Available online at www.sciencedirect.com

SCIENCE DIRECT®

European Journal of Operational Research 150 (2003) 370–394

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Production, Manufacturing and Logistics

Optimization of multi-feeder (depot) printed circuit board manufacturing with error guarantees

Burak Kazaz [a], Kemal Altınkemer [b,*]

[a] *School of Business, University of Miami, 414 Jenkins Building, Coral Gables, FL 33124, USA*
[b] *Krannert Graduate School of Management, Purdue University, 1310 Krannert Building, West Lafayette, IN 47907-1310, USA*

Received 25 October 2001; accepted 28 May 2002

Abstract

This paper considers an integrated optimization problem enhancing productivity in printed circuit board (PCB) manufacturing. The problems of assigning component types to feeder locations and sequencing component placements on the PCB are simultaneously formulated in a mathematical model. Our model differs from earlier studies by allowing component types to be placed in multiple feeders. Although such flexibility adds complexity to the original problem, we develop an integrated solution that has promising results. We develop an integrated algorithm that finds the optimal solution when the optimal solutions for the multi-depot vehicle routing problem (MDVRP) are given. Otherwise, given an ε-approximation algorithm for the MDVRP, our integrated solution has a theoretical ε-error guarantee for PCB problem. The effectiveness of the integrated approach is shown with extensive computational experiments.
© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Optimization; ε-Approximation, modeling; PCB manufacturing; Multi-depot vehicle routing problem

1. Introduction

Printed circuit board (PCB) manufacturing entails challenging optimization problems. In this paper, two problems, the assignment of component types (chips) to feeder locations in a computerized numerically control machine (CNC) and the sequencing of the placement of these components on the PCB, are considered together. Although the decisions regarding these problems are stated to be dependent on each other (see [4,5,8,18]), most previous research has studied them independently as part of interrelated problems. One motivating factor for our research is that these problems need to be formulated in a single model in order to obtain system-wide optimal solutions and increase the productivity of PCB manufacturers. In this paper, we propose an integrated approach that accounts for both of these problems simultaneously. Our study differs from earlier works by introducing the flexibility to assign a component type to multiple feeders. Although this flexibility adds complexity to the formulation and the solution method, our results

* Corresponding author. Fax: +1-765-494-1526.
E-mail addresses: bkazaz@miami.edu (B. Kazaz), kemal@mgmt.purdue.edu (K. Altınkemer).

0377-2217/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0377-2217(02)00529-5

indicate that the benefits are plenty. While the sequencing of the placement of components on a PCB resembles the vehicle routing problem (VRP), having a component type located in several feeders creates multiple number of depots. Therefore, it becomes necessary to consider the multi-depot vehicle routing problem (MDVRP) in the formulation. To accomplish our goal, we first structure the problem. By using optimization techniques, we develop an integrated method that finds the optimal solution if the MDVRP can be solved optimally. However, MDVRP belongs to NP-hard class of problems and it is unlikely to find optimal solution in polynomial time. Therefore, we develop an integrated solution method that has a theoretical ε-error guarantee under the presence of an ε-approximation algorithm for the MDVRP. We show the effectiveness of our integrated algorithm by testing it on various PCB types.

There are many different processing designs used in the surface mount technology CNCs that produce the PCBs. The CNC machine that is studied in this problem is a widely used one with a rotary head. The rotary head enables the machine to pick up a certain number of components (of the same type) from a feeder and mount them on the PCB in one tour. A CNC has three main parts: a board that the PCB is placed on, a feeder locator where the components are located, and a head that picks up components from feeders and mounts them on the board. The head uses a nozzle to grasp the component located in the feeder. Since each component type requires a different nozzle, the head cannot mount various component types on the board in the same tour. The head changes the nozzle at the tool magazine which is generally located at the mid-point of the feeder locator. Most widely used heads have the capability of picking up more than just one component at a time, and are called "rotary heads." This capability of rotary heads resembles a VRP and differs from other technologies used in the PCB manufacturing. For example, a single-head CNC can only pick up one component at a time and resembles a Chinese postman problem (see [4]). It is also different from technologies such as the double-head CNC (or the dual-head CNC) which picks up components and mounts them on the board using two independent heads. As the technology varies, so does the formulation and complexity.

Fig. 1 illustrates an example CNC studied in this paper as well as exemplifying the intertwined decisions of assigning component types to feeders and routing the head to mount the components on the PCB. In this example, there are two component types, 1 and 2, and eight possible feeder locations, and the rotary head can pick up four components of the same type at a time. The PCB requires seven components of type 1 and three of type 2 to be mounted. The example shows that component type 1 is assigned to feeders two and eight while component type 2 is located only in feeder 4. Assigning component types to *multiple feeders* is

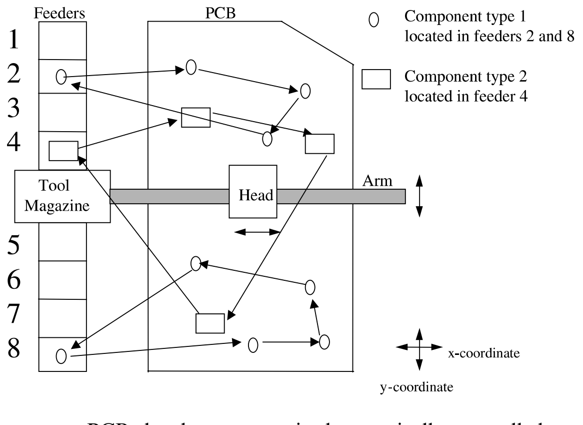
Feeders
PCB
1
2
3
4
5
6
7
8
Tool Magazine
Head
Arm
Component type 1 located in feeders 2 and 8
Component type 2 located in feeder 4
x-coordinate
y-coordinate

Fig. 1. The mounting of components onto PCBs by the computerized numerically controlled machines.
(*Note:* The arm moves in the y-axis whereas the head moves in the x-axis. Furthermore, component type 1 is assigned at two different feeders.)

the flexibility that our model provides while previous models in literature restrict each component type to be located in one and only one feeder. Our model allows both single and multiple-feeder assignments. This study sheds light to most PCB manufacturers' concern about whether their throughput can increase by allowing multiple feeders for a component type. In the figure, the head picks up three components of type 1 from feeder two and mounts them on the board by following the route indicated with arcs. Then, the head travels to feeder eight and picks up four more and mounts them on the PCB. Next, the head goes to feeder 4, changes the nozzle and picks up four components of type 2. After completing the tour as indicated, the example PCB is completed. It should be noted that component type 1 could have been located at a single feeder, such as at feeder five. However, it is visually evident that the head would have to make longer trips if the head capacity is equal to four. The impact of such flexibility becomes even more significant when the number of components from each component type increases. Similarly, it can be less costly to assign component type 2 at feeders four and seven. Therefore, the PCB manufacturing problem requires an integrated formulation and a solution method for both the assignment and the VRP using multiple feeders for a component type.

The head of the CNC can move both in horizontal and in vertical directions simultaneously. This corresponds to a Chebychev metric for distance (cost) calculations. Therefore, the distance between points i and j on the PCB for component type k, c_{ij}^k, can be written as

$$c_{ij}^k = \max\{|x_i - x_j|, |y_i - y_j|\},$$

where x_i, x_j represent the horizontal and y_i, y_j are the vertical coordinates, respectively. The distance between any two points is the maximum of the distances in either horizontal or vertical axes. The head also moves up and down in the z-coordinate in order to mount the component on the PCB. However, due to the importance of precision in the mounting task, it does not move in this axis while moving in horizontal and vertical axes. Since the total number of components that are mounted is constant for a given PCB, these movements exhibit a constant for the total z-coordinate movements and it can be dropped from consideration.

The paper is organized in the following way: Section 2 contains a literature review. In Section 3, we present the mathematical model and the integrated algorithm designed to develop feasible solutions for the PCB manufacturing problem. Structural and theoretical developments are also included in this section. Next, computational experiments that show the efficiency of the integrated algorithm and the benefits of using multi-feeder assignments over the single-feeder approach are presented in Section 4. Finally, conclusions are provided in Section 5, as are future research directions.

2. Literature review

The PCB manufacturing problem is best described in Ball and Magazine [4] by using an insertion technology CNC. They define the following three subproblems in the PCB manufacturing problem:

(i) allocation of component types to machines,
(ii) allocation of component types to feeders at each machine,
(iii) pick-and-placement sequencing.

Many studies examined the PCB manufacturing problem as part of interrelated papers. An extensive survey of problems studied in PCB manufacturing can be found in [6]. While Ball and Magazine [4] focus on the third subproblem, Drezner and Nof [8] concentrate on the component assignment. For a different technology, an insertion machine, Gavish and Seidmann [13], McGinnis et al. [17], Or and Demirkol [18], and Leipala and Nevalainen [16] state the importance of an integrated approach. In our paper, we study a

CNC with a rotary head that can pick a certain number of components at a time. This resembles the VRP, however, since a component type can be assigned to more than one feeder location, the formulation requires the use of the MDVRP. Examples of MDVRP studies include [1,7,19]. Because the rotary head is a different technology, none of these studies can provide a solution for the technology presented here. Altınkemer et al. [3] provides a solution, however, it assumes that each component type is assigned to a single feeder. In a slightly different setting, Crama et al. [5] considers multiple-feeder assignment for a component type while using a rotary-head CNC, however, their solution heuristic cannot assign a component type to more than two-feeder locations. One motivating factor for our study is that our formulation relaxes this assumption, therefore it results in a different formulation and a solution technique. It should be noticed that relaxing such an assumption complicates the problem and increases the number of binary variables in the formulation factorially.

3. The integrated model and a heuristic

In this section, we present a formulation for the PCB manufacturing problem. By using Lagrangian relaxation, we decompose our formulation into two subproblems: an assignment-like and a multi-depot vehicle routing-like problem. This analysis enables us to develop a lower bound for the PCB problem. An integrated algorithm is inspired by the same structural analysis. Next, we show that given an ε-approximation for the MDVRP, our integrated algorithm has an error bound that is less than or equal to ε.

The following two assumptions are made in order to keep the problem tractable:

1. The head completes the tour by returning to the feeder location before going to the tool magazine.
2. The travel time between two coordinates is approximately linear.

We begin our analysis by presenting the mathematical model used for the PCB manufacturing problem. It corresponds to an integer programming formulation.

3.1. The mathematical model

An integer programming model is used to formulate the PCB manufacturing problem. One set of integer variables corresponds to the decisions regarding the assignment of component types to feeder locations. A second set of integer variables are defined in a manner which is similar to traditional VRP formulations. The decisions regarding the next point on which a component will be mounted on the PCB are determined by these variables. The objective function aims to minimize the total distance traveled by the rotary head so that the throughput can be increased in the system. The mathematical model is as follows.

