A New Dynamic Point Specification Approach To Optimise Surface Mount Placement Machine in Printed Circuit Board Assembly

Masri Ayob And Graham Kendall
Automated Scheduling, Optimisation and Planning (ASAP) Research Group,
University of Nottingham, School of Computer Science and IT
Nottingham NG8 1BB, UK
Phone: +44-115-846-6525 Fax: +44-115-951-4254 E-mail: mxa|gxk@cs.nott.ac.uk

Abstract
Some factors that significantly contribute to the overall assembly efficiency of the placement machines are robot motion control, sequence of placement points and feeder setup. Many techniques have been developed to improve the sequence of placement points and/or the feeder setup for the PCB assembly process. However, a limited number of works have reported on improving the robot motion control. This paper proposes a revised dynamic pick-and-place point (DPP) specification approach called Chebychev DPP (CDPP). We formulate a problem for a placement machine that is a type of cartesian robot which is able to move in both X and Y directions concurrently. The formulations are constructed based on the triple objectives of minimising robot assembly time, feeder movements and PCB table movements. Experimental results show that our CDPP is superior to Wang's DPP approach in terms of robot assembly time, feeder movements and PCB table movements.
Keywords: Modelling, Optimisation, Electronic Assembly, Printed Circuit Board Assembly, SMT.

1. Introduction

Hundreds of electronic components have to be placed onto the printed circuit board (PCB) by using a surface mount placement machine to produce a complete PCB. Since the placement machines are expensive, improving their efficiency is highly desirable. Some factors which contribute to the overall assembly efficiency of the placement machine is the robot motion control, the sequence of placement points, and the feeder slot assignment [1].

There has been a lot of previous work to improve the sequence of placement point and/or feeder slot assignment of the PCB assembly process. For example, Wang, Nelson and Tirpak [2] applied a genetic algorithm (GA) to optimise the feeder slot assignment problem for multi-station surface mount technology (SMT) placement machines. They found that the GA performed as well as a human expert. Kumar and Li [3] optimised the feeder setup and component placement sequence by using an integer programming approach. They reported a 25% time saving over some of the techniques currently use in industry such as type-writer method together with greedy assignment algorithm and S-shape method together with greedy assignment algorithm. However, only a few works have been reported on improving the robot motion control such as Su et al. [4]. They proposed a dynamic pick-and-place (DPP) point to avoid robot's waiting time. The approach allows the robot to pick and place a component at any location rather than a fixed pickup and placement (FPP) locations. The placement sequence was determined using a traveling salesman problem (TSP) method and the feeder slots are randomly arranged. They found that the DPP approach was superior to the FPP approach.

Wang, Ho and Cannon [5] also proposed a heuristic based on the DPP approach. They sequenced the placement operations based on the X coordinate starting with the smallest X value, then the larger Y coordinate if more than one point had the same X value. Components are assigned to the feeder slot such that the total exchange frequency of all adjacent slot pairs have the maximum value. Fu and Su [1] simultaneously arranged the placement sequence and feeder slots based on the DPP approach by applying a genetic algorithm, simulated annealing and tabu search to solve the problem. They gained better performance than Wang et al. [6]. Recently, Hop and Tabucanon [7] proposed a new heuristic algorithm to improve on the approach of Wang [5] based on the fact that assembly time depends on the relative position of pickup and placement points (DPP model). The approach considered the trade-off between the strategies of assembling by area and assembling by component type in order to reduce the feeder carrier and PCB table movements, as well as reducing assembly cycle time. Results show that the approach was superior to Wang's approach, in terms of total assembly time saving. In other work, Hop and Tabucanon [8] proposed an extended dynamic point specification approach named as EDPP. The EDPP model determines the pickup and placement coordinate on the PCB based on global view of the point relationship in the system. The EDPP consideres the movement of the robot arm, the movement of the PCB table and the movement of the feeder carrier as a way of reducing the assembly cycle time. If the feeder carrier (or PCB table) can move fast enough to position the required point at the required pickup (or placement) location, the EDPP model may allow the feeder carrier (or PCB table) to pass over the required point and stop at the point where the feeder carrier (or PCB table) can provide better robot movement. This means that the EDPP is willing to pay an extra cost for the robot travel in order to gain better feeder movement or PCB table movement for the next assembly cycle. The EDPP model obtained better assembly cycle time compared to DPP model.

