Available online at www.sciencedirect.com
ScienceDirect
Discrete Optimization 4 (2007) 232–256
DISCRETE OPTIMIZATION
www.elsevier.com/locate/disopt

A model to optimize placement operations on dual-head placement machines

Wilbert E. Wilhelm[a,*], Nilanjan D. Choudhry[b], Purushothaman Damodaran[c]

[a] *Department of Industrial Engineering, Texas A&M University, TAMUS 3131, College Station, TX 77843-3131, United States*
[b] *Consumer Lending- Marketing HSBC USA, 2700 Sanders Avenue, Prospect Heights, IL 60070, United States*
[c] *Industrial and Systems Engineering, Florida International University, 10555 W Flagler Street, EC3172 Miami, FL 33174, United States*

Received 6 April 2006; received in revised form 13 November 2006; accepted 16 November 2006
Available online 22 December 2006

Abstract

Dual-head placement machines are important in the assembly of circuit cards because they offer the capability to place large components accurately. This paper presents a novel column-generation approach for optimizing the placement operations of a dual-head placement machine with the ultimate goal of improving the efficiency of assembly operations. Research objectives are a model that reflects relevant, practical considerations; a solution method that can solve instances within reasonable run times; and tests to establish computational benchmarks. Test results demonstrate the efficacy of our optimization approach on problems of realistic size and scope.

Keywords: Dual head placement machines; Surface mount technology; Column generation; Pick and place; Electronics assembly; Branch and bound; PCB assembly; Production planning; Throughput rate optimization; Constrained shortest path problem

1. Introduction

The dual-head (also called dual-gantry) placement machine (DHPM) plays an important role in circuit card (CC) assembly because it offers the capability of placing large and/or odd-shaped components (e.g., ball grid arrays, quad flat packs, column grid arrays, and flip-chips) with a high degree of accuracy. Most industrial assembly lines incorporate one or two DHPMs and several turret-type placement machines (TTPMs), which place small components (e.g., resistors and capacitors) very rapidly. Their importance is highlighted by the number of companies that market them, including Universal, Samsung, and Yamaha. The trend is for new CC designs to incorporate more components with finer leads and smaller pitches, so the need for the accuracy offered by DHPMs will increase in the future. The DHPM comprises a number of inter-related mechanisms and operates according to an intricate logic that creates challenges for process planners.

* Corresponding author. Tel.: +1 979 845 5493.
E-mail addresses: wilhelm@tamu.edu (W.E. Wilhelm), nilanjan.d.chowdhury@us.hsbc.com (N.D. Choudhry), damodarp@fiu.edu (P. Damodaran).

1572-5286/$ - see front matter
doi:10.1016/j.disopt.2006.11.006

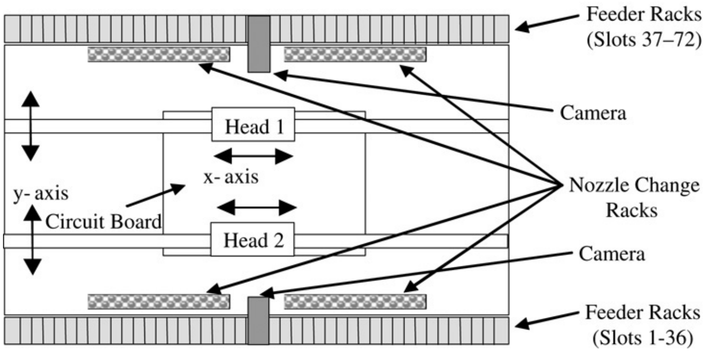

Fig. 1. Top-down view of a DHPM.

The purpose of this paper is to address this challenge by presenting a novel approach for optimizing the placement operations of a DHPM. The ultimate goal of this research is to provide a means of balancing workloads assigned to heads on DHPMs to promote efficiency by maximizing the throughput rate a line can achieve in assembling a given type of CC (e.g., [34]). With that goal in mind, we adopt the novel decomposition of process planning initiated by Wilhelm, Arambula, and Choudhry [38], yielding a series of four problems:

(P1) Assign component types (CTs) to heads and to feeder slots associated with each head (a CT is a set of physically identical components as the next subsection describes in detail).
(P2) Prescribe CT picking combinations (CTPCs) to minimize picking time.
(P3) Prescribe specific components placed on each placing step to minimize placement time.
(P4) Prescribe the sequence of picking, placing, and nozzle-change steps to balance workloads.

The DHPM presents challenges to modeling philosophy and to available modeling constructs. A primary question is, how can operations be abstracted to model them in sufficient detail to provide meaningful results? Based on the experience that has been reported in studies of other placement machines, a unified model that addresses all issues would not be solvable. Thus, decomposition is necessary. In fact, because operations are so extensively inter-related, it appears that this particular decomposition uniquely leads to meaningful problems that can be solved, as our research shows, in reasonable times to prescribe process plans. This paper presents a method to solve (P3) that can be used either alone to optimize placement operations or to complement algorithms that solve (P1), (P2), and (P4) to balance workloads. Placement time can be expected to dominate the times involved in picking and changing nozzles, so (P3) plays a very important role in balancing workloads.

Research objectives are (1) a model that reflects relevant, practical considerations; (2) a solution approach that can solve instances effectively; and (3) tests to establish computational benchmarks for the approach. No prior research has proposed an approach to prescribe placement decisions for the DHPM. Objective (2) highlights the need to prescribe a solution within a run time that will encourage process planners to implement the approach. Tests are needed to evaluate the robustness of the approach relative to different inputs, which might be prescribed by the solutions to (P1) and (P2).

This introduction comprises three subsections that describe DHPM placement operations, review relevant literature, and overview the paper, respectively.

1.1. DHPM placement operations

Fig. 1 depicts a DHPM [25] in a top-down view that shows a CC, heads, spindles, nozzles, feeder racks, cameras, and nozzle change racks and pads. Each head contains four spindles arranged on its centerline with adjacent spindles 40 mm apart. Each spindle uses a nozzle to grasp a component. Each head picks from a rack containing slots where feeders can be located. On a picking step the head moves to pick (up to 4) prescribed components from feeder slots, then moves to the camera and displays each component by positioning and holding it while the camera checks alignment. Subsequently, the placing step involves the head moving from the camera to the CC and placing the components one at a time. The CC is held in a fixed position while the heads move to populate it. Upon completing the placing step, the head moves to the nozzle storage rack to change and/or swap nozzles, if necessary, and then positions along the feeder rack to begin the next picking step. A round comprises picking, placing and nozzle-changing steps. This section describes some of the practical considerations that a model must address to prescribe placement operations.

Each CT comprises a set of components that are identical physically. Components of one CT are staged in a feeder that can accommodate its width. Each feeder is installed in one or more feeder slot(s) (depending upon its width) in a rack served by one of the heads. A camera occupies slots at the center of the feeder rack, essentially dividing it into two racks: the left-side rack $r = 1$ and the right-side rack $r = 2$.

The components that comprise a CT are physically identical, but individual components differ in their (x, y) locations on the CC. To place a component, the head must position the spindle, holding it at the correct (x, y) location. The placement operation itself requires the head to lower a spindle (along the z axis), apply a force that is necessary to place the particular CT, release the vacuum that holds the component, and raise the spindle. The placing step ends when components picked on the associated picking step have been placed; the head then positions to begin the next picking step, changing and/or swapping nozzles if necessary.

The head moves along the beam (in the x direction) and the beam moves along its guide rails (in the y direction). The beam can move much faster than the head, so different kinematics parameters (e.g., acceleration, velocity, and deceleration) govern their movements. Kinematics parameters are independent of the components the head is grasping and of components placed previously. The head and beam move simultaneously in the x and y directions and the centerline of the head (on which spindles are mounted) remains parallel to the x axis at all times.

1.2. Literature review

Process planning for CC assembly has been the focus of a number of prior studies; McGinnis, Ammons, Carlyle, Cranmer, Depuy, Ellis, Tovey, and Xu [33] and Nof et al. [34] provide overviews. Most studies focus on TTPMs (e.g.,[40]), single-arm robotic systems (e.g., [34]), and other systems unrelated to the DHPM (e.g., [35,30]). Only a few studies have investigated machines related to the DHPM and none has addressed the DHPM that we study in this paper.

Process planning typically involves prescribing five inter-related decisions [14,16]: (D1) partition CCs into families requiring similar CTs and prescribe a sequence for assembling CCs within each, (D2) assign each CT to a placement machine, (D3) assign each CT to a feeder slot(s) on the assigned machine, (D4) sequence component placement, and (D5) prescribe a retrieval plan for CTs assigned to more than one feeder. Production control deals with (D1); (D5) is required only if a CT is assigned to several feeders. We assume that an oracle prescribes decisions (D2), (D3), and (D5); we focus on (D4) as it relates to the DHPM. These decisions have been studied individually and in combination, for example, Ahmadi and Kouvelis [4]; Grotzinger [26] and Ahmadi, Ahmadi, Matsuo, and Tirupati [1]; and Ahmadi, Grotzinger, and Johnson [3] studied (D2), (D3), and (D5), respectively. Our approach uniquely identifies the set of decisions that are required to prescribe the placement operations of a DHPM.

Ahmadi, Grotzinger and Johnson [2] studied the DYNAPERT MPS500 placement machine, which has two heads mounted on a single arm. Heads are able to operate independently of each other and the CC is mounted on an xy table that moves to position the card for each placement. Chan and Mercer [12] addressed another type of dual-head machine that automatically loads each component into a head and employs an xy table to position the board for placement.

Crama, Kolen, Oerlemans and Spieksma [17] devised a hierarchical set of heuristics to prescribe decisions (D1)–(D4) for a related type of machine that has a single head with three spindles (in the terminology of this paper), each of which uses a nozzle to grasp components. Spindles always pick and place in the same sequence, 1–2–3. The worktable holds the card in fixed position and provides a feeder on each of its sides. More recently, Altinkemer, Kazaz, Köksalan and Moskowitz [6] studied a Quad 4000 series placement machine with a rotary head mounted on an arm (see also [31]). The head picks a certain number of components from the feeder rack, then the arm moves, positioning the head to place components. Several new papers (e.g., [36,11,28]) proposed heuristics for a placement machine that has a single head with multiple spindles (according to our terminology).

Most approaches decompose process planning into a set of related problems and use a heuristic to solve each. Grotzinger [26] emphasized this, noting that the “growing consensus in the literature with respect to this hierarchical decomposition approach and the sub-problems to be addressed by it . . . ”. For example, Crama, Flippo, van de Klundert and Spieksma [15] devised a polynomial time approach for the component retrieval problem (D5). We follow the lead of prior research in decomposing process-planning decisions but propose a unique set of four related problems for the DHPM: (P1), (P2), (P3), and (P4). Conforming to this philosophy, this paper focuses on (P3) placement operations.

We note that such a hierarchical decomposition has been used in other contexts as well. For example, in the flexible manufacturing setting, the inter-related problems of assigning tools to a capacitated magazine and sequencing jobs have been solved sequentially (e.g., [7]); small instances can also be solved by a unified model (e.g., [32]).

In the paper most related to this one, Wilhelm et al. [38] recently proposed an approach to optimize picking operations on a DHPM. Their analysis identified five types of picks that spindles can make: *gang pick*, *no-move pick*, *multiple pick*, *eclectic pick*, and *no pick*. On each picking step, the four spindles pick using some combination of the five types of picks to form a component type picking combination (CTPC). The importance of a CTPC to placing is that components picked on a picking step must be placed on the subsequent placing step (see (A2)). The five types of picks are relevant to placing in that (i) a *no pick* means that a spindle does not pick a component; (ii) a gang pick may pick 2, 3, or 4 components simultaneously; and (iii) other picks result in components being picked at different times. If 1, 2, or 3 spindles *no pick*, they cannot place on the subsequent placing step. If components are picked at the same time, the DHPM controller requires them to be placed according to the spindle closest to the camera in the associated CTPC. Finally, if components are picked at different times, the DHPM controller requires that they be placed in the same order in which they are picked.

Wilhelm et al. [38] devised a column-generation approach [37] to optimize (P2). It uses a general integer set-covering master problem along with sub-problems that are specially-constructed, resource-constrained shortest path problems. These sub-problems generate columns, each of which defines a CTPC p, by prescribing the set of CTs it picks, C_p; the number of components of CT c it picks, $e_{cp} \in \{0, 1, 2, 3, 4\}$; the order in which spindle $s \in \{1, 2, 3, 4\}$ picks, $o_{sp} \in \{0, 1, 2, 3, 4\}$; the CT spindle s picks, $\bar{c}_{sp} \in C_p$; and the time for the head to pick components and display each to the camera, $\hat{\Theta}_p$. Their master problem minimizes total picking time, $\sum_{p \in P} \hat{\Theta}_p x_p^*$, by prescribing an optimal set of CTPCs, P^*, and the number of times each must be used to pick all components, x_p^*. The solution prescribed by their model (i.e., x_p^*, C_p, e_{cp}, o_{sp}, and $\bar{c}_{sp}$ for $p \in P^*$) can be used as the input to the model we present in this paper; alternatively, these inputs can be provided by some other suitable source.