Notation:

K	the total number of component types
L	the total number of feeder locations; $L > K$
$n(k)$	number of components of type k
$V(k)$	the node set of component type k including the feeder locations (starting nodes) $V(k) = \{1, \ldots, n(k), \ldots, n(k) + L\}$
$\widetilde{V}(k)$	the node set excluding the indices of feeder locations $\widetilde{V}(k) = \{L + 1, \ldots, n(k) + L\}$
Q	the maximum number of components from one component type that the head can pick up at a time (equivalently, the maximum number of components that can be mounted in one tour)
c_{ij}^k	the cost of traveling (distance) from point i to point j ($i < j$ and $i \neq j$, $i \in V(k)$, $j \in \widetilde{V}(k)$) for component type k, $c_{ii}^k = \infty$; $c_{ij}^k = c_{ji}^k$ for $i \in (V(k) - \widetilde{V}(k))$ and $j \in \widetilde{V}(k)$

d_l the round-trip distance between the feeder location l and the tool magazine

$m(k)$ the minimum number of tours the arm needs to make for component type k, $m(k) = \lceil n(k)/Q \rceil$

U the maximum number of feeders that a component type can be assigned to (the minimum of either one more than the difference between the total number of feeders, L, and the total number of component types, K, i.e. $U = L - K + 1$, or the number of components of a component type k, i.e. $U = n(k)$).
$U = \min\{L - K + 1, n(k)\}$

Decision variables:

$$y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}} = \begin{cases} 1 & \text{if component type } k \text{ is assigned to } n \text{ feeder locations}: \ l_{k_1}, l_{k_2}, \ldots, l_{k_n}, \\ 0 & \text{otherwise,} \end{cases}$$

$$x^k_{ij} = \begin{cases} 1 & \text{if arc } (i,j) \text{ is traversed for component type } k;\ i < j \text{ and } i \neq j,\ i \in V(k),\ j \in \widetilde{V}(k), \\ 0 & \text{otherwise,} \end{cases}$$

$$x^k_{ji} = \begin{cases} 1 & \text{if arc } (j,i) \text{ is traversed for component type } k;\ i < j \text{ and } i \neq j,\ i \in \left(V(k) - \widetilde{V}(k)\right), \\ & \quad j \in \widetilde{V}(k), \\ 0 & \text{otherwise.} \end{cases}$$

It should be noted that this variable represents the return arc from the PCB to the feeder location.
$z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ = total cost of assigning component type k to n feeder locations: $l_{k_1}, l_{k_2}, \ldots, l_{k_n}$.

Objective function:

The objective is to minimize the total distance traveled by the rotary head. When a component type is assigned to a feeder location, the head travels from the tool magazine to the feeder and picks up a certain number of components, tours on the board mounting these components, and returns to the feeder location before visiting the tool magazine for the next component type. In order to understand the total distance traveled by the head, consider the following example. Suppose a component type k is assigned to two-feeder locations l_{k_1} and l_{k_2}, thus $y^{k,2}_{l_{k_1},l_{k_2}} = 1$. First, the head travels the round-trip distances between the tool magazine and these two feeders, $d_{l_{k_1}}$ and $d_{l_{k_2}}$. $z^{k,2}_{l_{k_1},l_{k_2}}$ is the total of the distances traveled on the board when component type k is located in feeders l_{k_1} and l_{k_2}. Therefore, the total distance traveled by the head for component type k becomes the sum of $z^{k,2}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}}$. In the objective function, this total distance is multiplied by a binary decision variable, $y^{k,2}_{l_{k_1},l_{k_2}}$. The term $\sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L-1} \sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}} (z^{k,2}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}})$ represents the total cost (distance) of assigning each component type $k = 1, \ldots, K$ to all possible two-feeder combinations. It should be noted that a component type can be assigned to at least to one and at most U feeders. Therefore, the objective function consists of the sum of the total distances that the head travels from all of the feeder combinations (e.g. one-feeder assignment to U feeder assignment).

$$\text{(P1):} \quad Z1^* = \operatorname{Min} Z1 = \sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}} \left(z^{k,1}_{l_{k_1}} + d_{l_{k_1}} \right) + \sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L-1} \sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}} \left(z^{k,2}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}} \right)$$
$$+ \cdots + \sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L-U+1} \sum_{l_{k_2}=l_{k_1}+1}^{L-U+2} \cdots \sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} \left(z^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} + d_{l_{k_1}} + d_{l_{k_2}} + \cdots + d_{l_{k_U}} \right). \tag{1}$$

Constraints:

- At most one component type will be assigned at each feeder location; while doing so, all feeder assignment combinations (assigning to one feeder, two feeders, up to U feeders) for each component type should be considered. It should also be noted that a component type cannot be assigned to more than U feeder locations:

$$\sum_{k=1}^{K} y_l^{k,1} + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-1} y_{l_{k_1},l}^{k,2} + \sum_{l_{k_2}=l+1}^{L} y_{l,l_{k_2}}^{k,2}\right) + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-2}\sum_{l_{k_2}=l_{k_1}+1}^{l-1} y_{l_{k_1},l_{k_2},l}^{k,3} + \sum_{l_{k_1}=1}^{l-1}\sum_{l_{k_3}=l+1}^{L} y_{l_{k_1},l,l_{k_3}}^{k,3} + \sum_{l_{k_2}=l+1}^{L-1}\sum_{l_{k_3}=l_{k_2}+1}^{L} y_{l,l_{k_2},l_{k_3}}^{k,3}\right) + \cdots + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{l-U+2}\cdots\sum_{l_{k_{U-1}}=l_{k_{U-2}}+1}^{l-1} y_{l_{k_1},l_{k_2},\dots,l}^{k,U} + \cdots + \sum_{l_{k_2}=l+1}^{L-U+2}\sum_{l_{k_3}=l_{k_2}+1}^{L-U+3}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y_{l,l_{k_2},\dots,l_{k_U}}^{k,U}\right) \leqslant 1 \quad \text{for each feeder location } l = 1,\dots,L. \tag{2}$$

- Each component type should be assigned to a feeder configuration (combination of feeders). In a feeder configuration, a component type can be assigned to at least one feeder and at most U feeders:

$$\sum_{l_{k_1}=1}^{L} y_{l_{k_1}}^{k,1} + \sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y_{l_{k_1},l_{k_2}}^{k,2} + \cdots + \sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y_{l_{k_1},l_{k_2},\dots,l_{k_U}}^{k,U} = 1$$
$$\text{for each component type } k = 1,\dots,K. \tag{3}$$

- The cost of component type assignment is calculated by touring cost:

$$\sum_{h=1}^{n}\sum_{j=L+1}^{n(k)+L} c_{l_{k_h},j}^{k} x_{l_{k_h},j}^{k} + \sum_{h=1}^{n}\sum_{j=L+1}^{n(k)+L} c_{l_{k_h},j}^{k} x_{j,l_{k_h}}^{k} + \sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L} c_{i,j}^{k} x_{i,j}^{k} = z_{l_{k_1},l_{k_2},\dots,l_{k_n}}^{k,n}$$
$$\text{for each feeder location combination } (l_{k_1}, l_{k_2},\dots,l_{k_n}) \text{ and component type } k = 1,\dots,K. \tag{4}$$

- The minimum number of arcs departing from feeder locations should be greater than or equal to the minimum number of tours:

$$\sum_{h=1}^{n}\sum_{j=L+1}^{n(k)+L} x_{l_{k_h},j}^{k} \geqslant m(k) y_{l_{k_1},l_{k_2},\dots,l_{k_n}}^{k,n} \quad \text{for each feeder location combination } (l_{k_1}, l_{k_2},\dots,l_{k_n}) \text{ and component type } k = 1,\dots,K. \tag{5}$$

- There should be at least one outgoing arc from a feeder if a component type is assigned to it:

$$\sum_{j=L+1}^{n(k)+L} x_{lj}^{k} \geqslant \left(y_l^{k,1} + \left(\sum_{l_{k_1}=1}^{l-1} y_{l_{k_1},l}^{k,2} + \sum_{l_{k_2}=l+1}^{L} y_{l,l_{k_2}}^{k,2}\right) + \cdots + \left(\sum_{l_{k_1}=1}^{l-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{l-U+2}\cdots\sum_{l_{k_U}}^{l-1} y_{l_{k_1},l_{k_2},\dots,l}^{k,U} + \cdots + \sum_{l_{k_2}=l+1}^{L-U+2}\sum_{l_{k_3}=l_{k_2}+1}^{L-U+3}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y_{l,l_{k_2},\dots,l_{k_U}}^{k,U}\right)\right) \quad \text{for each feeder location } l = 1,\dots,L \text{ and component type } k = 1,\dots K. \tag{6}$$

- The number of outgoing arcs from each feeder should be equal to the number of incoming arcs to that feeder:

$$\sum_{j=L+1}^{n(k)+L} x_{lj}^{k} = \sum_{j=L+1}^{n(k)+L} x_{jl}^{k} \quad \text{for each feeder location } l = 1,\dots,L \text{ and component type } k = 1,\dots,K. \tag{7}$$

- The total number of outgoing arcs from each point on the PCB and incoming arcs to that point should be equal to two:

$$\sum_{l=1}^{L} x_{lj}^{k} + \sum_{i=L+1}^{j-1} x_{ij}^{k} + \sum_{i=j+1}^{n(k)+L} x_{ji}^{k} + \sum_{l=1}^{L} x_{jl}^{k} = 2 \quad \text{for each point } j = L+1,\ldots,n(k)+L \text{ on the PCB for component type } k = 1,\ldots,K. \tag{8}$$

- The total number of outgoing arcs from the feeders to points on the PCB and from the points to other points on the PCB should be equal to the total number of components of that type:

$$\sum_{i=1}^{n(k)+L-1} \sum_{j=i+1}^{n(k)+L} x_{ij}^{k} = n(k) \quad \text{for each component type } k = 1,\ldots,K. \tag{9}$$

- Subtour elimination constraints as shown in [11]:

$$\sum_{i\in S(k)} \sum_{i<j\in S(k)} x_{ij}^{k} \leqslant |S(k)| - L_{S(k)} \quad \text{for each component type } k = 1,\ldots,K, \quad \forall S(k) \subset \widetilde{V}(k),\ L_{S(k)} \geqslant 1,\ |S(k)| \geqslant 2. \tag{10}$$

- Integrality and non-negativity constraints:

$$y_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n} = \{0/1\} \quad \text{for each component type } k = 1,\ldots,K \text{ and feeder location combination } (l_{k_1},l_{k_2},\ldots,l_{k_n}), \tag{11}$$

$$x_{ij}^{k} = \{0/1\} \quad \text{for each component type } k = 1,\ldots,K \text{ and arc } (i,j), \tag{12}$$

$$z_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n} \geqslant 0 \quad \text{for each component type } k = 1,\ldots,K \text{ and feeder location combination } (l_{k_1},l_{k_2},\ldots,l_{k_n}). \tag{13}$$

The above formulation is a combination of assignment-like and multi-depot vehicle routing-like problems where the decision variables $y_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n}$ represent the decisions of the assignment problem and variables x_{ij}^{k} represent the decisions of the MDVRP. Sets (2), (3) and (11) are the constraints of the assignment-like problem. Sets (5)–(10), (12) and (13) represent the constraints of MDVRP for each component type k. Set (10) is subtour elimination constraints where $L_{S(k)}$ represents an optimal solution of a one dimensional bin packing problem where bins have length Q and each item to be packed in the bins has a weight of one unit. A detailed description of these constraints can be found in [11,15]. Set (4) determines the total travel distance (cost) of each component type. It should be noted here that it is not hard to show that problem (P1) is NP-hard since MDVRP belongs to the class of NP-hard problems (see [10]). Thus, it is less likely to obtain the optimal solution for a large-scale problem in polynomial time. Therefore, we next focus on generating an integrated algorithm for the above formulation. In order to investigate the efficiency of this algorithm, we also develop a lower bound.