This paper proposes an improved DPP specification approach called Chebychev DPP (CDPP). The DPP model will try to maintain the fixed pickup and placement location as much as possible unless this leads to robot idling. Therefore, the DPP model may still have unnecessary movement. Hence, in our CDPP, we try to eliminate the unnecessary movement by looking forward to the next PCB coordinate (or feeder slot) when determining the current pickup (or placement) location. We formulate a problem for a placement machine that is a type of cartesian robot which has a single head equipped with a single nozzle. The formulations are constructed based on the triple objectives of minimising robot assembly time, feeder movements and PCB table movements. The main difference between our CDPP model and the previous DPP (and EDPP) is that our CDPP calculates the robot arm movement distance as the maximum of the movement in Y or the movement in X (a chebychev distance) since our robot arm can move in X-axis and Y-axis concurrently, whilst the previous DPP (and EDPP) calculate the robot arm movement as a euclidean distance.

2. FPP Background

Robotic assembly problems can be classified into two types based on the robot motion being either a fixed robot motion between pickup and placement points (FPP) or a dynamic robot motion between pickup and placement points (DPP) [5]. In the FPP model, the feeder carrier can move in the X-axis to position a required component at the fixed pickup location, the PCB table can move freely in X-axis and Y-axis to position a PCB coordinate at the fixed placement location but the robot arm can only move in the Y-axis between fixed pickup and placement locations. Since the robot arm only moves between these two fixed locations, there may exist an undesirable robot waiting time. Some works that have been conducted based on the FPP model [3, 9].

3. DPP Background

In the DPP model, both the feeder carrier and PCB table only move in X-axis whilst the robot arm moves in the Y-axis, in optimal conditions. This occurs when the feeder carrier or PCB table can move within 'free' movement time. In other words, the optimal condition occurs when the feeder carrier and PCB table can move to the best pickup point and placement point before the robot arm arrives. Otherwise, the robot arm moves at an angle from the Y-axis to catch the feeder carrier or PCB table to avoid robot idling [1]. The PCB table and robot arm, or the feeder carrier and robot arm, will stop and meet at the dynamically assigned interception location at the same time [4]. This situation is known as robot interception.

To more clearly describe the DPP model and our CDPP model, the following notations are used (most of them adopted from [5]):

- CT : the cycle time to assemble all components;
- N : the number of placement points;
- K : the number of component types (each feeder slot holds multiple copies of one component type);
- $c(i)_{x,y}$: the i^{th} x,y coordinate on the PCB which will have the i^{th} component be placed there;
- $f(i)_x$: the feeder pickup coordinate of the i^{th} assembly sequence. The $f(0)_x$ is defined as the center of the first pickup location (referring as the origin coordinate). For all *i*, $f(i)_y=0$ as the feeder slot can only move in the X direction;
- $b(i)_x$: the placement coordinate of the i^{th} assembly sequence which is the x,y offset from the origin coordinate $(f(0)_x)$. For all *i*, $b(i)_y=c(i)_y$ as the PCB table can only move in the X direction ;
- V_r : the robot speed (average);
- V_b : the PCB table speed (average);
- V_f : the feeder speed (average);
- T_p : the time for picking up a component;
- T_i : the time for placing a component;
- F_{ab} : the exchange frequency between component of type *a* and *b*;
- Fm(i) : the moving distance and direction of feeder (positive sign means the feeder moves to the left, negative otherwise) to position the i^{th} component at the i^{th} pickup location, $f(i)_{x,y}$;
- Tm(i) : the moving distance and direction of PCB table (positive sign means the PCB table moves to the left, negative otherwise) to position the i^{th} PCB coordinate at the i^{th} placement location, $b(i)_{x,y}$;
- $s_{i-1,i}$: the slot distance between feeder slot for i^{th} and and $(i-1)^{th}$ component in assembly sequence (positive sign means the ith slot is located at the right side of $(i-1)^{th}$ slot, negative otherwise);
- $c_{i-1,i}$: the distance between the i^{th} and the i^{th}*-1* points on the PCB board (positive sign means the X-coordinate of the i^{th} point on PCB is bigger than the i^{th}*-1* point, negative otherwise);
- $d_{f(i),b(i)}$: the distance between $f(i)_x$ and $b(i)_x$ where the distance is measured as a Euclidean distance in DPP or a Chebychev distance in CDPP ;
- D_x : the interception distance in X-axis (positive sign means the robot arm moves to the right, negative otherwise).

