

A triple objective function with a Chebychev dynamic pick-and-place point specification approach to optimise the surface mount placement machine

Masri Ayob ^{*}, Graham Kendall

*Automated Scheduling, Optimisation and Planning (ASAP) Research Group, University of Nottingham,
School of Computer Science and IT, Nottingham NG8 1BB, UK*

Received 1 October 2002; accepted 1 September 2003
Available online 7 May 2004

Abstract

Optimisation can play a major role in improving the throughput of surface mount placement machines. Most previous work has reported on improving only the assembly cycle time. The movement of the feeder carrier and PCB table are not always factors which are minimised. In this paper we introduce a triple objective function with a Chebychev dynamic pick-and-place approach to optimise the sequential pick and place machine. We are focusing on improving the feeder setup. The aims are to minimise the robot assembly time, the feeder movements and the PCB table movements. To provide flexibility to our approach, we integrate three weighted parameters into the triple objective function such that one can vary the importance of each factor to be minimised. Experimental results show that our approach gives good robot assembly time and less movement of the feeder carrier and PCB table.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Scheduling; Optimisation; SMD placement machine; Printed circuit board assembly

1. Introduction

A printed circuit board (PCB) consists of copper wire printed on a board which makes up a circuit for electronic devices. Hundreds, or even thousands, of electronic components are placed onto the PCB by using a surface mount device (SMD) placement machine to produce a complete PCB. Modern electronic products increasingly demand smaller PCB's, which are often produced using an automated assembly process in order to produce the high volumes required [1]. To remain competitive, manufacturers have to continually improve the efficiency of their assembly line [2,3]. Reducing PCB assembly time to increase the efficiency of robotic assembly machines has become a critical issue within the electronics industry. Moreover, because of the high purchase price, typically more than \$300,000, and operating costs of the SMD placement machine,

^{*} Corresponding author. Tel.: +44-115-846-6525; fax: +44-115-951-4254.

E-mail addresses: mx@cs.nott.ac.uk (M. Ayob), gk@cs.nott.ac.uk (G. Kendall).

the surface mount technology (SMT) lines are designed so that the SMD placement machines are the limiting resource or “bottleneck” [4]. They are often the focus of productivity improvement efforts in the electronic manufacturing company. The robot motion control, sequence of placement points, and magazine slot assignment, are some of the determining factors of assembly efficiency [5]. Furthermore, automated assembly can provide consistent quality and can increase the rate of production [6].

Various approaches have been attempted to improve the sequence of placement points and/or feeder slot assignment for the PCB assembly process [7]. Egbelu et al. [8] investigated assigning components to feeder slots and sequencing component placement onto the PCB in order to minimise the total assembly cycle time. They classify machines depending upon whether the PCB table and the feeder carrier are stationary or not. They developed rules such as the centroid rule, the proportion rule, the partition rule and the weighted rule to initially assign components to feeder slots. To obtain the component insertion sequence, they model the problem as a travelling salesman problem (TSP). Finally, by having the component insertion sequence and the initial feeder setup, Egbelu et al. [8] converted the component slot assignment into a quadratic assignment problem (QAP) where the cutting plane and exchange heuristic were used. Results show that the assembly cycle time were minimised when both the feeder carrier and PCB table are able to move. Ahmadi and Mamer [9] considered two interdependent sub-problems. These being the problem of sequencing the component types for placement and the problem of scheduling the movement between points on the PCB. They model the problems as a collection of interdependent travelling salesman problems. The sequencing of the component types was transformed to the problem of finding the cheapest way to traverse the required arc, starting and ending at the placement device’s resting position. The implementation process shows that the approximation of the problem by a sequence of TSPs was adequate to produce significant increases in throughput. Some works related to feeder setup and/or component placement sequencing can be found in [10–13].

Most of the published work only aims to minimise the assembly cycle time. The movements of the feeder carrier and the PCB table are not normally a factor to be minimised. In the case where the feeder carrier and the PCB table are moveable, we should also consider minimising their movements. Hence, our work proposes a new objective function that attempts to minimise the assembly cycle time together with the minimisation of the movement of the feeder carrier and the PCB table. By integrating the three minimisation factors, we can still achieve an assembly cycle time as good as those which only aims to minimise the assembly cycle time. Moreover, we can gain improved feeder carrier movement and PCB table movement. Reducing these movements may also prolong the life cycle of placement machine even if it does not affect the throughput rate of the machine.

Hop and Tabucanon [14] tried to minimise the assembly cycle time together with minimisation of the feeder carrier movement and the PCB table movement. They proposed a new heuristic which they called a multiple criteria approach and improve on the results of Wang [15]. The approach was based on the fact that assembly time depends on the relative position of pickup and placement points (DPP model). The multiple criteria approach uses the trade-off between two strategies:

- (i) Minimise the PCB table travel distance, where the strategy is ‘assemble by area’.
- (ii) Minimise the feeder rack travel distance, where the strategy is ‘assemble by component type’.

The idea is to place the same component types as close to each other as possible. The algorithm starts with an initial solution using Wang’s method [15]. In Wang’s approach, the component placement sequence starts with components which have smaller X coordinates (and larger Y if duplicate X ’s). Wang et al. [15] assigned feeders to slots such that the total exchange frequency of all adjacent slot pairs has a maximum value. Hop and Tabucanon [14] construct an initial feeder setup and a placement sequence using Wang’s method. Then a new feeder setup is constructed based on assemble by component type. Next, the assembly sequence is generated based on the principle of assemble by area. These two steps are iterative, and will be

terminated when there is no improvement in assembly time or the number of iterations reaches a pre-defined limit. Computational results indicated that the new approach performed better than the approach by Wang in terms of total assembly time. The reduction in time ranged between 4% for lower density board, to 28% for higher density board. It showed that when the board density increases, the chances of a robot arm waiting for the PCB table movement will reduce. However, Hop and Tabucanon did not measure the saving of the feeder carrier movement and the PCB table movement. They only hoped that their strategy was able to reduce the PCB table movement and feeder carrier movement.

This paper uses a Chebychev dynamic pick-and-place (CDPP) approach which we proposed in [7]. We apply the triple objective function to the CDPP approach. The approach will eliminate the unnecessary movement by looking forward to the next PCB coordinate when determining the current pickup location and looking forward to the next feeder slot when determining the current placement location. Instead of only searching for a minimum assembly cycle time, we are also looking for a minimum assembly cycle time and reducing feeder carrier movement and PCB table movement as far as possible.