Since the problem has a special structure, the formulation can be decomposed into two subproblems when constraint set (4) is relaxed. The corresponding Lagrangian variables are denoted by $\alpha_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n}$ (unrestricted in sign). The resulting Lagrangian relaxation is presented as $(P1^{\mathrm{LR}})$.

$$\begin{aligned}
(P1^{\mathrm{LR}}(\boldsymbol{\alpha})):\ \mathrm{Min}\, Z1^{\mathrm{LR}} = {} & \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}}\left(z^{k,1}_{l_{k_1}} + d_{l_{k_1}}\right) \\
& + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L}\alpha^{k,1}_{l_{k_1}}\left(-\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_1},j}x^k_{l_{k_1},j} - \sum_{j=L+1}^{n(k)+L} c^k_{l_{k_1},j}x^k_{j,l_{k_1}} - \sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L} c^k_{i,j}x^k_{i,j} + z^{k,1}_{l_{k_1}}\right) \\
& + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}}\left(z^{k,2}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}}\right) + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L}\alpha^{k,2}_{l_{k_1},l_{k_2}} \\
& \times\left(-\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{l_{k_h},j} - \sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{j,l_{k_h}} - \sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L} c^k_{i,j}x^k_{i,j} + z^{k,2}_{l_{k_1},l_{k_2}}\right) \\
& + \cdots + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} \\
& \times\left(z^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} + d_{l_{k_1}} + d_{l_{k_2}} + \cdots + d_{l_{k_U}}\right) \\
& + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L}\alpha^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(-\sum_{h=1}^{U}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{l_{k_h},j}\right. \\
& \left. - \sum_{h=1}^{U}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{j,l_{k_h}} - \sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L} c^k_{i,j}x^k_{i,j} + z^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}\right) \\
\text{s.t.}\quad & (2),(3),(5)\text{–}(13)
\end{aligned} \tag{14}$$

where $\alpha^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ is unrestricted in sign. Recall that in a Lagrangian relaxation the formulation can be represented as $\max_\alpha\{Z1^{\mathrm{LR}}(\boldsymbol{\alpha})\} \leqslant Z1^*$ subject to suitable sign restrictions on $\boldsymbol{\alpha}$. In this case, $\boldsymbol{\alpha}$ is unrestricted in sign. $\boldsymbol{\alpha}^*$, which maximizes the Lagrangian relaxation, $Z1^{\mathrm{LR}}$, gives the tightest lower bound to the original problem [9,14]. Furthermore, (1) is minimized when $z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ is equal to the sum of the distances traveled by the head, $\sum_{h=1}^{n}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{l_{k_h},j} + \sum_{h=1}^{n}\sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j}x^k_{j,l_{k_h}} + \sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L} c^k_{i,j}x^k_{i,j}$. When this is the case, the value of $z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ is equal to the optimal cost of the MDVRP when component type k is assigned to feeder locations $l_{k_1}, l_{k_2}, \ldots, l_{k_n}$, denoted as $z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$. Constraint sets (5) and (6) tie all the assignment and touring decisions. If constraint set (5) is enumerated by assigning each $y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ value to be equal to 1 and solving for the MDVRP, one can obtain all $z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$ values.

3.2. The integrated algorithm

In this algorithm, we first solve a MDVRP for each component type at every possible feeder location combination $(l_{k_1}, l_{k_2}, \ldots, l_{k_n})$. The feasible solution obtained from the MDVRP is used as the cost of assigning the component type to the particular feeder locations. By using the feasible solution value as the cost of assigning component types to feeder locations, an assignment problem can be solved. The optimal solution of this assignment problem is a solution to the original problem.

Problem (P1) presents a special structure. The variables of the assignment-like subproblem, $y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$, and the variables of the multi-depot vehicle routing-like subproblem, x^k_{ij}, are tied by constraint sets (5) and (6). When constraint set (5) is enumerated by assigning each $y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}} = 1$ and solving a MDVRP, we obtain the optimal sum of the distances traveled, $z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}$, when component type k is located in feeder locations $l_{k_1}, l_{k_2}, \ldots, l_{k_n}$. This special structure inspired the proposed algorithm that is developed for the original integrated formulation of (P1).

```
The integrated algorithm:
Step 1.
  for each $k$ {for each component type}
    for $l_{k_1}, l_{k_2}, \dots, l_{k_n}$ {for each possible combination of feeder locations}
      Find the heuristic solution of the MDVRP when component type k
        is assigned to feeder locations $l_{k_1}, l_{k_2}, \dots, l_{k_n}$.
        $\mathrm{DP}^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ = the cost of the MDVRP when component type $k$
        is located in feeders $l_{k_1}, l_{k_2}, \dots, l_{k_n}$.
        $\mathrm{DP}^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}} = z^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}} + \sum_{h=1}^{n} d_{l_{k_h}}$
        (Note that $d_{l_{k_h}}$ is a constant and does not depend on the touring decision)
Step 2. Solve the assignment problem by using $\mathrm{DP}^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ as the cost of assigning component type k to
feeder locations $l_{k_1}, l_{k_2}, \dots, l_{k_n}$.
```

$$\text{Min } \overline{Z1} = \sum_{k=1}^{K}\sum_{n=1}^{U}\sum_{h=1}^{n} \mathrm{DP}^{k,h}_{l_{k_1},l_{k_2},\dots,l_{k_h}} y^{k,h}_{l_{k_1},l_{k_2},\dots,l_{k_h}} \tag{15}$$

s.t. (2), (3).

While constraint set (2) ensures that at most one component type is assigned at each feeder, constraint set (3) guarantees each component type is assigned to feeders, in particular, to at least one and at most U feeders. We can solve this assignment problem by using an efficient assignment code.

We next present structural properties pertinent to the integrated algorithm developed to solve the PCB manufacturing problem by using a rotary-head CNC. We begin our analysis with a theorem showing that if the MDVRP is solved optimally and the cost is used as the cost of assigning component type k to feeder locations $l_{k_1}, l_{k_2}, \dots, l_{k_n}$, $\mathrm{DP}^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$, then the proposed integrated algorithm finds the optimal solution for the original problem.

Theorem 1. *The assignment problem with cost parameters equal to* $z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ (*optimal MDVRP cost when component type k is assigned to feeder locations* $l_{k_1}, l_{k_2}, \dots, l_{k_n}$) *solves the original problem* (*P*1) *to optimality.*

Proof. See Appendix A. □

The MDVRP is an NP-hard problem, and may not be solved optimally in polynomial time for large-scale versions. Therefore, we show that if the feasible solution generated for the MDVRP with an error of ε is used as the cost of assigning a component type to feeder locations, then the original problem (P1) has an error bound of ε as well. This is important because it establishes a theoretical error bound for the proposed integrated algorithm.

Theorem 2. *Suppose there exists* $z^{k,n,\mathrm{f}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ (*feasible solution cost of the MDVRP when component type k is assigned to feeder locations* $l_{k_1}, l_{k_2}, \dots, l_{k_n}$) *where*

$$\frac{z^{k,n,\mathrm{f}}_{l_{k_1},l_{k_2},\dots,l_{k_n}} - z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}}{z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}} \leqslant \varepsilon$$

($z^{k,n,\mathrm{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ *is the optimal MDVRP cost when component type k is assigned to feeder locations* $l_{k_1}, l_{k_2}, \dots, l_{k_n}$) *for every component type k in feeder locations* $l_{k_1}, l_{k_2}, \dots, l_{k_n}$. *If these* $z^{k,n,\mathrm{f}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ *values are used in the integrated heuristic, then the error for the integrated algorithm for* (*P*1) *is less than or equal to ε as well, i.e.*

$$\frac{Z1^{\mathrm{f}} - Z1^{*}}{Z1^{*}} \leqslant \varepsilon$$

($Z1^{\mathrm{f}}$ *is the feasible solution generated by using* $z^{k,n,\mathrm{f}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ *values and* $Z1^{*}$ *is the optimal solution to problem* ($P1$)).

Proof. See Appendix A. □

This theorem establishes a theoretical error guarantee under the presence of an ε-approximation algorithm for the MDVRP. Later, we show that this result can also be extended to empirical error guarantees.

The following theorem shows that relaxing the assumption of assigning all components of a component type into the same feeder location enables us to improve the solution found for the PCB problem. The models found in the literature consider the component types as assigned to only one feeder. Our formulation and solution algorithm provides the flexibility of assigning a component type into multiple feeders. The following theorem proves that the solution obtained from the proposed algorithm of this study yields improved results, or at least of the quality of solutions obtained from earlier work.

Theorem 3. *The objective function value of* ($P1$) *is less than or equal to that of the problem when component types are assigned to only one location.*

Proof. See Appendix A. □

The theoretical error guarantee of Theorem 2 can be extended to empirical error bounds. Cordeau et al. [7] present a tabu search heuristic for the MDVRP and compare it empirically with other heuristics. Their experiments show that the tabu search heuristic provides approximately at least 1% better results. In our analysis, we used the heuristic presented in [2]. If their algorithm is used to develop the feasible MDVRP cost values, $z^{k,n,\mathrm{f}}_{l_1,l_2,\dots,l_n}$, then as shown in the following remark, the empirical percentage error of the integrated algorithm would be improved.

Remark 4. Suppose there exists an empirical ε-approximation algorithm for the MDVRP. Hence, from Theorem 2, the integrated algorithm is an empirical ε-approximation algorithm for the original problem (P1) as well.

These results highlight another important aspect of Theorem 2 regarding future improvements in the solutions of the MDVRP. Whenever a new algorithm that performs better than the available MDVRP algorithms is established, the error guarantee of the integrated algorithm for the PCB problem also improves. This increases the significance of the result obtained through Theorem 2. The above remark demonstrates that the proposed algorithm of this paper is also practical. When a promising algorithm for the MDVRP is used, our proposed method provides solutions of at least the same quality of results, and is likely to yield better performances. This is because the solutions from the MDVRP usually have varying error percentages. When these solutions are used in the proposed algorithm, the integrated solution might contain the results of better performance MDVRP solutions. This means that the solution for the PCB manufacturing problem is likely to have a better performance than the MDVRP. In the worst case scenario, the solution of the proposed algorithm has the same performance of the MDVRP.

Remark 5. If the maximum empirical error bound of the MDVRP is $\varepsilon_{\max}$, then the maximum empirical error bound of the integrated algorithm is less than or equal to $\varepsilon_{\max}$.

This remark is a result of Theorem 2. When there are varying error gaps in the solutions of the MDVRP, the solution of the proposed algorithm is bound by the worst error gap, and is likely to have smaller error gaps. Therefore, the integrated algorithm's worst case error guarantee depends on that of the MDVRP.

We next develop a solution procedure by using Lagrangian relaxation. This procedure establishes the basis for our computational experiments.

3.3. Lagrangian relaxation

Lagrangian relaxation for problem (P1) provides further insight for the structural properties which are explored in our experimentation. We begin our analysis by relaxing several constraint sets. First, we divide constraint set (10) into two parts as follows:

$$\sum_{i\in S(k)}\sum_{i<j\in S(k)} x^k_{ij} \leqslant |S(k)|-1 \quad \text{for each component type } k=1,\dots,K,$$
$$\forall S(k)\subset \widetilde{V}(k),\ L_{S(k)}=1,\ |S(k)|\geqslant 2, \tag{10a}$$

$$\sum_{i\in S(k)}\sum_{i<j\in S(k)} x^k_{ij} \leqslant |S(k)|-L_{S(k)} \quad \text{for each component type } k=1,\dots,K,$$
$$\forall S(k)\subset \widetilde{V}(k),\ L_{S(k)}\geqslant 2,\ |S(k)|\geqslant 2. \tag{10b}$$

Next, we relax constraint sets (4)–(6), (8) and (10b) by using the following Lagrangian multipliers respectively:

$\alpha^{k,n}(k=1,\dots,K; n=1,\dots,U)$, $\quad\alpha^{k,n}$ is unrestricted in sign;

$\phi^{k,n}(k=1,\dots,K; n=1,\dots,U)$, $\quad\phi^{k,n}\leqslant 0$;

$\xi^{k,n}(k=1,\dots,K; n=1,\dots,U)$, $\quad\xi^{k,n}\leqslant 0$;

$\beta^{j,k}(j=L+1,\dots,n(k)+L; k=1,\dots,K)$, $\quad\beta^{j,k}$ is unrestricted in sign;

$\pi^{k,S(k)}\Big(k=1,\dots,K;\forall S(k)\subset\widetilde{V}(k), L_{S(k)}\geqslant 2, |S(k)|\geqslant 2\Big)$, $\quad\pi^{k,S(k)}\leqslant 0$.

