INT. J. PROD. RES., 1993, VOL. 31, NO. 12, 2835–2856

An evaluation of heuristics for allocating components to kits in small-lot, multi-echelon assembly systems

J. F. CHEN† and W. E. WILHELM‡§

The kitting problem in multi-echelon assembly systems is to allocate on-hand stock and anticipated future deliveries to kits to minimize total cost, consisting of job earliness, job tardiness, and in-process holding costs. This paper describes the kitting problem and compares the performance of three heuristics, two that are commonly used in industry and a new one, to resolve it. Computational experience demonstrates that the new heuristic outperforms the others, finding the optimal solution in 16 of 24 test problems and averaging just 0·6% above the optimum for the 24 problems. It is expected that the new heuristic will find application in large-scale problems encountered in industry. Solutions will facilitate time-managed flow control, prescribing kitting decisions that promote cost-effective performance to schedule.

1. Introduction

Before assembly operations begin, small-lot assembly systems commonly gather all required components in a kit, then 'launch' (i.e. 'release') it to initiate production. Operating according to the logic of material requirements planning (MRP), each subassembly is assigned a due date to be kitted, planning adequate time to complete the end-product by its due date. The entire assembly schedule thus relies upon the ability to compose each kit on its due date. This is often impossible, however, because there may not be sufficient stock on hand to meet scheduled requirements. Even though MRP sets vendor due dates to support the assembly schedule, unexpected events affect deliveries, disrupting kitting operations and, therefore, the assembly schedule.

This problem is particularly prevalent in the electronics industry. For example, each circuit card may require several hundred components, so that a kit for a lot of circuit cards may require several thousand components of various types. Some components, such as integrated circuits, may be subject to yield loss; others, to transportation delays from offshore vendors. Safety stocks might be used to hedge against these problems, but the high cost of electronic components makes this an unattractive alternative.

In multi-echelon assembly systems, kits may require components from vendors as well as subassemblies produced upstream. Thus, a disruption at one echelon may readily propagate to other echelons, affecting the co-ordinated flow of materials and the assembly schedule. For example, consider the three-echelon system of subassemblies depicted in Fig. 1. At the top echelon, circuit cards are assembled from

Revision received January 1993.

† Feng Chi University, Tai Chung, Taiwan.
‡ Department of Industrial Engineering, Texas A&M University, College Station, TX 77843-3131, USA.
§ To whom correspondence should be addressed.

0020-7543/93 $10·00 © 1993 Taylor & Francis Ltd.

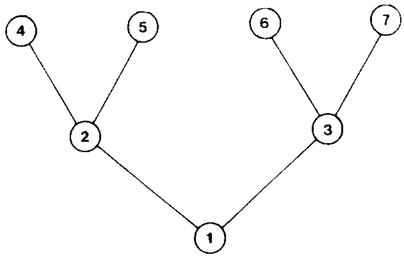

Figure 1. A three-echelon binary assembly tree.

vendor-supplied components. The circuit cards are then assembled into subassemblies which are, in turn, assembled into an end-product such as a computer. A kitting delay at any echelon can affect material flow, increasing the cost of in-process inventories (e.g. some circuit cards may have to wait for the completion of others that are used in the same subassembly) and leading to poor schedule performance.

The kitting problem in multi-echelon assembly systems requires on-hand stock and promised deliveries to be allocated to kits to minimize total cost, including job earliness, job tardiness, and the cost of holding in-process inventories, which may result from poor co-ordination of material flows. Part availabilities present important limitations that must be observed. In addition, shop capacities must be considered so that bottlenecks are not created by kitting an inappropriate set of subassemblies. The primary decision to be made is the day on which each kit is to be composed (we assume that each completed kit is immediately launched into production).

The *purpose* of this paper is to evaluate the efficacy of three heuristics in resolving the kitting problem for multi-echelon, small-lot assembly systems. More specifically, *research objectives* are (1) a new heuristic which exploits problem structure to prescribe solutions, and (2) empirical evaluation of the new heuristic in comparison with two methods commonly used to make kitting decisions in industry.

The kitting problem is prevalent in industry, but has been the focus of relatively little research. The most closely related study is by Wilhelm and Chen (1990) who formulated the problem and devised an optimizing approach employing Lagrangian relaxation. Their Lagrangian problem is decomposable into subproblems related to independent jobs. They incorporated several pre-processing steps, an efficient dynamic programming algorithm to resolve the subproblems, dominance properties to enhance efficiency, and a specialized branching rule in their branch and bound algorithm. They showed that this approach outperforms OSL, a standard mathematical programming package, on a set of 24 test problems. While this optimizing approach was successful in application to a set of problems of modest size, it is expected that heuristic methods are needed to deal with truly large-scale problems found in some industries. This paper addresses that need.

Other studies are less closely related, but we review several below to indicate the directions that they have taken. The release (i.e. launch) of jobs into a job shop (Irastorza and Deane 1974, Roderick 1990) and into manufacturing cells (Gershwin *et al.* 1984, Han and McGinnis 1988) has been studied, but no prior research has addressed kitting in multi-echelon assembly. OPT (Goldradt 1980) apparently prescribes kitting decisions, but little can be said about this procedure, since it is a proprietary computer code.