For every pickup and placement operation, movement of the robot must occur. However, the movements of the feeder carrier and the PCB table are dependent on the next pickup or placement point respectively. Hence, the aim of our work is to increase the optimal robot movement in order to minimise the assembly cycle time, CT, which is a function of the total robot traveling distance divided by the robot speed, plus the total pickup and placement time. Thus the aim is;

$$CT = \min \sum_{i=1}^{N} \left[\frac{d_{f(i),b(i)}}{V_r} + \frac{d_{b(i),f(i+1)}}{V_r} \right] + NT_i + NT_p \qquad (1)$$

when i=N, then f(N+1)=f(N)

Wang, Ho and Cannon [5] argue that the shortest robot traveling distance occurs when both $d_{f(i),b(i)}$ and $d_{b(i),f(i+1)}$ involve no robot arm movement in the X direction. In the DPP approach, the optimal pickup happens when equation (2) is true, that is when the total time taken for the robot arm to move from the pickup point *f(i-1)* to the placement point *b(i-1)*, to place the $(i-1)^{th}$ component and to move from the placement point *b(i-1)* to the next best possible pickup point *f'(i)*; is greater than the time taken for the feeder carrier to bring the i^{th} component from location *F(i)* to the best pickup location *f'(i)* where the best pickup point is the case when $f'(i)_x=b(i-1)_x$.

$$\frac{d_{f(i-1),b(i-1)}}{Vr} + T_i + \frac{d_{b(i-1),f'(i)}}{Vr} \geq \frac{d_{F(i),f'(i)}}{V_f} \qquad (2)$$

Similarly, the optimal placement occurs when equation (3) is satisfied, that is when the total time taken for the robot arm to move from the placement point *b(i-1)* to the pickup point *f(i)*, to pick the i^{th} component and to move from the pickup point *f(i)* to the next best possible placement point *b'(i)*; is greater than the time taken for the PCB table to bring the i^{th} placement point from location *B(i)* to the best placement location *b'(i)* where the best placement location is the case when $b'(i)_x = f(i)_x$.

$$\frac{d_{b(i-1),f(i)}}{Vr} + Tp + \frac{d_{f(i),b'(i)}}{Vr} \geq \frac{d_{B(i),b'(i)}}{V_b} \quad (3)$$

When equation (2) does not hold, that is when the robot arm can reach point *f'(i)* before the feeder carrier can arrive at point *f'(i)*, then instead of moving in *Y* direction from the *b(i-1)* to *f'(i)* and wait for the feeder carrier at *f'(i)*, the robot arm will move at an angle of *Y* from the *b(i-1)* to pick the i^{th} component at the interception location, *f"(i)*. Both, the robot arm and the feeder carrier will meet at *f"(i)* and stop moving at the same time. This condition is represented by equation (4).

$$\frac{d_{f(i-1),b(i-1)}}{Vr} + T_i + \frac{d_{b(i-1),f''(i)}}{Vr} = \frac{d_{F(i),f''(i)}}{V_f} \quad (4)$$

Similarly, when equation (3) does not hold, the robot arm will move an angle of *Y* from the *f(i)* to place the i^{th} component onto the i^{th} PCB coordinate at the interception location, *b"(i)*. Both, the robot arm and the PCB table will meet at *b"(i)* and stop moving at the same time. This condition is represented by equation (5).

$$\frac{d_{b(i-1),f(i)}}{Vr} + T_p + \frac{d_{f(i),b''(i)}}{Vr} = \frac{d_{B(i),b''(i)}}{V_b} \quad (5)$$

4. A CDPP Formulation

In this work we model the PCB assembly problem for a sequential pick and place machine which is a type of cartesian robot is able to move in X-axis and Y-axis concurrently. The nozzle grasps a component from the feeder carrier and then mounts it on the PCB. The PCB table and feeder carrier can only move in the X-axis to position the placement coordinate of PCB and component pickup coordinate of feeder carrier, respectively. The robot, PCB table and feeder carrier can move simultaneously. The robot travels between feeder carrier and PCB table for picking and placing a component, respectively.

