COMPONENT FIXTURE POSITIONING/SEQUENCING FOR PRINTED CIRCUIT BOARD ASSEMBLY WITH CONCURRENT OPERATIONS

JAVAD AHMADI

IBM Systems Technology, Austin, Texas

REZA AHMADI

University of California, Los Angeles, California

HIROFUMI MATSUO and DEVANATH TIRUPATI

The University of Texas at Austin, Austin, Texas

(Received February 1990; revisions received March 1991, April, November 1993, January 1994; accepted February 1994)

This paper considers the problem of positioning component fixtures on the carriers of computer, numerically controlled dual delivery machines used for populating printed circuit boards with surface mounted technology. This reel positioning problem (**RPP**) is one of a series of optimization problems that are critical for improving system productivity and realizing the full potential of concurrent operations. We formulate the **RPP** as a mathematical program and establish its complexity. Since the problem is NP-complete we focus on the development of heuristics. Our solution procedure was prompted by engineering considerations that included concerns for minimizing the changes in the carrier direction and total movement. We also present encouraging results with test problems. The method has been implemented and achieved 7 to 8% reductions in cycle time.

There has been a growing interest in the modeling and analysis of manufacturing systems in the electronic and semiconductor industry. This research has been motivated by the developments in the design of computer numerically controlled (CNC) machines. Some CNC machines can perform several operations concurrently, thereby significantly increasing the productivity (commonly called burst rate). However, Ahmadi, Grotzinger and Johnson (1988) note that realization of high-burst rates requires efficient use of concurrent operations, and in the absence of optimal (or near-optimal) planning and scheduling, the systems degenerate into performing operations serially, which results in low production rates. Ball and Magazine (1988) point out that while there is a considerable amount of literature dealing with the design of printed circuit board and VLSI circuitry (for details, see Soukup 1981, Hu and Kuh 1985, and the references therein), increased productivity may be obtained by focusing on the manufacturing issues. For research in this area, see Ahmadi (1986), Lofgren and McGinnis (1986a, b), Tang (1986), and Tang and Denardo (1988a, b).

This paper focuses on an optimization problem that is crucial to the improvement of productivity in printed circuit boards (PCB) assembly operations. We consider a CNC dual delivery placement machine for populating PCBs with surface mounted technology (SMT) components and examine the problem of positioning reels of components on the carrier. The motivation for this work comes from operations of the Dynapert MPS 500 machine at the IBM plant in Austin. This system was studied by Ahmadi, Grotzinger and Johnson (1988), who presented a hierarchy of three optimization problems relating to productivity improvements. These authors examined two of the three problems in detail, and presented solution procedures. This paper deals with the third problem, which relates to the positioning of component fixtures on the component carrier. We formulate this problem, termed the reel positioning problem (**RPP**), as a mathematical program and establish its computational complexity. Since **RPP** is NP-complete, we focus on the development of heuristics. Our solution procedure was prompted by engineering considerations that included concerns for minimizing the carrier direction changes and total carrier movement. Ease of implementation was another major requirement for the proposed algorithms. We also present results with test problems that suggest that the procedures developed require modest computational effort and are suitable for implementation. In addition, these results indicate that over the range of problem parameters we have experienced, the solution procedures have a potential to significantly increase the burst rates. These procedures were implemented and resulted in 7–8% improvements in cycle time. These improvements are relative to the procedures previously in place and are due to the reel positions determined by our heuristics.

While the problem studied is specific to this application, it has features common to other systems with concurrent operations. For example, while the Dynapert MPS 500 is no longer in operation at the IBM Austin

Subject classifications: Manufacturing, automated systems: populating printed circuit boards. Production/scheduling, approximation/heuristics; sequencing; concurrent operations.
Area of review: MANUFACTURING, OPERATIONS AND SCHEDULING.

0030-364X/95/4303-0444 $01.25

plant, the heuristics developed in this paper have found application in other machines with concurrent operations, such as the Panasonic MK1, Dynapert MPS 2500, and MV. Hence, we expect the proposed methods to have a wider application.

The remainder of the paper is organized as follows. In the following section we describe the manufacturing process using CNC machines for pick-and-place operations and introduce the reel positioning problem. We also present a brief review of the framework of optimization problems in these types of systems, as described by Ahmadi, Grotzinger and Johnson (1988). In Section 2 we focus on the reel positioning problem with one group of components. Extensions to multiple groups are discussed in Section 3. Computational results illustrating the applications of the procedures developed in this paper are presented in Section 4. We conclude with a summary in Section 5.

1. DESCRIPTION OF THE PROCESS

Figure 1 illustrates a Dynapert MPS 500, a computer numerically controlled dual delivery pick-and-place machine at the IBM plant in Austin. The key components of the machine include the work board, the placement arm, pick-and-place heads, and the component delivery carriers. These are capable of independent movements, as shown in Figure 1. Exploitation of these concurrent movements requires coordination of the movements. See Figure 2, reproduced from Ahmadi, Grotzinger and Johnson (1988), for a schematic description of the main features of the pick-and-place operation. For further details of the process the reader is referred to that paper and to Grotzinger (1988). The pick-and-place heads are mounted on the two ends of a fixed length arm which can move between two fixed position in the y-direction only. The heads are

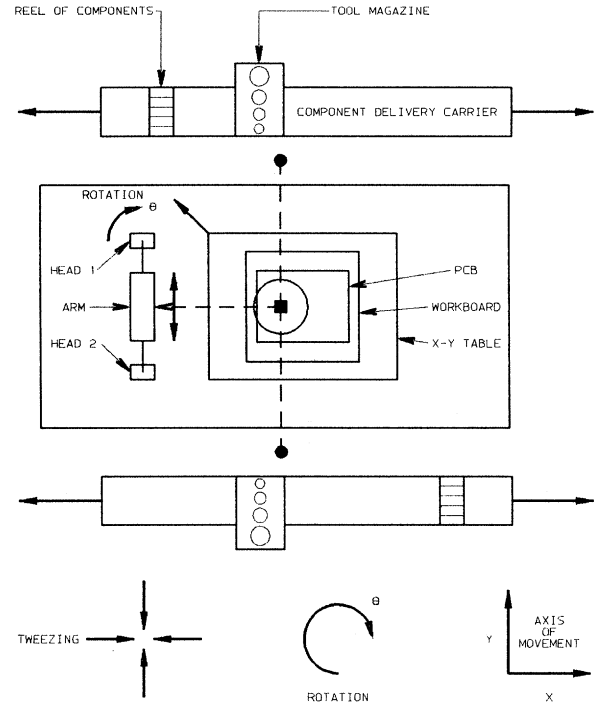

Figure 2. Main features of the pick-and-place machine.

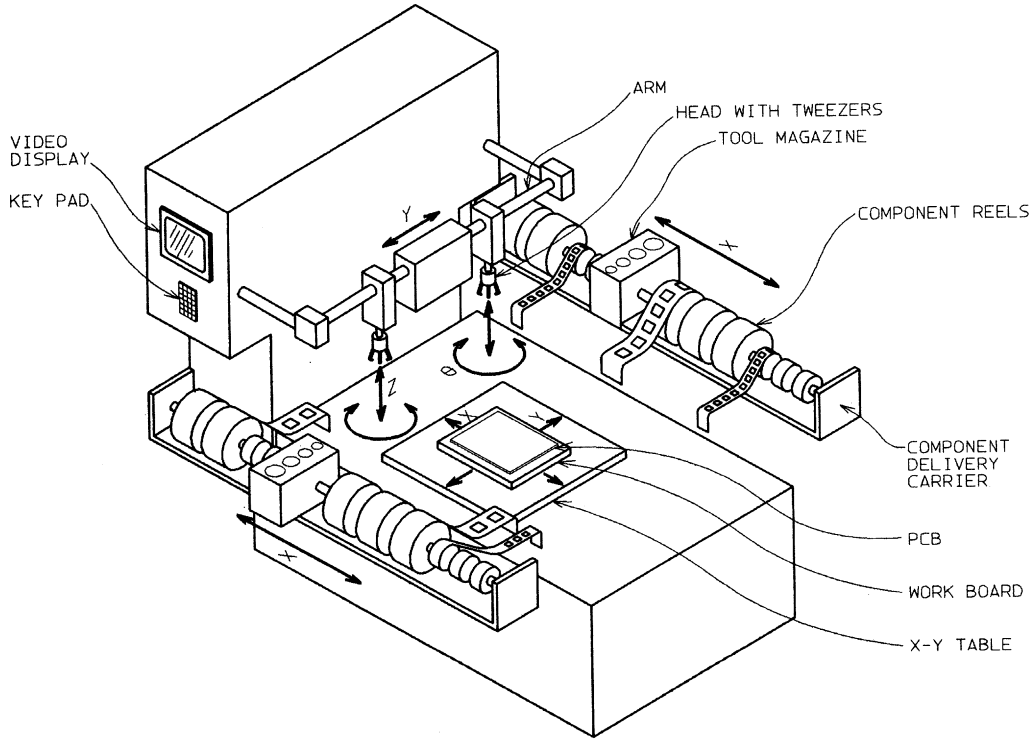

Figure 1. Dual delivery pick-and-place machine (Dynapert MPS 500).