Each job might be viewed as a project; and each subassembly as an activity. Thus, the kitting problem could be interpreted as a resource-constrained, multi-project scheduling problem. The literature on project scheduling (Hax and Candea 1983) deals with resource constraints either by focusing on project duration or on the total cost of multiple projects. In the latter case, resources are treated (Slowinski 1981) as doubly constrained with availability limited each period and over the life of a project, non-storable but available in limited quantities each time period, or storable items which may be consumed over the life of a project. However, none of these treatments accommodates 'parts' which are delivered over the duration of the horizon and which can be held for use in later periods. Thus, heuristics devised for project scheduling are not directly applicable to the kitting problem.

Research on the kitting problem is justified, because of the importance of assembly operations, which comprise some 40% of total manufacturing cost (Funk 1986). One prime application area, the electronics industry, accounts for 15% of the US gross national product (US Department of Commerce 1990). Thus, there is ample opportunity to achieve industrial benefits from such research.

This paper is organized in four sections. Fundamental concepts are discussed in the first section, including the context of kitting, the model formulated by Wilhelm and Chen (1990), and the two heuristics (H2 and H3) commonly used in industry. The new heuristic (H1) is detailed in the second section. Computational evaluation is described in the third section, and conclusions are given in the final section.

2. Fundamentals

This section describes the kitting problem in some detail. The basic structure of the decision environment is discussed in the first subsection, and a mathematical model is presented in the second subsection to provide an explicit statement of the problem. Finally, the two heuristics used in industry are described. Notation is defined as it is presented in the text and is summarized in the appendix for reader convenience.

2.1. *Structure of the environment*

This subsection describes the kitting problem in the context of small-lot, multi-echelon, multi-product assembly systems. The presentation deals with information related to jobs, the structure of each job, subassemblies and part relationships, shop operations, the kitting schedule, and part availabilities.

Jobs. The goal is to assemble a set of jobs, J. Each job, $j \in J$, represents a customer order that requires a single end-product. The due date of job j is denoted γ_j ($\gamma_j \geq 0$); and its lot size, Q_j ($Q_j \geq 1$). A kit for subassembly ij must, therefore, contain a sufficient number of components to assemble N_j final assemblies. A number of customer orders may require the same end-product, but each is modelled individually to allow flexibility in the kitting schedule.

Job structure. The bill of materials for each job may be described by an assembly tree in which each node has at most one successor (denoted by the set $K(i,j)$). Each node represents a subassembly. $PATH(m,i,j)$ is the set of nodes in the path from node m to node i (inclusive) for job j. Node ij is in echelon e_{ij} ($e_{ij} = |PATH(i,1,j)|$). $M(i,j)$ [$A(i,j)$] is the set of immediate [all] predecessors of node i. Job j is composed of the set of nodes $I(j)$ where $I(j)=\{i|i\in\{1j\}\cup A(1,j)\}$ (i.e. $I(j)$ is the set of nodes for job j consisting of node $1j$ and all of its predecessors).

Subassembly and part relationships. The final assembly of each job is completed at echelon 1. $SE(e,j)$ denotes the set of subassemblies of job j in echelon e (for $e=1,\ldots,E_j$). We assume that subassemblies in echelon E_j are composed exclusively of parts supplied from vendors. Subassembly ij in echelon $e_{ij}<E_j$, is composed of subassemblies $m\in M(i,j)$ and requires quantity q_{ijp} of part p, where q_{ijp} reflects the total requirements of order quantity Q_j.

Shop operations. Subassembly ij is assembled in shop s_{ij}. We denote the set of all subassemblies of job j that are assembled in shop s by $SS(s,j)$.

The capacity of shop s ($s=1,\ldots,S$) is measured in terms of the number of kits that can be launched into it each day, R_{sd}. This measure appears to be acceptable, since small-lot production involves many lots, none of which dominate the processing capability of an entire shop. More detailed measures of capacity (involving, for example, the number of lots in process or the processing time requirements of each lot) could be incorporated at the cost of making the model somewhat more difficult to resolve.

The planned lead time of subassembly ij in shop s_{ij} is denoted σ_{ij}. This planned lead time is (assumed to be) an integer number of days that can be estimated by simulation, queueing models, or methods used to determine planned lead times for MRP systems.

The kitting schedule. It is assumed that the kitting policy for each day over the horizon $d=1,\ldots,D$ is to be prescribed and that each completed kit is immediately launched into production. Using binary *decision variables* X_{ijd}

$$X_{ijd}=1 \text{ if subassembly } i \text{ on job } j \text{ is kitted on day } d\text{; 0 otherwise}$$

the day on which subassembly ij is kitted is

$$\sum_{d=1}^{D} d\mathrm{X}_{\mathrm{ijd}}$$

Precedence relationships are invoked by

$$\sum_{d=1}^{D} dX_{ijd} \geq \sum_{d=1}^{D} dX_{mjd}+\sigma_{mj} \qquad \text{for } m\in M(i,j)$$

in which σ_{mj} is the planned lead time for subassembly mj. Operation mj completes on day

$$\sum_{d=1}^{D} dX_{mjd}+\sigma_{mj}-1$$

so that subassembly mj could be kitted for subassembly ij on the following day, according to our convention for accounting time.