We agree with Wang et al. [5] that the shortest robot traveling distance occurs when no robot movements occur in the X-axis, but in the case where the robot can move in *X* and *Y* direction simultaneously, the optimal robot movement can still be preserved even if the robot has to move in *X* direction as long as the movement in *Y* takes longer time than the *X* movement. In our CDPP approach, the optimal robot moves indicate that the feeder carrier and/or PCB table can move within free movement time and/or the movement of robot in *Y* takes longer than the movement in *X*. Our approach differs from [1, 5, 10] as we allow the robot to move in *X* and *Y* direction simultaneously whenever necessary, even in the case where the feeder and PCB table can move within free movement time. When robot interception occurs, we allow the robot arm to move in X-axis and Y-axis concurrently and the traveling time is dictated by the maximum of X or Y distance. If the Y distance is greater than the X distance, the robot still performs an optimal movement, even though the feeder and/or PCB table are not fast enough to bring the best pickup/placement point at a specific time. By allowing the robot to move in X direction, even though in the optimal movement, we can increase the chance of an optimal movement for the next placement or pickup operation.

By default, in optimal feeder carrier movement this is the case when equation (2) is true, f(i) = f'(i) while f'(i)x = b(i-1)x (refer to Fig.1). Similarly, in optimal PCB table movement, it is assumed that b(i) = b'(i) and $b'(i)_x = f(i)_x$ (refer to Fig.2). This means that in order to test the equation (2) or (3), we assume the robot only moves in Y-axis from b(i-1) to f'(i), or from f(i) to b'(i) respectively.

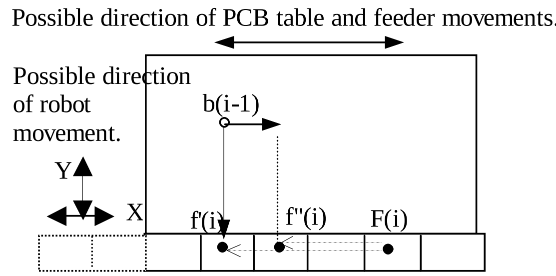

Fig.1 A CDPP model for determining pickup location f(i), the feeder will carry the component from F(i) to f'(i) if robot does not need to move in the X-axis or from F(i) to f"(i) otherwise.

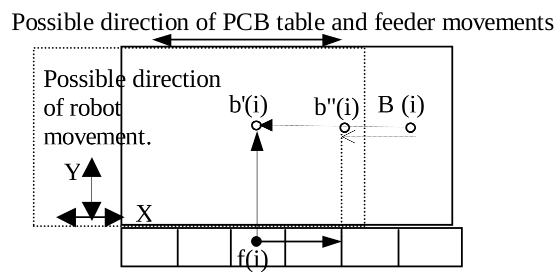

Fig.2 A CDPP model for determining placement location b(i), the PCB table will move to position the placement point from B(i) to b'(i) if robot does not need to move in the X-axis or from B(i) to b"(i) otherwise.

In optimal pickup (equation (2) is true), we will consider two cases (case 1 and case 2) to determine the pickup location, f(i):

Case 1:

The robot arm moves simultaneously in the X-axis and Y-axis to pick a component at pickup location f(i) (where f(i)=F(i) in this case) and the feeder carrier does not move at all if the X distance between the i^{th} *and* i^{th}*-1 PCB coordinate,* $c_{i-1,i}$ *is greater than* $d_{F(i),f'(i)}$*; and the* i^{th} *PCB coordinate is located in the pickup direction; and the value of the Y coordinate of the* i^{th} *PCB coordinate* $(c(i)_y)$ *is greater than the absolute value of* $d_{F(i),f'(i)}$.

Then,

$D_x = d_{F(i),f'(i)}$; $f(i)_x = b(i-1)_x + D_x$;

$Fm(i) = F(i)_x - f(i)_x = [F(i)x] - [b(i-1)x + (F(i)_x - b(i-1)_x)] = 0$

then there is no feeder movement.

Case 2:

The robot arm does not move in the X-axis to pick a component at pickup location f(i) if case 1 is not satisfied but equation (2) is true;

Then,

$D_x=0$; $f(i)_x=b(i-1)_x$; (similar to Wang's approach)

$Fm(i)= F(i)_x-f(i)_x$.

When equation (2) is false, then we consider case 3 and case 4 to determine pickup point f(i). When the equation (2) does not hold, the robot arm and feeder carrier movement time can be expressed by the following equation:

$$\frac{d_{f(i-1),b(i-1)}}{Vr}+Ti+\frac{d_{b(i-1),f''(i)}}{Vr}\geq\frac{d_{F(i),f''(i)}}{V_f} \quad (6)$$

In all conditions for these cases, the robot arm has to move in the *X* axis.