capable of making a vacuum pickup, rotating, "tweezing" and making a reverse vacuum placement. (Tweezing is an operation to center and position a component after it is picked up by the vacuum nozzle.) The two component delivery carriers move in the x-direction only. Each carrier has a tool magazine that can hold four pickup-and-placement nozzles (tools) and sixty slots for accommodating reels of components. Component pickup by the head requires that the carrier should move and position the slot with the required components in the fixed pick position. The workboard can move in both the x and y directions and should be aligned under the head to perform the placement operation. Typically, a pick-and-place operation requires these steps:

i. movement of the arm to place the head in the pick position;
ii. movement of the carrier to position the component reel in the pick position;
iii. pick operation by the head, rotation and tweezing of the component;
iv. movement of the arm to locate the head in the place position;
v. movement of the board to align for placement operation; and
vi. placement operation.

Steps i–vi are repeated for each pick-and-place operation alternating between the two sides. Our description assumes that no nozzle change is required. Nozzle changes are usually necessary when components belong to different groups. In that case, the tool magazine needs to be in the pick position to facilitate nozzle changes. Nozzle changes represent unproductive movements and tend to reduce the production rate.

To realize the full benefits of the machine, it is necessary to synchronize the concurrent operations of the two carriers and the head such that, while one head is performing the pick operation, the other is engaged in the placement operation. In this scheme, the production rate is maximized by minimizing the cycle time. Here, cycle time is the time between successive pick operations. In an ideal situation, cycle time is the sum of two elements: the time required for the arm head to move between the two fixed positions (movements i and iv above), and the fixed time required for vertical motion of the heads (movements iii and vi). The "ultimate" burst rate specified by the vendor of the dual delivery pick-and-place machine is based on this idealized situation. However, in reality, the concurrent movements (those of carriers ii and board v) are not always completed simultaneously and result in significant delays which substantially increase the cycle time. Furthermore, all movements of the board and carrier are frozen during the pick-and-place operations (movements iii and vi). Thus, the cycle time is determined by the maximum of the arm, the board, and the carrier movements.

Ahmadi, Grotzinger and Johnson (1988) identify a hierarchical framework consisting of three optimization problems to improve system productivity and to increase the production rate, as follows.

1a. **The Component Allocation Problem**: to determine the number of feeders of each component type assigned to a machine.
1b. **The Partitioning Problem**: to assign each component to one of the two carriers along with the corresponding sets of nozzles.
2. **The Placement Sequencing Problem**: to specify the sequence of component placements, taking into consideration 1b.
3. **Reel Positioning Problem** (**RPP**): to assign reels of components to slots on the carriers. We assume that the components have been assigned to the carriers and that the placement sequence is given. The objective now is to determine the reel positions that maximize the system production rate.

In this scheme the carrier will become a bottleneck if the time required for its movement is larger than the maximum of the arm and board movements. Note that the arm movement essentially remains the same for each pick-and-place operation and can be considered constant. For small boards (typically 1″ × 4″) in which the arm movement dominates the board movement the window of time available for carrier movement may be considered constant. (In our application, over 60% of the boards were small.) On the other hand, with large boards (typically 10″ × 14″), the corresponding time window is a variable and depends on the sequence of pick-and-place operations. However, in practice, estimating these times is a cumbersome process and, in most cases, the time for the board movement is also approximated as a constant. Hence, we assume that the time for maximum of the arm and the board movements is constant and is denoted by α. For ease of exposition, we scale and express α in units of slot width, i.e., it represents the number of slots that could be covered by the carrier during the arm and board movements. (Likewise, we express reel widths in units of slot widths.) The objective is to position the reels such that each carrier movement can be achieved in a time not greater than α. In case such an assignment cannot be realized, the objective is to minimize the net movement in excess of α. Clearly, the assumption regarding α is an approximation. However, most of the proposed procedures (with one exception) can be modified to accommodate variability in this parameter. (These aspects are discussed in detail in later sections.)

Typical values of α range from 1 to 5 in our application. More specifically, in one instance (defined by the types of PCBs) the variation in α may be described in the following manner:

α	1	2	3	4	5
% of cases	40	25	25	5	5

These values are consistent with those observed by Grotzinger, who encountered α values up to 6.

Drezner and Nof (1984), Ball and Magazine (1988), Leipala and Nevalainen (1989), and Hu and Carter (1989) are among several researchers who have examined component insertion and sequencing problems encountered in electronic assembly operations. For example, Ball and Magazine describe three optimization problems for improving the productivity in the manufacture of PCB assemblies. These are similar to, but different from, problems considered in this paper. While Ball and Magazine have studied systems with conventional pin-and-hole technology, we focus on the surface mounted technology with concurrent operations. Moreover, while these authors are concerned with the determination of the sequence of insertions, we assume that the pick-and-place sequence has been specified, and focus on the positioning of reels on the carrier. The problem considered in Grotzinger is motivated by the IBM system and is similar to the one described in this paper. He formulated the problem as a mixed integer program with the objective of minimizing "the maximum number of carrier slots transversed over the set of all required carrier moves." He solved the problem with 30 reels and total reel widths of 51 slots, and concluded that "this strongly suggests that the explicit incorporation of excess delay into the objective function should yield nearly delay-free positioning of the reels." The problem was solved using the standard software package MPSX/MIP 370. One motivating factor for our work comes from the inability of this package to handle larger problems with different objective functions. In addition, we consider a more general version of the problem that has potential applications to other systems with concurrent operations.

Also note the similarity between our problem and some machine layout problems considered in Chan and Francis (1979) and Heragu and Kusiak (1988). The problems are alike because the locations of the machines in the layout problem and positions of the reels in **RPP** determine the required movements. However, the nature of concurrent operations introduces additional complexity. For the special case when $\alpha = 0$, the **RPP** reduces to the layout problem described and solved in Chan and Francis. However, as we discuss later, we focus on the problems where $\alpha \geq 2$, although the special case $\alpha = 0$ is also easy to handle by the properties presented in the paper. The reel positioning problem can also be viewed as a special case of the general one-dimensional space allocation problem addressed by Picard and Queyranne (1981), Wong (1983), Romero and Sanchez (1990), and Kouvelis, Chiang and Yu (1992). Picard and Queyranne establish the computational complexity of several special cases of the one-dimensional space allocation problem, and develop an exact algorithm using dynamic programming for the general problem. They conjecture that this latter problem is NP complete. Our Proposition 1, which establishes NP completeness of **RPP**, provides additional evidence in this regard. The **RPP** is similar to the "root-in first and return to root constraint" problem of Picard and Queyranne with the root position analogous to the tool position in the **RPP**.

2. REEL POSITIONING PROBLEM WITH ONE GROUP OF COMPONENTS

In this section, we describe the **RPP** with one group of components and examine solution procedures. First, we present an algebraic formulation of the problem (Section 2) and discuss its computational complexity (subsection 2.1). Since the problem is NP-complete, our focus is on the development of heuristics. In subsection 2.2 we discuss the implications of restricting the set of feasible solutions to oriented solutions (defined later). An algorithm to obtain an optimal oriented solution to **RPP** is presented in subsection 2.3.