The resulting formulation, denoted by, $P1^{\mathrm{LR1}}$, is as follows:

$$(P1^{\mathrm{LR1}}(\boldsymbol{\alpha},\boldsymbol{\phi},\boldsymbol{\xi},\boldsymbol{\beta},\boldsymbol{\pi})):\ \text{Min}\quad Z1^{\mathrm{LR1}}$$
$$\begin{aligned}
&= \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}}\Big(z^{k,1}_{l_{k_1}}+d_{l_{k_1}}+m(k)\phi^{k,1}_{l_{k_1}}\Big)+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L}\left(\alpha^{k,1}_{l_{k_1}}\left(-\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_1},j}x^k_{l_{k_1},j}-\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_1},j}x^k_{j,l_{k_1}}\right.\right.\\
&\quad\left.\left.-\sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L}c^k_{i,j}x^k_{i,j}+z^{k,1}_{l_{k_1}}\right)-\phi^{k,1}_{l_{k_1}}\left(\sum_{j=L+1}^{n(k)+L}x^k_{l_{k_1}j}\right)\right)+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L}y^{k,2}_{l_{k_1},l_{k_2}}\Big(z^{k,2}_{l_{k_1},l_{k_2}}+d_{l_{k_1}}+d_{l_{k_2}}\\
&\quad+m(k)\phi^{k,2}_{l_{k_1},l_{k_2}}\Big)+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L}\left(\alpha^{k,2}_{l_{k_1},l_{k_2}}\left(-\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{l_{k_h},j}-\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{j,l_{k_h}}\right.\right.\\
&\quad\left.\left.-\sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L}c^k_{i,j}x^k_{i,j}+z^{k,2}_{l_{k_1},l_{k_2}}\right)-\phi^{k,2}_{l_{k_1},l_{k_2}}\left(\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L}x^k_{l_{k_h}j}\right)\right)+\cdots\\
&\quad+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L}y^{k,U}_{l_{k_1},l_{k_2},\dots,l_{k_U}}\Big(z^{k,U}_{l_{k_1},l_{k_2},\dots,l_{k_U}}+d_{l_{k_1}}\\
&\quad+d_{l_{k_2}}+\cdots+d_{l_{k_U}}m(k)\phi^{k,U}_{l_{k_1},l_{k_2},\dots,l_{k_U}}\Big)
\end{aligned}$$

$$+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L}\left(\alpha^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(-\sum_{h=1}^{U}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{l_{k_h},j}-\sum_{h=1}^{U}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{j,l_{k_h}}\right.\right.$$
$$\left.\left.-\sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L}c^k_{i,j}x^k_{i,j}+z^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{kn}}\right)-\phi^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(\sum_{h=1}^{U}\sum_{j=L+1}^{n(k)+L}x^k_{l_{k_h}j}\right)\right)$$
$$+\sum_{k=1}^{K}\sum_{l=1}^{L}\xi^{k,l}\left(\left(y^{k,1}_l+\left(\sum_{l_{k_1}=1}^{l-1}y^{k,2}_{l_{k_1},l}+\sum_{l_{k_2}=l+1}^{L}y^{k,2}_{l,l_{k_2}}\right)+\cdots+\left(\sum_{l_{k_1}=1}^{l-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{l-U+2}\cdots\sum_{l_{k_U}}^{l-1}y^{k,U}_{l_{k_1},l_{k_2},\ldots,l}\right.\right.\right.$$
$$\left.\left.\left.+\cdots+\sum_{l_{k_2}=l+1}^{L-U+2}\sum_{l_{k_3}=l_{k_2}+1}^{L-U+3}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L}y^{k,U}_{l,l_{k_2},\ldots,l_{k_U}}\right)\right)-\sum_{j=L+1}^{n(k)+L}x^k_{lj}\right)+\sum_{j=L+1}^{L+n(k)}\sum_{k=1}^{K}\beta^{j,k}\left(2-\sum_{l=1}^{L}x^k_{lj}\right.$$
$$\left.-\sum_{i=L+1}^{j-1}x^k_{ij}-\sum_{i=j+1}^{n(k)+L}x^k_{ji}-\sum_{l=1}^{L}x^k_{jl}\right)+\sum_{k=1}^{K}\sum_{\substack{\widetilde{\forall S(k)\subset V(k)}\\ L_{S(k)}\geqslant 2,|S(k)|\geqslant 2}}\pi^{k,S(k)}\left(|S(k)|-L_{S(k)}-\sum_{i\in S(k)}\sum_{i<j\in S(k)}x^k_{ij}\right)$$

s.t. (2), (3), (7), (9), (10a), (11)–(13).

Problem ($P1^{\mathrm{LR1}}$) is now separable into two subproblems. One of the problems formed by this Lagrangian relaxation is an assignment-like problem. It contains Lagrange multipliers $\alpha^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{kn}}$, $\phi^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{kn}}$, $\xi^{k,l}$, $\beta^{j,k}$, and $\pi^{k,S(k)}$, and decision variables $y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{kn}}$ and $z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{kn}}$.

$$(P1^{\mathrm{LR1A}})=\mathrm{Min}\quad Z1^{\mathrm{LR1A}}$$
$$=\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L}\left[z^{k,1}_{l_{k_1}}\left(y^{k,1}_{l_{k_1}}+\alpha^{k,1}_{l_{k_1}}\right)+y^{k,1}_{l_{k_1}}\left(d_{l_{k_1}}+m(k)\phi^{k,1}_{l_{k_1}}+\xi^{k,l_{k_1}}\right)\right]$$
$$+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L}\left[z^{k,2}_{l_{k_1},l_{k_2}}\left(y^{k,2}_{l_{k_1},l_{k_2}}+\alpha^{k,2}_{l_{k_1},l_{k_2}}\right)\right.$$
$$\left.+y^{k,2}_{l_{k_1},l_{k_2}}\left(d_{l_{k_1}}+d_{l_{k_2}}+m(k)\phi^{k,n}_{l_{k_1},l_{k_2}}+\xi^{k,l_{k_1}}+\xi^{k,l_{k_2}}\right)\right]$$
$$+\cdots+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L}\left[z^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}+\alpha^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\right)\right.$$
$$\left.+y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(d_{l_{k_1}}+d_{l_{k_2}}+\cdots+d_{l_{k_U}}+m(k)\phi^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}+\xi^{k,l_{k_1}}+\xi^{k,l_{k_2}}+\cdots+\xi^{k,l_{k_U}}\right)\right]$$
$$+\sum_{j=L+1}^{L+n(k)}\sum_{k=1}^{K}2\beta^{j,k}+\sum_{k=1}^{K}\sum_{\substack{\widetilde{\forall S(k)\subset V(k)}\\ L_{S(k)}\geqslant 2,|S(k)|\geqslant 2}}\pi^{k,S(k)}\left(|S(k)|-L_{S(k)}\right)$$

s.t. (2), (3), (11), (13).

The first subproblem, $P1^{\mathrm{LR1A}}$, can be solved by using an efficient assignment algorithm.

The second subproblem, $P1^{\mathrm{LR1B}}$, becomes a degree-constrained minimal spanning tree problem with the decision variables x^k_{ij}.

$$(P1^{\mathrm{LR1B}})=\mathrm{Min}\quad Z1^{\mathrm{LR1B}}$$
$$=\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L}\left(\alpha^{k,1}_{l_{k_1}}\left(-\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_1},j}x^k_{l_{k_1},j}-\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_1},j}x^k_{j,l_{k_1}}-\sum_{i=L+1}^{n(k)+L-1}\sum_{j=i+1}^{n(k)+L}c^k_{i,j}x^k_{i,j}\right)-\phi^{k,1}_{l_{k_1}}\left(\sum_{j=L+1}^{n(k)+L}x^k_{l_{k_1}j}\right)\right)$$
$$+\sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L}\left(\alpha^{k,2}_{l_{k_1},l_{k_2}}\left(-\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{l_{k_h},j}-\sum_{h=1}^{2}\sum_{j=L+1}^{n(k)+L}c^k_{l_{k_h},j}x^k_{j,l_{k_h}}\right.\right.$$

$$
\begin{aligned}
&- \sum_{i=L+1}^{n(k)+L-1} \sum_{j=i+1}^{n(k)+L} c^k_{i,j} x^k_{i,j} \Bigg) - \phi^{k,2}_{l_{k_1},l_{k_2}} \left(\sum_{h=1}^{2} \sum_{j=L+1}^{n(k)+L} x^k_{l_{k_h} j} \right) \Bigg) \\
&+ \sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L-U+1} \sum_{l_{k_2}=l_{k_1}+1}^{L-U+2} \cdots \sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} \Bigg(\alpha^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} \Bigg(- \sum_{h=1}^{U} \sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j} x^k_{l_{k_h},j} \\
&- \sum_{h=1}^{U} \sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j} x^k_{j,l_{k_h}} - \sum_{i=L+1}^{n(k)+L-1} \sum_{j=i+1}^{n(k)+L} c^k_{i,j} x^k_{i,j} \Bigg) - \phi^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} \left(\sum_{h=1}^{U} \sum_{j=L+1}^{n(k)+L} x^k_{l_{k_h} j} \right) \Bigg) \\
&+ \sum_{k=1}^{K} \sum_{l=1}^{L} \xi^{k,l} \left(- \sum_{j=L+1}^{n(k)+L} x^k_{lj} \right) + \sum_{j=L+1}^{L+n(k)} \sum_{k=1}^{K} \beta^{j,k} \left(- \sum_{l=1}^{L} x^k_{lj} - \sum_{i=L+1}^{j-1} x^k_{ij} - \sum_{i=j+1}^{n(k)+L} x^k_{ji} - \sum_{l=1}^{L} x^k_{jl} \right) \\
&+ \sum_{k=1}^{K} \sum_{\substack{\forall S(k) \subset \widetilde{V}(k) \\ L_{S(k)} \geqslant 2, |S(k)| \geqslant 2}} \pi^{k,S(k)} \left(- \sum_{i \in S(k)} \sum_{i<j \in S(k)} x^k_{ij} \right) \\
&\text{s.t.} \quad (7),\ (9),\ (10a),\ (12).
\end{aligned}
$$

The second subproblem, $P1^{\mathrm{LR1B}}$, can be solved by using an augmented Lagrangian relaxation algorithm, which was originally introduced by Gavish [12], and later efficiently utilized in the delivery problem by Altınkemer and Gavish [2].

3.4. The subgradient optimization procedure

In this section we present a subgradient optimization procedure that is used for generating a lower bound. The objective function of the Lagrangian problem, $Z1^{\mathrm{LR}}$, for a given set of Lagrange multipliers, $\boldsymbol{\alpha}$, $\boldsymbol{\phi}$, $\boldsymbol{\xi}$, $\boldsymbol{\beta}$, and $\boldsymbol{\pi}$, gives a lower bound to the objective function value of the original problem (P1).