The remainder of the paper is organised as follows: In Section 2 we explain the general features of surface mount device (SMD) placement machines and the type of machine we have studied here. Section 3 briefly describes the fixed pick-and-place model. Dynamic pick-and-place (DPP) model and their formulations are discussed in Section 4. A proposed new Chebychev DPP (i.e. CDPP) is described in Section 5. Detailed formulations of CDPP are also shown in Section 5. In Section 6, we give a description of the objective function that aims to minimise the assembly cycle time together with the minimisation of the movement of the feeder carrier and the PCB table. Computational experiments are then reported in Section 7. Section 8 presents our conclusion.

2. Surface mount device (SMD) placement machine

SMD placement machines are categorised into five categories based on their specification and operational methods. These are dual delivery, multi-station, turret style, multi-head and a sequential SMD placement machine [16]. In general, each SMD placement machine has a feeder carrier (sometimes called a feeder magazine), PCB table (or worktable), head(s), nozzle(s) (or gripper(s)) and a tool magazine. The PCB table, the feeder carrier and the head(s) can either be moveable or fixed. A typical feeder carrier consists of either several tape reels or vibratory ski slope feeders, or both [17]. The feeder reels, that is tapes holding electronics components; or vibratory ski slope feeders are positioned in the feeder slots according to the arrangement given by the feeder setup [16]. Usually, larger components are supplied by tray feeders [18]. Some machines allow a tray to be placed into the machine feeding area whilst the others have an automatic tray-handling unit. Normally, the components are transported from the feeders to the placement position on the PCB using vacuum nozzle(s) that are placed at the end of the head(s) [18]. The PCB table is required to support the PCB, in a locked position, during placement operations. Some machines may have more than one PCB table. The PCB table(s) can be a stationary, use a conveyor system, or use an X – Y motion table [16].

This work only focuses on a problem of a sequential pick and place machine that has a single arm with a single head equipped with a single nozzle. The robot (that is the arm and head) is able to move in both X and Y directions concurrently to pick and place a component. The placement machine uses a nozzle to grasp a component from the feeder carrier and then mounts it onto the PCB. The feeder carrier and the PCB table are moveable in the X -axis to position the component pickup coordinate and the placement coordinate of the PCB, respectively. The robot, PCB table and feeder carrier can move concurrently. The robot travels between feeder carrier and PCB table for picking and placing a component, respectively.

3. Fixed pick-and-place (FPP) model

In the FPP model, the feeder carrier can move horizontally (along the X -axis) to position a required component at the required pickup location. The PCB table can move, concurrently in the X/Y -axis to position a PCB coordinate at the required placement location. The robot arm can only move in the Y -axis between fixed pickup and placement locations [5] (see Fig. 1). Since the robot arm only moves between these two fixed locations, and the speed of the PCB table, the feeder carrier and the robot arm varies and, furthermore, their travelling distances also differs, there may exist an undesirable robot waiting time.

4. Dynamic pick-and-place (DPP) model

In the DPP model, both the feeder carrier and PCB table are still moveable but only in the X -axis, whilst the robot arm moves in the Y -axis, in optimal conditions [7]. This case only happens when the feeder carrier or PCB table can move within 'free' movement time, that is when the feeder carrier and PCB table can move to the best pickup and placement point before the robot arm arrives. However, when the robot arm can arrive at the best pickup location before the feeder carrier can bring the required component to the best pickup location, or when the robot arm can arrive at the best placement location before the PCB table can position the required PCB coordinate at the best placement location, then the robot arm has to move at an angle from the Y -axis to catch the feeder carrier or PCB table at a dynamically allocated coordinate to

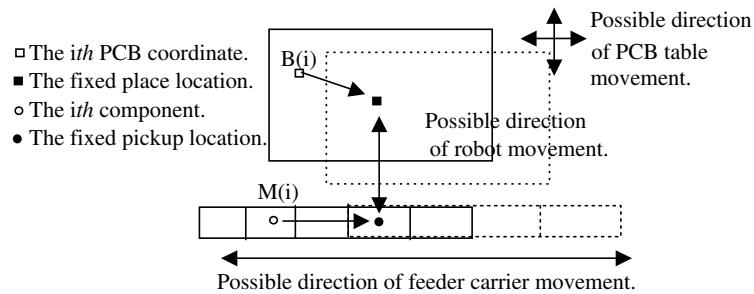


Fig. 1. The FPP model.

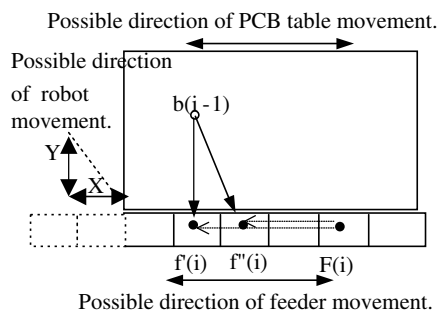


Fig. 2. The DPP model for determining pickup location $f(i)$, the feeder will carry the component from $F(i)$ to $f'(i)$ if no robot interception occurs or from $F(i)$ to $f''(i)$ otherwise.

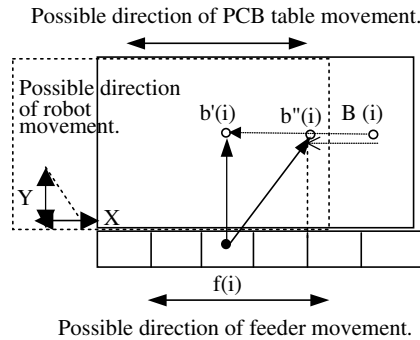


Fig. 3. The DPP model for determining placement location $b(i)$, the PCB table will move to position the placement point from $B(i)$ to $b'(i)$ if no robot interception occurs or from $B(i)$ to $b''(i)$ otherwise.

avoid robot idling [5]. Both the PCB table and the robot arm, or the feeder carrier and the robot arm, will stop and meet at the dynamically assigned interception location at the same time [19]. This situation is defined as a robot interception. In the DPP model, the robot arm can pick and place a component at any location along the feeders and the PCB, respectively. Figs. 2 and 3 demonstrate how the pickup and placement location are determined in the DPP model.

The following notations are used (most of them adopted from [15]) to describe the DPP and the CDPP models:

- CT: the cycle time to assemble all components;
- N : the number of PCB points;
- K : the number of component types (each feeder slot holds one component type only);
- $c(i)_{x,y}$: the i th x, y coordinate on the PCB which will have the i th component placed there;
- $f(i)_x$: the feeder pickup coordinate of the i th assembly sequence. The $f(0)_x$ is defined as the centre of the first pickup location (referred to 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 , $b(i)_y$ 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 components of type a and b ;
- $F_m(i)$: 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, $f(i)_{x,y}$;
- $T_m(i)$: the moving distance and direction of the 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 $(i-1)$ th component in the assembly sequence (positive sign means the i th 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 $(i-1)$ th points on the PCB board (positive sign means the X -coordinate of the i th point on PCB is bigger than the $(i-1)$ th 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);
- $B(i)$: the location of the i th PCB coordinate when the robot arm is placing the $(i - 1)$ th component at $b(i - 1)$;
- $F(i)$: the feeder slot location of the i th component when the robot arm is picking up the $(i - 1)$ th component at $f(i - 1)$.