In the **RPP** we assume that the positioning of reels on each of the two carriers is independent and that the pick-and-place sequence is specified so that reel positions determine the carrier movement. Without loss of generality, we also assume that the reels are numbered in accordance with this sequence. With one group of components, tool and nozzle changes are unnecessary. Consequently, only two carrier movements involving the tool magazine are required. The first occurs at the start of the pick-and-place sequence. The operation terminates with the final move to the tool magazine to prepare for the nozzle change for the next group of components. (We consider the multigroup problem in Section 3.) To simplify our presentation, in the remainder of this section we assume that the tool magazine is at position 0, i.e., at the end of the tool carrier. (The implications of these assumptions are discussed in Section 5.)

Our formulation does not recognize explicitly the presence of duplicate components. While this is a limitation of our solution approach for **RPP**, it was not a serious problem in our application. First, when a board requires several units of the same component, they are inserted consecutively. Second, multiple feeder reels of the same component are placed adjacent to each other. As a result, the carrier remains in one position during the insertion of the duplicate components. This scheme is motivated by the desire to minimize the carrier movement.

Our objective is to determine the reel position that minimizes the total carrier movement in excess of α for each pick-and-place operation. We use the following notation.

n: the number of reels to be positioned;

t_i: the width of reel i, $i = 1, 2, \ldots, n$; $t_0 = t_{n+1} = 0$ (as mentioned earlier, t_i is in units of slot width);

R_i: the starting position of reel i; reel i will occupy slots $R_i, R_i + 1, \ldots R_i + t_i - 1$ (the tool magazine is assumed to be in position 0, i.e., $R_0 = R_{n+1} = 0$);

α: the time required to complete the concurrent operations;
s: the carrier speed;
y_{ij}: an indicator variable that takes the value 1 if reel i precedes j and 0, otherwise.

Component pickup by the head requires that the center of the reel be aligned in the pick position. Hence, the time for carrier movement between operations i and $i + 1$ is given by

$$\frac{1}{s}\left|R_i + \frac{t_i}{2} - R_{i+1} - \frac{t_{i+1}}{2}\right|.$$

Note that the average carrier speed, s is an approximation. This is reasonable as long as the acceleration and deceleration effects are not significant. Hence, we assume, without any loss of generality, that s is one. The cycle time for operation i is thus given as

$$\max\left\{\alpha, \left|R_i + \frac{t_i}{2} - R_{i+1} - \frac{t_{i+1}}{2}\right|\right\}.$$

The problem is to find a sequence of reels that minimizes the sum of the cycle times for completing the series of pick-and-place operations. The reel positioning problem is formulated as follows.

RPP

$$Z = \text{Min} \sum_{i=0}^{n} \max\left\{\alpha, \left|R_i + \frac{t_i}{2} - R_{i+1} - \frac{t_{i+1}}{2}\right|\right\}$$

subject to

$$R_i + t_i \leq R_j + Ly_{ji} \qquad \text{for all } i, j, i \neq j$$
$$y_{ij} + y_{ji} = 1 \qquad \text{for all } i, j, i \neq j$$
$$R_0 = R_{n+1} = 0$$
$$R_i \in \left\{0, 1, \ldots \sum_{j=1}^{n} t_j\right\}, \qquad \text{for all } i$$
$$y_{ij} \in \{0, 1\} \qquad \text{for all } i, j, i \neq j,$$

where L is a sufficiently large number.

2.1. Computational Complexity of RPP

The integer programming formulation of **RPP** suggests that it is difficult to solve optimally. Proposition 1 establishes that the **RPP** is binary NP-complete.

Proposition 1. *The* **RPP** *is binary NP-complete.*

Proof. See Appendix A.

2.2. Oriented Solutions to the RPP

The result above suggests it is unlikely one can develop a polynomially bounded algorithm to find an optimal solution. Hence, our objective is to present efficient heuristics that provide "good" solutions to **RPP**. We focus on "oriented" solutions to the **RPP**. An oriented solution to **RPP** is a feasible solution to the problem in which all the required carrier movements are accomplished by changing the direction of movement exactly once. All feasible solutions that are not oriented involve some back tracking and require at least three changes of direction of movement. We refer to such solutions as *nonoriented*. Example 1 illustrates oriented and nonoriented solutions.

The choice of oriented solutions to the **RPP** is motivated by engineering issues. These included concerns for minimizing the changes in the direction of carrier movement and for minimizing the total carrier movement. CNC machines are precision equipment and limiting the mechanical movement will improve their reliability. Hence, reducing the absolute movement of the carrier is a desirable characteristic. Excess movement implies avoidable wear and tear that could lead to misalignment of the feeder location. Ease of implementation was another important characteristic of the proposed method.

Oriented solutions have other desirable features as well. In deriving these results, we make use of the fact that there exists an optimal solution to the **RPP** in which there are no inserted empty slots. Thus, it is sufficient to consider solutions in which reels are positioned adjacently. Hence, specifying the sequence in which the reels are placed on the carrier is equivalent to determining the slot positions assigned to the reels. Let $[i]$ denote the ith reel when they are arranged in order of increasing reel widths, i.e., $t_{[1]} \leq t_{[2]} \leq \cdots \leq t_{[n]}$. The following properties illustrate the optimality of oriented solutions for specific cases.

Property 1. Consider the **RPP** in which the reel widths are all equal, i.e., $t_i = t$, $i = 1, 2, \ldots n$. Then there exists an oriented solution that is optimal to **RPP**.

Property 2. If $\alpha \leq (t_{[1]} + t_{[2]})/2$, then there exists an oriented solution that is optimal to **RPP**.

Property 3. If $\alpha \geq (t_{[n]} + t_{[n-1]})$, then there exists an oriented solution that is optimal to **RPP**. Properties 1 and 3 can be established by considering an oriented solution in which reels along the forward and backward movements alternate. The corresponding sequence is $(1, n, 2, n - 1, \ldots)$. That is, place reel 1 in position 1, reel n in position 2, reel 2 in position 3, etc. It is easy to see that this provides an optimal solution. Property 2 follows directly from these observations:

i. The absolute movement, excluding the moves to and/or from the tool magazine, is at least α.
ii. The total absolute carrier movement is minimized by any oriented solution with the largest reel ($[n]$) positioned last.

While oriented solutions have several attractive properties, restricting our attention to this class of solutions does not guarantee optimality to **RPP**. Example 1 demonstrates that an optimal solution to **RPP** can be a nonoriented solution.

Example 1. Consider an instance of **RPP** with the following parameters: $n = 6$, $t_1 = 4$, $t_2 = 4$, $t_3 = 2$, $t_4 = 12$, $t_5 = 1$, $t_6 = 10$, and $\alpha = 12$. The best oriented solution to this problem yields an optimal value of 84.5. However, this solution is not optimal to **RPP**. The optimal value of 84 is realized by a nonoriented solution. Figure 3

illustrates the best oriented solution and the optimal solution to the problem.

We next present a procedure for solving **RPP** in which the solutions are restricted to be oriented solutions (**RPP1**). An optimal solution to **RPP1** is referred to as an optimal oriented solution (OOS) and the corresponding objective function value is denoted by $Z1$.

Error Bounds on the Solution OOS as an Approximation to RPP

A trivial lower bound on Z, the objective value for **RPP**, is obtained as $(n + 1)\alpha$. Hence, an *a posteriori* error bound on the **OOS** solution is given by $(Z1 - Z) \leq Z1 - (n + 1)\alpha$. Proposition 2 provides an a priori error bound on the solution **OOS**. The proof of this proposition can be found in Appendix A.

Proposition 2. $Z1/Z \leq 1.5$.

2.3. An Algorithm for Solving RPP1

In this section, our focus is on the sequence in which the reels are placed, rather than the slots assigned to the reels. In what follows, "position" of reel i refers to its position in this sequence. For example, position 1 implies that the reel is next to the tool magazine ($R_0 = 0$), while position n implies that the reel is placed farthest from the tool magazine. In this development we rely on the fact that an optimal solution to **RPP1** can be obtained as a shortest path in a related layered network. The procedure (illustrated later in Example 2) is as follows.

Algorithm for RPP1

STEP 1. Construct a layered network $G = (N, E)$ with the following characteristics:

Layers: The number of layers in the network is $n + 2$. Layers 0 and $n + 1$ include only one node each that corresponds respectively to the source node S, and sink node T. Here, the source node represents the tool magazine in position 0. Layers 1 through n correspond to reels 1 through n.

