Assembly Time Minimization of a Particular Placement Machine

ALI FUAT ALKAYA[1], EKREM DUMAN[2]
[1]Computer Science and Engineering Department, Marmara University, 34722, Istanbul, Türkiye
[2]Industrial Engineering Department, Doğuş University, 34722, Istanbul, Türkiye

Abstract: - A new algorithm for a component placement machine is proposed in this study. The new algorithm is compared with former approaches on synthetically generated benchmark instances. It outperforms the former approaches by 3.28% on printed circuit board with 100 components to be placed. We prove that this new approach works better in a typical PCB mounted in the industry, so can be applicable to the considered type of machine.

Key-Words: - Printed Circuit Board Assembly, Traveling Salesman Problem, Placement Machine

1 Introduction

The use of automated placement machines and optimization issues emerging from them has attracted the interest of researchers for a few decades. Since a small percentage of improvement can bring huge amounts of benefit in terms of money and time to the manufacturers, it still is worth to study on. To obtain maximum performance from these machines, it is beneficial to solve the inherent optimization problems emerging from them.

Basically, the operations of these machines yield four basic problems [1]. These are, allocation of component types to machines, determination of board production sequence, allocation of component types to feeder cells (also called feeder configuration problem) and determination of component placement sequence. In many other studies this list is extended or shortened but the last two have great influence and hence importance in optimizing the PCB machines [2,3]. All of these problems are interdependent, that is solution of one affects the other. Depending on the principles of the machine, some may be trivially solved while in most cases they yield NP-Complete problems. Hence a solution aiming to achieve the optimum in all problems simultaneously is very hard. In this study, we investigate a machine type whose operations involve feeder configuration and placement sequencing problems.

In a previous study, Duman [4] modeled the operations of a component placement machine with rotational turret and stationary component magazine. After modeling the operations of this machine, two problems are formulated; the placement sequencing problem and feeder configuration problem. In the study, it is shown that the placement sequencing problem can be modeled and solved as a classical TSP whereas the feeder configuration problem is solved in an ideal way by a proposed procedure. The proposed algorithm is called Assembly Time Minimization Algorithm (ATMA). The results show that it is an improvement in terms of total assembly time for this type of machines when compared with the solution approach used up to that date.

In this study, we propose an improved version of the (ATMA) algorithm called inverse ATMA (iATMA). This newly proposed procedure outperforms ATMA in terms of total assembly time. Two algorithms are compared on randomly generated PCB data and about 2% improvement is gained on the average. In the next section, we give the problem definition and an overview of the working principles of the analyzed machine type. Section 3 gives a summary for ATMA. In Section 4, we give detailed explanation for iATMA. Section 5 gives the details of comparison study and Section 6 includes the summary and conclusions of this study.

2 Machine Working Principles and Problem Formulation

The particular machine type investigated is TDK brand, model RX-5A SMD placement machine.

2.1 Working Principles

The machine has a rotational turret which includes 72 heads. These heads take the components from the component magazine and during the rotation of the turret the head reaching the placement location places the component on the PCB. The component magazine is stationary, has a circular structure placed behind the machine (Figure 1).

By the time the next head reaches the placement location, the PCB is aligned to the exact position where placement will occur. This alignment is achieved by the simultaneous movements of the board carrier which has Chebyshev distance measure.

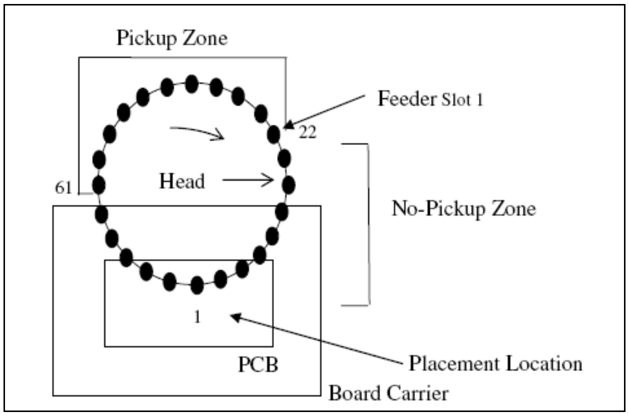

Figure 1

Another important property of the investigated machine is that it can handle component types of different weight. To cope with this each head is equipped with three suction nozzles compatible with different weight categories. Mechanically, there are two sets of operations that follow each other sequentially and the operations in each set are performed simultaneously. Operations in the first set includes: i) turret rotates and the next placement head comes over the PCB, ii) board carrier aligns the new placement point under the placement head, iii) placement heads rotate if necessary to align the suction nozzle carrying the component to be placed. In the second set there are: i) the placement head moves down, makes the placement and moves up, ii) heads in the pickup zone that are above the appropriate component tapes move down and pick up components.