The subgradient optimization procedure is utilized to estimate the vectors of $\boldsymbol{\alpha}$, $\boldsymbol{\phi}$, $\boldsymbol{\xi}$, $\boldsymbol{\beta}$, and $\boldsymbol{\pi}$. It has effective applications in a variety of problems including [1–3,12]. Letting $x^k_{ij}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \boldsymbol{\xi}_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$, $y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \boldsymbol{\xi}_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$, and $z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \boldsymbol{\xi}_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$ be the optimal solution of the Lagrangian problem $Z1^{\mathrm{LR1}}$ for a fixed vector of $(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \boldsymbol{\xi}_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$ at the mth subgradient iteration, the subgradient directions are calculated as follows:

$$
\begin{aligned}
\gamma^{k,n}\left(\alpha^{k,n}_m\right) = z^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}} &- \sum_{k=1}^{K} \sum_{l_{k_1}=1}^{L-U+1} \sum_{l_{k_2}=l_{k_1}+1}^{L-U+2} \cdots \sum_{l_{k_U}=l_{k_{U-1}}+1}^{n} \Bigg(- \sum_{h=1}^{U} \sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j} x^k_{l_{k_h},j} - \sum_{h=1}^{U} \sum_{j=L+1}^{n(k)+L} c^k_{l_{k_h},j} x^k_{j,l_{k_h}} \\
&- \sum_{i=L+1}^{n(k)+L-1} \sum_{j=i+1}^{n(k)+L} c^k_{i,j} x^k_{i,j} \Bigg) \quad \text{for each } k = 1,\ldots,K,\ n = 1,\ldots,U, \text{ and} \\
&\text{feeder location combination of } (l_{k_1}, l_{k_2}, \ldots, l_{k_n});
\end{aligned}
$$

$$
\gamma^{k,n}\left(\phi^{k,n}_m\right) = m(k) y^{k,n}_{l_{k_1},l_{k_2},\ldots,l_{k_n}} - \sum_{h=1}^{n} \sum_{j=L+1}^{n(k)+L} x^k_{l_{k_h} j} \quad \text{for each } k = 1,\ldots,K,\ n = 1,\ldots,U, \text{ and}
$$

feeder location combination of $(l_{k_1}, l_{k_2}, \ldots, l_{k_n})$;

$$\gamma^{k,l}(\xi_m^{k,l}) = \left(y_l^{k,1} + \left(\sum_{l_{k_1}=1}^{l-1} y_{l_{k_1},l}^{k,2} + \sum_{l_{k_2}=l+1}^{L} y_{l,l_{k_2}}^{k,2} \right) + \cdots + \left(\sum_{l_{k_1}=1}^{l-U+1} \sum_{l_{k_2}=l_{k_1}+1}^{l-U+2} \cdots \sum_{l_{k_U}}^{l-1} y_{l_{k_1},l_{k_2},\ldots,l}^{k,U} + \cdots + \sum_{l_{k_2}=l+1}^{L-U+2} \sum_{l_{k_3}=l_{k_2}+1}^{L-U+3} \cdots \sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y_{l,l_{k_2},\ldots,l_{k_U}}^{k,U} \right) \right) - \sum_{j=L+1}^{n(k)+L} x_{lj}^k$$

for each feeder location $l = 1,\ldots,L$ and component type $k = 1,\ldots K$;

$$\gamma^{j,k}(\beta_m^{j,k}) = 2 - \sum_{l=1}^{L} x_{lj}^k - \sum_{i=L+1}^{j-1} x_{ij}^k - \sum_{i=j+1}^{n(k)+L} x_{ji}^k - \sum_{l=1}^{L} x_{jl}^k \quad \text{for each } j = L+1,\ldots,n(k)+L,\ k = 1,\ldots,K;$$

$$\gamma^{k,S(k)}(\pi_m^{k,S(k)}) = |S(k)| - L_{S(k)} - \sum_{i\in S(k)} \sum_{i<j\in S(k)} x_{ij}^k$$

for each $k = 1,\ldots,K$; $\forall S(k) \subset \widetilde{V}(k)$, $L_{S(k)} \geqslant 2$, $|S(k)| \geqslant 2$.

The vector multipliers for the $m+1$st subgradient iteration are found by:

$$\begin{aligned}
\alpha_{m+1}^{k,n} &= \alpha_m^{k,n} + t_m\gamma^{k,n}(\alpha_m^{k,n}),\\
\phi_{m+1}^{k,n} &= \min\{0, \phi_m^{k,n} + t_m\gamma^{k,n}(\phi_m^{k,n})\},\\
\xi_{m+1}^{k,l} &= \min\{0, \xi_m^{k,l} + t_m\gamma^{k,l}(\xi_m^{k,l})\},\\
\beta_{m+1}^{j,k} &= \beta_m^{j,k} + t_m\gamma^{j,k}(\beta_m^{j,k}),\\
\pi_{m+1}^{k,S(k)} &= \min\{0, \pi_m^{k,S(k)} + t_m\gamma^{k,S(k)}(\pi_m^{k,S(k)})\},
\end{aligned}$$

where

$$t_m = s_m \frac{\overline{Z1} - Z1^{\mathrm{LR1}}(\boldsymbol{\alpha},\boldsymbol{\phi},\boldsymbol{\xi},\boldsymbol{\beta},\boldsymbol{\pi})}{\|\boldsymbol{\alpha},\boldsymbol{\phi},\boldsymbol{\xi},\boldsymbol{\beta},\boldsymbol{\pi}\|^2} \quad \text{for all } k = 1,\ldots,K,\ j = L+1,\ldots,n(k)+L,$$
$$\forall S(k) \subset \widetilde{V}(k),\ L_{S(k)} \geqslant 2,\ |S(k)| \geqslant 2.$$

$\overline{Z1}$ is an upper bound of the objective function value of the original problem that corresponds to the feasible solution obtained from the heuristic. s_m is a scalar whose value is halved whenever no improvement in $Z1^{\mathrm{LR1}}$ is observed in a predetermined number of iterations. Below, we explain the procedure in detail:

Step 1. Initialization:

Set the upper bound for the feasible solution value to the heuristic solution ($\overline{Z1}$ = feasible solution value obtained from the heuristic) and choose initial values of $\boldsymbol{\alpha}_0$, $\boldsymbol{\phi}_0$, $\boldsymbol{\xi}_0$, $\boldsymbol{\beta}_0$, $\boldsymbol{\pi}_0$ for the Lagrange multipliers. Set $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}_0$, $\boldsymbol{\phi}^* = \boldsymbol{\phi}_0$, $\boldsymbol{\xi}^* = \boldsymbol{\xi}_0$, $\boldsymbol{\beta}^* = \boldsymbol{\beta}_0$, $\boldsymbol{\pi}^* = \boldsymbol{\pi}_0$, *non-improvement* counter IMP $= 0$, the iteration counter $m = 0$, the step size $s_m = s$ (the value is generally between 0 and 2 as suggested in [9]). We use zero as the initial values of all Lagrangian multipliers in our computer experiments.

Step 2. Solving the Lagrangian problem:

Solve the Lagrangian subproblem $P1^{\mathrm{LR1}}(\boldsymbol{\alpha}_m,\boldsymbol{\phi}_m,\boldsymbol{\xi}_m,\boldsymbol{\beta}_m,\boldsymbol{\pi}_m)$ using the current multipliers $\boldsymbol{\alpha}_m$, $\boldsymbol{\phi}_m$, $\boldsymbol{\xi}_m$, $\boldsymbol{\beta}_m$, and $\boldsymbol{\pi}_m$ and obtain the values of

$$Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}_m,\boldsymbol{\phi}_m,\boldsymbol{\xi}_m,\boldsymbol{\beta}_m,\boldsymbol{\pi}_m),\quad x_{i,j}^k(\boldsymbol{\alpha}_m,\boldsymbol{\phi}_m,\boldsymbol{\xi}_m,\boldsymbol{\beta}_m,\boldsymbol{\pi}_m),\quad y_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n}(\boldsymbol{\alpha}_m,\boldsymbol{\phi}_m,\boldsymbol{\xi}_m,\boldsymbol{\beta}_m,\boldsymbol{\pi}_m),\quad \text{and}$$
$$z_{l_{k_1},l_{k_2},\ldots,l_{k_n}}^{k,n}(\boldsymbol{\alpha}_m,\boldsymbol{\phi}_m,\boldsymbol{\xi}_m,\boldsymbol{\beta}_m,\boldsymbol{\pi}_m).$$

Step 3. Testing and updating the parameters:

3.1. If $Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$ is greater than the current $Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}^*, \boldsymbol{\phi}^*, \boldsymbol{\xi}^*, \boldsymbol{\beta}^*, \boldsymbol{\pi}^*)$, then set $Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}^*, \boldsymbol{\phi}^*, \boldsymbol{\xi}^*, \boldsymbol{\beta}^*, \boldsymbol{\pi}^*) = Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$.

3.2. If $x^k_{i,j}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$, $y^{k,n}_{l_{k_1}, l_{k_2}, \ldots, l_{k_n}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$, and $z^{k,n}_{l_{k_1}, l_{k_2}, \ldots, l_{k_n}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m)$ are feasible for problem P1, compute the corresponding value of $Z1$, and if it is less than $\overline{Z1}$, then set $\overline{Z1} = Z1$.

3.3. If $Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}_m, \boldsymbol{\phi}_m, \xi_m, \boldsymbol{\beta}_m, \boldsymbol{\pi}_m) < Z1^{\mathrm{LR1}}(\boldsymbol{\alpha}^*, \boldsymbol{\phi}^*, \boldsymbol{\xi}^*, \boldsymbol{\beta}^*, \boldsymbol{\pi}^*)$, set IMP $=$ IMP $+ 1$. If the value of IMP has reached a prespecified limit, set $m = m + 1$, $s_m = s_{m-1}/2$, $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}^*$, $\boldsymbol{\phi}_m = \boldsymbol{\phi}^*$, $\xi_m = \boldsymbol{\xi}^*$, $\boldsymbol{\beta}_m = \boldsymbol{\beta}^*$, and $\boldsymbol{\pi}_m = \boldsymbol{\pi}^*$.

3.4. Check the termination conditions. The algorithm terminates whenever the total number of iterations exceeds a prespecified limit, the step size s_m becomes exceedingly small, or when the values of overestimate and of the Lagrangian are acceptably close, i.e. the algorithm has converged within tolerance limit.

Step 4. Updating the multipliers:

The multipliers for the next step are computed as follows:

$$\begin{aligned}
\alpha^{k,n}_{m+1} &= \alpha^{k,n}_m + t_m \gamma^{k,n}\left(\alpha^{k,n}_m\right)\\
\phi^{k,n}_{m+1} &= \min\left\{0, \phi^{k,n}_m + t_m \gamma^{k,n}\left(\phi^{k,n}_m\right)\right\}\\
\xi^{k,l}_{m+1} &= \min\left\{0, \xi^{k,l}_m + t_m \gamma^{k,l}\left(\xi^{k,l}_m\right)\right\}\\
\beta^{j,k}_{m+1} &= \beta^{j,k}_m + t_m \gamma^{j,k}\left(\beta^{j,k}_m\right)\\
\pi^{k,S(k)}_{m+1} &= \min\left\{0, \pi^{k,S(k)}_m + t_m \gamma^{k,S(k)}\left(\pi^{k,S(k)}_m\right)\right\}
\end{aligned}$$

Step 5. Recursion:

Set $m = m + 1$, and go to Step 2.

We now can develop further structural properties of our solution procedure in light of the analysis presented in Section 3.2. Remark 5 stated that the maximum empirical error bound of the Integrated Algorithm is equal to that of the MDVRP. One algorithm for the MDVRP is presented in [1] where a k-iterated tour partitioning algorithm is used to develop feasible solutions for a given set of multiple depots. If this algorithm is used in generating solutions for our integrated algorithm, we present the worst case error guarantee. We also state that the resulting algorithm is non-polynomial.