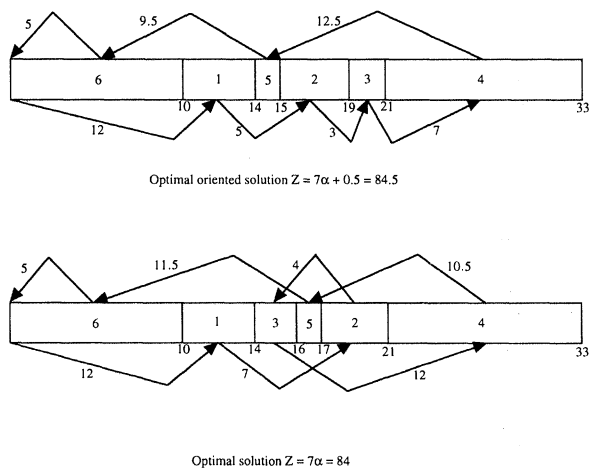

Figure 3. An optimal oriented solution and an optimal solution (nonoriented) to example 1 with $\alpha = 12$.

Nodes (N): The nodes other than the source and the sink node in the network are identified by a triplet (i, j, k) where i denotes the reel number or, equivalently, the layer number, j the position of the reel in the sequence, and k the position of the first reel encountered in the backward movement after i had been crossed. For an illustration of this notation, consider the following partial sequence in a ten reel example: 1, 2, 10, 3, 9, 4, ... The node corresponding to reel 3 would be denoted by (3, 4, 3), indicating that reel 3 is in position 4 and position 3 contains the first reel (reel 10 in this case) encountered in the backward movement after reel 3 is crossed. In a similar manner the node for reel 4 in the sequence is denoted by (4, 6, 5). Note that for a given (i, j, k), the reel occupying position k can be determined uniquely and it is reel $n + i - j + 1$. It can be verified that the set of nodes

$$N = \{S\} \cup \{T\}$$
$$\cup \{(i, j, k) | i = 1, 2, \ldots n, j = i, k = 0\}$$
$$\cup \{(i, j, k) | i = 1, 2, \ldots, n - 1, \quad j = i + 1 \ldots n, k = j - i, \ldots j - 1\}.$$

Directed Arcs (E): Each directed arc in the network connects a pair of triplets from adjacent layers, and represents part of a feasible assignment of reel positions. The following directed arcs are incident from layer $i - 1$ to (i, j, k) for $i = 2, 3, \ldots n$: for $j > k + 1$, $\{(i - 1, j - 1, k) \to (i, j, k)\}$, and for $j = k + 1$, $\{(i - 1, j', k') \to (i, j, k) | j' = i - 1, \ldots, j - 2$ and any k' such that $(i - 1, j', k')$ is feasible$\}$. Node S is connected to all nodes in layer 1. Node T is connected to all nodes with $j = n$.

Costs: Let $C(x)$ denote the cost of arc x in E. Then, the cost of each arc can be written as:

$$C((i, n, k) \to T) = 0 \tag{1}$$

$$C((i - 1, j - 1, k) \to (i, j, k)) = \max(\alpha, (t_{i-1} + t_i)/2) + \max\left(\alpha, (t_i + t_{i+1})/2 + \sum_{\ell=i-j+k+1}^{i-1} t_\ell\right)\delta(j - n)$$

$$\text{for } j > k + 1 \text{ and } i = 2, 3, \ldots n; (i, j, k) - (1, 1, 0) \tag{2}$$

$$C((i - 1, j', k') \to (i, j, j - 1)) = \max\left(\alpha, (t_{i-1} + t_i)/2\right) + \sum_{\ell=n+i-j+1}^{n+1-j'-1} t_\ell + \max(\alpha, (t_i + t_{i+1})/2)\delta(j - n) + \sum_{\ell=n+i-j+1}^{n+i-j'-2} \max(\alpha, (t_\ell + t_{\ell+1})/2) + \max\left(\alpha, (t_{n+i-j'-1} + t_{n+1-j'})/2 + \sum_{\ell=i-j'+k'}^{i-1} t_\ell\right)$$

$$\text{for } i = 1, 2, \ldots, n; (i, j, k) \neq (1, 1, 0), \tag{3}$$

where $t_0 = t_{n+1} = 0$, $\delta(j - n) = 1$ if $j = n$, $\delta(j - n) = 0$ if $j \neq n$, $S \equiv (0, 0, 0)$ and $\sum_{\ell=a}^{b} y_\ell = 0$ for $b < a$.

Equation (1) states that the costs of all arcs incident to node T are zero. For $j > k + 1$, reels $i - 1$ and i are adjacent. The first term of (2) represents the forward movement. When $j = n$, the second term is added to account for the backward movement from reel i to reel $i + 1$. In between the two reels, there are $j - k - 1$ reels in the forward sequence (i.e., reels $i - 1 - (j - k - 2)$ through $i - 1$). For $k = j - 1$, there are $j - j' - 1$ reels in the backward sequence (i.e., reels $n + i - j + 1$ through $n + i - j' - 1$) in between reels $i - 1$ and i. Therefore, the first term of (3) represents the forward movement from reel $i - 1$ to reel i. For $j = n$, the second term is added to represent the backward movement from reel i to reel $i + 1$, which are adjacent reels. The third term represents the backward movement from reel $n + i - j + 1$ to reel $n + i - j' - 1$. There are $j' - k'$ reels in the forward sequence (i.e., reels $i - j' + k'$ through $i - 1$) in between reel $n + i - j' - 1$ and reel $n + (i - 1) - j' + 1$ (i.e., the reel occupying the kth position). Hence, the fourth term represents the corresponding backward movement from reel $n + i - j' - 1$ to reel $n + (i - 1) - j' + 1$. Note that we use $i - 1 = j' = k' = 0$ in (3) for the arcs incident from S. In this representation a path from S to T specifies a sequence of reels and the path length corresponds to the completion time of all operations.

STEP 2. The shortest path between the source and sink node determines the optimal oriented solution and solves **RPP1**.

The number of nodes in the network described above is $0(n^3)$. Since $0(n)$ arcs are incident to each node, and since the calculation of each cost coefficient requires $0(n)$ computations, the forward dynamic programming algorithm on the network requires $0(n^5)$ computations to determine the shortest path from S to T. This procedure to determine **OOS** does not require α to be constant. Thus, the cost/item in Step 1 may be suitably modified for cases in which α is variable and depends on the sequence of pick-and-place operations.

Example 2. Consider the following 4-reel problem: $t_1 = 2$, $t_2 = 4$, $t_3 = 3$, $t_4 = 6$, and $\alpha = 4$. The layered network to determine **OOS**, the corresponding sequence, and the objective function value are shown in Figure 4. The optimal sequence is given by 4, 1, 3, 2 with a total movement of 27.5.

3. THE REEL POSITIONING PROBLEM WITH MULTIPLE REEL GROUPS [MRPP]

In this section we consider extensions of the **RPP** when the components belong to multiple reel groups. A change of tools is required for the pick-and-place operation when the components belong to different groups. Such tool changes require a considerable amount of time, which reduces the system productivity significantly. Thus, in practice, components belonging to a reel group are inserted in sequence. This reduces the number of tool changes and minimizes unproductive tool change time. We assume that the pick-and-place sequence is such that all components within a group are inserted consecutively. For a further discussion of this problem and a justification of this assumption, see Ahmadi, Grotzinger and Johnson (1988). Note that since a tool change is required following the insertion of components in a group, alignment of the tool magazine in the pick position is necessary at the start and at the termination of component insertions in each group. Thus, a reel group may be defined as a maximal subset of reels which can all be placed without having to change any tools. In addition, we assume that the sequence in which the component groups are inserted is also specified. Our objective is to determine the reel positions for all components to minimize the cycle time and thereby increase the system productivity.