When the robot moves from the $(i - 1)$ th pickup location on the feeder carrier, $f(i - 1)$, to the $(i - 1)$ th placement point on the board, $b(i - 1)$, the feeder carrier concurrently moves to position the next component at the pickup location $f'(i)$ on the feeder carrier (refer to Fig. 2). Similarly, when the robot moves from the $(i - 1)$ th placement point, $b(i - 1)$, to pickup the i th component at $f(i)$, the PCB table simultaneously moves from the placement point $b(i - 1)$ to position the i th PCB coordinate, $c(i)_{x,y}$ point, at the best placement location, $b'(i)$ (refer to Fig. 3). Since the robot arm, the feeder carrier and the PCB table move at different speeds and vary in the distance they need to travel, the next possible pickup and placement locations are dynamically determined to avoid robot idling. If the feeder carrier can move fast enough to bring the i th component from $F(i)$ to $f'(i)$, then the next possible pickup location is $f'(i)$ where the X coordinates of $b(i - 1)$ and $f'(i)$ are the same and $f(i) = f'(i)$ (refer to Fig. 2). Otherwise, the robot arm will pick the i th component when it meets the feeder carrier at the interception location, $f''(i)$ and $f(i) = f''(i)$. Similarly, in determining the next placement location, if the PCB table can move fast enough to position the i th PCB coordinate, $c(i)_{x,y}$, from $B(i)$ to $b'(i)$, then the next possible placement point is $b'(i)$ where the X coordinates of $b'(i)$ and $f(i)$ are the same and $b(i) = b'(i)$ (refer to Fig. 3). Otherwise, the robot arm will place the i th component onto the i th PCB coordinate when it meets the PCB table at the interception location, $b''(i)$ and $b(i) = b''(i)$. For every pickup and placement operation, the movement of the robot arm is unavoidable. Hence, the time taken for the robot arm movement is a major contribution to the total CT (by ignoring the other factors such as machine down time, nozzle change etc.). However, the effect of the movements of the feeder carrier and the PCB table can be eliminated. If the pickup and placement sequence are optimally scheduled, and a location of where the robot arm should pick and place a component are well designed, then the PCB table and the feeder carrier may move within a free movement time (i.e. they can move within the time taken for the robot arm movement) and their movements do not affect the CT. Hence, one of the aims 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

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

Wang et al. [15] claimed that the shortest robot traveling distance occurred 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 occurs when Eq. (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 $f'(i)_x = b(i - 1)_x$.

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

The optimal placement occurs when Eq. (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 $b'(i)_x = f(i)_x$.

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

When Eq. (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 waiting for the feeder carrier at $f'(i)$, the robot arm will move at an angle from the Y -axis from $b(i-1)$ 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 Eq. (4).

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

Similarly, when Eq. (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 Eq. (5).

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

5. A Chebychev dynamic pick-and-place (CDPP) model

The CDPP model allows the robot to move in X and Y direction simultaneously. Interestingly, the optimal robot movement can still be preserved even if the robot has to move in the X direction as long as the movement in Y takes longer time than the X movement. This means that the CDPP approach can move the robot in the X direction without adding extra cost to the assembly cycle time, CT. On the contrary, the extended dynamic pick-and-place (EDPP) approach proposed by Hop and Tabucanon [20] will pay an extra cost when the robot moves in the X -axis. The EDPP model arranged the pickup and placement locations on the PCB based on the global view of the point relationship in the system. The movement of the robot arm, the movement of the PCB table and the movement of the feeder carrier were being considered to reduce the assembly cycle time. When the feeder carrier (or PCB table) can move fast enough and arrive at the required pickup (or placement) location earlier than the robot arm, then 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 placement (pickup) location for the next movement. This means that the EDPP was 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. Hop and Tabucanon [20] formulated three different cases; fixed PCB table, dynamic feeder carrier and robot arm; fixed feeder carrier, dynamic PCB table and robot arm and dynamic PCB table, feeder carrier and robot arm. The EDPP model obtained better assembly cycle time compared to DPP model designed by Su et al. [19].

Our CDPP approach differs from [15,19–21] as we allow the robot to move in the X and Y direction concurrently whenever necessary, even in the case where the feeder carrier (or PCB table) can move within free movement time. That is, when the feeder carrier (or PCB table) can arrive at the best current pickup (placement) location earlier than the robot arm. Initially, we consider the best current pickup (placement) location as defined by Wang et al. [15]. That is, the pickup (placement) operation which involves no robot

motion in the X -axis. When the feeder carrier is not able to bring the i th component to the i th pickup location, $f'(i)$, or when the PCB table is not able to bring the i th placement point to the i th placement location, $b'(i)$, then robot interception occurs (refer to Figs. 4 and 5). When the robot interception occurs, Fu and Su [5] allow the robot arm to move at an angle from the Y -axis to catch the feeder carrier or the PCB table at a certain point, while our approach moves the robot arm in X -axis and Y -axis simultaneously and the traveling time is dictated by the maximum of X or Y distance (Chebychev distance). If the Y distance is greater than the X distance, the robot arm still performs an optimal movement, even though the feeder and/or PCB table are not fast enough to bring the best pickup/place point on time without the movement of robot arm in the X -axis. Allowing the robot to move in the X direction, even though in the optimal movement, this may also increase the chance of an optimal movement for the next placement or pickup operation [7].

Originally, we assumed that the optimal pickup happens if Eq. (2) is true and the optimal placement occurs when Eq. (3) is true. An example of these movements is shown in Figs. 4 and 5, respectively. In fact, in the CDPP approach, the optimal pickup or optimal placement can still be preserved even though Eq. (2) or Eq. (3) do not hold as long as the movement of the robot arm in Y takes longer than the movement in X . By default, in optimal feeder carrier movement this is the case when Eq. (2) is true, the i th pickup point is $f'(i)$ (that is $f(i) = f'(i)$) while $f'(i)_x = b(i-1)_x$. Similarly, in optimal PCB table movement, it is assumed that the i th placement point is $b'(i)$ (that is $b(i) = b'(i)$) and $b'(i)_x = f(i)_x$. This means that in order to test Eq. (2) or (3), we assume the robot only moves in the Y -axis from $b(i-1)$ to $f'(i)$, or from $f(i)$ to $b'(i)$ respectively.