Remark 6. If the polynomial k-iterated tour partitioning algorithm in [1] is used to generate the feasible solution for the MDVRP, then the worst case error guarantee for problem (P1) is $4 - (3/2Q)$ where Q is the capacity of the rotary head and $Q \geqslant 3$. Furthermore, the size of the assignment problem increases factorially in the number of feeder combinations. Therefore, the resulting algorithm would be non-polynomial.

The computational complexity of the integrated algorithm is therefore dependent on the number of components that are mounted on the PCB from each component type. The assignment of the proposed method has a computational complexity of $\mathrm{O}(K \times L^U)$. For each component type, the assignment problem considers the combinations of assigning to one feeder, to two feeders, up to U feeders. The integrated algorithm completes the assignment of component types to feeders in such a way that all the components are mounted on the PCB in one iteration. This provides the flexibility of having some feeders to be empty. However, it does not allow multiple iterations. As a result, the computational complexity of the integrated algorithm is $\mathrm{O}\left(K \times L^U + \sum_{k=1}^{K} (n(k))^3\right)$. It should be noted that the second term, $\mathrm{O}\left(\sum_{k=1}^{K} (n(k))^3\right)$, is the dominating factor and comes from the computational complexity of the MDVRP. This is certainly a computationally demanding method. However, this is not unusual for NP-Hard problems such as the VRP, and most solution methods for the VRP have similar computational complexities. Furthermore, some of the exact methods use branch-and-bound procedures that have computational complexity of $O(2^N)$ where N represents the number of components to be placed on the PCB. In the next section, we show the results of a computational study.

4. Computational experiments

In this section, we present the computational experiments that show the benefits of using the integrated approach provided in Section 3. We use twenty different examples of PCBs manufactured on a CNC with a rotary head as explained in the Introduction. Table 1 presents the characteristics of each PCB. For example, the first PCB has 11 component types and 1400 components in total, and PCB 4 has 27 component types and 1272 components. In order to represent the real characteristics of a PCB, we define five classes of component types. Class I components are allocated on a vertical line, Class II on a horizontal line, Class III on a diagonal line, Class IV on a circular line, while a majority of them are on random points on the board as Class V component types. The first PCB, for example, has two component types from Class I that are allocated on a vertical line. We have at most three component types from each of the first four classes on a PCB. Furthermore, we do not allow more than 24 components to be allocated on a single line. For example, the first component type of Class I has 120 components that need to be mounted on the first PCB. These components are located on five different vertical lines and the location of these lines are randomly determined. Class V component types represent those with completely random locations on the board which is rectangular with dimensions of 500 units of length and 200 units of width. We use uniform random generation to populate these components on the board. In the first PCB, there are four Class V component types, two of them with 240 components each and the other two with 80 components each. As stated in Table 1, we assume that there are 30 possible feeder locations for component type assignments in all of our experiments. It should be noted that these PCBs are fairly large as the total number of components to be mounted is relatively high.

In our computational analysis we compare single-feeder and multiple-feeder assignments for component types and show that multi-feeder assignments can improve the total cost (by reducing the total distance traveled). In the single-feeder assignment analysis, each component type is assigned to only one feeder. The feasible solutions for this approach are generated by using the method presented in Altınkemer et al. [3] where we solve a VRP for each component type at each feeder location. The multi-feeder assignment approach of this paper enables component types to be allocated in more than one feeder. In our experiments, we restrict each component type to be allocated to at most three feeders in order to reduce the computational time necessary to develop feasible solutions. Therefore, we solve a MDVRP for each combination of two and three feeders for component types. When a component type is assigned to adjacent feeders the savings from multi-feeder analysis would be small. Therefore, we do not allow adjacent feeder locations for feeder assignments of a component type. For this reason, when two-feeder and three-feeder assignments are considered, we require at least five feeders in between each feeder in the combination. This restriction reduces the number of MDVRPs solved for each component type. When two-feeder assignments are considered for a component type, there is a total of 325 alternative feeder assignment combinations. Similarly, when three-feeder assignments are investigated for a component type, there is a total of 1540 different alternatives. In total, we solve a MDVRP for 1865 different multi-feeder assignment alternatives for each component type considered in multi-feeder analysis.

In our computational analysis, we report the average, maximum, and minimum "percentage error gaps" for each approach. We experiment with ten replications for each PCB type and record the percentage error gap. The percentage error gap is calculated as the difference between the costs of the integrated algorithm (feasible solution) and the lower bound divided by the value of the lower bound. It can be mathematically expressed as follows:

$$\text{Percentage error gap} = \frac{\overline{Z1} - Z1^{\mathrm{LR1}}}{Z1^{\mathrm{LR1}}} \times 100\%$$

Table 1
Characteristics of 20 different PCB types and the associated number of components from each component type

Example PCB Type	Total no. of components on PCB	Total no. of component types on PCB	Total no. of feeders	Class I component types			Class II component types			Class III component types			Class IV component types		
				1	2	3	1	2	3	1	2	3	1	2	3
1	1400	11	30	120	80	0	120	120	0	80	80	0	160	0	0
2	1368	22	30	100	56	40	120	60	48	96	60	0	96	48	12
3	1304	21	30	80	76	0	96	84	0	80	44	0	80	32	12
4	1272	27	30	80	64	40	120	60	24	80	40	24	60	60	24
5	1240	19	30	100	80	0	120	60	0	80	60	0	80	60	0
6	1208	22	30	80	56	40	60	60	0	80	28	28	80	28	28
7	1144	26	30	100	36	0	60	36	24	60	40	28	72	28	28
8	1080	11	30	160	0	0	120	0	0	60	60	0	120	0	0
9	1054	20	30	92	48	0	128	18	0	128	20	0	128	20	0
10	1002	20	30	72	28	28	124	12	12	68	20	20	68	20	20
11	950	13	30	80	60	0	60	30	0	60	40	0	60	40	0
12	898	18	30	84	48	0	36	36	6	48	44	0	44	40	8
13	872	27	30	100	16	12	54	12	6	66	12	8	66	12	8
14	820	20	30	96	12	12	48	6	6	40	40	0	32	32	16
15	808	27	30	96	12	8	42	12	6	32	32	16	32	32	16
16	784	13	30	84	12	12	54	6	0	80	0	0	80	0	0
17	760	27	30	76	12	12	48	6	6	56	16	8	56	16	8
18	736	25	30	68	12	12	36	18	6	48	24	8	48	24	8
19	700	11	30	80	0	0	48	6	6	80	0	0	80	0	0
20	560	27	30	24	24	16	24	18	6	24	24	16	24	24	16

				Class V component types														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1400	11	30	80	240	240	80	0	0	0	0	0	0	0	0	0	0	0
2	1368	22	30	80	120	72	48	120	72	40	40	16	16	8	0	0	0	0
3	1304	21	30	80	80	120	72	48	96	88	32	32	32	24	16	0	0	0
4	1272	27	30	40	40	72	48	72	48	40	40	40	40	32	20	40	16	8
5	1240	19	30	40	40	120	72	48	80	80	40	40	24	16	0	0	0	0
6	1208	22	30	40	40	72	72	48	48	56	56	80	80	48	0	0	0	0
7	1144	26	30	32	32	36	144	48	48	128	24	24	44	32	16	8	8	8
8	1080	11	30	140	120	120	72	60	48	0	0	0	0	0	0	0	0	0
9	1054	20	30	40	40	40	40	40	40	40	40	40	40	40	32	0	0	0
10	1002	20	30	80	168	48	80	80	30	16	8	0	0	0	0	0	0	0
11	950	13	30	80	200	160	40	40	0	0	0	0	0	0	0	0	0	0
12	898	18	30	40	40	72	112	80	80	40	40	0	0	0	0	0	0	0
13	872	27	30	48	32	24	32	32	32	32	24	32	32	64	32	32	32	20
14	820	20	30	80	80	64	16	80	64	16	40	40	0	0	0	0	0	0
15	808	27	30	32	32	16	48	48	48	8	48	48	48	16	24	16	24	16
16	784	13	30	80	112	24	112	48	80	0	0	0	0	0	0	0	0	0
17	760	27	30	56	16	32	32	32	16	32	16	48	32	32	16	32	24	24
18	736	25	30	40	16	16	8	48	56	64	64	32	16	16	32	16	0	0
19	700	11	30	80	80	80	80	80	0	0	0	0	0	0	0	0	0	0
20	560	27	30	24	24	16	24	24	16	48	48	32	16	16	8	8	8	8

Note: Ten different PCBs are generated from each PCB type for computational experiments.

where $\overline{Z1}$ is the feasible solution value obtained from the integrated approach and $Z1^{LR1}$ is the value of the lower bound for the multi-feeder PCB problem as presented in Section 3.1. We use the subgradient optimization procedure in our computational experiments, as explained in Sections 3.3 and 3.4.

The computational results in Table 2 show that the integrated algorithm that uses multi-feeder assignments reduces the total cost significantly. The integrated algorithm of Section 3.2 has average percentage error gaps ranging between 1.68% and 7.04%. The minimum and maximum error gaps are recorded as 1.62% and 7.10%, respectively. Considering the combinatorial complexity of the problem and the size of the PCBs considered, these results are favorable. Table 2 also presents the results of the integrated algorithm where each component type is assigned to a single feeder. The average percentage error gap ranges between 1.87% and 19.36% with a minimum and maximum of 1.84% and 19.52%, respectively. Our experiments conclude that in all of the PCB types, the multi-feeder approach has a lower percentage error gap than the single-feeder approach. This is illustrated with the last column of Table 2 where we report the average percentage improvement that the multi-feeder solution value provides over the single-feeder feasible solution. We only report the average percentage improvement because the minimum and maximum percentage improvements are close to the averages. The multi-feeder assignment improves the feasible solution value by at least 0.15% and at most 10.32% in our experiments. We observe that as the number of components from a component type increases, our VRP algorithm does not generate the same quality of feasible solutions resulting in higher percentage error gaps. This is typical of most combinatorial problems including the VRP. However, when these component types with a higher number of components were divided into multiple feeders, the solutions improve dramatically. In particular, we find our Class V component types (randomly located on the board) to provide higher savings with MDVRP solutions (versus VRP solutions) than other classes of component types. Therefore, in our multi-feeder analysis we allow only Class V component types to be assigned to multiple (two or three) feeders. This is because the VRP solutions for the first four classes provide relatively superior results with less savings from multiple-feeder assignments. We conjecture that this is an outcome of the algorithm that we use to develop feasible solutions. Therefore, we cannot conclude that it is better to allocate randomly located component types into multiple feeders than those located on lines. However, it is easy to conclude that a multi-feeder assignment yields higher savings when there are relatively less component types (compared to the total number of available feeders) with a high number of components from each type. PCB types 1, 8, 11, 16 and 19 are good examples of this observation. When the total number of component types is close to the number of available feeders, the savings from multi-feeder assignments reduce. Similarly, PCB types 4, 9, 13, 15, 17 and 20 present smaller savings from multi-feeder assignments. It should be noted that the parallel savings algorithm of Altınkemer [1] is used to develop feasible solutions for the individual MDVRP solutions. We conjecture that if the MDVRP algorithm presented in [7] were used, our percentage error gaps would have been improved. Nonetheless, the results presented in this paper are the first of its kind and serve as a basis for future algorithms. A final observation is made regarding the utilization of feeders. In almost all problems, we achieve the maximum utilization of feeders in optimal solutions. This indicates that locating component types in multiple feeders can significantly improve the quality of the solution for large PCBs as experimented in this study. Component types that were assigned to single feeders had a higher percentage of error gaps than those located in multiple feeders. This enables us to conclude that utilizing more feeder locations for component types that have higher number of components can be significant in terms of the quality of the solution.