When a component of heavier type is picked up by any of the head, the rotation speed of the turret is reduced. For the machine type that is under analysis, we have four discrete speed values corresponding to four different weight categories. These are 0.20, 0.23, 0.33 and 0.40 seconds per rotational movement of distance one head. The other movable part of the machine, the board carrier, has a speed of 120 mm per second in both x and y directions. In [4], a more detailed explanation of the operations of this machine can be found.

2.2 Problem Formulation

Basically, the goal is to optimize the PCB assembly time of the machine considered, but this is can be achieved by an efficient solution of the placement sequencing and feeder configuration problems.

When one has a deeper look at the whole problem, he can easily see that what makes it complicated is the varying speed of the turret. Also, he can easily see that the feeder configuration problem vanishes if the rotational turret has a unique speed value for all weight categories (i.e. the turret has a uniform rotational speed throughout the whole assembly process), and the problem turns out to be only a placement sequencing problem. If we let t_{ij} be the time between the completion of consecutive placements at points i and j, then it can be calculated by using the following definitions.

$t^0 =$ component placement time (including placement head moving down and up time),

$t_{ij}^x =$ board carrier movement time in x direction between points i and j,

$t_{ij}^y =$ board carrier movement time in y direction between points i and j,

$t^t =$ turret time (turret rotation time required for the next placement head to arrive over the PCB),

Then, t_{ij} is given by the following expression

$$t_{ij} = t^0 + \max\{t_{ij}^x, t_{ij}^y, t^t\} \qquad (1)$$

Observe that the formulation turns out to be a TSP with Chebyshev distance measure. On the other hand, given a placement sequence, the objective of the feeder configuration problem is to find the optimum positioning of the component tapes within the magazine so that, the number of slower steps taken by the by the turret time is minimized.

3 Previous Work (ATMA)

The placement sequencing problem (for a given feeder configuration) should be regarded as the main problem since its solution directly gives the PCB assembly time. Accordingly, Duman regards the feeder configuration problem as the auxiliary problem [4]. The variable and complicated nature of t^t not only makes the placement sequencing problem difficult but also makes the TSP formulation infeasible. However, it turns out that, from the placement machine investigated, this change in the t^t values makes placement sequences of mixed light and heavy components quite inefficient. Accordingly, it seems to be a good idea to place all of the lighter components first and then the heavier ones. This way, the t^t values corresponding to each weight category would be constant and it would be possible to use the TSP formulation to find the placement sequences within each weight category. This is the idea behind ATMA.

The given solution procedure includes firstly finding TSP routes for each weight category, connecting these routes and then given these placement sequence the optimal configuration is obtained by assigning the component types to the feeder locations in the order of first appearance in the placement sequence. The routes are obtained using Convex Hull and Or-Opt algorithms [5].

Algorithm ATMA:

i) Find TSP routes for each weight category. Call the route for weight category 1 as Route 1, and so on.
ii) Connect Route 2 to the last point of Route 1 through the shortest connection. Rearrange Route 2 to make the connection point as the home city.
iii) Repeat Step 2 to connect Route 3 to the modified Route 2 and Route 4 to the modified Route 3.
iv) Apply FAP to find the feeder configuration.
v) Recalculate the assembly time using the modified TSP routes and the tt values found through FAP.

Feeder Assignment Procedure (FAP):

i) Assign sufficient number of slots to each group of components where group 4 takes the closest slots to the PCB, group 3 takes the set of next closest slots, and so on.
ii) For the internal arrangement of each group, assign component types to feeder slots in the order of their first appearance in the placement sequence.

4 Proposed Algorithm (iATMA)

In this study, we propose an improved version of the ATMA algorithm called iATMA. The idea behind iATMA is similar to ATMA and so it is very similar to it in terms of the steps. But the basic difference that it proposes is in the order of placement of component groups. Both algorithms mount the components in groups of their weight categories. ATMA places the component groups starting from group 1 to group 4, but iATMA inverses this process such that it starts mounting the components from group 4 to group 1. Below, only steps 2 and 3 of iATMA are given because the rest is the same as ATMA.

ii) Connect Route 3 to the last point of Route 4 through the shortest connection. Rearrange Route 3 to make the connection point as the home city.
iii) Repeat Step 2 to connect Route 2 to the modified Route 3 and Route 1 to the modified Route 2.

We propose no new ideas for FAP because it already gives ideal solutions.