Part availability. The traditional inventory classification scheme identifies A, B, and C classes of items (e.g. Hax and Candea 1984). We consider a new class, A^+, which designates parts (e.g. integrated circuits) that are likely to disrupt planned schedules. Representing technological capabilities, it is assumed that each A^+ part is assembled by only one of the S shops. Kitting schedules are prescribed only for the set P of A^+ parts, since only A^+ parts require this additional management attention. However, the cost of holding a subassembly, say ij, as in-process inventory should include the cost of constituent A^+, A, B, and C parts as well as the total contributed value built into all subassemblies $m \in M(i,j)$.

The quantity of part p $\{p|p \in P\}$ on hand, n_{p0}, is known when the kitting policy is prescribed. Vendors have promised to deliver quantity n_{pd} of part p on day d, and these parts may be kitted starting on day $d+1$. Thus, the cumulative number of part p available for use through day d is

$$N_{pd} = \sum_{t=0}^{d-1} n_{pt} \qquad \text{for } d = 1, \ldots, D$$

and the total to be available over the horizon, N_{pD}, must be sufficient to complete all scheduled jobs.

3. The model

This subsection describes the model of the kitting problem formulated by Wilhelm and Chen (1990). First, the objective function is discussed.

The objective is to minimize total cost which consists of job earliness, job tardiness, and subassembly holding cost. Cost parameters H_{1j}, job earliness cost/day, and T_j, job tardiness cost/day, penalize performance that deviates from schedule, while H_{ij}, the cost/day of holding subassembly ij, penalizes poor material flow co-ordination from one echelon to the next.

H_{ij} reflects the contributed value added by each subassembly operation, and the value of the parts required by subassembly ij. This holding cost is assessed for each day that subassembly ij must wait to be kitted to produce subassembly $k \in K(i,j)$. H_{1j} is the cost of holding a finished end-product until its due date.

If assembly $1j$ is kitted on day $d = \gamma_j - \sigma_{1j} + 1$, job j would be completed on its due day γ_j. Thus, earliness is incurred if $1j$ is kitted on day $d < \gamma_j - \sigma_{ij} + 1$, and the cost of earliness is

$$H_{1j} \max\{0, \gamma_j - \sigma_{ij} + 1 - d\} \tag{1}$$

Tardiness is incurred if assembly $1j$ is kitted on day $d > \gamma_j - \sigma_{ij} + 1$, and the cost of tardiness is

$$T_j \max\{0, d - (\gamma_j - \sigma_{ij} + 1)\} \tag{2}$$

The cost of holding subassembly $mj \in M(i,j)$, given that subassembly ij is kitted on day d is

$$H_{mj}\left[d - \sum_{t=1}^{d-\sigma_{mj}} (t + \sigma_{mj}) X_{mjt}\right] \tag{3}$$

and the cost of holding subassembly $ij \in A(1,j)$ if it is kitted on day d is

$$H_{ij}\left[\sum_{t=d+\sigma_{ij}}^{D} t X_{kjt} - (d + \sigma_{ij})\right] \qquad \text{for } k \in K(i,j) \tag{4}$$

in which the bracketed term in expression (3) [(4)] defines the number of days over which subassembly mj [ij] is held. The cost of holding subassembly ij over the planned lead time, σ_{ij}, is not included, since this cost is incurred regardless of the kitting schedule. Objective function coefficients $C_{ij}(d)$, which incorporate job earliness, job tardiness, and in-process holding costs, may be determined using equations (1)–(4) (Wilhelm and Chen 1990).

The kitting problem may be formulated as

Problem P_0: Minimize

$$Z_0 = \sum_{j \in J} \sum_{i \in I(j)} \sum_{d=1}^{D} C_{ij}(d) X_{ijd} \tag{5}$$

Subject to:

$$\sum_{j \in J} \sum_{i \in SS(s,j)} X_{ijd} \leq R_{sd} \qquad \text{for } d=1, \ldots, D \qquad s=1, \ldots, S \tag{6}$$

$$\sum_{j \in J} \sum_{i \in I(j)} \sum_{t=1}^{d} q_{ijp} \mathrm{X}_{ijt} \leq N_{pd} \qquad \text{for } d=1, \ldots, D-1; p \in P \tag{7}$$

$$\sum_{j} \sum_{i \in I(j)} \sum_{t=1}^{D} q_{ijp} X_{ijt} = N_{pD} \qquad \text{for } p \in P \tag{8}$$

$$\sum_{d=1}^{D} (d+\sigma_{mj}) X_{mjd} \leq \sum_{d=1}^{D} d X_{ijd} \qquad \text{for } j \in J;\ i \in I(j);\ m \in M(i,j) \tag{9}$$

$$\sum_{d=1}^{D} X_{ijd} = 1 \qquad \text{for } j \in J;\ i \in I(j) \tag{10}$$

$$X_{ijd} = 0 \text{ or } 1 \qquad \text{for } j \in J;\ i \in I(j);\ d=1, \ldots, D \tag{11}$$