To demonstrate the relationship between (P2) and (P3), suppose (P2) involves four CTs, each comprising 10 components, and that the solution to that problem prescribes a single CTPC, $|P^*| = 1$, that gang picks all four CTs simultaneously on each of $x_1^* = 10$ picks. The (P3) problem must now prescribe which of the individual components (from each CT) to place on each of 10 placing steps. Individual components can be placed in a total of 10^4 combinations, so (P3) is not a trivial problem. More complex cases are, of course, even more challenging.

Another related paper by Choudhry, Wilhelm, Vasudeva, Gott, and Khotekar [13] presents a heuristic to solve (P1), an optimizing method to solve (P4), and reports computational experience, which shows that the algorithm proposed by Wilhelm et al. [38] for (P2); the approach presented in this paper for (P3); and the method they describe for (P1) and (P4) do, in fact, balance workloads quite well, achieving the ultimate objective of promoting efficiency by balancing workloads to maximize the throughput rate a line can achieve in assembling a given type of CC. This approach solves (P1) with a heuristic because it can do no better — the impact that the assignment of CTs to feeder slots has on workload balance cannot be determined until (P2), (P3), and (P4) have all been solved. While sub-optimal, this approach optimizes (P2), (P3), and (P4) in turn, each given the solution(s) to earlier problem(s), in a sense prescribing the best possible solution for each problem.

1.3. Paper overview

This paper comprises four sections. Section 2 formulates our model and Section 3 describes our column-generation solution approach. Section 4 reports computational experience and Section 5 summarizes conclusions and recommendations for future research.

2. Model formulation

This section formulates our model, which reflects the practical considerations described in Section 1. Each placing step deals with the CTs associated with one CTPC, which can be prescribed by (P2). A CTPC specifies the nozzle type fitted to each spindle, the CT each spindle picks, and the order in which spindles pick. Placing-step time is determined as the sum of times for the head to move from the camera to the CC and place components sequentially, each at its specific (x, y) location. The objective of (P3) is to minimize the total time to place all components by prescribing a set of specific components to be placed on each placing step (for which the CTs are defined by the associated CTPC). This objective is important because placing operations account for a substantial portion of the workload assigned to a head and thus constitute a challenging aspect of balancing workloads.

This section comprises three subsections. The first two provide a modeling structure by describing our assumptions and relating operating rules for placement. The third subsection presents the binary integer, set covering model that we use as our master problem. We define notation as we present it and also summarize it in Table 1 for reader convenience.

2.1. Assumptions

We structure (P3) by invoking five assumptions:

(A1) An oracle solves problems (P1) and (P2) to provide inputs to problem (P3).
(A2) The DHPM controller specifies the sequence in which spindles pick and place on a round.
(A3) The DHPM controller ensures that the two heads on a DHPM will not collide.
(A4) To perform each movement, the head – and, independently, the beam – accelerates from a rest, then travels at a constant velocity and, finally, decelerates to stop.

(A1) reflects our unique decomposition of process-planning decisions. We invoke (A2) because the DHPM machine controller can, apparently, not be changed and because our industrial collaborators were steadfast in adhering to that logic. Assumption (A3) recognizes the algorithms incorporated in the machine controller to assure that heads do not collide as they move through their rounds. (A4) provides a basis that allows us to model the kinematics of DHPM movements.

2.2. Operating rules

Wilhelm et al. [38] specified several operating rules to manage travel time along the feeder rack:

(R1) A head picks all components from its left-half rack ($r = 1$) then picks all components from its right-half rack ($r = 2$).
(R2) Each picking step starts with the head at its feeder-rack position farthest from the camera and requires the head to move towards the camera on successive picks.
(R3) A head does not cross any point on a feeder rack more than once on a picking step.

If a CTPC included CTs that are located in slots in both $r = 1$ and $r = 2$ racks, the head would incur the inefficiency of traveling across the camera — a distance of 80 mm — to reach CTs in the second half rack; then, it would incur additional inefficiency as it backtracks — an inefficient operation that would violate rule (R3) — to the camera to view components. Rules (R2) and (R3) promote efficiency and (R1) allows decomposition so that picking (P2) and placing (P3) operations can be analyzed in smaller, independent sub-problems, one for each hmr combination where $m \in M$, $h \in H_m$ and $r \in R_{hm}$ in which

M is the set of machines
H_m is the set of heads on machine m
R_{hm} is the set of racks associated with head h on machine m.

For example, if the line incorporates two DHPMs ($M = \{m : m = 1, 2\}$), we solve a set of eight (P3) problems, because each DHPM has two heads ($H_m = \{h : h = 1, 2\}$) and each head has two racks ($R_{hm} = \{r : r = 1, 2\}$) according to (R1).

2.3. The model

Our model requires inputs that may be obtained as outputs from (P2): x^*_p, C_p, e_{cp}, o_{sp}, and $\bar{c}_{sp}$ for $p \in P^*$. The output of our (P3) model prescribes the individual components $i \in I$ (i.e., with coordinates (x_i, y_i)) placed on each step t and the total time the step requires to move from the camera and place prescribed components, $\bar{\Theta}_t$. To populate the CC, (P3) must place CTPC p on each of x^*_p steps, so that it prescribes a total of $\sum_{p \in P^*} x^*_p$ steps.

We solve a (P3) model for each hmr combination. Suppressing subscripts hmr to facilitate presentation and defining decision variable $u_t = 1$ if placing step t is prescribed and 0 otherwise, we now present our (P3) model:

Table 1
Notation

Indices	
h	Heads on a DHPM
i, j	Individual components on a circuit card
m	DHPMs on the assembly line
p	CTPCs
r	Racks associated with a head
t	Placing steps
Parameters	
$\bar{a}_{it}$	$= 1$ if placing step t places component i, 0 otherwise
c_i	CT for component i
$\bar{c}_{sp}$	CT picked by spindle s in CTPC p $\quad \bar{c}_{sp} \in C_p$ for $s \in \{1, 2, 3, 4\}$
e_{cp}	Number of components of CT $c \in C_p$ picked in CTPC p
o_{sp}	Order in which spindle s picks in CTPC p $\quad o_{sp} \in \{0, 1, 2, 3, 4\}$ for $s \in \{1, 2, 3, 4\}$
x_p^*	Number of times the CTPC should be used to minimize total picking time
(x_i, y_i)	Co-ordinates at which component i is located on the circuit card
$\hat{\Theta}_p$	Time for the head to pick all components in CTPC p and display each for camera to view
$\bar{\Theta}_t$	Time for placing step t
Sub-problem parameters	
a	Arc in a sub-problem network
i	Component represented by the node at which arc a originates
j	Component represented by the node to which arc a points
q_a	Reduced cost associated with arc a
s	Spindle (=1, 2, 3, 4)
s_{ℓ_i}	Spindle that grasps component i
s_{ℓ_j}	Spindle that grasps component j
β_a	Resource required by the operation represented by arc a
γ_j	Dual variable associated with the jth constraint of type (2)
τ	Time limit for a placing step
Time parameters	
$(\tilde{a}_x, \tilde{v}_x, \tilde{d}_x)$	Acceleration, velocity and deceleration rate for the head to move along the x axis
$(\tilde{a}_y, \tilde{v}_y, \tilde{d}_y)$	Acceleration, velocity and deceleration rate for the beam to move along the y axis
D_{ij}^x	Distance the head moves along the x axis for successive placements of components i and j $=(x_j - x_i - 40(s_{\ell_j} - s_{\ell_i}))$ if the head moves from left to right (i.e., $x_i < x_j$), and $= -(x_j - x_i - 40(s_{\ell_j} - s_{\ell_i}))$ if the head moves from right to left (i.e., $x_i > x_j$).
D_{ij}^y	Distance the head moves along the y axis for successive placements of components i and $j = \lvert y_i - y_j \rvert$
$g_x(D_{ij}^x)$	Function that applies kinematic relationships to determine move time along the x axis
$g_y(D_{ij}^y)$	Function that applies kinematic relationships to determine move time along the y axis
$\bar{\theta}_{ij}$	Time for the head to move from the position with spindle s_i centered at (x_i, y_i) to the position with spindle s_j centered at (x_j, y_j) and place a component of CT c_j.
Sets	
C_p	CTs picked by CTPC $p \in P^*$
H_m	Heads on machine m
I	Components that populate the circuit card
M	Machines
P^*	CTPCs prescribed by (P2)
R_{hm}	Racks associated with head h on machine m
T	Index set representing all possible combinations in which components can be placed
T^*	Placing steps prescribed by the solution to (P3) $T^* \subseteq T$
T_p	Placing steps that represent all possible combinations in which components associated with CTPC p can be placed
T_p^*	Placing steps that represent the combinations in which components associated with CTPC p are prescribed by the solution to (P3) $T_p^* \subseteq T_p$
Decision variables	
u_t	$= 1$ if placing step t is prescribed, 0 otherwise

$$\text{Min } Z = \sum_{t \in T} \bar{\Theta}_t u_t \tag{1}$$

$$\text{st. } \sum_{t \in T} \bar{a}_{it} u_t \geq 1 \quad i \in I \tag{2}$$

$$\sum_{t \in T_p} u_t = [x_p^*] \quad p \in P^* \tag{3}$$

$$u_t \in \{0, 1\} \quad t \in T. \tag{4}$$

The objective function (1) minimizes the total time for all placing steps. Inequalities (2) assure that each individual component i is placed. Equalities (3) assure that (P3) employs each of the CTPCs the number of times prescribed by input $[x_p^*]$. Rather than artificially constraining (P3), constraints (3) assure the necessary integration of picking and placing operations. They guarantee that the pre-determined set of CTPCs P^* will be used for both picking and placing, as desired. As an example, suppose that four CTs can be gang picked and that three of them comprise 5 components; and the fourth, 6. $P^* = \{1, 2\}$, CTPC $p = 1$ gang picks the four CTs and $[x_1^*] = 5$, while CTPC $p = 2$ picks just one component of type 4 and $[x_2^*] = 1$. Without (3), (P3) could use CTPC $p = 2$ six times and CTPC $p = 1$ five times, each time placing only 3 components. Clearly, this lack of consistency between picking and placing is not feasible — once picked, components must be placed. Finally, constraints (4) impose binary restrictions.

T is the index set of columns that represents all of the combinations in which individual components can be placed; its cardinality is huge. Similarly, T_p comprises the large number of combinations in which CTPC p can be placed, so that $|T_p|$ is large ($T = \cup_{p \in P} T_p$). Our model prescribes subsets $T_p^* \subseteq T_p$ and $T^* \subseteq T$, where $T^* = \cup_{p \in P^*} T_p^*$, $|T_p^*| = x_p^*$, and $|T^*| = \sum_{p \in P^*} x_p^*$. Note that $T_p \cap T_{p'} = \emptyset : p, p' \in P^*, p \neq p'$.

The column of coefficients associated with decision variable u_t, $\begin{bmatrix} \bar{\Theta}_t & \bar{\mathbf{a}}_t^T \end{bmatrix}^T$, where $\bar{\mathbf{a}}_t$ is an $|I|$-dimensional column vector comprising elements $\bar{a}_{it}$ for $i \in I$, is not known explicitly *a priori*; rather, it is generated by a sub-problem as described in Section 3. The advantage of column generation is that columns can be generated as needed instead of enumerating all of them explicitly. Because $|T|$ is huge, enumerating columns does not offer an effective way to solve (P3).

We solve the linear relaxation of model (1)–(4) to obtain a bound at each node in the branch-and-bound tree over which we search for an optimal solution. A number of applications (e.g., cutting stock [23,24]) and vehicle routing [22,18,21] have demonstrated that the set-covering model (i.e., (1), (2) and (4)) can be optimized effectively using column generation [37]. The set-covering formulation is efficacious because it is known [18,10] to have a tight linear relaxation (i.e., the gap between the optimal solution to the integer problem and its linear relaxation tends to 0 as $|I|$ increases, facilitating solution). However, side constraints (3), which assure continuity with the solution to (P2), may lead to somewhat larger gaps in certain cases.

Model (1)–(4) is novel in three ways: column generation has not been used previously to prescribe process plans, DHPM placing operations have not been modeled previously, and specially formulated sub-problems have not been used to generate "good" placement steps previously.

3. Solution approach

This section presents three subsections that describe our sub-problems, our sub-problem solution methods, and our column-generation approach.

3.1. Sub-problems

Sub-problem p employs a network to model CTPC $p \in P_{hmr}^*$; we solve a constrained shortest path problem (CSPP) on that network to generate a column, which prescribes coefficients $\bar{\Theta}_t$ and $\bar{a}_{it}$ (for $i \in I$). Fig. 2 depicts the fundamental concepts on the directed, acyclic network $G_{hmr}(\bar{N}, \bar{A})$ in which $\bar{N}$ is the set of nodes and $\bar{A}$ is the set of (directed) arcs.