In order to see the improvement that iATMA provides, consider the following example. Let N=100, i.e. 100 components will be assembled with the following weight categories: N_1=80, N_2=10, N_3=5 and N_4=5 where increasing index value identifies heavier component categories. After configuring the feeder slots as stated in the FAP, using ATMA, it can be found that the number of rotational steps of the turret in each speed category (0.20, 0.23, 0.33, 0.40 s) as 50, 15, 10 and 25, respectively if continuous assembly process of the PCB assembly is considered.

If the iATMA is to be used in this example, the number of rotational steps of the turret in each speed category would be 60, 10, 5 and 25.

If we assume that the *x-y* movements of the board carrier can be completed within the allowed turret time, t^t, then, the sum of the 'number of placements in a speed category *i*' plus $N*t^0$ can be defined as the Lower Bound (LB) for the PCB assembly time. Clearly, the LB values of the algorithms is a comparison criterion.

Comparing the performances of the iATMA and ATMA with respect to LB, we see that for the above example ATMA would populate a PCB in 26.75 seconds while iATMA would populate the same PCB in 25.95 seconds. This gives about a 3% improvement.

Group	n_i	Types of components	N_i	Placement sequence (given)
1 (lightest)	4	X,Y,Z,T	8	X,Y,Z,T,T,Z,Y,X
2	3	P,Q,R	7	P,Q,R,P,Q,R,P
3	3	K,L,M	7	K,K,L,M,L,M,K
4 (heaviest)	3	A,B,C	6	A,B,B,A,C,A
			N=26	

Table 1

In order to trace the steps in the algorithm and movements of the turret, we constructed the following hypothetical small rotational turret system (see Table 1).

Say that, the placement sequences are already determined by using the appropriate TSP solving algorithm (Convex-Hull and Or-Opt). As stated before, larger index values stand for heavier component groups. Next, using FAP we assign a sufficient number of slots to each group of components, where group 4 takes the closest slots to the PCB, group 3 takes the set of next closest slots, and so on, and for the internal arrangement of each group we assign component types to feeder slots in the order of their first appearance in the placement sequence. This is illustrated in Figure 2.

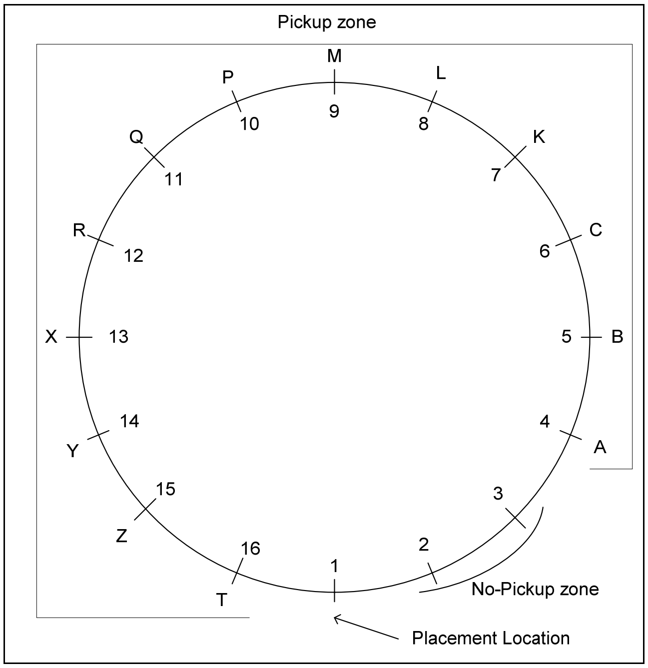

Figure 2

In this hypothetical rotational turret system, we have 16 heads, one being the placement head, the next two are in the no-pickup zone, and the rest 13 form the pickup zone.

If one applies the two algorithms, namely iATMA and ATMA, he will reach the results in Table 2.

	ATMA	iATMA
SC_1	0	6
SC_2	10	7
SC_3	10	7
SC_4	8	8

Table 2

where SC_i denotes the number of rotational steps of the turret in a speed category i. If the weight categories have the same t^t values that we used in the above example, then 5.5% improvement will be obtained by using iATMA (with respect to LB).