The objective as noted in relation (5) is to minimize the total cost. Shop capacities, which limit the number of kits that can be launched each day, are imposed by constraint (6). Inequality (7) assures that the kitting schedule observes part availabilities, so that no more than N_{pd}—the cumulative number of parts on hand initially and planned for delivery on days $1, \ldots, d-1$—can be used in kits composed through day d. Constraint (8) assures that part deliveries satisfy all requirements over the horizon of D days. Inequality (9) invokes precedence relationships, so that all predecessors $m \in M(i,j)$ must be assembled before subassembly ij can be kitted. Constraint (10) ensures that each subassembly is kitted exactly once during the planning horizon, and (11) requires decision variables to be binary integers.

In a related study, Fisher (1972, 1973) used Lagrangian relaxation to resolve a scheduling problem which did not incorporate resource availability constraints of the form in constraint (7). He defined decision variables $y_{jq} = 1$ if schedule q is used for job j, and 0 otherwise, so schedules were generated 'offline' rather than by the model as done in problem P_0.

3.1. *Two heuristics used in industry*

One heuristic commonly used in industry 'dedicates' on-hand stock to specific kits; the other requires kits to 'compete' daily until all required parts are available. In both heuristics, each kit is assigned a 'due day' on which it becomes a candidate for the allocation of parts. In general, the due day of a kit is determined by the due day of its associated end-product. Thus, each kit has a priority based on its due day and, perhaps, other business considerations according to which it is allocated parts.

In the 'dedicate' heuristic, H2, on-hand parts are allocated (i.e. dedicated) to the priority kit. A partially completed kit (i.e. the dedicated parts that are on-hand) must wait for future deliveries of missing components from vendors or upstream production/assembly operations.

In the 'compete' heuristic, H3, parts are allocated to a kit only when **all** the parts required by the kit are on hand, so that a kit must compete daily until it finds all required components in stock. When two kits compete for the same parts, the priority kit has a competitive advantage to be assigned parts that are on hand.

In actual operations, partially completed kits may be launched into production to initiate assembly operations. However, this practice incurs additional costs (future deliveries must be specially handled to 'catch up' with the kit) and should be discouraged, particularly in the automated assembly environment. Thus, we assume that no partial kits are launched into assembly.

4. Heuristic H1

Since constraints (6) and (10) form an assignment problem, P_0 could be viewed as an assignment problem with side constraints (APSC). Since the APSC is known to be NP-hard (Mazzola and Neebe 1986), the kitting problem is NP-hard; that is, only exponential time optimizing algorithms are known for this class of problems. Thus, when dealing with a large instance, in the worst case, it is not possible to obtain an optimal solution in a reasonable time.

A linear programming (LP) approximation approach might be applied in this situation. However, it is expected that when part availability constraints are very 'dense' (i.e. each constraint contains many decision variables and each decision variable appears in many constraints), many of the decision variables may have fractional values in an LP relaxation. Thus, to obtain a feasible integer solution, it is likely that a rounding procedure would have to be applied over and over again, resulting in computational inefficiency. A heuristic which is not based on LP thus appears to be an attractive means of prescribing good solutions for large instances of the kitting problems that might be found in some industries. In order to develop a good heuristic for the kitting problem, four goals must be sought simultaneously: (1) a good kitting sequence for jobs; (2) low job tardiness costs; (3) low job earliness costs; and (4) low subassembly holding costs. Since each job may have unique cost parameters and planned lead times, kitting jobs according to earliest-due-date-first may not result in a solution of high quality. In our new heuristic, H1, we use a procedure to determine job kitting sequence. In the first $|J|$ passes, jobs are scheduled in increasing order of slack. The slack of job j is defined as

$$SLA_j = \gamma_j + 1 - \max_{mj \in SE(E_j, j)} \sum_{aj \in PATH(mj, 1j)} \sigma_{aj}$$

in which the maximum operator defines the longest series of planned lead times to complete the end-product among all subassemblies in echelon E_j. In terms of MRP, this term gives the planned lead time for job j. Subassemblies of a selected job are then sequenced according to earliest due day, $IDUE(i,j)$, first:

$$IDUE(i,j) = \gamma_j + 1 - \sum_{aj \in PATH(ij, 1j)} \sigma_{aj}$$

in which the summation gives the planned lead time for subassembly ij. Each pass determines $ORDER(j)$, the sequence position in the final schedule, for the job j with the lowest potential tardiness cost ($ORDER(j) = 1, \ldots, |J|$).

To achieve goal (2), subassemblies should be kitted as early as possible. However, this may result in high job earliness and/or subassembly holding costs, which may be incurred when a subassembly is completed and there are not enough parts to kit its immediate successor, and thus conflicts with goals (3) and (4). So, each time all subassemblies of a job are scheduled, the kitting times for all subassemblies of that job are right shifted in time (i.e. postponed) to avoid holding costs.