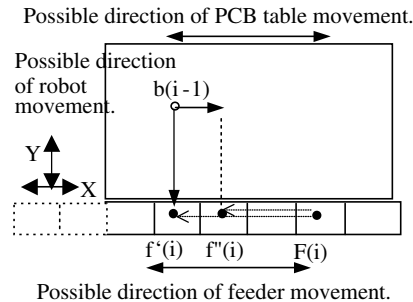


Fig. 4. CDPP model for determining pickup location $f(i)$, the feeder will carry the component from $F(i)$ to $f'(i)$ if robot does not need to move in the X -axis or from $F(i)$ to $f''(i)$ otherwise.

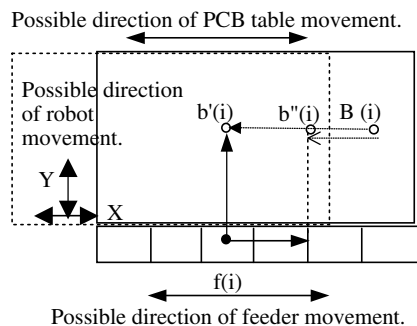


Fig. 5. CDPP model for determining placement location $b(i)$, the PCB table will move to position the placement point from $B(i)$ to $b'(i)$ if robot does not need to move in the X -axis or from $B(i)$ to $b''(i)$ otherwise.

The detailed formulation of the CDPP approach has been presented in [7]. However, to clearly discuss our triple objective function with the CDPP approach, it may be beneficial to rephrase the CDPP formulation in this paper.

The detailed calculation of Eq. (2) for the CDPP formulation are:

- (a) $d_{f(i-1),b(i-1)} = \max(d_{f(i-1)_x,b(i-1)_x}, d_{f(i-1)_y,b(i-1)_y})$; Since the robot can move simultaneously in both the X and Y axis, we use the Chebychev distance and calculate the distance between the pickup point $f(i-1)$ and the placement point $b(i-1)$ as the maximum of the movement in the X and Y direction.
- (b) $d_{b(i-1),f'(i)} = d_{b(i-1)_y,f'(i)_y}$ where $f'(i)_y = 0$; Initially, it is assumed that the robot only moves in the Y direction from the placement point $b(i-1)$ to the next pickup point $f'(i)$.
- (c) $F(i) = F(i)_x = f(i-1)_x + s_{i-1,i}$; $F(i)$ is a relative feeder slot location (referring to the origin point) containing the i th component in the assembly sequence when location $f(i-1)$ and $b(i-1)$ are already known.
- (d) $d_{F(i),f'(i)} = d_{F(i)_x,b(i-1)_x} = F(i)_x - b(i-1)_x$; A distance between the point $F(i)$ and the next possible pickup point $f'(i)$. Initially, the next possible pickup is assumed as $f'(i)$ that is $f(i) = f'(i)$ where $f'(i)_x = b(i-1)_x$. Originally, the feeder carrier should move up to this distance to have an optimal movement. A positive value means the $F(i)$ is located at the right side of $b(i-1)$, negative otherwise.

The detailed calculations of Eq. (3) for the CDPP formulation are

- (a) $d_{b(i-1),f(i)} = \max(d_{b(i-1)_x,f(i)_x}, d_{b(i-1)_y,f(i)_y})$; Since the robot can move simultaneously in both the X and Y axis, we use the Chebychev distance and calculate the distance between the placement point $b(i-1)$ and the pickup point $f(i)$ as the maximum of the movement of the robot arm in the X and Y direction.
- (b) $d_{f(i),b'(i)} = d_{f(i)_y,b'(i)_y} = d_{f(i)_y,c(i)_y}$ where $f(i)_y = 0$ and $b' = c(i)_y$; Initially, it is assumed that the robot only moves in the Y direction from the pickup point $f(i)$ to the next placement point $b'(i)$.
- (c) $B(i) = B(i)_{x,y}$ where $B(i)_x = b(i-1)_x + c_{(i-1)_x,(i)_x}$ and $B(i)_y = c(i)_y$; $B(i)$ is a relative coordinate on the PCB (referring to the origin point) where the i th component has to be placed on the PCB during the i th assembly sequence when location $b(i-1)$ and $f(i)$ are already known.
- (d) $d_{B(i),b'(i)} = d_{B(i)_x,f(i)_x} = B(i)_x - f(i)_x$; A distance between the point $B(i)$ and the next possible placement point $b'(i)$. Initially, the next possible placement is assumed as $b'(i)$ that is $b(i) = b'(i)$ where $b'(i)_x = f(i)_x$. Originally, the PCB table should move up to this distance to have an optimal move.

In order to simplify our terms used in the CDPP formulation we introduce four variables:

- (a) $p = \frac{d_{f(i-1),b(i-1)}}{V_r} + T_i$ is the total time taken by the robot arm to move from the pickup point $f(i-1)$ to the placement point $b(i-1)$ together with the time to place the $(i-1)$ th component.
- (b) $q = \frac{d_{b(i-1),f(i)}}{V_r} + T_p$ is a total time taken by the robot arm to move from the placement point $b(i-1)$ to the pickup point $f(i)$ together with the time to pickup the i th component.
- (c) $R = pV_f$ is a distance made by the feeder carrier by moving from coordinate $F(i)$ to position the i th component for the next pickup operation, at p time.
- (d) $Q = qV_b$ is a distance made by the PCB table by moving from coordinate $B(i)$ to position the i th coordinate on the PCB for the next placement operation, at q time.

In optimal pickup (Eq. (2) is true), the CDPP considers 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 $(i-1)$ th PCB coordinate $c_{i-1,i}$ is greater than $d_{F(i),f'(i)}$; and the i th PCB coordinate is located in the direction in which the robot arm is moving; 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

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

$$F_m(i) = F(i)_x - f(i)_x = [F(i)_x] - [b(i-1)_x + (F(i)_x - b(i-1)_x)] \\ = 0 \quad \text{then there is no feeder movement.}$$

An example of this case 1 is shown in Fig. 6.