In this study, we also propose the formulation of the number of rotational steps of the turret in each speed category i, (SC_i). Assume that we partitioned the component types into s groups according to their weight values, with group 1 being the lightest and group *s* being the heaviest. Then the following equations can be used to calculate SC_i.

$$SC_s = \min\{N, N_s + NPZ\} \tag{2}$$

$$SC_i = \max\left\{0, \min\left\{N_i, \sum_{j=1}^{i} N_j - NPZ\right\}\right\} \tag{3}$$

$$\text{for } 1 \le i \le s-1$$

where N denotes the total number of components to be populated, N_i represents the number of components to be placed in group *i* and NPZ stands for the number of heads that are in no-pickup zone with $NPZ \ge 0$. It is worth to note that the formulation has no recurrence relation and it is a general formula which works for any value of s for $1 \le s \le N$. For the special case when *s=1*, i.e. there is only a unique weight category, we consider this unique group as the heaviest and apply Equation 2. Also, for *i=1*, Equation 3 can be reduced to $SC_1 = \max\{0, \min\{N_1, N_1 - NPZ\}\}$ which is equal to $SC_1 = \max\{0, N_1 - NPZ\}$ since $NPZ \ge 0$.

5 Conceptual Analysis and Comparison of the Algorithm

The proposed iATMA is guaranteed to bring a lower LB than ATMA for the component placement problem. The following analysis proves this statement.

A data generator that produces random printed circuit boards is implemented. The position, type of the component and which category it belongs is randomly created. The methodology for this generator is the same as the one explained by Duman. The total number of components to be placed (N) is prespecified and it is given to the data generator. We studied with 4 different N values, 100, 200, 300 and 400. For N=100, the number of component types in groups 2, 3 and 4 are determine uniformly between 1 and 5, where each of them is placed one, two or three times (with equal probabilities) on the PCB. The rest of the components were in group 1 that is composed of 40 component types and the placement number of each one is probabilistically equal. For the larger problems (N=200, 300 or 400), the number of component types in each group is kept constant but their placement numbers are increased proportionally with N. The board that these components are placed is assumed to have dimensions 250mm x 300mm and for the sake of pure randomness it is assumed that no two components can be placed on the same coordinate.

When the above data generation model is analyzed, one can easily see that in a typical PCB with 100 components (N=100), number of types of components and number of components to be placed for each group (N_i and n_i values) can be expected as follows: n_1=40, n_2=3, n_3=3, n_4=3, N_1=82, N_2=6, N_3=6 and N_4=6 on the machine described in [4]. Applying the formulas for calculating number of rotational steps of the turret in a speed category *i* (SC_i) values for ATMA and iATMA given above, the results in Table 3 can be obtained easily.

i	n_i	N_i	SC_i (ATMA)	SC_i (iATMA)
1	40	82	56	62
2	3	6	9	6
3	3	6	9	6
4	3	6	26	26
Lower Bound			31.64	31.16

Table 3

The lower bound for this board is computed as 56*0.25 + 9*0.28 + 9*0.38 + 26*0.45 = 31.64 seconds for ATMA and 31.16 for iATMA similarly. This brings a 0.48 second improvement, that is, the assembly time can be improved 0.48 seconds, which means 1.5% improvement on the average.

The same analysis can be applied when N=400 and one can obtain the following similar results in Table 4.

i	n_i	N_i	SC_i (ATMA)	SC_i (iATMA)
1	40	340	314	320
2	3	20	23	20
3	3	20	23	20
4	3	20	40	40
Lower Bound			111.68	111.20

Table 4

It is interesting to note that the improvement that ATMA guarantees remains constant when N increases (which is 0.48 seconds), and so relative improvement decreases to 0.4%.

Up to this point, the expected benefits of algorithms are examined conceptually. Below, we will give simulation results.

For the simulation study, the formerly introduced data generator is implemented and run to generate 100 different PCBs with varying N values from 100 to 400. The results are given in Table 5.

N	ATMA		iATMA	
	Assembly Time	Lower Bound	Assembly Time	Lower Bound
100	40.011	31.725	41.084	31.235
200	67.527	58.493	67.729	57.987
300	92.778	84.959	93.323	84.475
400	118.910	111.523	119.390	111.021

Table 5

These values are exactly the same as the above conceptual analysis. The lower bound for iATMA is about 0.48 seconds less than ATMA's lower bound values for all N. But the results also indicate a problem for iATMA. They indicate that even though iATMA is theoretically better than ATMA in terms of lower bound values, it is not superior than ATMA when placing the components for any N. So, as a result, it is proved that lower theoretical bounds can be obtained by placing groups of components in the reverse order (heaviest to lightest) but iATMA failed at reaching these lower bounds and even gave worse results. The possible cause of this situation is that in iATMA we face with the case $\max\{t_{ij}^x, t_{ij}^y\} > t^t$ for consecutive placement operations more than we face with it in ATMA. That is, the travel time from placement location to another is greater than turret time for more number of cases than predicted. If this travel time were smaller than t^t, than we can achieve lower assembly times. The deeper analysis given below clarifies this discussion.