To summarize, heuristic H1 schedules jobs one-at-a-time. In each of the first $|J|$ passes, jobs are scheduled in increasing order of slack, SLA_j, and $ORDER(j)$ of one job, j, in the final sequence is determined. Each subassembly of each job is assigned a due day, $IDUE(i,j)$. Subassemblies of a selected job are then sequenced with the earliest $IDUE(i,j)$ first. After all subassemblies of a selected job are scheduled, the kitting days for all subassemblies of that selected job are right shifted in time to reduce holding costs. The remaining jobs are rescheduled again and $ORDER(j)$ of another job, j, is fixed at the next iteration. The same procedure iterates until $ORDER(j)$ is determined for each job $j \in J$. Then, the kitting days of all subassemblies are determined in the $(|J|+1)$th pass, scheduling job j according to $ORDER(j)$. Heuristic H1 is detailed in the next subsection. Then, the computational complexity of H1 is analysed. This section concludes with a numerical example that demonstrates H1.

4.1. *The H1 procedure*

The heuristic determines $KIT(i,j)$, the scheduled kitting day for subassembly ij. Before presenting H1, we define several additional elements of notation. $POSITION$ $(=|J|, \ldots, 1)$ is a counter that identifies how many sequence positions remain to be considered. $FINAL$ is a code set to 0 for the first $|J|$ passes, then to 1 for the final pass which prescribes the kitting schedule (i.e. $KIT(i,j)$ for all i and j). Π_1 and Π_2 are subsets of J, the set of jobs. TC_j, the tardiness cost of sequencing job j to complete on day D, is defined as

$$TC_j = T_j \max[0, D + \sigma_{1j} - \gamma_j - 1]$$

First, we present H1 in succinct algorithmic form, then we describe each step from an intuitive perspective.

Step 0. Initialization

0-A. Set $\Pi_1 = \Pi_2 = J$, $POSITION = |J|$, $FINAL = 0$

0-B. For each $j \in J$,
set $IDUE(1,j) = \gamma_j + 1 - \sigma_{1j}$, $SLA_j = \gamma_j + 1 - \max_{mj \in SS(E_j,j)} \Sigma_{aj \in PATH(mj,1j)} \sigma_{aj}$, and for each $ij \in A(1,j)$ set $IDUE(i,j) = IDUE(k,j) - \sigma_{ij}$, in which $kj \in K(i,j)$.

Step 1. Schedule all jobs in set Π_1

1-A. If $FINAL = 0$, select $j^* = \text{argmin}_{j \in \Pi_1} \{SLA_j\}$
breaking ties by selecting the one with the smallest order quantity first.
Otherwise, select $j^* = \text{argmin}_{j \in \Pi_1} \{ORDER(j)\}$.

1-B. Select subassembly i^* as the one with the earliest $IDUE(i,j^*)$ (i.e. $i^* = \text{argmin}_{i \in I(j^*)} \{IDUE(i,j^*)\}$).

Break ties by selecting subassembly $i^* = \text{argmin}_{i \in I(j^*)} \{H_{ij^*}\}$.

Set $KIT(i^*,j^*)$ as early as possible without violating constraints (6)–(9). (Update the right-hand sides of constraints (6)–(9) every time a kitting schedule for a subassembly is determined.)

1-C. Select subassembly i^* as the one with the latest $KIT(i,j^*)$ [i.e. $i^* = \text{argmax}_{i \in I(j^*)} \{KIT(i,j^*)\}$].

Break ties by selecting i^* as the subassembly $i^* = \text{argmax}_{i \in I(j^*)} \{H_{ij^*}\}$.

1-C-1. For $i^* = 1$

If $KIT(i^*,j^*) < IDUE(i^*,j^*)$, set $KIT(i^*,j^*)$ as close to (but not later than) $IDUE(i^*,j^*)$ as possible, observing (6)–(9). Otherwise, $KIT(i^*,j^*)$ is not changed. Go to step 1-D.

1-C-2. For $i^* > 1$

If $KIT(i^*,j^*) < KIT(k,j^*) - \sigma_{i^*,j^*}$, in which $k \in K(i^*,j)$, set $KIT(i^*,j^*)$ as close to (but not later than) $[KIT(k,j^*) - \sigma_{i^*,j^*}]$ as possible, observing (6)–(9). Otherwise, $KIT(i^*,j^*)$ is not changed. Go to step 1-D.

1-D. Set $\Pi_1 \leftarrow \Pi_1 - \{j^*\}$. If $\Pi_1 \neq \emptyset$, go to step 1.

Otherwise, $(\Pi_1 = \emptyset)$, set $D = \max_j \{KIT(1,j)\}$.

1-E. If $FINAL = 0$, go to step 2.

Otherwise $(FINAL = 1)$, stop.

Step 2. Determine the ORDER *of job* j+ *for final scheduling*

2-A. For $j \in \Pi_2$, calculate $TC_j = T_j \max[0, D + \sigma_{1j} - \gamma_j - 1]$.