Providing the flexibility of multi-feeder assignments come at a cost of computational time. The model presented in Section 3 suggests solving a MDVRP for each possible feeder combination for every component type. This can increase the computational needs exponentially. In order to illustrate the benefits of multi-feeder assignments without extensive computational time, we restrict the number of

Table 2
Computational experiments comparing single-feeder versus multi-feeder assignments over 20 PCB types

Example PCB type	Total no. of components on PCB	Total no. of component types on PCB	Total no. of feeders	Single feeder			Multiple feeders			Avg. % impovement from using multiple feeders
				Avg. % error gap	Min. % error gap	Max. % error gap	Avg. % error gap	Min. % error gap	Max. % error gap	
1	1400	11	30	**19.36**	19.24	19.52	**7.04**	6.97	7.10	10.32
2	1368	22	30	**11.58**	11.50	11.68	**6.90**	6.84	6.97	4.19
3	1304	21	30	**9.75**	9.70	9.81	**6.55**	6.43	6.67	2.91
4	1272	27	30	**7.08**	7.03	7.13	**6.33**	6.17	6.48	0.70
5	1240	19	30	**8.33**	8.24	8.39	**6.07**	5.84	6.30	2.09
6	1208	22	30	**6.96**	6.92	7.00	**5.85**	5.66	6.03	1.04
7	1144	26	30	**8.50**	8.42	8.59	**5.57**	5.33	5.80	2.70
8	1080	11	30	**12.67**	12.52	12.78	**5.46**	5.36	5.55	6.40
9	1054	20	30	**5.51**	5.46	5.55	**5.35**	5.28	5.44	0.15
10	1002	20	30	**12.25**	12.13	12.36	**5.00**	4.94	5.10	6.46
11	950	13	30	**16.09**	15.89	16.23	**4.70**	4.63	4.82	9.81
12	898	18	30	**6.41**	6.37	6.46	**4.42**	4.35	4.57	1.87
13	872	27	30	**4.55**	4.49	4.60	**4.31**	4.24	4.47	0.24
14	820	20	30	**6.65**	6.58	6.71	**4.16**	4.09	4.25	2.33
15	808	27	30	**4.11**	4.07	4.15	**3.92**	3.81	4.01	0.18
16	784	13	30	**7.99**	7.90	8.07	**3.35**	3.24	3.44	4.29
17	760	27	30	**3.01**	2.97	3.07	**2.87**	2.73	2.97	0.23
18	736	25	30	**4.23**	4.19	4.28	**2.42**	2.24	2.56	1.73
19	700	11	30	**6.02**	5.99	6.08	**1.78**	1.75	1.81	3.99
20	560	27	30	**1.87**	1.84	1.90	**1.68**	1.62	1.74	0.18

Notes: (1) Average percentage error gap, minimum percentage error gap, and maximum percentage error gap are calculated after 10 experiments for each PCB. (2) Average percentage improvement from using multiple feeders is calculated by dividing the difference between feasible solution values of single-feeder assignment and multi-feeder assignment problems by the feasible solution value of the single-feeder assignment problem.

component types for consideration. Although our approach is computationally demanding in an exhaustive search, the computational time for our experiments is relatively modest for such complexity. Table 3 provides a detailed summary of the CPU times for both the single-feeder approach utilizing VRP solutions and the multi-feeder approach using MDVRP solutions. The number of VRPs solved is determined by multiplying the number of component types in each PCB with the total number of feeders (30 in our experiments). In order to reduce the total computational time, we allow only Class V component types to be assigned to multiple feeders. We first calculate the number of feeders available for multi-feeder assignments. For example, PCB 4 has 27 component types and 30 feeders. It means that we have at most three component types that can be assigned to multiple feeders for PCB 4. Therefore, we solve MDVRPs for only three component types, the ones that have the highest number of (components from Class V component types) on PCB 4. The number of component types considered for multi-feeder assignment for each PCB type is also given in Table 3. This number, multiplied with 1852 different feeder assignment alternatives, gives the total number of MDVRPs solved for each PCB type. We report the average CPU time that was necessary to complete the multi-feeder analysis for each PCB type. Next, we provide the average, minimum, and maximum CPU times for the entire problem. The total CPU time includes the single-feeder, multi-feeder, and the assignment procedure CPU times. While the maximum computation time is recorded as 47,346.6 CPU seconds on a Pentium IV 1.7 GHz computer, the minimum computation time is 8,835.3 CPU seconds. At this point a PCB manufacturer has to decide on a

Table 3
Analysis of computational time in experiments

Example PCB type	Total no. of components on PCB	Total no. of component types on PCB	Total no. of feeders	No. of VRPs solved	No. of component types considered for MDVRP	No. of MDVRPs solved	Single-feeder average CPU time	Multi-feeder average CPU time	Average total CPU time	Min. total CPU time	Max. total CPU time
1	1400	11	30	330	3	5595	3453.2	43,659.4	47,162.7	46,997.3	47,346.6
2	1368	22	30	660	5	9325	2230.4	44,129.2	46,409.5	46,211.1	46,587.7
3	1304	21	30	630	5	9325	2029.7	40,495.5	42,573.1	42,361.2	42,710.8
4	1272	27	30	810	2	3730	1268.0	12,223.6	13,539.2	13,445.6	13,632.9
5	1240	19	30	570	4	7460	1912.3	18,431.6	20,390.4	20,213.3	20,522.4
6	1208	22	30	660	4	7460	1456.4	17,347.7	18,848.5	18,741.4	18,972.5
7	1144	26	30	780	2	3730	2286.7	13,943.5	16,273.5	16,143.1	16,364.8
8	1080	11	30	330	4	7460	2618.6	21,789.5	24,450.7	24,290.8	24,611.9
9	1054	20	30	600	3	5595	1092.8	11,289.9	12,424.1	12,316.7	12,586.9
10	1002	20	30	600	4	7460	3149.0	18,104.0	21,292.9	21,134.7	21,428.8
11	950	13	30	390	3	5595	3077.4	27,211.9	30,328.0	30,276.4	30,491.2
12	898	18	30	540	4	7460	3078.1	20,919.4	24,035.4	23,911.8	24,182.3
13	872	27	30	810	2	3730	1110.4	10,274.4	11,421.3	11,345.2	11,534.1
14	820	20	30	600	3	5595	1391.1	12,371.0	13,797.7	13,673.5	13,924.5
15	808	27	30	810	3	5595	1135.0	9864.9	11,034.3	10,895.2	11,176.6
16	784	13	30	390	4	7460	1946.2	18,759.2	20,739.8	20,586.9	20,881.3
17	760	27	30	810	2	3730	1117.3	8276.0	9426.6	9323.7	9510.3
18	736	25	30	750	4	7460	1107.1	14,402.1	15,541.7	15,437.0	15,703.6
19	700	11	30	330	5	9325	1657.2	16,793.6	18,481.9	18,392.1	18,588.2
20	560	27	30	810	3	5595	1088.7	7801.0	8920.0	8835.3	8997.6

Notes: (1) No. of VRPs solved is calculated by multiplying the total no. of component types with the total no. of feeders. (2) For each component type considered for MDVRP, there is a total of 1865 two-feeder and three-feeder assignment combinations. (3) No. of MDVRPs solved is calculated by multiplying the no. of component types considered for MDVRP with 1865. (4) Total CPU Time includes the CPU times of single-feeder and multi-feeder analysis as well as the solution of the assignment problem.

trade-off between the quality of the solution and the time to obtain it. Our proposed multi-feeder approach can benefit the manufacturers who produce large quantities of the same PCB. Considering that the multi-feeder PCB problem is solved once for millions of PCBs manufactured for large manufacturers, this computational time is a relatively small effort on behalf of the manufacturer. Alternatively, if the manufacturer does not produce large quantities of the same PCB our method becomes costly in time. Thus, it is better for such a manufacturer to impose the single-feeder restriction in order to save computational time. It should also be noted here that as the number of components in a component type increases, the computational time increases exponentially as can be observed in all VRP and MDVRP problems.

5. Conclusions

The PCB manufacturing problem presents a series of optimization problems which need to be formulated and solved together in order to increase productivity. The integrated approach presented in this paper simultaneously accounts for the problems of component assignment to feeders and the sequencing of placement while minimizing the total head movement.

We prove that when the optimal solution for the MDVRP is known, our proposed algorithm finds the optimal solution for the PCB manufacturing problem which belongs to NP-hard class. However, the MDVRP is also an NP-hard problem, and it is unlikely to obtain optimal solutions in polynomial time. Therefore, we theoretically show that our integrated algorithm provides a feasible solution with an error bound less than or equal to the *maximum* error bound of the MDVRP. Furthermore, if all the error bounds of the MDVRP are equal to ε, then our proposed algorithm has an ε-error guarantee. Otherwise, the error guarantee can be determined by the largest error bound in the MDVRP. We present the results of a computational study that support these theoretical findings. The computational analysis shows that there are significant benefits when component types are assigned to multiple feeders for densely populated PCBs. Such flexibility brings additional computation time. It can be argued that solving a MDVRP for each component type and possible feeder locations is a time-consuming process. Reducing the number of component types to be considered for multi-feeder assignments decreases the computational needs dramatically. Furthermore, the additional computational time is justified by solving the integrated problem once for extremely large quantities of a PCB type with the same setup. Therefore, our proposed method is beneficial for manufacturers who produce large quantities of the same PCB. When this is the case, this integrated algorithm is not repetitive and does not require extensive effort on the part of the manufacturer. For other manufacturers, we suggest imposing the single-feeder restriction for each component type in order to save computational time.

This paper can be extended in two ways. First, a linear time in distance (cost) for the head movement is assumed between points on the PCB in our paper. A study that investigates non-linear travel time could result in better estimates. Furthermore, our study is designed around a technology that uses rotary heads. Therefore, the solutions do not apply to all machines used for PCB manufacturing. However, our paper opens the channels for utilizing optimization techniques in other technologies used in PCB manufacturing.

Acknowledgements

We would like to thank the three anonymous referees for their insightful comments.