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 Eq. (2) is true. Then

$$D_x = 0;$$

$$f(i)_x = b(i-1)_x \quad (\text{similar to Wang's approach});$$

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

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

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

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 considers when Eq. (2) is false. If this case is satisfied, then 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 case 3 is not satisfied and Eq. (2) is 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

$$D_x = \frac{[\text{abs}(d_{F(i),f'(i)}) - R] V_r}{V_r + V_f};$$

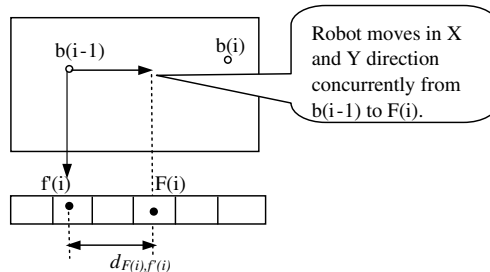


Fig. 6. An example of case 1 where the robot moves in X and Y direction simultaneously from $b(i-1)$ to $F(i)$ to pick the i th component whilst the feeder carrier does not move at all.

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

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 two cases when Eq. (3) is true (optimal movements):

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 $(i+1)$ th components is greater than $d_{B(i),b'(i)}$, and the feeder slot containing the $(i+1)$ th component is located in the direction in which the robot arm is moving; and the value of the Y coordinate of the i th PCB coordinate ($c(i)_y$) is greater than the absolute value of $d_{B(i),b'(i)}$. Then

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

$$b(i)_x = f(i)_x + D_x;$$

$$T_m(i) = B(i)_x - b(i)_x = 0 \quad \text{then there is no PCB table movement.}$$

Case 6: The robot only moves in the Y -axis to place a component at placement location $b(i)$ if Eq. (3) is true but case 5 is not satisfied.

Then

$$D_x = 0;$$

$$b(i)_x = f(i)_x \quad (\text{similar to Wang's approach});$$

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

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

$$\frac{d_{b(i-1),f(i)}}{V_r} + T_p + \frac{d_{f(i),b(i)}}{V_r} \geq \frac{d_{B(i),b(i)}}{V_b}. \quad (7)$$

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

Case 7: Similar to the case 5 except this case is considered when Eq. (3) is false and Eq. (7) is true. If this case is satisfied, then 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, Eq. (3) is false and Eq. (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

$$D_x = \frac{(\text{abs}(d_{B(i),b'(i)}) - Q)^* V_r}{V_r + V_b};$$

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

$$b(i)_x = f(i)_x - D_x \quad \text{if } c_{i-1,i} \text{ is negative;}$$

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

6. Triple objective formulation

Wang et al. [15] developed a heuristic for feeder setup by placing components in feeder slots such that all adjacent slots have maximum exchange frequency pickups involving multiples of the same component type. The exchange frequency is a counter that counts the exchange frequency between two different component types for succeeding pickups. For example, if the i th placement sequence involves g component type followed by z component type for the $(i + 1)$ th placement sequence, the exchange frequency between component type g and z is counted as '1' ($F_{gz} = 1$). Wang et al. [15] converted a feeder setup problem into a traveling salesmen problem by representing the slots as nodes and the exchange frequency between each slot pair as the length of an arc connecting the two nodes. In order to convert the problem of feeder setup into a traveling salesman model, Wang et al. [15] subtracts all the exchange frequencies between each slot pair from a large number (a number larger than all exchange frequency values). Hence, they arrange the feeder slots by searching for the minimum total exchange frequency. By doing this, Wang et al. [15] assumed that minimising the total exchange frequency can lead to the minimum assembly cycle time (CT).

However, Hop and Tabucanon [14] argued that assigning component feeders to slots based on maximum exchange frequency of the component types, may increase the feeder rack movement if components are not assembled by their type. Contrarily, our experimental results (refer to Table 2) show that Wang's approach for feeder setup can gain better (i.e. less) feeder movement. However, we also found that minimising (actually maximising) the total exchange frequency does not necessarily lead to the minimum CT. To show this, we ran 20 tests on a small data set using full enumeration to determine the optimum feeder setup. The placement sequences are generated based on Wang's approach. We defined three objective functions which are to minimise the exchange frequency, the CT and the triple objective function. Our triple objective function aims to minimise the CT whilst also minimising the feeder and PCB table movement. However, the main objective remains to minimise the CT but it would be beneficial if we can also minimise the feeder and PCB table movement. Reducing these movements may prolong the life cycle of placement machine even if it does not affect the throughput rate of the machine. By running a few preliminary tests we find that there is not much difference in CT among different feeder setups but a lot of variance in PCB table and feeder movements. If the triple objective function is determined without normalisation, then the output will be biased to have a better feeder movement and PCB table movement rather than better CT. Hence we normalise as follows:

Changes in CT:

$$\Delta C = \frac{\text{new}C - \text{old}C}{\text{new}C};$$

Changes in PCB table movement:

$$\Delta P = \frac{\text{new}P - \text{old}P}{\text{new}P};$$

Changes in feeder movement:

$$\Delta M = \frac{\text{new}M - \text{old}M}{\text{new}M},$$

where C , P and M are referring to CT, PCB table movement and feeder movement respectively. ΔC is calculated as a difference of CT between the current solution and the best solution previously obtained. Likewise, ΔP and ΔM are calculated in a similar way. A negative value means that the new solution is better than the best solution previously obtained. To introduce more flexibility into our triple objective function, we incorporate the changes of CT, PCB table movement and feeder movement with weighted parameters. The weighted parameters will determine the importance of the CT, PCB table movement and feeder movement. Therefore, the triple objective function is

$$F = \min (W_c * \Delta C + W_p * \Delta P + W_m * \Delta M), \quad (8)$$

where W_c , W_p and W_m are the weighted parameters for CT, PCB table movement and feeder movement respectively. By adapting the weighted parameters, we can gain better CT without ignoring the feeder movement and PCB table movement. Of course, if one assumes that CT is the most important factor, then the W_c parameter can be set to a suitably large value to ensure that the algorithm will search for a better CT and then when there are more than one solution having the same value of CT, the algorithm will select the better PCB table movement and feeder movement.

In Eq. (8), we are looking for the minimum value of F . F is calculated as the sum of changes in the CT, PCB table movement and feeder carrier movement and the multiplication of their weighted parameters. The comparison is made with the best solution that was previously obtained in the enumeration. A positive value of F means that the current solution is worse and zero value indicates that the current solution is similar to the best solution obtained in terms of the F value. However, a negative value of F shows that the current solution is superior to the best solution been obtained. If a negative value of F is obtained, then the best solution is updated with the new solution.

7. Implementation, testing and results

Since our work is focusing upon improving the feeder setup, we follow the method used in Wang et al. [15] in determining the component placement sequence. To schedule 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, we use three approaches:

- (a) In minimising (actually maximising) the exchange frequency, components are assigned to a specific feeder slot such that the total exchange frequency of all adjacent slot pairs has the maximum value (as in Wang's approach). The feeder setup problem is converted to a traveling salesman problem (TSP) 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. [15]):
 - Generate a K by K matrix of exchange frequency for each pair of component types based on the previously obtained component placement sequence.
 - Symmetrically add, $F_{gz} + F_{zg}$, where the exchange frequency between component type g and type z is fixed regardless of whether the pickup order is from component type g to type z or vice versa.
 - 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.
 - Assign components to feeder slots by applying any heuristic that can be applied to the traveling salesman problem.