The problem formulation **RPP** presented in Section 2 is easy to extend to the multiple reel group case. For each reel group, we introduce two dummy components with reel length zero and positioned at location 0 ($R = 0$). The two dummy reels correspond to the tool magazine and represent the first and last movements in the insertion sequence. The multiple reel group problem can now be formulated as an instance of **RPP** with $(n + 2m)$ components, where n is the total number of reels and m is the number of tool groups. This modification will assure tool changes following a switch in a component reel group. These comments indicate that the multiple tool group problem (referred to as **MRPP** in the remainder of this section) is at least as difficult as the single

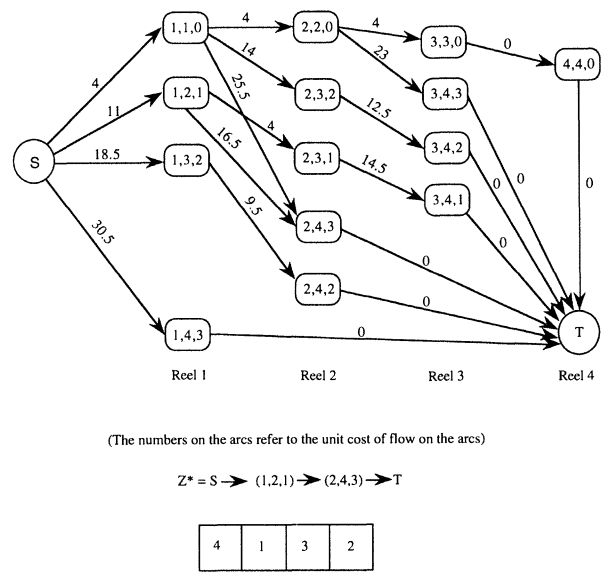

Figure 4. The network to solve **RPP1** for example 2.

group problem. From the results of Proposition 1 it follows that **MRPP** is also NP-complete. Hence, our objective in this section is to examine heuristics that are likely to provide good solutions to **RPP**.

In the development of heuristics to address the **MRPP** positioning problem, we focus on "nested" solutions. In nested solutions, components within a reel group are placed consecutively on the tool carrier. We also assume that the positions of the components within a group are determined in accordance with the oriented solutions described in the previous section. Such solutions have the attractive property that they tend to minimize the total absolute movement of the carrier. As mentioned earlier, this is a desirable feature from an engineering viewpoint. We refer to this restricted version of the problem as **MRPP1**.

3.1. An Algorithm to Solve MRPP1

In this section we present a pseudopolynomial algorithm to solve **MRPP1**. This procedure is based on the functional equation presented in Lawler and Moore (1969) to address a variety of single machine scheduling problems. Before describing the application of the procedure to **MRPP1** we briefly describe the problem considered by Lawler and Moore. They consider an n-job scheduling problem in which each job can be processed in either of two modes. The processing time for job j is a_j units if it is performed in mode 1 and b_j units if it is assigned to mode 2. A loss of $\lambda_j(t)$ is incurred if job j is completed in mode 1 at time t. The corresponding loss function for completing in mode 2 is $\beta j(t)$. The objective is to determine the assignment of jobs to each mode and the sequence of jobs to minimize the total penalty. The authors present an efficient dynamic programming formulation of the problem that requires nT computational steps, where n is the number of jobs and T the maximum time required to complete all the jobs. Lawler and Moore show that a number of single and multiple machine scheduling problems can be solved efficiently using this procedure. The common thread in all these applications is that jobs can be partitioned into two classes. The jobs in one class will be performed in a predetermined order, while the jobs in the other class may be processed in arbitrary order either preceding or following the jobs in the first class. The "modes" correspond to the two classes and the assignment of jobs to the two classes is determined by the dynamic programming procedure.

The multiple reel positioning problem with nested solutions is similar to the single machine weighted tardiness problem with common deadline (WTP) presented in Lawler and Moore. Note that in any feasible solution to **MRPP1**, the component groups can be partitioned into three sets:

i. Set A comprises reel groups positioned completely within α units from the tool magazine;
ii. Set B consists of reel groups positioned after α units from the tool magazine; and
iii. Set C consists of a reel group whose starting position is not greater than α and whose ending position is after α.

The application of the dynamic programming formulation of Lawler and Moore to **MRPP1** relies on these characteristics of the problem:

i. In an optimal solution to **MRPP1**, reel groups in set A may be placed in any arbitrary sequence.
ii. In general, the optimal sequence of reels in set C depends on α.
iii. In an optimal solution to **MRPP1**, reel groups in set B are sequenced in order of increasing width of the reels in each group.

Property i is straightforward. Properties ii and iii are a direct consequence of the following observation. Consider two alternative starting positions (say, a1 and a2) for a reel group, with $a1$, $a2 > \alpha$. For any given sequence of component positions, the total costs of carrier movement for the reel group in the two cases differ by a constant $(2(a2 - a1))$ which is independent of the reel lengths and depends only on the starting positions.

There are a number of similarities between the above properties and those of the weighted tardiness problem. For example, each group of components is similar to a job and the common deadline is analogous to α in **MRPP1**. Similar to i, jobs completed before the deadline are sequenced in order of increasing ratio of processing time to penalty parameter. We note that given the reel group in set C, the problem **MRPP1** reduces to partitioning the remaining reel groups into sets A and B. This can be determined using the functional equation presented in Lawler and Moore. Thus, **MRPP1** can be solved by enumerating all the alternatives for the reel group in set C. Since each alternative requires mα steps, the overall computation requires an order of $m^2\alpha$ number of steps. The details of the dynamic programming formulation and the computational procedure are presented in Appendix B.

The nested solution to **MRPP1** is not necessarily optimal for **MRPP** and it is easy to construct examples to illustrate this drawback of nested solutions. In an extended version of this paper (see Ahmadi et al. 1992) we present "meshing heuristics" to improve the nested solution. The quality of improvements obtained by these heuristics are illustrated by the computational experiments described in the next section.

4. COMPUTATIONAL RESULTS

In this section we present results of computational experiments that illustrate the application of the procedures described in this paper. The primary objective is to examine the quality of the approximate solutions provided by the heuristics. A second objective is to demonstrate

that the procedures require modest computational effort and are suitable for addressing realistic problems. Since an exhaustive examination of all the factors requires a large number of problems, we tested the algorithms on a set of randomly generated examples. We now describe the experiments in detail.

4.1. Experiments With One Reel Group

In this experiment, our objective was to evaluate the quality of the optimum oriented solution (**OOS**). We used a branch-and-bound procedure to explore improvements to the **OOS**. In this procedure, we modified Balas' zero-one algorithm (Balas 1965, 1967) and incorporated the lower bound of $(n + 1)\alpha$. The **OOS** was used as the initial solution for the branch-and-bound algorithm. Because of the computational time and storage limitations of the implicit enumeration procedure, the test problems in this experiment were limited to 30 reels. In the design of this experiment, we considered the following factors:

i. Number of reels: Five levels, with 10, 15, 20, 25 and 30 reels.
ii. Reel widths (t_i) were chosen from the discrete uniform distribution over the intervals [1, 10] to ensure a wide range of problem instances.
iii. Alpha (α): For each case specified by the number of reels and the corresponding reel lengths, we considered four levels of α: The four values of α were obtained in a two-step procedure. In the first step a parameter was calculated as $0.5t$, $1.0t$, $1.5t$ and $2.0t$, where $t = (\max_i t_i - \min_i t_i)/2$. Next, this parameter was rounded off to the nearest integer to yield a value for α. When the resulting value was either less than 2 (or greater than 8) it was adjusted to 2 (to 8). (Note that the problem is easy to solve for cases with $\alpha = 1$ using Property 2, while large values of α are considered in the second experiment. Also, by Property 3, **OOS** is optimal for large values of α.)

The value of α depends on many factors, such as precision and accuracy desired, reliability, and speed setting. Typically, to assure greater precision and reliability, slower operations are required resulting in larger values of α. In our application, the range of values for this parameter was between 1 and 5. Similarly, the reel widths were typically between 1 and 5. These parameter ranges give rise to instances for which the special cases described in Section 2 do not apply, and thus the heuristics developed in the paper are useful. For each set of parameters specified by the number of reels, reel lengths, and α-value, 20 examples were generated, giving a total of 400 problems.

In these problems, in addition to deriving the **OSS** and an optimal solution, we computed two benchmark solutions referred to as **BM1** and **BM2**. The objective in comparing the **OOS** with **BM1** and **BM2** was to examine the effect of α on the reel positions, and to provide a measure of benefits that may be derived from the procedures of this paper. Hence, the solutions **BM1** and **BM2** were based on procedures that ignored the parameter α. **BM1** is the solution obtained by a naive heuristic in which an oriented solution is constructed by positioning the widest reel last. The remainder of the sequence is developed by determining, in order, the reels in positions $n - 1$, $n - 2$, ... 1. At each stage, among the two alternatives that satisfy the oriented solution property, we chose to position the wider reel. Ties were broken in an arbitrary manner. **BM2** is obtained as the solution by applying the algorithm of Section 2 with the α value set at 0. **BM1** and **BM2** are optimal when α is zero and thus they may be regarded as alternative solutions in which this parameter is ignored.