Appendix A

Proof of Theorem 1. Consider formulation (P1) and relax constraint set (4) multiplying it by vector $\boldsymbol{\alpha}$ ($\boldsymbol{\alpha}$ is unrestricted in sign). The problem is composed of two subproblems: one assignment-like in variables, $y^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$, and one multi-depot vehicle routing-like in variables x^k_{ij}. The variables of these two subproblems are tied by constraint set (5). When each k and $(l_{k_1}, l_{k_2}, \dots, l_{k_n})$ combination of component types and feeder locations are assigned $y^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}} = 1$, a multi-depot vehicle routing-like problem can be solved. This provides the optimal cost of the MDVRP, $z^{k,n,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$, for component type k located in feeders $l_{k_1}, l_{k_2}, \dots, l_{k_n}$. When $z^{k,n,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$ is used in the assignment-like problem without constraint set (5), the optimal solution for the original problem (P1) can be obtained. □

Proof of Theorem 2. Let us consider the optimal solution for the assignment problem:

$$Z1^* = \text{Min} \quad \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}}\left(z^{k,1,\text{opt}}_{l_{k_1}} + d_{l_{k_1}}\right) + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}}\left(z^{k,2,\text{opt}}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}}\right) + \cdots$$
$$+ \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\dots,l_{k_U}}\left(z^{k,U,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_U}} + d_{l_{k_1}} + d_{l_{k_2}} + \cdots + d_{l_{k_U}}\right)$$

s.t.

$$\sum_{k=1}^{K} y^{k,1}_{l} + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-1} y^{k,2}_{l_{k_1},l} + \sum_{l_{k_2}=l+1}^{L} y^{k,2}_{l,l_{k_2}}\right) + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-2}\sum_{l_{k_2}=l_{k_1}+1}^{l-1} y^{k,3}_{l_{k_1},l_{k_2},l} + \sum_{l_{k_1}=1}^{l-1}\sum_{l_{k_3}=l+1}^{L} y^{k,3}_{l_{k_1},l,l_{k_3}}\right.$$
$$\left. + \sum_{l_{k_2}=l+1}^{L-1}\sum_{l_{k_3}=l_{k_2}+1}^{L} y^{k,3}_{l,l_{k_2},l_{k_3}}\right) + \cdots + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{l-U+2}\cdots\sum_{l_{k_{U-1}}=l_{k_{U-2}}+1}^{l-1} y^{k,U}_{l_{k_1},l_{k_2},\dots,l} + \cdots\right.$$
$$\left. + \sum_{l_{k_2}=l+1}^{L-U+2}\sum_{l_{k_3}=l_{k_2}+1}^{L-U+3}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l,l_{k_2},\dots,l_{k_U}}\right) \leqslant 1 \quad \text{for each feeder location } l = 1,\dots,L,$$

$$\sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}} + \sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}} + \cdots + \sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\dots,l_{k_U}} = 1$$

for each component type $k = 1,\dots,K$.

In the optimal solution, there will be only K of $y^{k,n}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$'s which are equal to 1. Then,

$$Z1^* = y^{1,n1,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n1}}} \times z^{1,n1,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n1}}} + y^{2,n2,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n2}}} \times z^{2,n2,opt}_{l_{k_1},l_{k_2},\dots,l_{k_{n2}}} + \cdots + y^{K,nK,\text{opt}}_{ll_{k_1},l_{k_2},\dots,l_{k_{nK}}} \times z^{K,nK,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{nK}}}.$$

Denote the optimal vector of $y^{k,n,*}_{l_{k_1},l_{k_2},\dots,l_{k_n}}$'s with:

$$y^{\text{OPT}} = \left(y^{1,n1,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n1}}}, y^{2,n2,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n2}}}, \dots, y^{K,nK,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{nK}}}\right).$$

Now, consider the solution for the assignment problem by using the feasible solution of the MDVRP. When the combination of

$$y^{\text{OPT}} = \left(y^{1,n1,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n1}}}, y^{2,n2,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{n2}}}, \dots, y^{K,nK,\text{opt}}_{l_{k_1},l_{k_2},\dots,l_{k_{nK}}}\right)$$

assignments (assignment of component types to feeders) is used, it will generate a feasible solution for the integrated algorithm. The cost of this solution, however, will be calculated by using the feasible solution costs of the MDVRP. The total assignment cost,

$$Z^{\mathrm{AP,F}} = y^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} \times z^{1,n1,\mathrm{f}}_{ll_{k_1},l_{k_2},\ldots,l_{k_{n1}}} + y^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} \times z^{2,n2,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} + \cdots + y^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} \times z^{K,nK,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}},$$

of this combination will be greater than or equal to the minimum cost by using the feasible solutions of the MDVRP, $Z1^{\mathrm{f}}$.

$$\begin{aligned} Z1^{\mathrm{f}} = \mathrm{Min}\quad & \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}}\left(z^{k,1,\mathrm{f}}_{l_{k_1}} + d_{l_{k_1}}\right) + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}}\left(z^{k,2,\mathrm{f}}_{l_{k_1},l_{k_2}} + d_{l_{k_1}} + d_{l_{k_2}}\right) + \cdots \\ & + \sum_{k=1}^{K}\sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}}\left(z^{k,U,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} + d_{l_{k_1}} + d_{l_{k_2}} + \cdots + d_{l_{k_U}}\right) \end{aligned}$$

s.t.

$$\begin{aligned} & \sum_{k=1}^{K} y^{k,1}_{l} + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-1} y^{k,2}_{l_{k_1},l} + \sum_{l_{k_2}=l+1}^{L} y^{k,2}_{l,l_{k_2}}\right) + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-2}\sum_{l_{k_2}=l_{k_1}+1}^{l-1} y^{k,3}_{l_{k_1},l_{k_2},l} + \sum_{l_{k_1}=1}^{l-1}\sum_{l_{k_3}=l+1}^{L} y^{k,3}_{l_{k_1},l,l_{k_3}}\right. \\ & \left. + \sum_{l_{k_2}=l+1}^{L-1}\sum_{l_{k_3}=l_{k_2}+1}^{L} y^{k,3}_{l,l_{k_2},l_{k_3}}\right) + \cdots + \sum_{k=1}^{K}\left(\sum_{l_{k_1}=1}^{l-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{l-U+2}\cdots\sum_{l_{k_{U-1}}=l_{k_{U-2}}+1}^{l-1} y^{k,U}_{l_{k_1},l_{k_2},\ldots,l} + \cdots\right. \\ & \left. + \sum_{l_{k_2}=l+1}^{L-U+2}\sum_{l_{k_3}=l_{k_2}+1}^{L-U+3}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l,l_{k_2},\ldots,l_{k_U}}\right) \leqslant 1 \quad \text{for each feeder location } l = 1,\ldots,L, \\ & \sum_{l_{k_1}=1}^{L} y^{k,1}_{l_{k_1}} + \sum_{l_{k_1}=1}^{L-1}\sum_{l_{k_2}=l_{k_1}+1}^{L} y^{k,2}_{l_{k_1},l_{k_2}} + \cdots + \sum_{l_{k_1}=1}^{L-U+1}\sum_{l_{k_2}=l_{k_1}+1}^{L-U+2}\cdots\sum_{l_{k_U}=l_{k_{U-1}}+1}^{L} y^{k,U}_{l_{k_1},l_{k_2},\ldots,l_{k_U}} = 1 \end{aligned}$$

for each component type $k = 1,\ldots,K$.

Next, the objective function of the assignment problem for the combination y^{OPT} is calculated. Notice that the combination y^{OPT} is also a feasible combination for the integrated algorithm. Then,

$$Z^{\mathrm{AP,F}} = y^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} \times z^{1,n1,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} + y^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} \times z^{2,n2,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} + \cdots + y^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} \times z^{K,nK,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}},$$

$$\begin{aligned} Z^{\mathrm{AP,F}} \leqslant{} & y^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} \times \left(z^{1,n1,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} + \varepsilon \times z^{1,n1,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}}\right) + y^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} \times \left(z^{2,n2,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} + \varepsilon \times z^{2,n2,\mathrm{f}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}}\right) \\ & + \cdots + y^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} \times \left(z^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} + \varepsilon \times z^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}}\right), \end{aligned}$$

$$\begin{aligned} Z^{\mathrm{AP,F}} \leqslant{} & \left(y^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} \times z^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} + y^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} \times z^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} + \cdots + y^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} \times z^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}}\right) \\ & + \varepsilon \times \left(y^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} \times z^{1,n1,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n1}}} + y^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} \times z^{2,n2,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{n2}}} + \cdots + y^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}} \times z^{K,nK,\mathrm{opt}}_{l_{k_1},l_{k_2},\ldots,l_{k_{nK}}}\right), \end{aligned}$$

$$Z^{\mathrm{AP,F}} \leqslant Z1^* + \varepsilon \times Z1^*.$$

We also know that the optimal solution value of the integrated algorithm, $Z1^{\mathrm{f}}$, will be less than or equal to the cost of this combination.

$$\begin{aligned} & Z1^{\mathrm{f}} \leqslant Z^{\mathrm{AP,F}}, \\ & Z1^{\mathrm{f}} \leqslant Z1^* + \varepsilon \times Z1^*, \\ & Z1^{\mathrm{f}} \leqslant Z1^* \times (1+\varepsilon), \\ & \frac{Z1^{\mathrm{f}}}{Z1^*} \leqslant 1 + \varepsilon, \\ & \frac{Z1^{\mathrm{f}} - Z1^*}{Z1^*} \leqslant \varepsilon. \qquad \square \end{aligned}$$

Proof of Theorem 3. The problem that the components of the same type are assigned to a single feeder, defined as problem (PR), has a feasible region which is only a portion of the feasible region of (P1). Therefore, the optimal solution of problem (PR) is also feasible for problem (P1) and its value can only be greater than or equal to the optimal objective function value of (P1). □

References

[1] K. Altınkemer, Topological design of ring networks, Computers and Operations Research 21 (4) (1994) 421–431.

[2] K. Altınkemer, B. Gavish, Parallel savings based heuristics for delivery problem, Operations Research 39 (3) (1991) 456–469.

[3] K. Altınkemer, B. Kazaz, M. Koksalan, H. Moskowitz, Optimization of printed circuit board manufacturing: Integrated modeling and algorithms, European Journal of Operational Research 124 (2000) 409–421.

[4] M. Ball, M. Magazine, Sequencing of insertions in printed circuit board assembly, Operations Research 36 (2) (1988) 192–201.

[5] Y. Crama, O. Flippo, J. van de Klundert, F. Spieksma, The assembly of printed circuit boards: A case with multiple machines and multiple board types, European Journal of Operational Research 98 (1997) 457–472.

[6] Y. Crama, J. van de Klundert, F. Spieksma, Production planning problems in printed circuit board assembly, Working Paper, 1999. Available from <http://www.cwi.nl/donet/publications.html>.

[7] J.F. Cordeau, M. Gendreau, G. Laporte, A Tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks 30 (1997) 105–119.

[8] Z. Drezner, S.Y. Nof, On optimizing bin picking and insertion plans for assembly robots, IIE Transactions 16 (3) (1984) 262–270.

[9] M.L. Fisher, The Lagrangian relaxation method for solving integer programming problems, Management Science 27 (1) (1981) 1–18.

[10] M.R. Garey, D.S. Johnson, Computers and Tractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.

[11] B. Gavish, Topological design of centralized computer networks: formulations and algorithms, Networks 12 (1982) 355–377.

[12] B. Gavish, Augmented Lagrangean based algorithms for solving CMST problems, IEEE Transactions on Communications 33 (12) (1983) 1247–1257.

[13] B. Gavish, A. Seidmann, Printed circuit boards assembly automation—formulations and algorithms, in: A. Midal (Ed.), Recent Developments in Production Research, Elsevier Science Publishers, Amsterdam, 1988, pp. 695–717.

[14] A.M. Geoffrion, Lagrangian relaxation for integer programming, Mathematical Programming Study 2 (1974) 82–114.

[15] G. Laporte, Y. Nobert, M. Desrochers, Optimal routing under capacity and distance restrictions, Operations Research 33 (1985) 1050–1073.

[16] T. Leipala, O. Nevalainen, Optimization of the movements of a component placement machine, European Journal of Operational Research 38 (1989) 167–177.

[17] L.F. McGinnis, J.C. Ammons, M. Carlyle, L. Cranmer, G.W. DePuy, K.P. Ellis, C.A. Tovey, H. Xu, Automated process planning for printed circuit card assembly, IIE Transactions 24 (4) (1992) 18–30.

[18] I. Or, E. Demirkol, Optimization issues in automated production of printed circuit boards: Operations sequencing and feeder configuration problems, Transactions on Operational Research 8 (1) (1996) 9–23.

[19] J.L. Yi, N.D. Simchi-Levi, Worst case analysis of heuristics for multi-depot capacitated vehicle routing problems, INFORMS Journal on Computing 2 (1990) 64–73.