$\bar{N}$ includes a (dummy) start node in level $\ell = 0$ and a (dummy) end node in level $\ell = L + 1$, where L is the number of spindles that pick/place in CTPC p ($L = 4$ if the CTPC includes 0 *no picks* since all 4 spindles pick; $L = 3$ if the CTPC includes 1 *no pick* since 3 spindles pick; overall, $L = 4, 3, 2, 1$ if the CTPC includes $0, 1, 2, 3$ *no picks*,

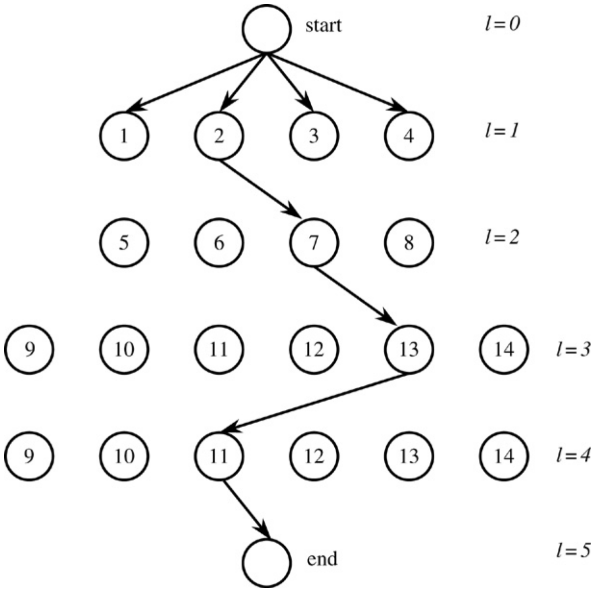

Fig. 2. An example sub-problem network.

respectively). Level $\ell = 1, \ldots, L$ represents the CT that the head places ℓ^{th}, so that the ordering of levels corresponds to the order in which spindles place, observing (A2). The nodes in a level represent the individual components that constitute the associated CT. Node i in level $\ell = 1, \ldots, L$ represents individual component i of CT $c_i \in C_p$ and must be placed at location (x_i, y_i) on the CC. For example, Fig. 2 depicts a case in which $L = 4$. Level $\ell = 1$ represents CT $c = 1$,which comprises individual components $1, \ldots, 4$. Similarly, level $\ell = 2$ represents CT $c = 2$, which comprises individual components $5, \ldots, 8$ and levels $\ell = 3$ and $\ell = 4$ represent CT $c = 3$, which comprises individual components $9, \ldots, 14$. This network represents a CTPC that uses two spindles to pick CT $c = 3$ (see [38] for more detail). The path from the start node to the end node represents a step that places individual components 2, 7, 13, and 11 (other arcs are not shown for clarity).

Each arc $a = (i, j) \in \bar{A}$ points from a node i in level $\ell = 0, \ldots, L$ to a node j in level $\ell + 1$. The network incorporates three types of arcs. Each type 1 arc points from the start node in level $\ell = 0$ to a node in level $\ell = 1$, each type 2 arc points from a node in level $\ell = 1, \ldots, L - 1$ to a node in level $\ell + 1$, and each type 3 arc points from a node in level L to the end node in level $L + 1$.

Each arc a of type 1 or 2 represents the time and set of resources needed to place the component, j, which is represented by the node to which a points. Each path from the start node to the end node includes one type 1 arc. We set the time $\bar{\theta}_{ij}$ associated with type 1 arc $a = (i, j)$ to be the time required for the head to move from the camera to position spindle s_{ℓ_j}, which is represented by level $\ell = 1$, at (x_j, y_j) and to place component j (see Section 1). We set the time $\bar{\theta}_{ij}$ associated with type 2 arc $a = (i, j)$ to be the time required for the head to move from the position with spindle s_{ℓ_i} at location (x_i, y_i) to the position with spindle s_{ℓ_j} at position $\left(x_j, y_j\right)$ and to place component j (see Section 1). Because type 3 arcs are included only as logical devices to reach the end node, we set the time for each type 3 arc to be $\bar{\theta}_{ij} = 0$.

The head moves simultaneously along the x and y axes, so the time required for the head-movement portion of duration $\bar{\theta}_{ij}$ may be calculated using

$$\max\{g_x(D^x_{ij}), g_y(D^y_{ij})\},$$

in which the move distance along the y axis is $D^y_{ij} = |y_i - y_j|$ and the move distance along the x axis, correcting for the 40 mm spacing of adjacent spindles, is

$$D^x_{ij} = (x_j - x_i - 40(s_{\ell_j} - s_{\ell_i})) \quad \text{if the head moves from left to right (i.e., } x_i < x_j\text{)}, \quad \text{and}$$
$$D^x_{ij} = -(x_j - x_i - 40(s_{\ell_j} - s_{\ell_i})) \quad \text{if the head moves from right to left (i.e., } x_i > x_j\text{)}.$$

The $g_x(D^x_{ij})(g_y(D^y_{ij}))$ function applies kinematics relationships to determine move time along the x (y) axis as the head travels on the beam (as the beam moves). We model a movement by starting with the head (beam) at rest, accelerating it to the constant velocity at which it travels, and decelerating it to stop. We use $(\tilde{a}_x, \tilde{v}_x, \tilde{d}_x)$ $((\tilde{a}_y, \tilde{v}_y, \tilde{d}_y))$ to denote the kinematics parameters for head (beam) movement in the x (y) direction. The equation of motion that gives the move time in either direction is

$$g(D_{ij}) = [2D_{ij} + \tilde{v}^2(1/\tilde{a} + 1/\tilde{d})]/(2\tilde{v}).$$

A move over a short distance ($D_{ij} \leq \bar{D} = 0.5\tilde{v}^2(1/\tilde{a} + 1/\tilde{d})$) does not allow the velocity to be achieved, so it involves only acceleration and deceleration. In such a case, the move time is given by

$$D_{ij} = [2D_{ij}(1/\tilde{a} + 1/\tilde{d})]^{1/2}.$$

We label arc a with the amount of resource required by the operation, β_a, and the appropriate reduced cost $q_a = \bar{\theta}_{ij} - \gamma_j$, in which $\bar{\theta}_{ij}$ is the time duration required to reposition and place component j and γ_j is the dual variable associated with the jth constraint of type (2) in the master problem. Type 3 arcs are exceptions that are labeled with values $\bar{\theta}_{ij} = 0$, $\beta_a = 0$, and $q_a = 0$. We update reduced costs each time sub-problems are solved, incorporating current (optimal) values of dual variables, γ_j for $a \in \bar{A}$. We define the resource requirement $\beta_a = \bar{\theta}_{ij}$ for arc a and interpret the resource constraint in the CSPP as a limit on the duration of a placing step, τ.

According to the linear programming optimality criterion [8], the solution to a sub-problem represents the non-basic column that has the smallest reduced cost among all non-basic columns associated with the related CTPC. It identifies an improving column that may enter the basis of the master problem if the solution value is negative. If the optimal solution value is non-negative for all sub-problems, the current solution to master problem (1)–(4) is optimal.

Sub-problem $SP(p)$ may now be stated using decision variables $w_a = 1$ if arc a is on the CSP, 0 else (for $a \in \bar{A}$):

$$SP(p): \text{Min } “C_t - Z_t” = \sum_{a \in \bar{A}} q_a w_a \tag{5}$$

$$\text{st.} \sum_{a \in \bar{A}_{(i.j)}} w_a - \sum_{a \in \bar{A}_{(j,i)}} w_a = b_i \quad i \in \bar{N} \tag{6}$$

$$\sum_{a \in \bar{A}} \beta_a w_a \leq \tau \tag{7}$$

$$w_a \in \{0, 1\} \quad a \in \bar{A}. \tag{8}$$

The objective function (5) minimizes the sum of the reduced costs $q_a = \bar{\theta}_{ij} - \gamma_j$ of prescribed arcs. Constraints (6) formulate the SPP as a network flow problem in which one unit of flow originates at the *start* node, travels across the shortest path in the network, and terminates at the *end* node. They require a flow balance at each node, so that the flow out of node i on the set of arcs $\bar{A}_{(i,j)}$ minus the flow into it on the set of arcs $\bar{A}_{(j,i)}$ equals b_i, where $b_{\text{start}} = +1$, $b_{\text{end}} = -1$, $b_i = 0$ for $i \in \bar{N} \setminus \{\{\text{start}\} \cup \{\text{end}\}\}$. Inequality (7) invokes the resource constraint, limiting the duration of any placing step generated. This limitation promotes efficiency by assuring that a head will not take long to perform a placing step, causing the second head to incur a lengthy idle time waiting to begin placing. While (5), (6) and (8) define a shortest path problem, which has the Integrality Property [37], (5)–(8) define a constrained shortest path problem, which does not have the Integrality Property, allowing our branch-and-price approach to provide tighter bounds that improve the effectiveness of our solution approach. We solve model (1)–(4) for each hmr combination; each of these “rack problems” includes a sub-problem of the form $SP(p)$ for each CTPC $p \in P^*_{hmr}$.

An optimal solution to $SP(p)$ prescribes the sequence in which components are placed; a series of arcs from the start node to the end node: $\Pi^* = \{a \in \bar{A} : w^*_a = 1\}$; parameters $\bar{\Theta}_t = \sum_{a \in \bar{A}} \beta_a w^*_a = \sum_{a \in \Pi^*} \bar{\theta}_{ij}$; and $\bar{a}_{jt} = 1$ if $a = (i, j) \in \Pi^*$, 0 else for $j \in I$. Resource limitation (7) assures that $\bar{\Theta}_t \leq \tau$. If “$C^*_t - Z^*_t$” < 0, the sub-problem solution defines an improving column, which may be entered into the basis of the master problem. Inequalities (2) allow a component to be placed more than once and a CTPC that includes multiple picks or an augmented gang pick could lead to placing the same component i several times on one step. We do not include arc (i, i) from level ℓ to level $\ell + 1$ (for $\ell = 2, \ldots, L - 1$) because it would represent placing component i twice on the same step. Component i could, however, be prescribed twice, for example, by a path that includes nodes representing component i in levels ℓ and $\ell + 2$ (for $\ell = 1, \ldots, L - 2$). Since each placement requires time, it would not be optimal to place a component more than once so that inequalities (2) hold at equality at an optimal solution. The next section describes how we solve each sub-problem, a CSPP, as a SPP.

3.2. Solving the CSPP as a SPP

The CSPP is NP-hard [27] but has been well researched (e.g., [22,9,18,21]) and several pseudo-polynomial-time dynamic programming algorithms are available [19,20,29]. We base our pseudo-polynomial time algorithm on that of Wilhelm, Damodaran and Li [39]. The algorithm uses a pseudo-polynomial time dynamic programming algorithm to construct a directed, acyclic expanded network on which the CSPP is solved as a SPP at each iteration using a polynomial-time algorithm [5]. The algorithm actually performs in a pseudo-polynomial time because the size of the expanded network may grow quickly.

Each node (arc) in the sub-problem network may be associated with a number of nodes (arcs) in the expanded network. Each node in the expanded network represents a path into the associated node in the sub-problem network and the cumulative resource requirement on that path. To construct the expanded network, we process levels in a sub-problem network from level $\ell = 0$ to level $\ell = L+1$ and nodes within each level from left to right. To extend the paths that lead to a node in the expanded network, each arc emanating from the associated node in the sub-problem network is augmented to reach the next level of the expanded network. We allow only augmented paths with cumulative resource requirements that do not exceed the τ limitation. After identifying nodes in the next level of the expanded network, we check the subset of these nodes that is associated with the same node in the sub-problem network. If the cumulative resource requirements on the paths into two or more of nodes in a subset are identical, we merge them into a single node. This combination manages the growth of the expanded network and leads to the pseudo-polynomial time complexity of the expansion method. Nodes in level ℓ of the expanded network represent nodes in level ℓ in the sub-problem network. Each node in level L of the expanded network represents a path through the sub-problem network that specifies a placing step, including the individual components placed, placement sequence, placement time, and the cumulative resource (time) required, which cannot exceed τ. We invoke resource constraints by disallowing any path with cumulative resource requirements that would exceed resource limitations so that the CSPP on the sub-problem network can be solved as a SPP from the start node to the end node in the expanded network.

3.3. Column generation details

We optimize our model using branch and bound (B&B). At each node n in the B&B search tree, we solve the linear relaxation of model (1)–(4), replacing (4) with

$$0 \leq u_t \leq 1 \quad t \in T.$$

To construct an initial basic feasible solution (BFS) at each B&B node, we define a set of placing steps, each of which places one component at a *Big_M* cost (i.e., time). The columns associated with this initial BFS, which comprise a basis of artificial variables, represent a feasible, albeit costly, solution.

At each iteration, we optimize the master problem over columns known explicitly before updating the reduced costs on all arcs in the expanded networks using the new q_a value on each arc in an expanded network that is associated with arc a in the sub-problem network. Our column generation strategy solves all sub-problems and enters the most improving column.

At each B&B node, we branch on the master problem variable with the largest fractional part, say u_t, conforming to the traditional method for branching on binary variables. Branching at node n creates child nodes n_1 and n_2 at which u_t is fixed to 0 and 1, respectively.