The computations were carried out on an HP Vectra RS/16, coded in Turbo Pascal. Summary results of the experiment presented in Table I demonstrate the excellent quality of the **OOS**. Out of 400 problems, **OOS** was optimal in 391 cases. The average error of the **OOS** relative to the optimal solution in the other 9 cases was 2.35%. The average relative error of the **OOS** over 400 problems was 0.053%. The maximum error was 3.2%. The table also provides some measures that compare the relative performance of **OOS**, **BM1**, and **BM2**. Over the 400 test problems, the average deviation of **BM1** (**BM2**) from optimal was 12.5% (11.8%). These results also indicate that the performance of **BM1** and **BM2** deteriorate with increasing values of α. Compared to the heuristics which ignore the parameter α, the procedures of Section 2 provide significant improvement (about 10%) in the cycle times. Since cycle time is directly related to burst rate and productivity, the methods of our paper have the potential to significantly increase the system capacity. Our experimental results also demonstrate the modest computational requirements of the heuristics. For example, for problems with $n = 30$, in contrast to the average time of over 30 minutes required for obtaining optimal solutions, the heuristics required only 26 seconds. In this context, it is interesting to note that the **OOS** provides a good initial solution for our branch-and-bound procedure. In a limited number of test problems with $n = 30$, in some cases we were unable to obtain optimum solutions after 3–5 hours of CPU time using standard software packages to solve the corresponding MIP problems.

4.2. Multiple Reel Groups

The second experiment relates to tests with components belonging to different reel groups. Recall that the heuristic for this case involves two stages. In the first stage an optimum nested solution (**NS**) is determined using the Lawler and Moore procedure described in subsection 3.1. In the second stage improvements are obtained by the "meshing heuristic" (described in Ahmadi et al.). The improved solution is referred to as **MP**. In this experiment, we generated 2,240 problems with the number of reel groups ranging

Table I
Summary of Experiments With One Tool Group

Number of Reels, n	$\alpha(1)$			$\alpha(2)$			$\alpha(3)$			$\alpha(4)$		
	OOS	**BM1**	**BM2**	**OOS**	**BM1**	**BM2**	**OOS**	**BM1**	**BM2**	**OOS**	**BM1**	**BM2**
10	19 [1]	1.07	1.08	20	1.12	1.09	19 [2]	1.12	1.14	20	1.14	1.12
15	20	1.12	1.10	20	1.08	1.13	20	1.13	1.09	18 [3,4]	1.16	1.14
20	20	1.13	1.09	20	1.11	1.07	19 [5]	1.12	1.08	20	1.17	1.18
25	19 [6]	1.08	1.12	19 [7]	1.16	1.14	20	1.12	1.16	20	1.14	1.13
30	20	1.14	1.08	19 [8]	1.08	1.09	20	1.16	1.17	19 [9]	1.15	1.16

OOS: The figures under column **OOS** denote the numbers of instances (out of 20) in which optimum oriented solution solved the corresponding **RPP**.

BM1 (**BM2**): **BM1** (**BM2**) denote the average % error of the solution obtained by procedure **BM1** (**BM2**) described in subsection 5.1.

These footnotes provide the % error of the **OOS** in the 9 instances in which it was not optimal: 1–2.5%, 2–3.2%, 3–1.1%, 4–2.4%, 5–1.8%, 6–2.6%, 7–3.1%, 8–1.9%, and 9–2.6%.

between 2 and 10. The design of the experiment was similar to that of subsection 3.1, except for one important difference. The number of reels in this experiment varied between 10 and 50, in contrast to a maximum of 30 reels in the previous case. For small problems (up to 30 reels), we obtained optimum solutions using a branch-and-bound procedure. For larger problems with more than 30 reels, we were unable to derive optimal solutions and compare the improvement of **MP** over **NS**.

The summary results in Tables II and III suggest that the heuristics perform very well. Table II presents the number of instances in which the heuristic solutions were optimal. It can be seen from these results that the nested solution is optimal in 80.2% (963 out of 1,200) of the cases. The meshing heuristic significantly improves the quality of the solution and **MP** is optimal in 91.1% (1,093 out of 1,200) of the cases. These results suggest that the heuristic is reasonable. Table III presents the improvement obtained by the meshing heuristic. It also indicates, for small problems, the errors of **NS** and **MP** relative to the optimum value. For problems with the number of reels not greater than 30, the average error of **NS** is 4.13% relative to the optimal solution. The meshing procedure improves the nested solution by 1.86% on the average. It is interesting to note that the improvements due to the meshing heuristic do not deteriorate with the problem size.

5. CONCLUSIONS

This paper describes an optimization problem that has significant implications for the efficient management of concurrent operations in the manufacture of printed circuit boards using surface mounted technology. The problem represents one of a series of optimization problems that needs to be addressed to realize high burst rates. The problem is NP-hard and, hence, intractable for designing efficient algorithms to obtain optimal solutions. The proposed heuristics take into account engineering concerns for minimizing the carrier movement. Our results indicate that the computational effort required is modest and the quality of the solution is reasonable. Compared with other methods that ignore the concurrent

Table II
Summary of Experiments With Multiple Tool Groups

Number of Reels, n	Number of Tool Groups, m	$\alpha(1)$		$\alpha(2)$		$\alpha(3)$		$\alpha(4)$	
		NS	**MP**	**NS**	**MP**	**NS**	**MP**	**NS**	**MP**
10	2	17	19	17	20	18	20	17	19
15	2	19	20	18	19	15	17	16	18
15	3	17	19	18	19	16	18	13	16
20	2	19	20	17	19	16	18	14	17
20	3	16	18	16	19	18	18	14	18
20	4	15	17	16	17	15	16	14	17
25	2	16	19	18	20	14	17	16	18
25	3	19	20	15	17	16	18	16	18
25	4	15	17	16	17	15	16	14	17
25	5	15	17	16	18	15	19	15	17
30	2	18	20	20	20	16	18	16	18
30	3	17	19	16	19	17	19	16	18
30	4	17	19	15	17	14	17	15	17
30	5	16	18	14	18	16	17	13	17
30	6	15	17	16	18	16	18	14	17

NS: **NS** refers to number of instances (out of 20) in which the nested solution was optimal for the corresponding **MRPP**.

MP: **MP** refers to the number of instances (out of 20) in which the solution provided by the meshing heuristic was optimal for the corresponding **MRPP**. The initial solution for meshing was the solution **NS**.

Table III
Summary of Experiments With Multiple Tool Groups

Number of Reels, n	Number of Tool Groups, m	$\alpha(1)$		$\alpha(2)$		$\alpha(3)$		$\alpha(4)$	
		NS	**MP**	**NS**	**MP**	**NS**	**MP**	**NS**	**MP**
10	2	1.04	0.02	1.03	0.03	1.02	0.02	1.04	0.02
15	2	1.01	0.01	1.02	0.01	1.04	0.02	1.05	0.03
15	3	1.04	0.02	1.05	0.04	1.04	0.02	1.05	0.03
20	2	1.02	0.02	1.03	0.01	1.06	0.03	1.05	0.03
20	3	1.05	0.03	1.03	0.02	1.04	0.02	1.06	0.04
20	4	1.04	0.02	1.04	0.02	1.05	0.03	1.05	0.02
25	2	1.03	0.02	1.02	0.01	1.07	0.04	1.06	0.03
25	3	1.01	0.01	1.04	0.02	1.05	0.03	1.06	0.03
25	4	1.04	0.03	1.02	0.03	1.04	0.02	1.04	0.02
25	5	1.04	0.02	1.05	0.02	1.06	0.02	1.07	0.04
30	2	1.02	0.02	1.00	0.00	1.05	0.03	1.05	0.02
30	3	1.04	0.02	1.04	0.02	1.03	0.02	1.06	0.03
30	4	1.04	0.03	1.05	0.03	1.06	0.02	1.05	0.03
30	5	1.05	0.03	1.06	0.03	1.04	0.01	1.04	0.02
30	6	1.04	0.01	1.03	0.01	1.04	0.02	1.07	0.04
40	2		0.04		0.03		0.02		0.03
40	3		0.03		0.02		0.03		0.04
40	4		0.02		0.03		0.01		0.03
40	5		0.04		0.02		0.03		0.02
40	6		0.05		0.03		0.03		0.02
40	7		0.04		0.02		0.04		0.03
50	2		0.03		0.04		0.03		0.02
50	3		0.04		0.02		0.01		0.04
50	4		0.03		0.02		0.04		0.03
50	5		0.04		0.03		0.01		0.03
50	6		0.03		0.02		0.04		0.02
50	7		0.02		0.01		0.03		0.02
50	8		0.03		0.04		0.03		0.03

NS: **NS** refers to the average ratio of the objective function value of the optimum nested solution to that of the optimal solution (the average over the number of instances in which **NS** was not optimal).