Case 3:

Similar to the case 1 except this case is consider when equation (2) false. If this case is satisfied, thus optimal movements are still preserved even though the feeder carrier is not fast enough to position the i^{th} component at the pickup location f'(i).

Case 4:

If the case 3 is not satisfied and equation (2) false, then the robot arm moves simultaneously in X and Y direction while the feeder carrier also moves concurrently in X direction to position the i^{th} component to the new relative pickup location. The robot arm stops moving in X when it meets the feeder carrier at f"(i).

Then;

$$Dx=\frac{\left(abs(d_{F(i),f'(i)})-(\frac{abs(d_{f(i-1),b(i-1)})}{V_r}+T_i)*V_f\right)*Vr}{V_r+V_f};$$

$f(i)_x=b(i-1)_x + D_x$ if $s_{i-1,i}$ is positive, or

$f(i)_x=b(i-1)_x - D_x$ if $s_{i-1,i}$ is negative ; $Fm(i)=F(i)_x-f(i)_x$;

In case 4, the optimal movements can still be preserved if the absolute value of D_x is less than the value of *Y* coordinate of the i^{th} PCB coordinate ($c(i)_y$).

Similarly, to determine the placement location *b(i)*, we will consider a few cases. When equation (3) is true (optimal movements), we consider these two cases (case 5 and case 6):

Case 5:

The robot moves simultaneously in the X-axis and Y-axis to place a component at placement location b(i) and the PCB table does not move at all if the distance between the feeder slots for the i^{th} and $i^{th}+1$ components is greater than $d_{B(i),b'(i)}$, and the feeder slot containing the $i^{th}+1$ component is located in the placing direction; and the value of Y coordinate of the i^{th} PCB coordinate ($c(i)_y$) is greater than the absolute value of $d_{B(i),b'(i)}$.

Then,

$D_x=d_{B(i),b'(i)}$; $b(i)_x=f(i)_x + D_x$;

$Tm(i)=B(i)_x-b(i)_x=0$ (there is no PCB table movement).

Case 6:

The robot only moves in the Y-axis to place a component at placement location b(i) if equation (3) is true but case 5 is not satisfied;

Then,

$D_x=0$; $b(i)_x=f(i)_x$; (similar to Wang's approach)

$Tm(i)= B(i)_x-b(i)_x$.

If equation (3) is false, then we consider case 7 and case 8 to determine the placement location b(i). When equation (3) does not hold, the robot arm and PCB table movement time can be expressed by the following equation:

$$\frac{d_{b(i-1),f(i)}}{Vr}+Tp+\frac{d_{f(i),b''(i)}}{Vr}\geq\frac{d_{B(i),b''(i)}}{V_b} \quad (7)$$

In all conditions for these cases, the robot has to move in X direction.

Case 7:

Similar to the case 5 except this case is considered when equation (3) false and equation (7) is true. If this case is satisfied, thus optimal movements are preserved even though the PCB table is not fast enough to position the i^{th} PCB coordinate at placement point b'(i).

Case 8:

If case 7 is not satisfied, equation (3) false and equation (7) is true, then the robot arm moves simultaneously in the X and Y direction while the PCB table also moves concurrently in the X direction to position the i^{th} PCB coordinate at the new relative placement position. The robot arm stops moving in X when it meets the placement location b"(i).

Then;

$$Dx=\frac{\left(abs(d_{B(i),b'(i)})-(\frac{abs(d_{b(i-1),f(i)})}{V_r}+T_p)*V_b\right)*Vr}{Vr+Vb};$$

$b(i)_x=f(i)_x + D_x$ if $c_{i-1,i}$ is positive, or

$b(i)_x=f(i)_x - D_x$ if $c_{i-1,i}$ is negative ; $Bm(i)=B(i)_x-b(i)_x$;

5. Methodology For Component Placement Sequencing And Feeder Setup

Since our work is focusing on improving the robot motion, we follow the method used in Wang et al. [5] in determining the component placement sequence and the feeder setup. This allows us to make a fair comparison with Wang's approach in our experiments. To ascertain the component placement sequence, the placement points are sequenced from left to right starting with the smallest X at the left lowermost corner of the PCB then with larger Y if more than one coordinate has the same value of X. To decide the feeder setup, components are assigned to a specific feeder slot such that the total exchange frequency of all adjacent slot pairs has the maximum value. The exchange frequency is an index that counts the exchange frequency between two different component types for succeeding pickups. For example, if the i^{th} placement sequence involves *a* component type followed by *b* component types for the $i^{th}+1$ placement sequence, the exchange frequency between component type *a* and *b* is counted as '1' ($F_{ab}=1$). The feeder setup problem is converted to a traveling salesman problem by associating a feeder slot as a node (or city) and the exchange frequency as the arc (or distance) connecting the two nodes (or cities).