At B&B child node n_1, the column associated with placing step t must be removed from the basis, and the expanded network must be modified so that step t cannot be obtained again as a solution. We adopt the "Bypass Algorithm" proposed by Wilhelm et al. [38] to prevent our algorithm from generating the column associated with u_t again. The goal is to disallow *the set of all arcs* on the path, Π_t^*, that defines the column associated with u_t but to *allow any subset* to be used to define another column. Starting at the dummy node in level $\ell = 0$, we traverse the arcs on path Π_t^* to identify the first node, v_{ℓ^*}, that has more than one predecessor in the expanded network. If each node on path Π_t^* has a single predecessor, we traverse the arcs on path Π_t^* to identify the first node, v_{ℓ^*}, that has more than one successor in the expanded network. In the special case in which each node on path Π_t^* has a single predecessor and a single successor, the reduced cost of any arc $a \in \Pi_t^*$ can be set to Big_M to exclude the path. The arc $a \in \Pi_t^*$ that points to node v_{ℓ^*}, $(v_{\ell^*-1}, v_{\ell^*})$, is on path Π_t^* but no other path from the start node to the end node, so that setting $q_{(v_{\ell^*-1}, v_{\ell^*})} = Big\_M$ prevents step t from being prescribed again as optimal. The algorithm then bypasses arc $(v_{\ell^*-1}, v_{\ell^*})$, augmenting arcs (v_{ℓ^*-1}, v) from node v_{ℓ^*-1} to each node v in levels $\ell = \ell^* + 1, \ldots, L$ that could be reached on any path $(v_{\ell^*-1}, v_{\ell^*}, \ldots, v)$ that starts with arc $(v_{\ell^*-1}, v_{\ell^*})$. These arcs bypass node v_{ℓ^*}, which is on path $(v_{\ell^*-1}, v_{\ell^*}, \ldots, v)$. The bypassing arc is assigned a reduced cost equal to the sum of the reduced costs that would otherwise be assigned to the arcs on path $(v_{\ell^*-1}, v_{\ell^*}, \ldots, v)$. The resulting, augmented network must be used at child node n_1 in the B&B tree to prescribe the path with the minimum reduced cost that excludes path Π_t^*. Each node that descends from node n_1 must include this bypass and may include additional bypasses as well, corresponding to other binary variables that are fixed to 0 (i.e., "$C_t - Z_t$" = 0). At each subsequent iteration of the master problem, we update the reduced cost on each arc, except, of course, the Big_M cost must be retained on each arc designated by Algorithm Bypass. Wilhelm et al. [38] provide a detailed justification of the Bypass Algorithm, showing that it excludes only path Π_t^*. Algorithm Bypass allows each sub-problem to be solved as a SPP, no matter how many binary variables are fixed to 0.

At B&B child node n_2, the column associated with placing step t must be retained in the solution to the master problem. While it is in the basis of the master problem, the sub-problem will not identify it again as improving because it will price out at 0. The associated variable must be retained in the master problem but it may become non-basic at its upper bound.

4. Computational evaluation

This section describes the experiments we use to evaluate the efficacy of our column generation approach. The purposes of our tests are to investigate the effects that CTPCs and fundamental parameters (i.e., factors) have on run time, to evaluate the robustness of our approach in the face of different CT assignments to feeder slots, to explore the relationship of run time to the tightness of our model, and to record benchmarks for use in future research.

We coded our program in C in the Watcom-C editor and performed all tests interfacing with MINTO 3.0 and CPLEX 4.0 on a Pentium III PC (667 MHz with 128 MB RAM). We note that our industrial collaborator enabled this study but required a nondisclosure agreement that does not allow us to relate certain details in this paper. This section describes the factors we used to evaluate our approach, the way we generated test instances, and computational results.

4.1. Experimental design

Our experiment addresses six factors, which represent different machine configurations, operating restrictions, and operating procedures. The experimental design assigns two levels to each of five factors to evaluate its effect over a range of possibilities.

Factor 1 addresses the logic by which CTs are assigned to DHPMs and to the feeder slots on each machine. We use three different heuristics, (level 1) H1, (level 2) H2, and (level 3) H3 as the levels of this factor to evaluate the robustness of our column generation procedure relative to different assignments of CTs to slots. The actual logic incorporated in these heuristics is not within the scope of this paper.

Factor 2 specifies the number of DHPMs: (level 1) 1 and (level 2) 2. Factor 3 defines the number of CTs and width of each. We select two levels to fill all slots on a single DHPM: (level 1) 32 CTs, each requiring 2 slots and (level 2) 64 CTs, each requiring 1 slot. When two DHPMs are used (level 2 of Factor 2) each of these sets of CTs fills only half of the available slots. Factor 4 designates the number of components of each CT: (level 1) 10 components and (level 2) a number generated from a discrete uniform distribution on [5, 15] (i.e., DU [5, 15]). Level 2 has an expected number of components equal to that specified by level 1 but introduces variability to test the ability of our column-generation approach to prescribe appropriate placing steps under different conditions. Factor 5 provides either (level 1) 2 or (level 2) 4 types of nozzles to each head, with 4 copies of each. In practice, each nozzle type can only pick CTs of specified width and may affect how frequently efficient gang picks can be used. Level 1 assigns a nozzle type to each CT using DU [1, 2]; and level 2, DU [1, 4], to provide a wider variety of nozzle types. Factor 6 assigns an orientation – denoted θ – to each CT. Each of the levels of this factor is an empirical distribution that we select to study requirements of different types. Level 2 generates more highly variable orientations and may limit the use of efficient gang picks, which require all CTs to have the same orientation.

Factor			Levels
1	Assignment of CTs to DHPMs and to slots on each machine		H1
			H2
			H3
2	Number of DHPMs		1
			2
3	Number of CTs		32 CTs, each 2 slots wide
			64 CTs, each 1 slot wide
4	Number of components of each CT		10
			DU[5, 15]
5	Nozzle type assigned to each CT		DU [1, 2]
			DU [1, 4]
		degrees	empirical probability distribution
6	Orientation assigned to each CT, θ:	0, 90, 270 or 180	0.4, 0.3, 0.2, 0.1
		0, 90, 270 or 180	0.25, 0.25, 0.25, 0.25

We select these six factors because they have significant influence over problems (P1)–(P4) and, hence, the workload balance that can be achieved and the run time required to do so. Some factors have an effect on several of these problems; others affect only one of them. Factor 1, the heuristic used to solve (P1), assigns CTs to feeder slots and affects the sizes of problems (P2)–(P4). Different logics used to solve (P1) assign CTs differently to feeder racks and may result in (P2)–(P4) instances that differ widely in terms of size and solvability. Factor 2, the number of DHPMs, influences the size of all problems and makes workload balancing more difficult as the number of heads increases. Fortunately, our approach decomposes (P3) into individual hmr rack problems so that several smaller problems can be solved instead of one large one. Factor 3, the number of CTs, affects the number of decisions that must be made to resolve (P1), the sizes of (P2) and (P3), and the difficulty involved in balancing workloads. Factor 4, the number of components of each CT, affects the total time required to pick and place. In particular, placing times can be expected to dominate picking times and nozzle changing times. Factor 5, the nozzle type assigned to each CT, affects the number of times the head must undertake the inefficient operation of changing nozzles and affects the workload balance that (P4) can achieve. Finally, Factor 6, CT orientation, affects the number of efficient gang picks that can be made in the solution to (P2).

4.2. Test instances

We generate each test instance by specifying the number of CTs (Factor 3) and, for each CT, the number of constituent components (Factor 4), the nozzle type required (Factor 5) and the orientation required (Factor 6). Each of the $2^5 = 32$ unique selections of Factors 2–6 characterizes a test instance, which we solve using each of the three heuristics to solve (P1), assigning CTs to DHPMs (Factor 2) and to feeder slots on each machine. We use the (P1) solution as an input to (P2) and the (P2) solution — x_p^*, C_p, e_{cp}, o_{sp}, and $\bar{c}_{sp}$ for $p \in P^*$ — as an input to (P3). Each test instance that involved 1 (2) DHPM(s) required solution of 4 (8) rack problems (1)–(4), resulting in a total of 384 rack problems.

4.3. Test results

Tables 2–4 record overall measures of performance associated with H1, H2, and H3, respectively. Columns 1–7 describe the instance and columns 8–13 summarize test results. We solve a (P3) problem for each hmr combination separately, but, to conserve space, we report composite results for all rack problems associated with an instance. The acronyms that head the columns of Tables 2–4 are defined below:

Table 2
Summary of results for (P3) using Heuristic H1

#	F1 H#	F2 #M	F3 #CT	F4 #C/CT	F5 #NT	F6 θ	#SP Solved	#Impr Cols	#Entrd Cols	#B&B Nodes	Total RT (s)	Max RT (s)
1	1	1	32	10	1	1	671	519	309	96	38.98	25.48
2	1	1	32	10	1	2	198	186	95	4	11.11	3.34
3	1	1	32	10	2	1	225	212	99	4	10.57	3.34
4	1	1	32	10	2	2	186	169	89	4	10.11	2.8
5	1	1	32	[5, 15]	1	1	3039	2996	514	528	115.68	78
6	1	1	32	[5, 15]	1	2	244	243	30	4	10.79	3.19
7	1	1	32	[5, 15]	2	1	1312	1305	211	236	56.76	47.93
8	1	1	32	[5, 15]	2	2	240	240	34	4	11.19	3.19
9	1	1	64	10	1	1	224	224	46	4	11.95	3.19
10	1	1	64	10	1	2	224	224	52	4	11.66	2.93
11	1	1	64	10	2	1	223	220	49	4	11.39	2.96
12	1	1	64	10	2	2	224	220	52	4	11.22	2.93
13	1	1	64	[5, 15]	1	1	262	262	29	4	3.91	3.18
14	1	1	64	[5, 15]	1	2	262	262	28	4	3.91	3.18
15	1	1	64	[5, 15]	2	1	209	209	30	3	5.18	3.21
16	1	1	64	[5, 15]	2	2	284	284	30	4	6.57	3.35
17	1	2	32	10	1	1	280	271	159	73	13.48	3.51
18	1	2	32	10	1	2	416	351	220	64	10.67	2.85
19	1	2	32	10	2	1	393	350	191	121	6.4	2.33
20	1	2	32	10	2	2	1237	1089	543	199	18.76	10.54
21	1	2	32	[5, 15]	1	1	492	460	157	162	11.37	4.28
22	1	2	32	[5, 15]	1	2	698	662	149	195	24.16	13.33
23	1	2	32	[5, 15]	2	1	600	591	862	100	14.67	13.33
24	1	2	32	[5, 15]	2	2	1847	1809	198	532	45.57	42.16
25	1	2	64	10	1	1	594	579	191	40	3.77	1.53
26	1	2	64	10	1	2	1895	1820	396	380	40.37	33.13
27	1	2	64	10	2	1	688	662	200	96	13.37	6.09
28	1	2	64	10	2	2	1269	1233	290	208	30.27	22.19
29	1	2	64	[5, 15]	1	1	830	793	245	151	11.09	2.2
30	1	2	64	[5, 15]	1	2	1927	1898	408	343	35.71	31.33
31	1	2	64	[5, 15]	2	1	757	745	155	88	4.59	2.94
32	1	2	64	[5, 15]	2	2	726	685	134	128	7.96	2.14

Column	Acronym	Description
1	#	Instance number
2	F1 H#	Factor 1: heuristic number (i.e., H1, H2, or H3)
3	F2 #M	Factor 2: number of DHPMs
4	F3 #CT	Factor 3: number of CTs (i.e., 32 or 64)
5	F4 #C/CT	Factor 4: number of components per CT
6	F5: #NT	Factor 5: nozzle type assignment
7	F6: θ	Factor 6: CT orientation
8	#SP Solved	Number of sub-problems solved
9	#Impr Cols	Number of improving columns generated
10	#Entrd Cols	Number of improving columns entered into the master problem
11	#B&B Nodes	Number of B&B nodes required to optimize all rack problems
12	Total RT	Total run time (secs.) to prescribe optimal solutions to all rack problems
13	Max RT	Maximum run time (secs.) to solve any rack problem

The run times reported in columns 12 and 13 do not include the (negligible) time required to expand the sub-problem networks, a one-time process.