MP: **MP** refers to the % improvement in the objective value as a result of meshing heuristic.

movements (i.e., $\alpha = 0$), the procedures have a significant potential to improve the burst rates of the process.

In the development of the solution procedures, we assumed that the tool magazine is located at one of the extremities of the carrier (position 0). However, this may not be true and the tool magazine may be in the middle of the carrier. In such cases, we suggest a two-phase approach to the **RPP**. In the first phase, the components may be assigned to either side of the magazine using procedures similar to the partitioning problem referred to in Section 1. In the second phase, the heuristics of this paper can be used to determine the reel positions. In the formulation of **RPP** we assumed that the value of α is the same for all pick-and-place operations, i.e., the time for the concurrent movements is independent of the specific pick-and-place operations. This was a reasonable assumption in our application in which the range of α values was small. However, note that in other applications this may not be the case, and it is possible for this parameter to vary with the pick-and-place operations. The formulation of **RPP** is easy to modify to model this feature. Also, for the problem with a single reel group, the procedure of Section 2 can be extended in a straightforward manner. However, for the case with multiple reel groups, the required modifications are not obvious. In particular, the Lawler and Moore procedure to determine the optimal nested solution does not extend to this situation. However, the branch-and-bound procedure of Rinnooy-Kan, Lageweg and Lenstra (1975) can be adapted to solve this problem. Further details of these extensions can be found in Ahmadi et al.

APPENDIX A

Proposition 1. *The problem* **RPP** *is NP-hard.*

Outline of Proof. Consider the set partitioning problem (**SPP**): Given $a_1, a_2, \ldots, a_n \in Z^+$, are there sets A and B such that $A \cup B = \{1, 2, \ldots, n\}$, $A \cap B = \emptyset$ and $\sum_{i \in A} a_i = \sum_{i \in B} a_i$. Let $L = \sum_{i=1}^{n} a_i$. Consider the following instance of **RPP** with $n + 4$ reels, $t_1 = 2$, $t_i = a_{i-1}$, $i = 2, 3, \ldots, n + 1$; $t_{n+2} = L + 8$, $t_{n+3} = 2$, and $t_{n+4} = L + 4$; and $\alpha = L + 5$. We prove the proposition by showing that **SPP** has a yes answer if and only if the corresponding **RPP** has an objective value of $(n + 5)\alpha$.

i. Assume that **SPP** has a yes answer, and $\sum_{i \in A} a_i = \sum_{i \in B} a_i$. Also, without loss of generality, assume that $n \in B$. Consider the following sequence of reels in

RPP: ($n + 4$, 1, all i such that $i - 1 \in A$ in any order, $n + 3$, all i such that $i - 1 \in B$ in any order, $n + 2$). It is easy to confirm that each carrier movement is not greater than α. For example, movement $0 \to 1$ is $L + 5 = \alpha$, movement $i \to i + 1$, $i = 1, 2, \ldots, n + 1$ is less than $L + 4 < \alpha$; $n + 1 \to n + 2$ is less than $L + 4 < \alpha$; $n + 2 \to n + 3$ and $n + 3 \to n + 4$ are $L + 5 = \alpha$; $n + 4 \to 0$ is $(L + 4)/2 < \alpha$. Hence the objective value in **RPP** is $(n + 5)\alpha$.

ii. Conversely, assume that the objective value of **RPP** is $(n + 5)\alpha$. This implies that an optimal sequence will have the following characteristics:

a. reel $n + 4$ is positioned first,
b. reel 1 is positioned second, and
c. reel $n + 2$ is positioned last.

It is easy to see that violation of either a, b or c will result in at least one movement greater than α. Consider movements $n + 2 \to n + 3$ and $n + 3 \to n + 4$. Let A1 (B1) denote the set of reels positioned between 1 and $(n + 3)$ ($n + 3$ and $n + 2$). The carrier movement for $n + 2 \to n + 3$ is $(L + 8)/2 + \sum_{i \in B1} t_i + 1$, and the carrier movement for $n + 3 \to n + 4$ is $1 + \sum_{i \in A1} t_i + 2 + (L + 4)/2$. Since each of the moves above is not greater than α, we have $\sum_{i \in B1} t_i \leq L/2$, and $\sum_{i \in A1} t_i \leq L/2$. Since,

$$\sum_{i \in A1 \cup B1} t_i = \sum_{i=1}^{n} a_i = L, \quad \sum_{i \in A1} t_i = \sum_{i \in B1} t_i = L/2.$$

Let $A = \{i - 1 | i \in A1\}$ and $B = \{i - 1 | i \in B1\}$, then $\sum_{i \in A} a_i = \sum_{i \in B} a_i = L/2$ holds.

Proposition 2. $Z1/Z \leq 1.5$, *where*

Z = *the optimal value of the of the objective function of* **RPP**, *and*

$Z1$ = *the optimal value of the objective function of* **RPP** *with the set of feasible solutions restricted to oriented solutions.*

Outline of Proof. Let C(no) denote the optimal sequence of reel positions to **RPP**. We assume that C(no) is nonoriented. (Otherwise, there is nothing to prove.) Let M denote the last reel in this sequence. In what follows, we refer to carrier movements from start to reel M as forward movement and the movements from reel M to the tool magazine as backward movement. The corresponding costs are denoted by ZF and ZB, respectively. That is, $ZF(\text{no}) + ZB(\text{no}) = Z(\text{no}) = Z$. We assume that $ZF(\text{no}) \geq ZB(\text{no})$.

Consider the following two sequences:

C(os): the optimum oriented solution with reel M occupying the last position.

C(osl): the oriented solution that minimizes the cost of forward movement with reel M in the last position.

Then it follows that

i. $Z \leq Z1 \leq Z(\text{os}) \leq Z(\text{osl})$,
ii. $ZF(\text{osl}) \leq ZF(\text{os})$,
iii. $ZF(\text{osl}) \leq ZF(\text{no})$.

We make the following observations regarding C(no), C(os) and C(osl)

a. $ZB(\text{osl}) \leq d + (n + 1 - M)\alpha$, where d is the distance of the center of reel M from the tool magazine (position 0), and
b. $ZB(\text{no}) \geq d$, $ZB(\text{no}) \geq (n + 1 - M)\alpha$

so that $ZB(\text{no}) \geq (d + (n + 1 - M)\alpha)/2$. Then a and b together imply that $ZB(\text{osl}) \leq 2ZB(\text{no})$.

From the results above it can be seen that

$$\begin{aligned} Z \leq Z1 \leq Z(\text{os}) \leq Z(\text{osl}) &= ZF(\text{osl}) + ZB(\text{osl}) \\ &\leq ZF(\text{no}) + 2ZB(\text{no}) \\ &= Z(\text{no}) + ZB(\text{no}) \\ &\leq 3/2\ Z(\text{no}) = 3/2\ Z. \end{aligned}$$

If $ZF(\text{no}) < ZB(\text{no})$, the result can be shown by similar arguments and by considering an oriented solution that minimizes the cost of backward movements.

APPENDIX B

This Appendix provides a brief summary of the dynamic programming algorithm developed in Lawler and Moore to address a variety of scheduling problems. Next, we present an application of this procedure to solve **MRPP1** to obtain an optimum nested solution.