For each group, number of components placed in each speed category can be summarized in Tables 6 and 7 for ATMA and iATMA. The considered PCB board is the typical PCB board with 100 components described above.

ATMA		SC_1	SC_2	SC_3	SC_4
Group 1	Out of 82	56	9	9	8
Group 2	Out of 6	-	-	-	6
Group 3	Out of 6	-	-	-	6
Group 4	Out of 6	-	-	-	6

Table 6

iATMA		SC_1	SC_2	SC_3	SC_4
Group 1	Out of 82	62	-	-	20
Group 2	Out of 6	-	6	-	-
Group 3	Out of 6	-	-	6	-
Group 4	Out of 6	-	-	-	6

Table 7

When generating the placements of components on a board the only consideration that we take into account is not to place two components on the same location. So the components in a group are randomly distributed over the board. When group 1 components are randomly distributed on the board and the TSP route for it is found, it is seen that the distance between two placements have reasonable length. That is, for most cases the travel time for this distance is not extremely greater than the turret time. But when other group components are randomly distributed on the board and the TSP route for it is found, it is seen that the distance between two placements do not have reasonable length. That is, for most cases the travel time of the carrier board for this distance is much greater than the turret time. So they can not be placed in the turret time and an excess time is unavoidable. In the first table above, we see that ATMA places these group 2, 3 and 4 components when the turret time is at maximum, i.e when the turret is at minimum speed that is during SC_4. But iATMA places group 2 components during SC_2 and group 3 components during SC_3, which is when turret is faster. So ATMA compensates the possible excess time more effectively than iATMA, hence obtains a really big advantage. On the other hand, iATMA places more number of group 1 components in SC_4 than ATMA. But this will imply a small benefit for ATMA because, for most cases the travel time for this distance is not much greater than the turret time.

To summarize, to minimize the excess time occurring while placing the group 2, 3 and 4 components, placing them in minimum speed (SC_4) brings an advantage to ATMA and so it performs better.

A research on the design of placement locations of the heavier components reveal the fact that the components in the same group are preferred to be placed more closely. Their placement locations are mostly very close with one or two of them are far away from the group. So the data generator is re-implemented to give PCB instances in which heavier groups of components are placed more closely. On a board with this type of design, ATMA is expected to give worse results than iATMA because the disadvantage for iATMA disappears in this type of PCB instances. In Table 8, we give the results of the comparison of the PCB assembly time obtained by ATMA and iATMA. The PCB instances are created according to the modified data generator and each value is an average of 100 different PCB instances.

N	ATMA		iATMA	
	Assembly Time	Lower Bound	Assembly Time	Lower Bound
100	35.880	31.572	34.704	31.099
200	61.525	58.201	60.387	57.720
300	87.266	84.664	86.194	84.192
400	113.821	111.722	112.744	111.233

Table 8

iATMA outperforms ATMA by 3.28% for N=100, by %1.85 for N=200, by %1.23 for N=300 and by 0.95% for N=400. This result could be expected because as N increases , the average x-y distances between component pairs decreases and it will be easier to arrange the placement sequence so that the board carrier movements can be completed within the free (turret) time by both iATMA and ATMA. Again it is worth to note that lower bound for iATMA is about 0.48 less than lower bound for ATMA, which is found by theoretical analysis.

6 Conclusion

In this study, we proposed a new algorithm for the solution of a problem faced in a particular PCB assembly machine. The basic idea of the newly proposed heuristic is to mount the components in reverse order of a previously proposed approach. This method brings an improvement from 0.95% to 3.28% for varying N values. The future work would be to investigate the possible improvement that may be obtained by redesigning the route for group 1 components using prize collecting TSP or other approaches.

References:

[1] Duman, E., Optimization Issues in Automated Assembly of Printed Circuit Boards, *PhD Thesis,* Boğaziçi University, 1998.

[2] Smed, J., Johnsson, M., Nevalainen O., A hierarchical classification scheme for electronics assembly problems, *Proceedings of TOOLMET Symposium—Tool Environments and Developments Methods for Intelligent Systems,* Oulu, Finland, 2000, pp.116–119.

[3] Crama, Y., Klundert, J.V.D., Spieksma, F.C.R., Production Planning Problems in Printed Circuit Board Assembly, *Discrete Applied Mathematics,* 123 (1-3), 2002, pp. 339-361.

[4] Duman E., Modelling the operations of a component placement machine with rotational turret and stationary component magazine, *Journal of the Operational Research Society*, 58, 2007, pp. 317-325.

[5] Or, I., Traveling Salesman type combinatorial problems and their relation to the logistics of blood banking, *PhD Thesis,* Nortwestern University, 1976.