2-B. Let $j+ = \text{argmin}_{j \in \Pi_2} \{TC_j\}$.

Break ties by setting $j+ = \text{argmax}_{j \in \Pi_2} \{IDUE(1,j)\}$.

Break any second tie by designating $j+ = \text{argmin}_{j \in \Pi_2} \{Q_j\}$.

Update $\Pi_2 \leftarrow \Pi_2 - \{j+\}$.

Set $ORDER(j+) = POSITION$, and decrement $POSITION \leftarrow POSITION - 1$.

2-C. If $POSITION > 1$, set $\Pi_1 = \Pi_2$ and go to step 1.

Otherwise $(POSITION = 1)$, go to step 3.

Step 3. Determine final schedule

Set $j+ =$ the index of the remaining job in set Π_2, $ORDER(j+) = 1$, $\Pi_1 = J$, $FINAL = 1$, re-initialize constraints (6)–(9), and go to step 1.

To begin, Π_1, Π_2, $POSITION$, and $FINAL$ are initialized (step 0-A). Subassembly due day and job slack are calculated in step 0-B. In step 1-A, a job is selected to determine its kitting schedule. In each of the first $|J|$ passes, the unscheduled job with the smallest slack SLA_j is selected first. In the $(|J|+1)$th pass, job j is selected according to $ORDER(j)$, which is determined during the first $|J|$ passes.

After selecting a job [$j^* \in \Pi_1$] to schedule in step 1-A, the procedure proposes a feasible (i.e. relative to constraints (6)–(9)) kitting schedule for all subassemblies of job j^* (step 1-B). The subassembly i^* with the earliest $IDUE(i,j)$ amongst all subassemblies of job j^* is scheduled first. A tie is broken by giving priority to the subassembly with the smallest in-process holding cost/day. The kitting schedule of i^* is set on the earliest day feasible with respect to constraints (6)–(9). In step 1-C, kitting schedules for all subassemblies of job j^* are refined, if possible, to reduce in-process holding cost. The subassembly i^* with the latest kitting day, $KIT(i^*,j^*)$, is rescheduled first. $KIT(1^*,j^*)$ is rescheduled to be as close as possible to γ_j (step 1-C-1) and $KIT(1^*,j^*)$, $i^* \neq 1$, is rescheduled to be as close as possible to the day when its immediate successor requires it [i.e. $KIT(k,j^*) - \sigma_{i^*,j^*}$] (step 1-C-2). In step 1-D, Π_1 is updated. If all jobs have been scheduled, the kitting horizon, D, relative to all jobs in Π_1, is determined. Otherwise, another job is selected to schedule. If the final schedule has been obtained (i.e. in the $(|J|+1)$th pass), H1 terminates in step 1-E. Otherwise, the procedure progresses to step 2 to determine $ORDER(j)$ of another job $j \in J$.

In step 2-A, the total tardiness cost for each job is calculated based on the assumption that it is kitted on day D. The job $j+$ which will incur the smallest tardiness cost if it is kitted on day D is selected to determine $ORDER(j+)$ (step 2-B). The $ORDER(j+)$ of job $j+$ is set to the current value of $POSITION$. If $ORDER(j)$ of all jobs (except one job) has not been determined, Π_1 is updated and H1 returns to step 1. Otherwise, $POSITION$ is set to 1 and the procedure progresses to step 3.

In step 3, $ORDER(j+)$ of job $j+$, the one that has not been determined in the first $|J|-1$ passes, is set to 1. $FINAL$ is set to 1 and H1 returns to step 1 for the $|J+1|$th pass to determine the final schedule. After the final schedules for all subassemblies have been determined, H1 terminates in step 1-D. The upper bound on the optimal objective value of problem P_0, Z_0, may then be obtained according to $KIT(i,j)$ for all i and j.

4.2. *Complexity of heuristic H1*

The complexity of H1 is analysed as follows:

Step 1

1-A: The job with the smallest slack [or $POSITION$] is selected first. This takes $O(|J|\log(|J|))$ time.

1-B: 0

1-C: Subassembly i^* with the earliest $IDUE(i,j)$ is chosen first. Hence, the complexity of step 1-C is $O(\max_j |I(j)| \log(|I(j)|))$.

1-D: Subassembly i^* with the latest $KIT(i,j)$ is selected first. This takes $O(\max_j |I(j)| \log(|I(j)|))$ time.

1-E: 0

Hence, the complexity of step 1 is

$$O(\max\{|J| \max_j[|I(j)| \log(|I(j)|)], |J| \log(|J|)\}).$$

Step 2

p	d	N_{pd}	p	d	N_{pd}	p	d	N_{pd}
1	1	0	3	5	7	5	9	22
1	2	0	3	6	11	5	10	22
1	3	0	3	7	11	6	1	14
1	4	7	3	8	11	6	2	14
1	5	7	3	9	11	6	3	14
1	6	7	3	10	11	6	4	26
1	7	7	4	1	10	6	5	26
1	8	11	4	2	16	6	6	26
1	9	11	4	3	16	6	7	26
1	10	11	4	4	24	6	8	26
2	1	0	4	5	24	6	9	26
2	2	0	4	6	24	6	10	26
2	3	0	4	7	24	7	1	10
2	4	7	4	8	24	7	2	10
2	5	7	4	9	24	7	3	10
2	6	7	4	10	24	7	4	14
2	7	7	5	1	10	7	5	14
2	8	11	5	2	10	7	6	22
2	9	11	5	3	22	7	7	22
2	10	11	5	4	22	7	8	22
3	1	0	5	5	22	7	9	22
3	2	0	5	6	22	7	10	22
3	3	0	5	7	22			
3	4	7	5	8	22			

Table 1. Part delivery schedule for the numerical example.