4.3.1. Factor effects on run time

First, we note that, although it is not a factor, the number of CTPCs prescribed by (P2) has a significant effect on run time. One sub-problem in the (P3) formulation represents each CTPC; a larger number of CTPCs thus results in

Table 3
Summary of results for (P3) using Heuristic H2

#	F1 H#	F2 #M	F3 #CT	F4 #C/CT	F5 #NT	F6 θ	#SP Solved	#Impr Cols	#Entrd Cols	#B&B Nodes	Total RT (s)	RT Max (s)
1	2	1	32	10	1	1	168	162	80	4	4.76	1.53
2	2	1	32	10	1	2	168	162	97	4	4.76	1.74
3	2	1	32	10	2	1	206	195	99	4	5.75	1.47
4	2	1	32	10	2	2	378	335	150	74	11.2	5.83
5	2	1	32	[5, 15]	1	1	389	346	133	74	11.6	5.83
6	2	1	32	[5, 15]	1	2	271	253	80	32	9.22	3.8
7	2	1	32	[5, 15]	2	1	284	277	74	24	8.42	2.96
8	2	1	32	[5, 15]	2	2	763	755	149	116	26.82	14.72
9	2	1	64	10	1	1	389	359	65	74	10.25	5.83
10	2	1	64	10	1	2	387	355	69	74	10.53	5.83
11	2	1	64	10	2	1	234	234	36	4	6.34	1.64
12	2	1	64	10	2	2	234	234	36	4	5.97	1.57
13	2	1	64	[5, 15]	1	1	429	414	71	56	16.47	9.59
14	2	1	64	[5, 15]	1	2	760	733	112	100	25.9	12.6
15	2	1	64	[5, 15]	2	1	261	261	30	4	7.47	2.71
16	2	1	64	[5, 15]	2	2	269	268	38	4	7.27	2.71
17	2	2	32	10	1	1	530	657	150	125	11	3.51
18	2	2	32	10	1	2	1750	1397	824	165	10.85	3.38
19	2	2	32	10	2	1	2794	2021	975	196	38.00	15.61
20	2	2	32	10	2	2	2788	2021	133	196	38.00	15.55
21	2	2	32	[5, 15]	1	1	676	644	238	233	14.4	7.58
22	2	2	32	[5, 15]	1	2	645	569	235	84	18.25	7.1
23	2	2	32	[5, 15]	2	1	3333	2537	1127	1599	47.3	16.1
24	2	2	32	[5, 15]	2	2	945	933	297	211	22.95	9.6
25	2	2	64	10	1	1	940	920	274	195	23.43	9.6
26	2	2	64	10	1	2	1474	1430	309	171	36.82	25.28
27	2	2	64	10	2	1	524	517	120	24	14.53	3.44
28	2	2	64	10	2	2	470	436	161	8	12.63	2.5
29	2	2	64	[5, 15]	1	1	510	489	109	16	13.91	2.26
30	2	2	64	[5, 15]	1	2	569	561	137	46	15.98	3.72
31	2	2	64	[5, 15]	2	1	1639	1612	476	670	42.41	23.86
32	2	2	64	[5, 15]	2	2	890	885	193	128	7.96	2.14

more sub-problems that must be solved, increasing run time. For example, instances 17–24 each involve several rack problems for which (P2) prescribes only 1 CTPC. For these instances, (P3) uses a gang pick of four components on each of 10 placing steps. Instances 8–16 each entail more CTPCs but require low run times; H1 and H3 assign CTs so that each of these rack problems is solved at the root node and H2 allows half of them to be solved at their root nodes (column 11 records that 4 B&B nodes were used, one for each rack problem).

The summary measures in Tables 2–4 highlight the effect of each factor on run time. We compare the two levels of each factor by adding the run times for instances that involve each level. By comparing the two sums for each factor, we gain the following insights.

Factor 1 (heuristic H1, H2, or H3) has a significant effect on run time: average run time per (P3) instance resulting from H1, H2, and H3 is 19.5, 16.9, and 39.3 s, respectively. Thus, H2 leads to (P3) instances that can be solved in less run time, but that is not to say that H2 is preferred because the heuristics must also be judged relative to their effects on workload balancing and that issue is beyond the scope of this paper. In assigning CTs to feeder slots, H1–H3 place different emphasis on such attributes as nozzle-type requirement, orientation, and CT width. H2 apparently yields less challenging (P3) instances. We conclude that our approach for solving (P3) is robust in that it is easily able to solve instances that result from quite different logics used to assign CTs to feeder slots.

Level 2 of Factor 2, number of DHPMs, requires longer run time than level 1 for all three heuristics. The reason is that 2 DHPMs typically involve more CTPCs and, therefore, we solve more sub-problems at each master-problem iteration.

Table 4
Summary of results for (P3) using Heuristic H3

#	F1 H#	F2 #M	F3 #CT	F4 #C/CT	F5 #NT	F6 θ	#SP Solved	#Impr Cols	#Entrd Cols	#B&B Nodes	Total RT (s)	Max RT (s)
1	1	1	32	10	1	1	168	162	84	4	4.76	1.53
2	1	1	32	10	1	2	174	154	87	4	10.54	4.03
3	1	1	32	10	2	1	176	300	88	68	16.21	8.69
4	1	1	32	10	2	2	224	207	112	4	10.32	2.91
5	1	1	32	[5, 15]	1	1	219	219	73	4	9.97	2.81
6	1	1	32	[5, 15]	1	2	230	230	56	4	9.36	2.77
7	1	1	32	[5, 15]	2	1	200	376	80	44	18.61	10.88
8	1	1	32	[5, 15]	2	2	196	372	154	44	18.83	10.88
9	1	1	64	10	1	1	232	230	50	4	10.18	2.6
10	1	1	64	10	1	2	224	221	56	4	10.17	2.56
11	1	1	64	10	2	1	224	218	56	4	10.1	2.53
12	1	1	64	10	2	2	224	224	56	4	10.17	2.55
13	1	1	64	[5, 15]	1	1	240	240	56	4	10.74	3.12
14	1	1	64	[5, 15]	1	2	275	275	23	4	11.91	3.16
15	1	1	64	[5, 15]	2	1	263	263	30	4	11.67	3.74
16	1	1	64	[5, 15]	2	2	251	251	23	4	10.21	3.06
17	1	2	32	10	1	1	121	113	113	8	6.23	1.09
18	1	2	32	10	1	2	107	99	99	8	5.68	0.76
19	1	2	32	10	2	1	130	122	122	8	6.46	1.2
20	1	2	32	10	2	2	124	116	116	8	6.28	1.17
21	1	2	32	[5, 15]	1	1	194	176	176	43	9.4	2.06
22	1	2	32	[5, 15]	1	2	253	224	224	49	10.82	3.32
23	1	2	32	[5, 15]	2	1	173	157	157	20	8.29	2
24	1	2	32	[5, 15]	2	2	131	123	123	10	7.22	1.18
25	1	2	64	10	1	1	3155	2755	1763	803	159.89	89.8
26	1	2	64	10	1	2	1535	1242	768	78	54.99	18.49
27	1	2	64	10	2	1	4274	3640	2007	1736	359.00	39.93
28	1	2	64	10	2	2	2748	2304	1224	820	140.49	48.74
29	1	2	64	[5, 15]	1	1	4651	3899	1395	145	107.92	35.75
30	1	2	64	[5, 15]	1	2	720	621	347	37	25.24	4.66
31	1	2	64	[5, 15]	2	1	3208	2949	1158	598	82.84	30.24
32	1	2	64	[5, 15]	2	2	2215	1654	900	128	82.72	25.36

Levels 1 and 2 of Factor 3, number of CTs, have little effect on run time relative to H1 and H2. This is somewhat counterintuitive because one would expect a larger number of CTs to require more CTPCs. We note that this result may be affected by the fact that many instances involving 1 DHPM (e.g., 2, 4, 8–16, 33–35, 43–44, 47, and 48) run quickly because each of the rack problems solve at the root node. However, relative to H3, the number of CTs has a significant effect on run time, especially in the case of 2 DHPMs. Instances 89–96 have exceptionally high run times because they involve assigning 64 CTs to 2 DHPMs and there are more CTPCs per rack, leading to more sub-problems and, thus, longer run times.

Table 3 shows that the two levels of Factors 4–6 have the same effect on run time when H2 is used. However, the two levels have significantly different effects when H1 is used. Level 2 of Factor 4, number of components per CT, has a much more pronounced effect on run time than level 1 does (when H1 is used). The reason for this is that, for level 2, (P2) may prescribe more CTPCs, increasing the number of sub-problems and, thus, run time. A larger number of components has both positive and negative influences. On the negative side, more components require more decisions, increasing run time. On the positive side, more components provide more opportunities to select good combinations for each placing step. Doubtlessly, these two influences underlie results but it is difficult to distinguish (a priori) when one will dominate the other. Level 1 of Factor 5, nozzle type assigned to each CT, has a somewhat stronger influence on run time than level 2 does (when H1 is used). Problem (P3), by itself, appears to provide no obvious reason for this difference, which we conclude follows from the logic that H1 and H2 use to assign CTs to feeder slots and the resulting differences in the nature of (P3) instances. We do expect, however, that Factor 5 would have a significant

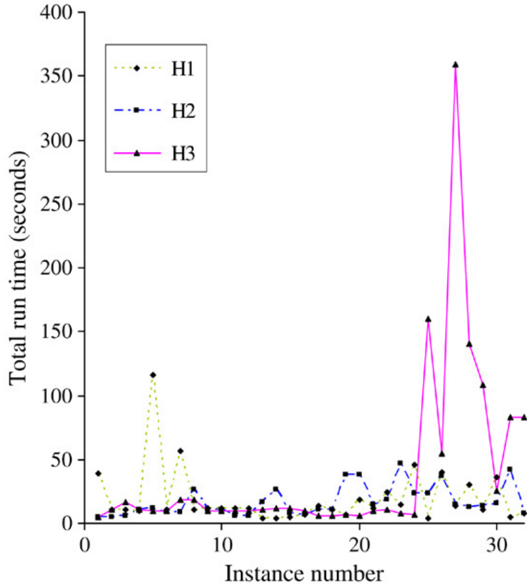

Fig. 3. Run time vs instance number and number of CTs for H1, H2, and H3.

effect on problem (P4) but that is beyond the scope of this paper. Factor 6, orientation requirement, does not have a significant influence on run time (for H1 and H2), although H1 takes somewhat longer to solve level 1 instances. In contrast, H3 poses (P3) problems that are highly sensitive to Factors 4, 5, and 6: level 1 of Factor 4, level 2 of Factor 5, and level 1 of Factor 6 each required significantly more run time than its respective complementary level. The logic used to assign CTs to feeder slots clearly has a significant effect on the structure of the (P3) problems that result.

Fig. 3 compares the run times that result from H1–H3 relative to the instance number and the number of CTs. H1 requires a longer-than-average run time to solve instance 5 because it requires a large number of sub-problems to be solved (3039) and a large number of B&B nodes (528). Similarly, H3 requires lengthy times to solve instances 25, 27, 28, and 29, increasing its average run time per instance. Overall, run times required to optimize (P3) instances are rather small. This suggests that it is relatively easy to identify a good combination of components for each placing step.

4.3.2. Overall performance measures

Columns 8–10 in Tables 2–4 demonstrate the performance of the column generation process. The most striking result is that the number of improving columns is almost as large as the number of sub-problems solved. This results because it is nearly always possible to select a set of individual components that form an improving column (i.e., a placing step). Even though a column in the current basis involves the placement of an individual component, no constraint prevents that component from being placed again (however, dual variable values would discourage this because it would not be optimal to place a component more than once). The number of columns entered, however, is much smaller because only one column is entered on each iteration. On the last iteration, which detects an optimal solution, all sub-problems are solved but no improving column is identified. Column 8 does not count this last round in reporting the number of sub-problems solved. As a result, columns 8 and 9 report the same number for several instances (e.g., 9, 10, 13–15).

Run time and the number of B&B nodes increase with the number of sub-problems and as one might expect. Finally, we note that the maximum run time associated with an instance typically dominates the run time for the set of associated rack problems.

4.3.3. Rack problems

Tables 5–7 provide detailed measures associated with individual rack problems and are headed by the following acronyms:

Table 5
Results for individual rack problems using heuristic H1

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
1	0	11.0	84	620	645	4314	2	2	0.2	84	620	629	4284	2
	1	0.0	84	620	626	4194	2	3	5.6	344	1970	1990	1260	7
2	0	0.0	104	620	623	4154	2	2	0.0	104	620	636	4254	2
	1	0.0	104	620	631	4155	2	3	0.0	104	620	629	4224	2
3	0	0.0	136	730	739	4226	3	2	0.0	104	620	636	4254	2
	1	0.0	104	620	631	4155	2	3	0.0	104	620	638	4214	2
4	0	0.0	104	620	636	4224	2	2	0.0	104	620	627	4185	2
	1	0.0	104	620	617	4094	2	3	0.0	104	620	615	3994	2
5	0	10.9	172	1112	1127	8185	4	2	0.0	428	3165	3149	2539	7
	1	5.4	254	1407	1434	7998	6	3	0.0	322	1924	1900	1307	6
6	0	0.0	431	2978	2964	2239	8	2	0.0	428	3165	3149	2539	7
	1	0.0	458	2938	2936	2123	8	3	0.0	273	1778	1802	1146	6
7	0	0.0	331	2281	2239	1739	6	2	0.0	428	3165	3149	2539	7
	1	0.0	458	2938	2936	2123	8	3	1.0	305	1705	1697	1100	6
8	0	0.0	454	2982	2970	2101	8	2	0.0	428	3165	3149	2539	7
	1	0.0	458	2938	2936	2123	8	3	0.0	195	1381	1377	9925	4
9	0	0.0	208	1240	1269	8408	4	2	0.0	428	3165	3149	2539	7
	1	0.0	208	1240	1261	8368	4	3	0.0	208	1240	1270	8408	4
10	0	0.0	208	1240	1260	8369	4	2	0.0	208	1240	1263	8379	4
	1	0.0	208	1240	1278	8518	4	3	0.0	208	1240	1268	8448	4
11	0	0.0	208	1240	1255	8370	4	2	0.0	208	1240	1260	8408	4
	1	0.0	208	1240	1266	8418	4	3	0.0	250	1450	1469	9500	5
12	0	0.0	208	1240	1268	8458	4	2	0.0	208	1240	1242	8250	4
	1	0.0	208	1240	1266	8418	4	3	0.0	208	1240	1253	8359	4
13	0	0.0	208	1240	1268	8458	4	2	0.0	829	6349	6379	5314	13
	1	0.0	642	3679	3727	2292	13	3	0.0	592	3278	3295	2003	13
14	0	0.0	208	1240	1268	8458	4	2	0.0	829	6349	6379	5314	13
	1	0.0	797	5629	5672	4181	14	3	0.0	592	3278	3295	2003	13
15	0	0.0	208	1240	1268	8458	4	2	0.0	797	5315	5307	4076	13
	1	0.0	506	2668	2700	1466	12	3	0.0	790	5380	5376	3993	14
16	0	0.0	208	1240	1268	8458	4	2	0.0	675	4576	4621	3425	12
	1	0.0	506	2668	2700	1466	12	3	0.0	958	5719	5748	4158	17
17	0	11.0	104	620	643	4294	2	4	0.1	42	310	306	2032	1
	1	11.0	104	620	643	4294	2	5	0.0	84	620	646	4324	2
	2	0.0	42	310	320	2132	1	6	0.0	42	310	306	2052	1
	3	0.0	42	310	320	2142	1	7	0.0	84	620	636	4244	2
18	0	0.0	42	310	320	2132	1	4	0.0	94	520	522	3065	2
	1	0.0	104	620	643	4294	2	5	0.1	42	310	306	2032	1
	2	0.0	74	520	540	3264	2	6	0.0	94	520	529	3175	2
	3	27.4	42	310	320	2142	1	7	0.0	94	520	521	3164	2
19	0	15.7	66	330	337	1236	3	4	0.0	94	520	517	3184	2
	1	15.7	119	18	14	3000	3	5	0.0	42	310	306	2032	1
	2	0.0	74	520	540	3264	2	6	15.7	94	520	529	3175	2
	3	0.0	94	520	534	3234	2	7	0.0	94	520	522	3175	2
20	0	0.0	66	330	337	1236	3	4	0.0	94	520	534	3234	2
	1	25.5	156	910	940	6108	3	5	0.0	84	620	624	4104	2
	2	0.0	74	520	540	3264	2	6	0.0	94	520	529	3175	2
	3	0.0	94	520	534	3234	2	7	0.0	94	520	522	3175	2
21	0	12.2	172	1112	1127	8185	4	4	20.3	62	690	693	6947	1
	1	9.2	113	868	884	5884	3	5	0.0	212	1464	1467	1119	4
	2	21.4	54	520	534	4630	1	6	11.8	54	520	534	4630	1
	3	19.1	58	602	604	5630	1	7	0.0	309	2130	2126	1476	6
22	0	1.9	182	1267	1251	8841	4	4	0.0	62	690	693	6947	1
	1	0.0	113	868	884	5884	3	5	8.6	212	1576	1552	1224	4
	2	0.0	54	520	534	4630	1	6	0.0	54	520	534	4630	1