In this approach we use a constructive heuristic (nearest neighbourhood) to assign components to feeder slots by choosing the first component type to be placed onto the PCB and assigning it to slot 0 (as an initial city in the TSP problem). Then, the rest of the components are assigned to the slots using the nearest neighbourhood heuristic to construct an initial feeder setup. Starting from initial feeder setup, we run a full enumeration with the aim of minimising the total exchange frequency of all adjacent slot pairs. In every iteration, the new solution is compared to the best obtained solution, in this case the solution which has the minimum exchange frequency.

- (b) In minimising the assembly cycle time (CT), components are assigned to a specific feeder slot such that the total CT is minimum. To simplify our work, we use the same constructive heuristic as in the first approach to generate an initial solution. Instead of searching for a minimum exchange frequency, we

directly search for the minimum CT. Again, we run a full enumeration to obtain an optimal CT (shown in Table 2; CT under Minimise CT). In each iteration, the CT of the current solution is compared to the best obtained CT.

- (c) Finally, in minimising the triple objective function (F), component feeders are arranged in the feeder slots such that the CT and the movement of PCB table and feeder carrier are minimised. Again, we use the same constructive heuristic to obtain the initial feeder setup. A full enumeration is performed to find the global optimal solution which has the minimum CT with a better movement of feeder carrier and PCB table. The optimal solution is dependent on the chosen value of weighted parameters (W_c , W_p and W_m). If W_c is a suitably large value, then the optimal solution will be the optimal CT but if there exist solutions with the same CT value, then the solution which has better feeder carrier and PCB table movement will be chosen.

In our experiments we assume that the feeder carrier and PCB table are positioned close to each other in order to minimise the robot arm travel distance [9]. We also assume that the components are already assigned to the placement machine and all components are the same size. In our experiments we choose $W_c = 20$, $W_p = 1$ and $W_f = 1$. The placement points are generated randomly. We apply the seven factors (Table 1) of parameters as used in [19,20]. The pick up and placement time are set as 0.5 unit time and the size of each feeder slot is 4 unit lengths. For the purpose of generating the random placement points, we choose the length and the width of the PCB as 40 and 10 unit length, respectively, such that the random placement points fall within these limits. The assembly points are chosen as 50 whilst the number of component types is 8 (as shown in Table 1). For simulating the assembly cycle time, feeder carrier movement and PCB table movement, we set the speed of the robot arm, feeder carrier and PCB table as 6, 5 and 4 unit distance/unit time respectively (as 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 component types. The computational results are shown in Table 2 and are obtained from twenty runs. New datasets are randomly generated in each test. The CDPP approach is used to determine the pickup and placement location during the component assembly process. The CT, feeder movement (FM) and PCB table movement (PM) are calculated based on the CDPP approach.

As we are dealing with a minimisation problem, smaller values are better (referring to Table 2, Figs. 7–10). The results shown in Table 2 and Fig. 7 clearly indicate that there is no relation between the exchange frequency and the CT. Fig. 7 shows that in all tests the minimum exchange frequency do not coincide with the minimum CT. In test 5, for example, the minimum exchange frequency is 22 with the CT being 147.07 unit time, whilst the minimum CT is 142.19 unit time obtained when the exchange frequency is 31. We suspect that we cannot reach the optimal CT if we are just looking for the minimum exchange frequency. Hence, assigning component feeders to slots based on exchange frequency can be considered as an incorrect strategy. These results are inconsistent with Wang et al. [15], where they claim that minimising (actually

Table 1
The seven factors of experimental parameters

Factors	Levels (low/high)
Number of assembly points (N)	50
Number of component types (K)	8
Length of PCB (BL)	40 (unit distance)
Width of PCB (BW)	10 (unit distance)
Speed of robot (V_r)	6 (unit distance/unit time)
Speed of feeder carrier (V_f)	5 (unit distance/unit time)
Speed of PCB table (V_b)	4 (unit distance/unit time)

Table 2

A comparison between three different objective functions that are minimising the CT, minimising the exchange frequency and minimising the triple objective function

Test	Minimise CT				Minimise exchange frequency				Minimise F			
	EF	CT*	FM	PM	EF*	CT	FM	PM	EF	CT	FM	PM
1	27	135.25	344.57	74.79	19	137.48	323.77	101.26	27	135.27	344.57	74.79
2	20	147.70	352.66	80.20	16	150.68	336.70	104.29	20	148.32	365.98	67.69
3	17	138.82	304.39	110.82	12	143.71	280.40	152.95	15	139.65	304.09	69.49
4	25	144.21	273.88	71.50	19	145.28	237.09	76.77	25	144.21	273.88	71.50
5	31	142.19	317.89	95.68	22	147.07	291.05	119.95	29	143.53	317.91	65.32
6	31	124.55	342.83	129.79	21	127.75	294.91	128.91	26	124.76	323.98	105.39
7	27	143.47	344.73	64.93	21	146.43	302.45	119.66	22	144.47	318.52	58.61
8	29	139.29	317.93	126.00	24	141.37	265.30	94.86	28	140.50	296.89	72.48
9	35	136.17	342.86	95.89	28	138.76	332.24	117.47	32	136.24	371.68	71.30
10	17	137.29	384.30	82.64	11	141.52	332.57	110.52	17	137.29	384.30	82.64
11	32	140.69	387.59	74.41	23	141.73	347.55	83.94	32	140.76	352.09	54.98
12	29	135.39	377.29	133.55	20	139.05	333.45	121.45	28	136.61	371.77	84.05
13	22	133.51	305.55	102.46	17	136.78	303.77	118.28	23	134.66	302.14	73.11
14	20	146.40	332.45	75.75	15	149.42	330.36	118.13	20	146.80	371.89	56.18
15	26	132.99	351.39	90.33	19	133.93	297.75	90.35	19	133.93	297.75	90.35
16	26	149.81	347.68	83.00	18	151.76	291.68	111.45	24	150.73	350.18	51.45
17	24	137.32	316.50	82.93	22	141.45	345.57	96.18	25	137.77	333.61	63.48
18	25	137.45	322.00	99.98	20	141.56	317.82	122.35	26	137.87	349.80	59.83
19	26	139.76	350.82	96.20	17	141.58	273.80	105.20	31	140.03	365.09	52.51
20	22	138.14	332.80	104.19	15	140.11	299.11	110.64	19	138.59	314.89	95.03
Ave	26	139.02	337.51	93.70	19	141.42	306.87	110.23	24	139.60	335.55	71.01

Note: EF = exchange frequency; CT = assembly cycle time; FM = the distance of feeder movement; PM = the distance of PCB table movement; CT* = optimum assembly cycle time; EF* = optimum exchange frequency.

