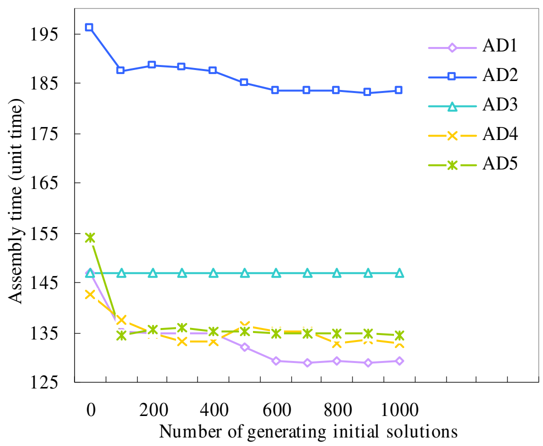


Fig. 10. Assembly time of AD1-AD5.

pickup with AD1,

AD3: Data by our proposed method with generating only one initial solution in the MLS,

AD4: Data by our proposed method with setting $\beta = 5 \times 10^{-3}$ in order to give the feeder efficiency a priority,

AD5: Data by our proposed method with setting $\gamma = 7.5$ in order to give the simultaneous pickup a priority,

Each assembly time was calculated by our placement machine simulator, and Fig.10 uses unit time of our simulator. The longest assembly time recorded by AD2. This indicates that the simultaneous pickup widely affects the assembly time. AD4 and AD5 indicate that the importance of balance of three evaluation values. Comparing AD1 with AD3, we see AD1 recorded 12 % improvement of its assembly time with enough generation of initial solutions in the MLS.

As to the computation time for AD1, AD2, AD4 and AD5, the MLS with 1000 initial solutions generated needs around 5 min with a personal computer (CPU: Pentium D 2.8GHz, Memory: 1GB, OS: Windows XP). AD3 which generates one initial solutions in the MLS needs around 10 sec. Considering that the assembly data is basically prepared before a production starts, we think 5 min is acceptably short (we do not target on-line calibration). These results show the effectiveness of the proposed method.

VII. CONCLUSION

In this paper, we propose a new method for reducing assembly time in PCB assembly by prioritizing efficient pickup operation of placement machines, simultaneous pickup.

The experimental results show that our method reduces the assembly time by maximizing the number of simultaneous pickups equal to the number of heads.

As a future work, we should improve the argorithm so that we can take nozzle changing, components height, and so on into consideration. Also we will consider using other major optimization method such as GA.

REFERENCES

[1] M. Ayob, P. Cowling, and G. Kendall, “Optimisation for Surface Mount Placement Machines,” *Proceedings of the 2002 IEEE International Conference on Industrial Technology (ICIT2002)*, vol. 1, pp. 498–503, 2002.

[2] Y. Crama, J. van de Klundert, and F. Spieksma, “Production Planning Problems in Printed Circuit Board Assembly,” *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 339–361, 2002.

[3] Y. Crama, O. Flippo, J. van de Klundert, and F. Spieksma, “The Assembly of Printed Circuit Boards: A Case With Multiple Machines and Multiple Board Types,” *European Journal of Operational Research*, vol. 98, pp. 457–472, 1997.

[4] K. Fujimura, K. Obu-Cann, and H. Tokutaka, “Optimization of Surface Component Mounting on the Printed Circuit Board Using SOM-TSP Method,” *Proceedings of the 6th International Conference on Newral Information Processing (ICONIP1999)*, pp. 131–136, 1999.

[5] M. Ayob and G. Kendall, “A New Dynamic Point Specification Approach to Optimise Surface Mount Placement Machine in Printed Circuit Board Assembly,” *Proceedings of the 2002 IEEE International Conference on Industrial Technology (ICIT2002)*, vol. 1, pp. 486–491, 2002.

[6] ——, “A Triple Objective Function with a Chebychev Dynamic Pick-and-Place Point Specification Approach to Optimise the Surface Mount Placement Machine,” *European Journal of Operational Research*, vol. 164, pp. 609–626, 2005.

[7] W. Wang, P. Nelson, and T. Tirpak, “Optimization of High-Speed Multistation SMT Placement Machines Using Evolutionary Algorithms,” *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 22, no. 2, pp. 137–146, 1999.

[8] E. Burke, P. Cowling, and R. Keuthen, “The Printed Circuit Board Assembly Problem: Heuristic Approaches for Multi-Headed Placement Machinery,” *Proceedings of the 2001 International Conference on Artificial Inteligence (IC-AI2001)*, pp. 1456–1462, 2001.

[9] M. Ayob and G. Kendall, “A Nozzle Selection Heuristic to Optimise the Hybrid Pick and Place Machine,” *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, pp. 1259–1264, 2004.

[10] P. Csaszar, T. Tirpak, and P. Nelson, “Optimization of a High-Speed Placement Machine Using Tabu Search Algorithms,” *Annals of Operations Research*, vol. 96, no. 1, pp. 125–147, 2000.

[11] W. Lee, S. Lee, B. Lee, and Y. Lee, “Genetic Optimization Approach to Operation of a Multi-Head Surface Mounting Machine,” *IEICE Transactions on Fundamentals of Electronics*, vol. E83–A, no. 9, pp. 1748–1756, 2000.