Table 5 (*continued*)

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
	3	9.2	113	868	884	5884	3	7	2.9	280	2045	2042	1675	5
23	0	0.0	26	121	122	4780	1	4	6.7	222	1559	1548	1139	4
	1	0.0	113	868	884	5884	3	5	0.0	45	293	291	1763	1
	2	0.0	373	2704	2714	2176	6	6	0.0	34	253	250	1263	1
	3	0.0	113	868	884	5884	3	7	0.0	280	2045	2042	1675	5
24	0	2.2	26	121	122	4780	1	4	0.0	222	1559	1548	1139	4
	1	4.2	113	868	884	5884	3	5	3.2	45	293	291	1763	1
	2	0.0	373	2704	2714	2176	6	6	0.0	34	253	250	1263	1
	3	0.0	113	868	884	5884	3	7	2.4	280	2045	2042	1675	5
25	0	6.4	26	121	122	4780	1	4	0.0	198	1140	1164	7438	4
	1	0.0	113	868	884	5884	3	5	0.0	198	1140	1164	7438	4
	2	0.0	198	1140	1164	7438	4	6	0.0	198	1140	1160	7408	4
	3	0.0	198	1140	1164	7438	4	7	0.0	198	1140	1167	7469	4
26	0	2.3	250	1450	1472	9461	5	4	4.8	198	1140	1164	7438	4
	1	0.0	136	730	741	4267	3	5	6.2	198	1140	1164	7438	4
	2	0.0	250	1450	1448	9212	5	6	0.0	354	2070	2139	1382	7
	3	0.0	146	830	840	5306	3	7	0.0	250	1450	1484	9560	5
27	0	5.3	188	1040	1053	6329	4	4	0.0	146	830	822	5276	3
	1	0.0	136	730	741	4267	3	5	0.0	198	1140	1164	7438	4
	2	3.7	146	830	845	5267	3	6	0.0	136	730	734	4249	3
	3	0.0	354	2070	2101	1368	7	7	0.0	146	830	845	5326	3
28	0	0.0	146	830	855	5376	3	4	0.0	146	830	831	5148	3
	1	0.0	198	1140	1144	7258	4	5	5.5	250	1450	1459	9312	5
	2	0.0	146	830	847	5316	3	6	0.0	136	730	734	4249	3
	3	0.0	188	1040	1055	6348	4	7	0.0	146	830	845	5326	3
29	0	1.4	146	830	855	5376	3	4	1.2	192	1555	1540	1352	3
	1	2.3	198	1140	1144	7258	4	5	2.1	250	1450	1459	9312	5
	2	0.0	146	830	847	5316	3	6	0.0	136	730	734	4249	3
	3	0.0	188	1040	1055	6348	4	7	0.0	146	830	845	5326	3
30	0	2.2	278	1974	1989	1361	6	4	6.6	192	1555	1540	1352	3
	1	0.0	198	1140	1144	7258	4	5	3.4	250	1450	1459	9312	5
	2	0.0	359	2458	2475	1814	7	6	0.0	136	730	734	4249	3
	3	7.2	188	1040	1055	6348	4	7	2.4	146	830	845	5326	3
31	0	0.0	278	1974	1989	1361	6	4	4.1	238	1499	1504	1109	4
	1	0.0	331	2253	2258	1707	6	5	0.0	222	1772	1744	1334	4
	2	0.0	433	2359	2402	1615	8	6	0.0	254	1793	1820	1263	5
	3	0.0	314	2089	2088	1504	6	7	6.5	201	1261	1244	9176	4
32	0	0.0	278	1974	1989	1361	6	4	0.0	238	1499	1504	1109	4
	1	2.4	331	2253	2258	1707	6	5	4.2	222	1772	1744	1334	4
	2	0.0	307	2028	2025	1405	6	6	0.0	254	1793	1820	1263	5
	3	2.2	314	2089	2088	1504	6	7	0.0	348	2587	2556	1866	7

Columns		Acronym	Description
1		#	Instance number
2	9	R	Rack number
3	10	Δ	$\%GAP = 100(Z_{IP} - Z_{LP})/Z_{LP}$
4	11	#NS	Number of nodes in all sub-problems networks
5	12	#AS	Number of arcs in all sub-problem networks
6	13	#NE	Number of nodes in all expanded networks
7	14	#AE	Number of arcs in all expanded networks
8	15	#S	Number of sub-problems (and CTPCs)

To conserve space, we record results for half of the rack problems for each instance in columns 2–8 and the other half in columns 9–15. For the 1 DHPM case, columns 2–8 (9–15) give results for the two racks associated with head 1 (2). For the 2 DHPM case, columns 2–8 (9–15) give results for the four racks associated with DHPM 1 (2). We provide a blank row just before instance 17 to separate the 1 and 2 DHPM cases. Columns 3 and 10 give $\Delta = \%GAP$

Table 6
Results for individual rack problems using heuristic H2

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
1	0	0.0	104	620	625	4144	2	2	0.0	104	620	628	4125	2
	1	0.0	104	620	628	4204	2	3	0.0	104	620	635	4204	2
2	0	0.0	104	620	636	4314	2	2	0.0	104	620	628	4125	2
	1	0.0	104	620	628	4204	2	3	0.0	104	620	627	4214	2
3	0	0.0	104	620	624	4115	2	2	0.0	104	620	632	4173	2
	1	0.0	104	620	629	4106	2	3	0.0	104	620	637	4235	2
4	0	0.0	104	620	625	4175	2	2	0.0	104	620	631	4185	2
	1	0.0	104	620	642	4234	2	3	1.2	156	930	922	6128	3
5	0	0.0	373	2246	2263	1575	7	2	0.0	104	620	631	4185	2
	1	0.0	104	620	642	4234	2	3	1.2	156	930	922	6128	3
6	0	0.0	373	2246	2263	1575	7	2	0.0	282	1877	1865	1404	5
	1	1.2	106	717	719	5275	2	3	0.0	204	1242	1252	8438	4
7	0	0.0	373	2246	2263	1575	7	2	0.0	94	455	470	2520	2
	1	0.0	474	3286	3360	2610	8	3	0.0	147	891	903	6035	3
8	0	0.0	142	803	813	4580	3	2	2.4	314	2151	2196	1746	5
	1	0.0	392	2401	2405	1586	8	3	3.0	283	1746	1786	1255	5
9	0	0.0	208	1240	1264	8369	4	2	0.0	438	2490	2512	1581	9
	1	0.0	490	2800	2851	1789	10	3	3.0	283	1746	1786	1255	5
10	0	0.0	208	1240	1264	8369	4	2	0.0	552	3210	3280	2116	11
	1	0.0	260	1550	1578	1039	5	3	3.0	283	1746	1786	1255	5
11	0	0.0	208	1240	1262	8467	4	2	0.0	552	3210	3280	2116	11
	1	0.0	208	1240	1260	8410	4	3	0.0	510	3000	3063	2003	10
12	0	0.0	260	1550	1577	1053	5	2	0.0	490	2800	2820	1770	10
	1	0.0	208	1240	1260	8410	4	3	0.0	344	1970	1983	1261	7
13	0	0.0	260	1550	1577	1053	5	2	0.0	490	2800	2820	1770	10
	1	0.0	414	2621	2655	1971	7	3	3.0	171	1045	1058	7661	3
14	0	0.0	260	1550	1577	1053	5	2	2.4	490	2800	2820	1770	10
	1	0.0	315	2196	2168	1831	5	3	0.0	331	2025	2047	1380	6
15	0	0.0	856	5925	5881	4389	15	2	0.0	490	2800	2820	1770	10
	1	0.0	315	2196	2168	1831	5	3	0.0	331	2025	2047	1380	6
16	0	0.0	856	5925	5881	4389	15	2	0.0	490	2800	2820	1770	10
	1	0.0	315	2196	2168	1831	5	3	0.0	331	2025	2047	1380	6
17	0	0.0	856	5925	5881	4389	15	4	3.7	42	310	321	2152	1
	1	0.0	315	2196	2168	1831	5	5	0.0	42	310	322	2142	1
	2	0.0	490	2800	2820	1770	10	6	0.0	84	620	624	4084	2
	3	3.7	42	310	321	2152	1	7	0.0	42	310	320	2152	1
18	0	2.4	104	620	638	4274	2	4	0.0	42	310	321	2152	1
	1	0.0	156	930	946	6278	3	5	10.2	42	310	322	2142	1
	2	1.6	42	310	326	2172	1	6	2.4	42	310	321	2142	1
	3	7.8	84	620	636	4264	2	7	1.2	96	630	635	3326	3
19	0	20.1	126	930	961	6386	3	4	12.1	126	930	948	6356	3
	1	0.0	126	630	640	3286	3	5	0.0	44	220	212	224	2
	2	0.0	42	310	326	2172	1	6	0.1	42	310	321	2142	1
	3	0.0	126	930	941	6226	3	7	0.0	96	630	635	3326	3
20	0	13.1	126	930	966	6456	3	4	0.0	126	930	948	6356	3
	1	4.0	96	630	627	3226	3	5	0.0	44	220	212	2240	2
	2	32.6	126	930	946	6336	3	6	0.0	42	310	321	2142	1
	3	0.0	84	620	641	4254	2	7	0.0	96	630	639	3266	3
21	0	2.6	126	930	966	6456	3	4	2.4	30	154	154	737	1
	1	4.2	50	444	456	3662	1	5	1.8	169	1079	1058	7634	3
	2	0.0	211	1118	1140	7352	4	6	8.6	177	981	977	6781	3
	3	1.3	50	444	448	3590	1	7	9.4	50	444	428	3434	1
22	0	0.0	305	1793	1785	1306	5	4	0.0	97	499	504	2779	2
	1	0.0	221	1508	1512	1126	4	5	5.8	169	1079	1058	7634	3
	2	1.2	214	1345	1352	8926	4	6	0.0	245	1670	1663	1270	4
	3	2.6	34	200	207	1106	1	7	8.6	30	154	154	737	1

Table 6 (*continued*)