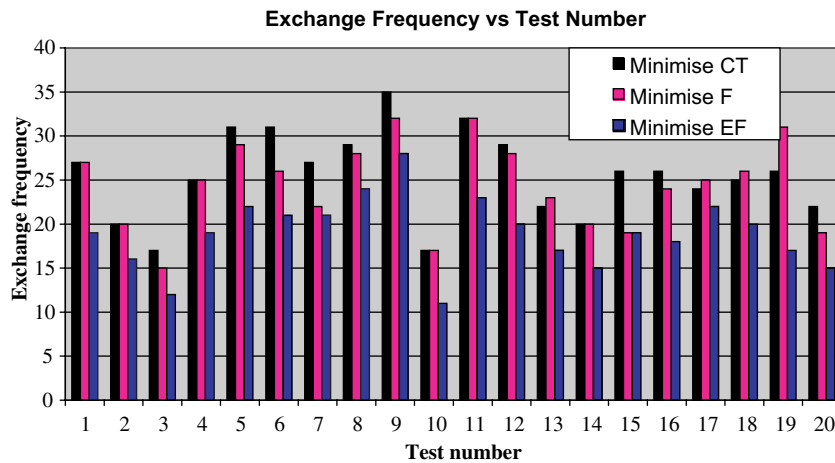


Fig. 7. A comparison among three objective function based on the exchange frequency.

maximising) the total exchange frequency can lead to the minimum CT. As such, they assigned component feeders to slots so that the total exchange frequency of all adjacent slot pairs has the minimum value (or actually maximum) assuming they would obtain the minimum CT.

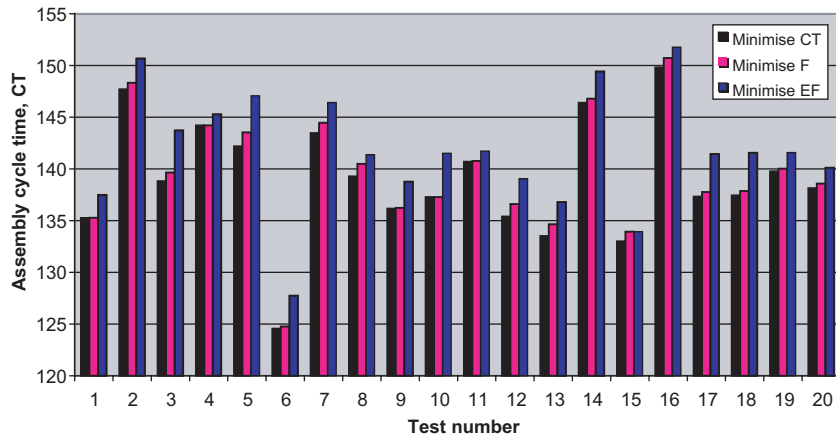


Fig. 8. A comparison among three objective function based on the CT.

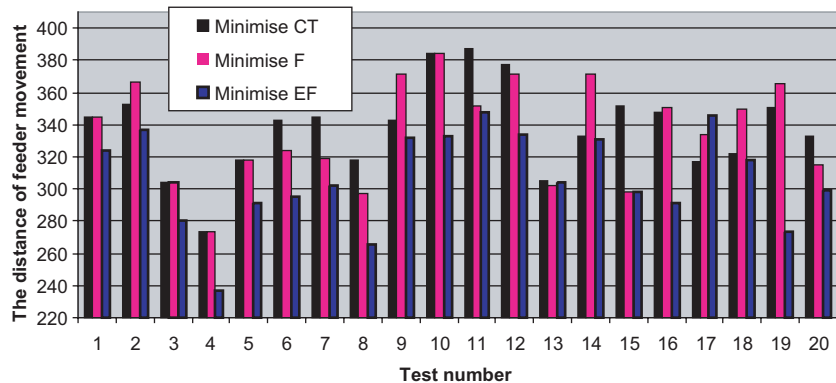


Fig. 9. A comparison among three objective function based on the distance of feeder movement.

The results also indicate that we can obtain near optimal CT if we use the triple objective function, F, in searching for the optimal solution. This is clearly shown in Fig. 8, where the triple objective function, F obtained an optimal CT in 25% of the tests (over 20 test) whilst the other 75% obtained near optimal CT. However, the strategy of minimising the exchange frequency is unable to obtain any optimal CT over 20 tests. In addition, by using the triple objective function, we can gain better movement of the PCB table and feeder carrier compared to the solution obtained by minimising CT (refer to Figs. 9 and 10). For example, in test 5, by minimising the CT, we obtain 142.19 unit time as the minimum CT (the optimal CT) with the movement distances of feeder carrier and PCB table being 317.89 and 95.68 unit distance, respectively. However, by using the F objective function, we obtain 143.53 unit time, 317.91 and 65.32 unit distance for the movement distance of the feeder carrier and PCB table, respectively (referring to test 5 in Table 2). In Fig. 10, we can observe that by minimising F, we obtained better PCB table movement in 80% of the tests. In fact, we will get an optimal CT with a good movement of PCB table and feeder carrier when the weighted parameter of CT, W_c , is set to a suitably large value. On the contrary, we can only achieve the optimal CT, but not always minimal PCB table and feeder movement when we only search for minimising the CT. Sometimes when there exist solutions with the same CT, we need other factors to determine which solution

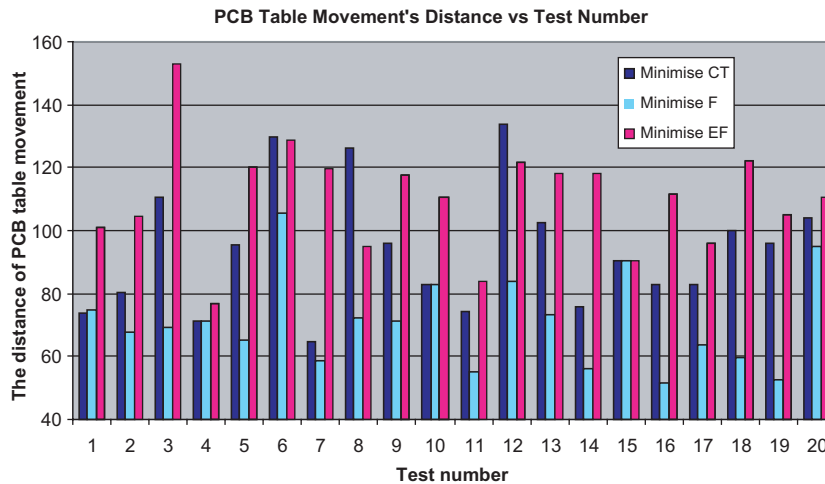


Fig. 10. A comparison among three objective function based on the distance of PCB table movement.

should be chosen. In this case, our triple objective function provides an advantage in choosing the best solution.