The algorithm of the feeder setup is (adopted from Wang et al [5]):

a) Generate a K by K matrix of exchange frequency for each pair of component type based on the previously obtained component placement sequence.
b) Symmetrically add, $F_{pq} + F_{qp}$, where the exchange frequency between component type *p* and type *q* is fixed regardless of whether the pickup order is from component type *p* to type *q* or otherwise.
c) Subtract from a large number (a number larger than all values in the matrix) in order to convert the feeder setup problem to a traveling salesman problem such that the aim is to find the shortest path.
d) Assign components to feeder slot by applying any heuristic that can be applied to the traveling salesman problem.

In this work we only use a constructive heuristic to arrange the feeder slots since our work only focuses on the robot motion control specification. However, we believe that by applying an even better heuristics in the feeder setup we can gain even better assembly cycle times by reducing the feeder and PCB table movements.

6. Testing And Results

In our experiments we assume that the PCB and feeder carrier are positioned adjacent to each other in order to minimise the robot arm travel distance [4]. The placement points are generated randomly. Components are assigned to a specific feeder slot such that the total exchange frequency of all adjacent slot pairs has the maximum value. We apply the seven factors (table 1) of parameters as used in [6]. The pick up and placement time are set as 0.5 unit time and the size of each feeder slot is 4 unit lengths. To demonstrate the performance of our approach, we choose the length of the PCB, the width of the PCB, the speed of robot arm, the speed of feeder carrier, the speed of PCB table as 40, 15, 12, 2.5 and 3 respectively (as shown in table 1). The assembly points are chosen as 50 or 100 while the number of component types are 5, 10, 20, 30 or 40 (also shown in table 1). We assume that all components use the same nozzle and the speed of robot arm, PCB table and feeder carrier are fixed for all components. The computational results are summarised in table 2 and are averaged over five runs.

Table 1 Seven factors of experimental design by [6]

Factors	**Levels (low/high)**
Number of assembly points (N)	50/100
Number of component types (K)	5/10/20/30/40
Length of PCB (BL)	40 (unit distance)
Width of PCB (BW)	15 (unit distance)
Speed of robot (Vr)	12 (unit distance/unit time)
Speed of feeder carrier (Vf)	2.5 (unit distance/unit time)
Speed of PCB table (Vb)	3 (unit distance/unit time)

The results show that our approach is superior to Wang's [5] in all tests. Our approach performs, an average, 3.29% better than Wang's when considering assembly cycle time, 55.54% improvement of optimal movements, shorter feeder movement distance (10.21% improvement) and shorter PCB movement distance (19.12% improvement) compared to Wang's approach. By reducing the assembly cycle time we can increase the throughput rate of surface mount placement machine. In addition, the life cycle of surface mount placement machine can be prolonged since our approach also minimises the movement of PCB table and feeder carrier.

Table 2 An average results of five runs

Combination		CT (assembly cycle time)			Optimal movement			Feeder movement distance			PCB table movement distance		
N	K	CDPP	WA	I (%)	CDPP	WA	I (%)	CDPP	WA	I (%)	CDPP	WA	I (%)
50	5	117.25	119.28	1.73	88	62.8	40.13	148.18	168.30	13.58	106.85	143.79	34.57
50	10	127.57	132.40	3.79	72.2	46.8	54.27	201.85	227.87	12.89	164.06	212.13	29.30
50	20	151.96	158.73	4.46	55.2	35.8	54.19	286.96	317.19	10.54	285.39	332.50	16.51
50	30	168.00	173.42	3.22	56.4	35.6	58.43	364.91	396.98	8.79	337.29	372.43	10.42
50	40	167.06	170.67	2.16	66.6	48.4	37.60	403.95	436.15	7.97	297.37	324.16	9.01
100	5	230.95	234.66	1.61	174.8	129	35.50	288.63	333.65	15.60	174.43	231.50	32.72
100	10	248.03	258.40	4.18	143	87.2	63.99	406.66	456.59	12.28	317.28	411.27	29.62
100	20	323.93	339.00	4.65	102.6	58.4	75.68	635.51	687.07	8.11	630.70	713.15	13.07
100	30	384.66	400.86	4.21	85.6	48.4	76.86	789.21	852.58	8.03	846.73	931.13	9.97
100	40	465.12	478.65	2.91	80	50.4	58.73	1019.92	1064.24	4.35	1104.91	1171.19	6.00
Average:				3.29			55.54			10.21			19.12