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
23	0	0.0	91	497	502	3092	2	4	0.0	97	499	504	2779	2
	1	3.2	82	503	514	2495	3	5	13.2	103	704	710	4195	3
	2	6.9	153	981	993	7316	3	6	4.5	32	26	32	32	3
	3	4.1	190	1485	1491	1252	3	7	2.1	30	154	154	7370	1
24	0	1.6	150	726	736	4504	3	4	0.0	97	499	504	2779	2
	1	2.9	82	442	449	2032	3	5	0.0	468	2790	2834	1881	9
	2	3.2	153	981	993	7316	3	6	0.0	245	1670	1663	1270	4
	3	7.8	190	1485	1491	1252	3	7	0.0	206	1537	1535	1339	3
25	0	0.0	208	1240	1270	8469	4	4	0.0	364	2170	2222	1470	7
	1	14.4	82	442	449	2032	3	5	0.0	468	2790	2834	1881	9
	2	1.2	153	981	993	7316	3	6	0.0	156	930	963	6396	3
	3	2.6	156	930	947	6386	3	7	12.2	156	930	945	6306	3
26	0	0.0	104	620	629	4144	2	4	0.0	364	2170	2222	1470	7
	1	2.5	82	442	449	2032	3	5	0.0	104	620	634	4205	2
	2	11.6	153	981	993	7316	3	6	3.6	364	2170	2218	1472	7
	3	0.0	364	2170	2195	1467	7	7	0.0	104	620	634	4254	2
27	0	0.0	208	1240	1261	8379	4	4	0.0	312	1860	1905	1272	6
	1	0.0	208	1240	1259	8409	4	5	0.0	208	1240	1270	8438	4
	2	0.0	104	620	629	4164	2	6	0.0	260	1550	1547	1032	5
	3	0.0	416	2480	2524	1685	8	7	4.8	208	1240	1279	8478	4
28	0	0.0	146	830	836	5236	3	4	0.0	104	620	636	4234	2
	1	0.0	208	1240	1268	8498	4	5	0.0	104	620	638	4275	2
	2	0.0	312	1860	1908	1268	6	6	0.0	104	620	623	4164	2
	3	0.0	146	830	842	5267	3	7	0.0	208	1240	1275	8428	4
29	0	0.0	465	2880	2868	1988	9	4	0.0	104	620	636	4234	2
	1	0.0	550	3853	3887	3063	9	5	0.0	414	2621	2658	1971	7
	2	0.0	312	1860	1908	1268	6	6	0.0	423	2414	2419	1550	9
	3	0.0	146	830	842	5267	3	7	4.8	172	1130	1130	8648	3
30	0	0.0	216	1170	1147	7574	4	4	0.0	98	512	530	3232	2
	1	0.0	550	3853	3887	3063	9	5	0.0	226	1448	1418	1061	4
	2	0.0	401	2603	2600	1780	8	6	0.0	317	2561	2552	1982	6
	3	0.0	146	830	842	5267	3	7	5.2	172	1130	1130	8648	3
31	0	0.0	320	1933	1897	1305	6	4	0.0	298	1854	1873	1333	5
	1	0.0	317	2590	2567	2220	5	5	0.0	569	3629	3666	2604	10
	2	0.0	401	2603	2600	1780	8	6	6.7	161	927	930	6080	3
	3	2.8	133	641	645	3647	3	7	0.0	367	2877	2858	2379	6
32	0	0.0	148	736	751	4460	3	4	0.0	230	1495	1473	1072	4
	1	7.8	317	2590	2567	2220	5	5	3.2	569	3629	3666	2604	10
	2	0.0	401	2603	2600	1780	8	6	0.0	161	927	930	6080	3
	3	8.4	169	1277	1275	9944	3	7	0.0	270	1540	1540	1038	5

for the rack problem, where Z_{LP} is the value of the optimal solution to the linear relaxation and Z_{IP} is the value of the optimal integer solution.

Δ is quite small for most rack problems, indicating that our model is tight. Approximately 71% of the rack problems have $\Delta = 0.0$ for H1–H3. On average (over the 192 rack problems), Δ is 1.9%, 1.65%, and 1.58% for H1, H2, and H3, respectively, so that H3 poses somewhat tighter rack problems than H2, and, in turn, H2 poses tighter rack problems than H1. For example, instances 13–16 each involve a large number of CTPCs (sub-problems) but have low run times because the Δ is small (see Tables 5–7) for instances with 1 DHPM (i.e., level 1). However, a few rack problems involve substantial gaps. Instances 24, 26, 28, and 30 each involve fewer CTPCs but have longer run times because Δ is typically large for at least one rack problem associated with each instance that involves 2 DHPMs (i.e., level 2), reflecting the fact that (P1) assigned more CTs to that rack. We conclude that the larger Δ values result from adding equalities (4) to the binary set-covering model (1), (2) and (4).

Instances 13–16 have appreciably more sub-problems with larger sub-problem networks because they represent cases in which (P2) prescribes a large number of CTPCs. On the other hand, (P2) prescribes only one CTPC for a number of rack problems that involve 2 DHPMs on which CTPCs can be dispersed so that each rack contains just a

Table 7
Results for individual rack problems using heuristic H3

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
1	0	0.0	104	620	626	4194	2	2	0.0	104	620	629	4115	2
	1	0.0	104	620	610	4115	2	3	0.0	104	620	634	4225	2
2	0	0.0	104	620	646	4284	2	2	0.0	104	620	625	4085	2
	1	0.0	104	620	636	4214	2	3	0.0	104	620	632	4246	2
3	0	0.0	104	620	627	4185	2	2	0.0	104	620	615	4094	2
	1	0.0	104	620	637	4215	2	3	0.0	104	620	635	4194	2
4	0	0.0	104	620	631	4224	2	2	0.0	104	620	627	4234	2
	1	0.0	104	620	636	4195	2	3	0.0	104	620	623	4064	2
5	0	0.0	104	620	631	4224	2	2	0.0	288	2118	2106	1661	5
	1	0.0	104	620	636	4195	2	3	0.0	309	2130	2126	1476	6
6	0	0.0	549	3585	3593	2606	10	2	0.0	394	2723	2704	2024	7
	1	0.0	104	620	636	4195	2	3	0.0	299	2025	2020	1613	5
7	0	0.0	549	3585	3593	2606	10	2	2.3	237	1502	1509	1180	4
	1	0.0	425	2858	2849	1992	8	3	0.0	237	1622	1614	1227	4
8	0	0.0	549	3585	3593	2606	10	2	4.3	237	1502	1509	1180	4
	1	0.0	324	2322	2317	1724	6	3	0.0	237	1622	1614	1227	4
9	0	0.0	416	2480	2512	1668	8	2	0.0	208	1240	1264	8388	4
	1	0.0	208	1240	1257	8379	4	3	0.0	208	1240	1265	8399	4
10	0	0.0	208	1240	1254	8260	4	2	0.0	208	1240	1253	8379	4
	1	0.0	208	1240	1262	8430	4	3	0.0	208	1240	1270	8478	4
11	0	0.0	208	1240	1246	8319	4	2	0.0	208	1240	1255	8300	4
	1	0.0	208	1240	1259	8329	4	3	0.0	208	1240	1257	8300	4
12	0	0.0	208	1240	1257	8312	4	2	0.0	208	1240	1269	8458	4
	1	0.0	208	1240	1256	8468	4	3	0.0	208	1240	1260	8419	4
13	0	0.0	208	1240	1257	8312	4	2	0.0	701	4626	4621	3451	12
	1	0.0	208	1240	1256	8468	4	3	0.0	208	1240	1260	8419	4
14	0	0.0	621	4561	4590	3315	12	2	0.0	692	4542	4588	3361	12
	1	0.0	208	1240	1256	8468	4	3	0.0	832	5640	5629	4100	15
15	0	0.0	621	4561	4590	3315	12	2	0.0	338	2099	2127	1392	7
	1	0.0	208	1240	1256	8468	4	3	0.0	746	5300	5342	4272	12
16	0	0.0	621	4561	4590	3315	12	2	0.0	338	2099	2127	1392	7
	1	0.0	208	1240	1256	8468	4	3	0.0	538	3240	3267	2227	10
17	0	0.0	42	310	313	2072	1	4	0.0	42	310	324	2152	1
	1	0.0	42	310	309	2032	1	5	0.0	42	310	308	2072	1
	2	0.0	42	310	304	1992	1	6	0.0	42	310	318	2092	1
	3	0.0	42	310	300	2062	1	7	0.0	42	310	313	2042	1
18	0	0.0	42	310	323	2152	1	4	0.0	42	310	303	1952	1
	1	0.0	42	310	323	2132	1	5	0.0	42	310	324	2152	1
	2	0.0	42	310	305	1992	1	6	0.0	42	310	315	2072	1
	3	0.0	42	310	318	2142	1	7	0.0	42	310	309	2042	1
19	0	0.0	42	310	314	2082	1	4	0.0	42	310	317	2142	1
	1	0.0	42	310	315	2122	1	5	0.0	42	310	299	1982	1
	2	0.0	42	310	324	2172	1	6	0.0	42	310	318	2102	1
	3	0.0	42	310	316	2102	1	7	0.0	42	310	314	2062	1
20	0	0.0	42	310	318	2122	1	4	0.0	42	310	320	2142	1
	1	0.0	42	310	315	2122	1	5	0.0	42	310	308	2092	1
	2	0.0	42	310	324	2172	1	6	0.0	42	310	311	2012	1
	3	0.0	42	310	314	2052	1	7	0.0	42	310	316	2062	1
21	0	3.2	79	559	573	3937	1	4	0.0	41	294	293	1876	1
	1	0.0	45	369	368	2659	1	5	0.0	54	520	517	4578	1
	2	1.6	89	673	681	5054	1	6	0.0	28	137	141	6040	1
	3	4.2	50	397	394	2814	1	7	0.0	46	379	378	2737	1
22	0	0.0	42	346	350	2342	1	4	0.0	50	460	463	3780	1
	1	0.0	42	346	350	2342	1	5	0.0	39	256	260	1570	1
	2	2.1	89	673	681	5054	1	6	11.2	86	628	626	4622	1
	3	6.5	99	927	929	7426	1	7	0.0	51	481	481	3995	1

Table 7 (*continued*)

#	R	Δ	#NS	#AS	#NE	#AE	#S	R	Δ	#NS	#AS	#NE	#AE	#S
23	0	0.0	39	186	187	8680	1	4	3.1	84	608	612	4382	1
	1	0.0	37	252	258	1503	1	5	0.0	47	351	359	2543	1
	2	4.3	38	260	262	1572	1	6	0.0	47	402	406	3007	1
	3	0.0	48	357	353	2380	1	7	0.0	44	285	287	1701	1
24	0	0.0	39	214	216	1329	1	4	0.0	33	198	204	8720	1
	1	0.0	37	252	258	1503	1	5	0.0	50	425	430	3412	1
	2	2.2	38	260	262	1572	1	6	0.0	45	354	355	2345	1
	3	0.0	48	357	353	2380	1	7	0.0	44	285	287	1701	1
25	0	0.0	84	620	632	4254	2	4	0.0	364	2170	2222	1470	7
	1	1.2	84	620	634	4144	1	5	5.2	84	620	630	4164	1
	2	6.5	84	620	628	4214	1	6	3.1	84	620	640	4254	1
	3	5.6	84	620	632	4234	1	7	6.7	84	620	636	4274	1
26	0	0.0	84	620	618	4054	1	4	1.1	364	2170	2222	1470	7
	1	0.0	84	620	630	4214	1	5	12.2	84	620	626	4204	1
	2	0.0	84	620	636	4254	1	6	16.7	84	620	640	4284	1
	3	11.3	84	620	636	4264	1	7	0.0	104	620	634	4254	2
27	0	0.0	84	620	629	4184	1	4	3.2	84	620	625	4164	1
	1	2.3	84	620	620	4154	1	5	3.1	84	620	635	4194	1
	2	2.1	126	930	948	6246	1	6	2.1	84	620	629	4124	1
	3	0.0	84	620	635	4264	1	7	17.2	126	930	953	6346	1
28	0	3.3	84	620	639	4234	1	4	3.1	84	620	632	4214	1
	1	0.0	84	620	628	4194	1	5	6.5	126	930	953	6396	1
	2	2.5	84	620	621	4214	1	6	2.4	126	930	944	6276	1
	3	0.0	126	930	953	6356	1	7	4.3	84	620	636	4244	1
29	0	0.0	465	2880	2868	1988	9	4	8.2	176	1382	1389	9404	1
	1	13.2	100	934	924	7376	1	5	3.2	85	654	648	4390	1
	2	12.2	72	488	499	2876	1	6	6.3	123	882	891	5827	1
	3	9.1	181	1416	1407	1058	1	7	4.1	91	764	770	5632	1
30	0	1.1	80	611	618	3929	1	4	0.0	98	512	530	3232	2
	1	2.6	77	561	564	3466	1	5	0.0	79	606	607	3764	1
	2	2.8	75	498	502	3028	1	6	8.1	94	835	824	6349	1
	3	12.1	88	707	712	4901	1	7	5.1	78	575	592	3760	1
31	0	5.2	126	882	884	6250	1	4	0.0	77	500	504	2958	1
	1	4.6	75	522	523	3145	1	5	0.0	71	489	496	2788	1
	2	6.5	143	1356	1360	1063	1	6	3.1	161	927	930	6080	3
	3	11.2	85	623	632	4246	1	7	2.2	94	746	758	5487	1
32	0	6.1	78	531	534	3227	1	4	0.0	76	549	553	3263	1
	1	0.0	75	574	575	3318	1	5	0.0	78	492	496	3117	1
	2	7.6	84	649	663	4550	1	6	2.1	161	927	930	6080	3
	3	5.6	127	923	917	6425	1	4	0.0	76	549	553	3263	1

few CTs. In particular, (P1) assigns a single CT to each of a number of racks, so each such rack problem requires just one sub-problem.