From Table 2 and Fig. 9 we also can see that in all tests the strategy of minimising the exchange frequency can provide a better feeder carrier movement compared to the other two strategies. Hence, we can conclude that assigning component feeders to slots based on minimising (actually maximising) the total exchange frequency of all adjacent slot pairs can provide better feeder carrier movement. Unfortunately, this result contradicts with Hop and Tabucanon [14] where they claimed (although show no evidence) that when the feeder assignment is arranged based on maximising the exchange frequency of component types, it will increase the feeder carrier movement if components are not assembled by their type. In fact, our approach does not assemble components based on their type. Of course, when the components are assembled based on component type, the movement of the feeder carrier can be eliminated but we may pay an extra cost in the robot arm movement, which may increase the CT. Since the robot arm movement is considered as 'unavoidable', and the movement of the feeder carrier and PCB table can be ignored if they can move within the 'free' movement time (that is the robot arm does not have to wait for them), assembling components based on their type may not be a good strategy as it may increase CT.

8. Conclusions

In this paper we have developed three strategies for feeder setup that assign component feeders to slots by minimising the assembly cycle time (CT), minimising the total exchange frequency and minimising the triple objective function. We found that minimising (actually maximising) the total exchange frequency can only provide better feeder carrier movement rather than a better CT.

We have introduced the triple objective function with a CDPP approach in our formulation to improve the feeder setup. The function aims to minimise the CT together with minimising the feeder carrier and PCB table movements. By using this strategy we found near optimal solutions with a better movement of feeder carrier and PCB table. In summary, minimising the triple objective function strategy can provide better solutions compared to the strategies of minimising the CT and minimising the exchange frequency.

Acknowledgements

This work has been supported by the Public Services Department of Malaysia and the National University of Malaysia.

References

- [1] D. Golding, PCB Assembly, *Assembly Automation* 15 (2) (1995) 10–13.
- [2] K.P. Ellis, F.J. Vites, J.E. Kobza, Optimizing the performance of a surface mount placement machine, *IEEE Transactions on Electronic Packaging Manufacturing* 24 (3) (2001) 160–170.
- [3] Y. Crama, J. van de Klundert, F.C.R. Spieksma, Production planning problems in printed circuit board assembly, *Discrete Applied Mathematics* 123 (1–3) (2002) 339–361.
- [4] T.M. Tirpak, P.C. Nelson, A.J. Aswani, Optimization of revolver head SMT machines using adaptive simulated annealing (ASA), *Electronics Manufacturing Technology Symposium*, 2000. Twenty-Sixth IEEE/CPMT International, 2000, pp. 214–220.
- [5] H.-P. Fu, C.-T. Su, A comparison of search techniques for minimizing assembly time in the printed wiring assembly, *International Journal of Production Economics* 63 (2000) 83–98.
- [6] W. Wang, P.C. Nelson, T.M. Tirpak, Optimization of high-speed multistation SMT placement machines using evolutionary algorithms, *IEEE Transactions on Electronics Packaging Manufacturing* 22 (2) (1999) 137–146.
- [7] M. Ayob, G. Kendall, A new dynamic point specification approach to optimise surface mount placement machine in printed circuit board assembly, *Proc. of the IEEE ICIT'02, Bangkok*, 11–14 December 2002, pp. 486–491.
- [8] P.J. Egbelu, C. Wu, R. Pilgaonkar, Robotic assembly of printed circuit boards with component feeder location consideration, *Production Planning and Control* 7 (1996) 162–175.
- [9] R.H. Ahmadi, J.W. Mamer, Routing heuristics for automated pick and place machines, *European Journal of Operational Research* 117 (1999) 533–552.
- [10] C. Klomp, J.J. van de Klundert, F.C.R. Spieksma, S. Voogt, The feeder rack assignment problem in PCB assembly: A case study, *International Journal of Production Economics* 64 (2000) 399–407.
- [11] K.P. Ellis, F.J. Vites, J.E. Kobza, Optimizing the performance of a surface mount placement machine, *IEEE Transactions on Electronic Packaging Manufacturing* 24 (3) (2001) 160–170.
- [12] L.P. Khoo, K.M. Loh, A genetic algorithms enhanced planning system for surface mount PCB assembly, *International Journal of Advanced Manufacturing Technology* 16 (2000) 289–296.
- [13] M.C. Leu, H. Wong, Z. Ji, Planning of component placement/insertion sequence and feeder setup in PCB assembly using genetic algorithm, *Journal of Electronic Packaging, Transactions ASME* 115 (1993) 424–432.
- [14] N.V. Hop, M.T. Tabucanon, Multiple criteria approach for solving feeder assignment and assembly sequence problem in PCB assembly, *Production Planning and Control* 12 (8) (2001) 736–744.
- [15] C. Wang, L.-S. Ho, D.J. Cannon, Heuristics for assembly sequencing and relative magazine assignment for robotic assembly, *Computers and Industrial Engineering* 34 (1998) 423–431.
- [16] M. Ayob, P. Cowling, G. Kendall, Optimisation for surface mount placement machines, *Proc. of the IEEE ICIT'02, Bangkok*, 11–14 December 2002, pp. 498–503.
- [17] J. Ahmadi, S. Grotzinger, D. Johnson, Component allocation and partitioning for a dual delivery placement machine, *Operations Research* 36 (1988) 176–191.
- [18] B. Bentzen, SMD placement, SMT in FOCUS, November 28, 2000 (September 25, 2002) (Downloadable from website http://www.smtinfocus.com/PDF/SMD_placement.pdf).
- [19] Y.-C. Su, C. Wang, P.J. Egbelu, D.J. Cannon, A dynamic point specification approach to sequencing robot moves for PCB assembly, *International Journal Computer Integrated Manufacturing* 8 (6) (1995) 448–456.
- [20] N.V. Hop, M.T. Tabucanon, Extended dynamic point specification approach to sequencing robot moves for PCB assembly, *International Journal of Production Research* 39 (8) (2001) 1671–1687.
- [21] C. Su, H. Fu, A simulated annealing heuristic for robotics assembly using the dynamic pick-and-place model, *Production Planning and Control* 9 (1998) 795–802.