2-A: 0

2-B: The job with the smallest TC_j is determined. This takes $O(|J|)$ time.

2-C: 0

Hence, the complexity of step 2 is $O(|J|)$ and the procedure takes $O(\max\{\max_j |I(j)| \log(|I(j)|),\ |J| \log(|J|)\})$ time for one pass. Since this heuristic procedure requires $|J|+1$ passes, the complexity of H1 is

$$O(|J| \max\{|J| \max_j |I(j)| \log(|I(j)|),\ |J| \log(|J|)\}).$$

4.3. *A numerical example*

A numerical example is presented in this subsection to demonstrate heuristic H1. Part delivery schedules are given in Table 1. Consider three, three-echelon, binary (i.e. $|M(i,j)|=2$ for all $j \in J$ and $i \in I(j) \backslash SE(E_j, j)$) networks. Initially, the kitting horizon is set at $D=9$. Values of parameters q_{ijp} for each A^+ part, H_{ij}, and σ_{ij} are given in Table 2. Table 3 lists values of parameters T_j and γ_j for each job $j \in J$.

Calculations are itemized below. The reader may best follow this presentation by reading it in parallel with the statement of the heuristic procedure.

Step 0

0-A. $\Pi_1 = \Pi_2 = \{1, 2, 3\}$, $POSITION = 3$, $FINAL = 0$

0-B. $SLA_1 = 1$, $SLA_2 = 4$, $SLA_3 = 2$,

j	i	p	q_{ijp}	H_{ij}	σ_{ij}
1	1	1	5	18	1
1	2	2	5	8	1
1	3	3	5	9	1
1	4	4	10	4	3
1	5	5	10	3	2
1	6	6	10	4	3
1	7	7	10	4	3
2	1	1	4	19	1
2	2	2	4	9	1
2	3	3	4	9	1
2	4	4	8	4	3
2	5	5	8	4	3
2	6	6	12	5	3
2	7	7	8	3	2
3	1	1	2	10	1
3	2	2	2	4	1
3	3	3	2	5	1
3	4	4	6	2	2
3	5	5	4	1	1
3	6	6	4	3	2
3	7	7	4	1	1

Table 2. Subassembly parameters for the numerical example.

j	T_j	γ_j
1	90	5
2	95	8
3	50	5

Table 3. Job parameters for the numerical example.

$IDUE(1,1)=5$, $IDUE(2,1)=4$, $IDUE(3,1)=4$, $IDUE(4,1)=1$, $IDUE(5,1)=2$, $IDUE(6,1)=1$, $IDUE(7,1)=1$, $IDUE(1,2)=8$, $IDUE(2,2)=7$, $IDUE(3,2)=7$, $IDUE(4,2)=4$, $IDUE(5,2)=4$, $IDUE(6,2)=4$, $IDUE(7,2)=5$, $IDUE(1,3)=5$, $IDUE(2,3)=4$, $IDUE(3,3)=4$, $IDUE(4,3)=2$, $IDUE(5,3)=3$, $IDUE(6,3)=2$, $IDUE(7,3)=3$.

Step 1

1-A. $(FINAL=0)$ $j^*=1$

1-B. $KIT(7,1)=1$, $KIT(6,1)=1$, $KIT(4,1)=1$, $KIT(5,1)=1$, $KIT(2,1)=4$, $KIT(3,1)=4$, $KIT(1,1)=5$.

1-C. $KIT(1,1)=6$, $KIT(3,1)=5$, $KIT(2,1)=5$, $KIT(7,1)=2$, $KIT(4,1)=2$, $KIT(6,1)=2$, $KIT(5,1)=3$.

1-D. $\Pi_1 = \{2,3\}$

1-A. $j^* = 3$

1-B. $KIT(4,3)=2$, $KIT(6,3)=1$, $KIT(5,3)=1$, $KIT(7,3)=4$, $KIT(2,3)=4$, $KIT(3,3)=5$, $KIT(1,1)=6$.

1-C. $KIT(1,3)=6$, $KIT(3,3)=5$, $KIT(2,3)=5$, $KIT(7,3)=4$, $KIT(4,3)=3$, $KIT(6,3)=3$, $KIT(5,3)=4$.

1-D. $\Pi_1 = \{2\}$

1-A. $j^* = 2$

1-B. $KIT(5,2)=1$, $KIT(4,2)=4$, $KIT(6,2)=4$, $KIT(7,2)=6$, $KIT(3,2)=8$, $KIT(2,2)=8$, $KIT(1,2)=9$.