4.3.4. Statistical analysis

We use Microsoft Excel 2003 to conduct a correlation analysis relative to H1–H3 (individually) with the goal of identifying which factors and two-way interactions of these factors are most related to response (i.e., run time). Because our data would give only 4 replications for three-way interactions (and even fewer for more interactions), we analyze only individual factors (F2, ..., F5) and their two-way interactions for which our data give 16 and 8 replications, respectively. Table 8 presents correlation coefficients associated with each of these factors and two-way interaction terms.

For H1, F3 (number of CTs) has the largest correlation (in absolute value) with run time; and the F3F4 (numbers of CTs and components in each CT) interaction, the second largest (in absolute value). This result is consistent with the logic of H1 and the observations we report above. Both correlation coefficients are negative, indicating that run time decreases as the numbers of CTs and components increase. Although this may appear to counter intuition, it is explained by the fact that the logic of H1 distributes CTs unevenly among racks for instances that involve fewer CTs,

Table 8
Coefficients of correlation between run times and terms

		Terms														
		F2	F3	F4	F5	F6	F2F3	F2F4	F2F5	F2F6	F3F4	F3F5	F3F6	F4F5	F4F6	F5F6
H1	r	−0.06	−0.28	−0.14	0.17	−0.06	−0.17	−0.11	0.10	0.02	−0.28	−0.04	−0.17	0.09	−0.11	0.08
	Rank	13	1	6	5	12	4	8	9	15	2	14	3	10	7	11
H2	r	0.52	−0.07	0.14	0.17	−0.03	0.29	0.24	0.53	0.29	0.09	−0.03	−0.07	0.20	0.08	0.06
	Rank	2	12	8	7	14	4	5	1	3	9	15	11	6	10	13
H3	r	0.40	0.42	0.15	−0.17	−0.18	0.66	0.41	−0.09	0.11	0.42	−0.08	0.12	−0.14	−0.02	−0.18
	Rank	5	3	9	8	6	1	4	13	12	2	14	11	10	15	7

r: correlation coefficient.

typically causing one rack problem to require a substantially longer run time that dominates the run times required by less heavily loaded racks. On average, the total run time (i.e., for all rack problems) for such an instance exceeds the total run time required by an instance for which CTs are distributed evenly among racks, even if the instance involves more CTs.

For H2, the F2F5 (number of DHPMs and theta distribution) interaction term has the largest correlation with run time; and F2 (number of DHPMs), the second largest. This result is consistent with the logic of H2, which forms super groups based on the theta distribution. The correlation coefficients are positive, indicating that run time increases with the number of DHPMs. One would expect this result (assuming that CTs are distributed evenly among racks) because each additional rack requires run time.

For H3, the F2F3 (numbers of DHPMs and CTs) interaction term has the largest correlation with run time. Because the logic of H3 distributes components evenly among all racks, it follows that the numbers of DHPMs and CTs would be most influential in determining the run time. Factors F2 (numbers of DHPMs), F3 (number of CTs), and two-way interactions F3F4 (number of CTs and components of each CT) and F2F4 (numbers of DHPMs and components of each CT) have similar, relatively large coefficients of correlation. As expected, these correlation coefficients are all positive, indicating that run time increases with each of these factors and two-way interactions.

5. Conclusions and recommendations for future research

This paper achieves its purpose, presenting a novel model for prescribing the placement operations of a DHPM. It makes research contributions by achieving its objectives. In particular, our model is able to address the broad range of relevant practical considerations. Objective (2) has led to an optimizing method that can solve problems of practical size and scope in run times that will facilitate implementation by process planners to promote the efficiencies of their assembly systems. Our computational tests fulfill their purposes by providing considerable insight into the influence that relevant factors have on run time, the robustness of our approach when different logics are used to assign CTs to feeder slots, the influence that the tightness of our model has on run time, and the solvability of our model. Because no prior study has addressed the DHPM, no alternative methods exist to compare with ours. Future studies can use our benchmark results to evaluate alternative methods vis-à-vis ours.

This paper focuses exclusively on placement operations and can be used in a standalone mode to optimize just placement operations in applications with such a need. A fertile opportunity for future research is to integrate our solution method with others that optimize (P1), prescribing the assignment of each CT to a DHPM, to a head on that machine, and to a feeder slot on a rack associated with that head; as well as (P2), prescribing picking steps; and (P4) prescribing the sequence of pick/place steps for each head, which determines the time required for nozzle changes. This expanded capability would address the ultimate goal of this research, balancing workloads assigned to heads.

It may be interesting to eliminate (A2) to study DHPMs that are capable of picking with one spindle sequence, then placing with a different one. We conjecture, however, that this change would not offer great potential for improving productivity because identifying good combinations of components to place on each round is not difficult (i.e., (P3) run times are low). We expect that the potential would increase as the number of components per CT reduces, placing a premium on identifying the best combinations of components on each placing step. Using different sequences to pick and place would require the DHPM controller to be redesigned. All possible placement sequences could be evaluated by defining, in association with each CTPC, up to $4! = 24$ networks as described in Section 3, one for each sequence in which spindles can place components. The column generation approach would then prescribe the optimal placement sequence as well. Gang picks are special cases in that a CTPC comprising two gang picks, each of 2 components, would require 4 networks to represent all possible spindle-placement sequences and an augmented gang pick of 3 (2) components would require $3! = 6$ ($2! = 2$) networks to prescribe placement sequence. These additional networks would add to the solution time required by each iteration, but each associated sub-problem can be solved quickly so that this approach appears to be practical. In fact, Wilhelm et al. [38] utilize 4! networks to represent all of the sequences in which spindles can pick. Our successful results recommend that column generation be investigated to prescribe process plans for other types of placement machines as well. Our research continues along these lines.

Acknowledgements

This material is based in part upon work supported by the Texas Advanced Technology Program on grant numbers ATP-036327-140 and 000512-0248-2001 and by the National Science Foundation on grant numbers DMI-9500211 and DMI-0217265. The authors gratefully acknowledge industrial collaborators Luis Giraldo, Gordon O'Hara, Richard Evans and Shelli Farr, who provided information that allowed us to formulate realistic models and devise test instances of realistic size and scope. We appreciate the efforts of Ashu Godse and Remi Salam, who contributed to the computer program used to implement the solution approach, and of several undergraduate research assistants: Jeffrey Doerr, Brian Kruppa, Christina McCann, Ryan Bowling, Kimberly Collier and Troy Schwartz. We also acknowledge two anonymous referees and an associated editor, whose comments allowed us to strengthen an earlier version of this paper.

References

[1] J. Ahmadi, R. Ahmadi, H. Matsuo, D. Tirupati, Component fixture positioning/sequencing for printed circuit board assembly with concurrent operations, Operations Research 43 (3) (1995) 444–457.
[2] J. Ahmadi, S. Grotzinger, D. Johnson, Component allocation and partitioning for a dual delivery placement machine, Operations Research 36 (1988) 176–191.
[3] J. Ahmadi, S. Grotzinger, D. Johnson, Emulating of concurrency in circuit card assembly system, International Journal of Flexible Manufacturing Systems 3 (1990) 45–70.
[4] R. Ahmadi, P. Kouvelis, Staging problem of a dual delivery pick-and-place machine in PCB assembly, Operations Research 42 (1) (1994) 81–91.
[5] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice-Hall, Upper Saddle River, NJ, 1993.
[6] K. Altinkemer, B. Kazaz, M. Köksalan, H. Moskowitz, Optimization of printed circuit board manufacturing: Integrated modeling and algorithms, European Journal of Operational Research 124 (2000) 409–421.
[7] S. Avci, M.S. Akturk, Tool magazine arrangement and operations sequencing on CNC machines, Computers & Operations Research 23 (11) (1996) 1069–1081.
[8] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, Linear Programming and Network Flows, 3rd ed., John Wiley & Sons, New York, 2005.
[9] J.E. Beasley, N. Christofides, An algorithm for the resource constrained shortest path problem, Networks 19 (1989) 379–394.
[10] J. Bramel, D. Simchi-Levi, On the effectiveness of set covering formulations for the vehicle routing problem with time windows, Operations Research 45 (2) (1997) 295–301.
[11] E.K. Burke, P.I. Cowling, R. Keuthen, New models and heuristics for component placement in printed circuit board assembly, in: Proceedings: IEEE International Conference on Information Intelligence, March 31–April 3, Rockville, MD, 2003, pp. 133–140.
[12] D. Chan, D. Mercer, IC insertion: An application of the traveling salesman problem, International Journal of Production Research 27 (10) (1989) 1837–1841.
[13] N. Choudhry, W.E. Wilhelm, B.R.T. Vasudeva, J. Gott, N. Khotekar, Process planning for circuit card assembly on a series of dual head placement machines, Working Paper, Department of Industrial and Systems Engineering, Texas A&M University, 2005.
[14] Y. Crama, O.E. Flippo, J. van de Klundert, F.C.R. 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.
[15] Y. Crama, O.E. Flippo, J. van de Klundert, F.C.R. Spieksma, The component retrieval problem in PCB assembly, The International Journal of Flexible Manufacturing Systems 8 (1996) 287–312.
[16] Y. Crama, J. van de Klundert, F.C.R. Spieksma, Production planning problems in printed circuit board assembly, GEMME 9925, University of Liege, Liege, Belgium, 1999.
[17] Y. Crama, A.W.J. Kolen, A.G. Oerlemans, F.C.R. Spieksma, Throughput rate optimization in the automated assembly printed circuit boards, Annals of Operations Research 26 (1990) 455–480.
[18] M. Desrochers, J. Desrosiers, M. Solomon, A new optimizing algorithm for the vehicle routing problem with time windows, Operations Research 40 (2) (1992) 342–354.
[19] M. Desrochers, F. Soumis, A generalized permanent labeling algorithm for the shortest path problem with time windows, INFOR 26 (3) (1988) 191–211.

[20] M. Desrochers, F. Soumis, A re-optimization algorithm for the shortest path problem with time windows, European Journal of Operational Research 35 (1988) 242–254.
[21] J. Desrosiers, Y. Dumas, M.M. Solomon, F. Soumis, Time constrained routing and scheduling, in: Handbooks in Operations Research and Management Science, in: M.E. Ball, T.L. Magnanti, C. Monma, G.L. Nemhauser (Eds.), Network Routing, vol. 8, North-Holland, Amsterdam, 1996, pp. 35–140.
[22] J. Desrosiers, F. Soumis, M. Desrochers, Routing with time windows by column generation, Networks 14 (1984) 545–565.
[23] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem, Operations Research 9 (1961) 849–859.
[24] P.C. Gilmore, R.E. Gomory, A linear programming approach to the cutting stock problem — Part II, Operations Research 11 (6) (1963) 863–888.
[25] J. Gott, W.E. Wilhelm, Dual-head placement machine for circuit card assembly optimization, Working Paper, Department of Industrial Engineering, Texas A&M University, 1999.
[26] S. Grotzinger, Feeder assignment models for concurrent placement machines, IIE Transactions 24 (4) (1992) 31–46.
[27] G.Y. Handler, I. Zang, Dual algorithm for the constrained shortest path problem, Networks 10 (1980) 293–310.
[28] J. Hong, W. Lee, S. Lee, B. Lee, Y. Lee, An efficient production planning algorithm for multi-head surface mounting machines using biological immune algorithm, International Journal of Fuzzy Systems 2 (1) (2000) 45–54.
[29] B. Jaumard, F. Semet, T. Vovor, Two-phased resource constrained shortest path algorithm for acyclic graphs, Les Cahiers du GERAD, Ecole des Hautes Etudes Commerciales, Montreal, Quebec, Canada, 1996.
[30] P. Ji, Y.F. Wan, Planning for printed circuit board assembly: State-of-the-art review, International Journal of Computer Applications in Technology 14 (1) (2001) 1–9.
[31] B. Kazaz, K. Altinkemer, Optimization of printed circuit board manufacturing: A multi-feeder assignment approach, Working Paper Series 1999, Loyola University, Chicago, IL, 1999.
[32] G. LaPorte, J.J. Salazar-Gonzales, F. Semet, Exact algorithms for the job sequencing and tool switching problem, IIE Transactions 36 (1) (2004) 37–45.
[33] 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.
[34] S.Y. Nof, W.E. Wilhelm, H. Warnecke, Industrial Assembly, Chapman and Hall, 1997.
[35] F. Spieksma, Y. Crama, J. van de Klundert, A bibliography of material on the assembly of printed circuit boards. http://www.math.unimaas.nl/PERSONAL/fritss/Bibliography/bibliogr.htm, 2000.
[36] W. Wang, P.C. Nelson, T.M. Tirpak, Optimization of high-speed multi-station smt placement machines using evolutionary algorithms, IIE Transactions on Electronics packaging Manufacturing (1999) 1–10.
[37] W.E. Wilhelm, A technical review of column generation in integer programming, Optimization and Engineering 2 (2) (2001) 159–200.
[38] W.E. Wilhelm, I. Arambula, N.N. Choudhry, A model to optimize picking operations on dual-head placement machines, IEEE Transactions on Automation Science and Engineering 3 (1) (2006) 1–15.
[39] W.E. Wilhelm, P. Damodaran, J. Li, Prescribing the content and timing of product upgrades, IIE Transactions 35 (7) (2003) 647–664.
[40] W.E. Wilhelm, P. Kiatchai, Circuit card assembly on tandem turret-type placement machines, IIE Transactions 35 (7) (2003) 627–646.