Note: WA=Wang's approach
I =Improvement over Wang's approach

7. Conclusion

In the DPP model, the pickup and placement points were dynamically changed based on the movement of the robot arm, the PCB table and the feeder carrier where all of them can vary their speed. Su et al. [4] argue that the DPP was superior to FPP. However, we found that the Wang's DPP model only considered the current movement. The Wang's DPP model tried to maintain the fixed pickup and placement location as much as possible unless this leads to robot idling. Hence, the DPP model may still had unnecessary movement. Thus, in our CDPP, we eliminated the unnecessary movement by looking forward to the next PCB coordinate when determining the current pickup location and looking forward the next feeder slot when determining the current placement location. We formulated a problem for a placement machine that was a type of cartesian robot which has a single head equipped with a single nozzle. The robot was able to move in both X and Y direction concurrently to pick and place a component. The robot, PCB table and feeder carrier can move simultaneously. The robot traveled between feeder carrier and PCB table for picking and placing a component, respectively. The formulations are constructed based on the triple objectives of minimising robot assembly time, feeder movements and PCB table movements. The main difference between our CDPP model and the previous DPP (and EDPP) was that our CDPP calculated the robot arm movement distance as the maximum of the movement in Y or the movement in X (a chebychev distance) since our robot arm can move in X-axis and Y-axis concurrently, whilst the previous DPP (and EDPP) calculated the robot arm movement as a euclidean distance. This work has shown an improvement compared to Wang's DPP approach. Therefore, we plan to apply this approach to more sophisticated placement machines (a type of cartesian robot) that has more than one head. We also plan to use different heuristics for the feeder setup in order to gain even better results.

References

[1] Fu, H. –P. and Su, C. –T., A comparison of search techniques for minimizing assembly time in printed wiring assembly, Int. J. of Production Economics, 63, 2000, pp. 83-98.

[2] Wang, W., Nelson, P.C. and Tirpak, T.M., Optimization of high-speed multistation SMT placement machines using evolutionary algorithms, IEEE Transactions on Electronics Packaging Manufacturing, Vol. 22(2), April 1999, pp. 137-146.

[3] Kumar, R. and Li, H., Integer programming approach to printed circuit board assembly time optimization, IEEE Transactions on Components, Packaging and Manufacturing Technology, Nov 1995. Vol.18(4), pp. 720-727.

[4] Su, Y.-C., Wang, C., Egbelu, P.J. and Cannon, D.J., A dynamic point specification approach to sequencing robot moves for PCB assembly, Int. J. Computer integrated Manufacturing, 8(6), 1995, pp. 448-456.

[5] Wang, C., Ho, L.-S. and Cannon, D.J., Heuristics for assembly sequencing and relative magazine assignment for robotic assembly, Computers and Industrial Engineering 34, 1998, pp. 423-431.

[6] Wang, C., Ho, L.S., Fu, H.P and Su, Y.C., A Magazine Assignment Heuristic for Robotic Assembly Using the Dinamic Pick and Place Approach, Int. J. of Industrial Engineering, 4(1), 1997, pp. 24-33.

[7] Hop, N.V and Tabucanon, M. T., Multiple criteria approach for solving feeder assignment and assembly sequence problem in PCB assembly, in the Production Planning and Control vol.12 no. 8, Dec 2001, pp. 736-744 .

[8] Hop, N.V and Tabucanon, M. T., Extended dynamic point specification approach to sequencing robot moves for PCB assembly, in the International Journal of Production Research Vol. 39(8), May 2001, pp. 1671-1687.

[9] Ahmadi, R.H. and Mamer, J.W., Routing heuristics for automated pick and place machines, European Journal of Operational Research 117, 1999, pp.533-552.

[10] Su, C. and H. Fu, A simulated annealing heuristic for robotics assembly using the dynamic pick-and-place model, in the Production Planning and Control 9, 1998, pp. 795-802.