Dynamic Programming Algorithm of Lawler and Moore

Consider the sequencing problem with the following characteristics:

Number of Modes	2 (say mode 1 and mode 2)
Number of Jobs n	each job requires one operation
Process time for job j	$a_j(b_j)$ if processed in mode 1 (2)
Costs	$\lambda_j(t)$ $(\beta_j(t))$ is the cost of completing job j at time t in mode 1 (2).

The objective is to determine the choice of mode for each job and the sequence in each mode to minimize the total cost. The following dynamic programming (DP) formulation solves the problem: Let $f(j, t)$ = the minimum total cost for the first j jobs subject to the constraint that job j is completed no later than time t. The DP recursion is given as:

$$f(j, t) = \min \begin{Bmatrix} (f(j, t - 1) \\ \lambda_j(t) + f(j - 1, t - a_j), \\ \beta_j(t) + f(j - 1, t - b_j) \end{Bmatrix}$$

$$j = 1, 2, \ldots n;\ t \geq 0.$$

The problem is solved by the calculation of $f(n, T)$, where T is a sufficiently large number.

Application of the DP to Solve MRPP1 (To Obtain the Optimum Nested Solution for the Multiple Reel Group Problem)

As described in subsection 4.1, this application is a direct adaptation of the DP algorithm to solve the single machine weighted tardiness problem with a common deadline, described in detail in Section 10 of Lawler and Moore (p. 82). The procedure is as follows:

STEP 1. The number of jobs is equal to the number of reel groups m. Order the reel groups according to decreasing a'_k, i.e., $a'_1 \geqslant a'_2 \geqslant \cdots \geqslant a'_m$, where a'_k is the sum of reel widths of reels in reel group k.

STEP 2. Solve the DP for each subset of $m - 1$ jobs (excluding the reel group in set C) with these parameters:

$$a_j = a'_j,\ \lambda_j(t) = 0,\ b_j = 0,$$

$$\beta_j(t) = 2\left(a_k + \sum_{i=j+1,\neq k}^{m} a_i + t - \alpha\right) + \gamma_{1j} + \gamma_{2j},$$

$$\text{if } a_k + \sum_{i=j+1,\neq k}^{m} a_i + t - \alpha \geqslant 0$$
$$= \infty, \text{ otherwise,}$$

where γ_{1j} and γ_{2j} correspond, respectively, to the absolute lengths of the first and the last carrier movements for reel group j when it is placed starting at position 0.

Let $f^{(k)}(m, t)$ denote the solution for the subset of jobs $(1, 2, \ldots k - 1, k + 1, \ldots m)$.

STEP 3. Let $\phi(k, t)$ denote the total of first and last carrier movement for reel group k given that the starting position of the group is at t and that the reels within the group are positioned to provide optimum oriented solution. Note that this may be done using the algorithm of subsection 3.3. (Recall that the sequence of reels within each group is fixed in accordance with the optimum oriented solution.)
Find the value of k and t for which

$$f^{(k)}(m, t) + \phi(k, t), \qquad \alpha - a'_k \leqslant t < \alpha, \text{ is minimum.}$$

The value of k in Step 3 determines the identity of the reel group in class C. The reel groups processed with cost $\lambda_j(t)$ correspond to class A and those with cost $\beta_j(t)$ correspond to class B.

The correctness of the procedure in Step 2 and the justification for the definition of $\beta_j(t)$ can be seen by considering the partition of the first j reel groups into classes A and B. If the total process time of reel groups in class A is t, the total processing time for reel groups in class B is $\sum_{i=1,\neq k}^{j} a_i - t$. It follows that if reel group j is assigned to class B, its starting time will be $a_k + \sum_{i=j+1,\neq k}^{m} a_i + t$ and $\beta_j(t)$ defines the corresponding cost.

ACKNOWLEDGMENT

The authors are grateful to the referees and to the associate editor for their constructive comments. Dr. Tirupati's research was partially supported by NSF grant DDM-8957189.

REFERENCES

AHMADI, J. 1986. Optimization Problems of Flexible Assembly Systems in Large Scale Electronic Parts Manufacturing. Unpublished Ph.D. Dissertation, The University of Texas at Austin.

AHMADI, J., R. AHMADI, H. MATSUO AND D. TIRUPATI. 1992. Component Fixture Positioning for Printed Circuit Board Assembly With Concurrent Operations. Department of Management Working Paper, The University of Texas at Austin.

AHMADI, J., S. GROTZINGER AND D. JOHNSON. 1988. Component Allocation and Partitioning for a Dual Delivery Placement Machine. *Opns. Res.* **36,** 176–191.

BALAS, E. 1965. An Additive Algorithm for Solving Linear Problems With Zero-One Variables. *Opns. Res.* **13,** 517–545.

BALAS, E. 1967. Discete Programming by the Filtering Method. *Opns. Res.* **13,** 915–957.

BALL, M. O., AND M. J. MAGAZINE. 1988. Sequencing of Insertions in Printed Circuit Board Assembly. *Opns. Res.* **36,** 192–201.

CHAN, A. W., AND R. L. FRANCIS. 1979. Some Layout Problems on the Line With Interdistance Constraints. *Opns. Res.* **27,** 952–971.

DREZNER, Z., AND S. Y. NOF. 1984. On Optimizing Bin Picking and Insertion Plans for Assembly Robots. *IIE Trans.* **162,** 262–270.

GROTZINGER, S. 1988. Positioning for a Dual Delivery Placement Machine. Research Report RC 13908, IBM Watson Research Center, Yorktown Heights, N.Y.

HERAGU, S. S., AND A. KUSIAK. 1988. Machine Layout Problems in Flexible Manufacturing Systems. *Opns. Res.* **36,** 258–268.

HU, X., AND M. W. CARTER. 1989. Optimal Component Sequencing to Minimize Total Production Time. Presented at CORS/TIMS/ORSA Meeting, Vancouver.

HU, T. C., AND E. S. KUH. 1985. Theory and Concepts of Circuit Layout. In *VLSI Circuit Layout: Theory and Design*, T. C. Hu and E. S. Kuh (eds.). IEEE, New York.

KOUVELIS, P., W. CHIANG AND G. YU. 1992. Optimal and Heuristic Procedures for Row Layout Problems in Automated Manufacturing Systems. *IIE Trans. (to appear).*

LAWLER, E. L., AND J. M. MOORE. 1969. A Functional Equation and Its Application to Resource Allocation and Sequencing Problems. *Mgmt. Sci.* **16,** 77–84.

LEIPALA, T., AND O. NEVALAINEN. 1989. Optimization of the Movements of a Component Placement Machine. *Eur. J. Opnl. Res.* **38,** 167–177.

LOFGREN, J. K., AND L. F. MCGINNIS. 1986a. Dynamic Scheduling for Flexible Printed Circuit Card Assembly. In *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, Atlanta, Ga.

LOFGREN, J. K. AND L. F. MCGINNIS. 1986b. Soft Configuration in Automated Insertion. In *Proceedings of the 1986*

IEEE International Conference on Robotics and Automation, San Francisco.

PICARD, J.-C., AND M. QUEYRANNE. 1981. On the One-Dimensional Space Allocation Problem. *Opns. Res.* **29,** 371–391.

RINNOOY KAN, A. H. G., B. J. LAGEWEG AND J. K. LENSTRA. 1975. Minimizing Total Cost in One-Machine Scheduling. *Opns. Res.* **23,** 908–927.

ROMERO, D., AND A. SANCHEZ. 1990. Methods for the One-Dimensional Space Allocation Problem. *Comp. Opns. Res.* **17**(5), 465–473.

SOUKUP, J. 1981. Circuit Layout. *Proc. IEEE* **69,** 1281–1304. (Reprinted in *VLSI Circuit Layout: Theory and Design*, T. C. Hu and E. S. Kuh (eds.). IEEE, New York.)

TANG, C. S. 1986. A Job Scheduling Model for a Flexible Manufacturing Machine. In *Proceedings of the IEEE Conference on Robotics and Automation*, San Francisco.

TANG, C. S., AND E. V. DENARDO. 1988a. Models Arising From a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches. *Opns. Res.* **36,** 767–777.

TANG, C. S., AND E. V. DENARDO. 1988b. Models Arising from a Flexible Manufacturing Machine, Part II: Minimization of the Number of Switching Instants. *Opns. Res.* **36,** 778–784.

WONG, C. K. 1983. *Algorithmic Studies in Mass Storage Systems*. Computer Science Press, Rockville, Md.