1-C. $KIT(1,2)=9$, $KIT(2,2)=8$, $KIT(3,2)=8$, $KIT(7,2)=6$, $KIT(6,2)=5$, $KIT(4,2)=5$, $KIT(5,2)=5$.

1-D. $\Pi_1 = \emptyset$, $D=9$

Step 2

2-A. $TC_1 = 90(9+1-1-5) = 360$, $TC_2 = 95$, $TC_3 = 200$

2-B. $j+ = 2$

2-C. $ORDER(2)=3$, $POSITION=2$

2-D. $\Pi_2 = \{1,3\}$

Step 1

1-A. $(FINAL=0)$ $j^* = 1$

1-B. $KIT(7,1)=1$, $KIT(6,1)=1$, $KIT(4,1)=1$, $KIT(5,1)=1$, $KIT(2,1)=4$, $KIT(3,1)=4$, $KIT(1,1)=5$.

1-C. $KIT(1,1)=6$, $KIT(3,1)=5$, $KIT(2,1)=5$, $KIT(7,1)=2$, $KIT(4,1)=2$, $KIT(6,1)=2$, $KIT(5,1)=3$.

1-D. $\Pi_1 = \{3\}$

1-A. $j^* = 3$

1-B. $KIT(4,3)=2$, $KIT(6,3)=1$, $KIT(5,3)=1$, $KIT(7,3)=4$, $KIT(2,3)=4$, $KIT(3,3)=5$, $KIT(1,1)=6$.

1-C. $KIT(1,3)=6$, $KIT(3,3)=5$, $KIT(2,3)=5$, $KIT(7,3)=4$, $KIT(4,3)=3$, $KIT(6,3)=3$, $KIT(5,3)=4$.

1-D. $\Pi_1 = \emptyset$, $D=6$

Step 2

2-A. $TC_1 = 90(6+1-1-5) = 90$, $TC_3 = 50$

2-B. $j+ = 3$

2-C. $ORDER(3)=2$, $POSITION=1$

2-D. $\Pi_2=\{1\}$

Step 3

$ORDER(1)=1$, $\Pi_1=\{1,2,3\}$, $FINAL=1$

Step 1

1-A. $(FINAL=1)$ $j^*=1$

1-B. $KIT(7,1)=1$, $KIT(6,1)=1$, $KIT(4,1)=1$, $KIT(5,1)=1$, $KIT(2,1)=4$, $KIT(3,1)=4$, $KIT(1,1)=5$.

1-C. $KIT(1,1)=6$, $KIT(3,1)=5$, $KIT(2,1)=5$, $KIT(7,1)=2$, $KIT(4,1)=2$, $KIT(6,1)=2$, $KIT(5,1)=3$.

1-D. $\Pi_1=\{2,3\}$

1-A. $j^*=3$

1-B. $KIT(4,3)=2$, $KIT(6,3)=1$, $KIT(5,3)=1$, $KIT(7,3)=4$, $KIT(2,3)=4$, $KIT(3,3)=5$, $KIT(1,1)=6$.

1-C. $KIT(1,3)=6$, $KIT(3,3)=5$, $KIT(2,3)=5$, $KIT(7,3)=4$, $KIT(4,3)=3$, $KIT(6,3)=3$, $KIT(5,3)=4$.

1-D. $\Pi_1=\{2\}$

1-A. $j^*=2$

1-B. $KIT(5,2)=1$, $KIT(4,2)=4$, $KIT(6,2)=4$, $KIT(7,2)=6$, $KIT(3,2)=8$, $KIT(2,2)=8$, $KIT(1,2)=9$.

1-C. $KIT(1,2)=9$, $KIT(2,2)=8$, $KIT(3,2)=8$, $KIT(7,2)=6$, $KIT(6,2)=5$, $KIT(4,2)=5$, $KIT(5,2)=5$.

1-D. $\Pi_1=\emptyset$, $D=9$

1-E. $(FINAL=1)$ STOP, bounds are given by the current values of D and $KIT(i,j)$.

In the next section, the runtime and solution quality of H1 are compared with those of H2 and H3.

5. Computational evaluation

In this section, a set of test problems is used to evaluate the computational characteristics of the heuristics. All of the experiments were performed on an IBM 3090 running the MVS-ESA operating system. The Lagrangian approach of Wilhelm and Chen (1990) was used to obtain optimal solutions, providing a basis of comparison to evaluate the quality of solutions prescribed by the heuristics. The matrix generators and heuristics were coded in FORTRAN.

5.1. *Test problems*

The test problems used by Wilhelm and Chen (1990) were used to evaluate the heuristics. Although generated randomly, they were designed to study the effects of five factors: (1) number of jobs, $|J|$; (2) number of echelons, E; (3) number of parts in class A^+, $|P|$; (4) number of different part types in class A^+ required by all subassemblies; and (5) part delivery time distribution. The magnitudes of $|J|$ and $|\Theta|$ affect the size of the problem, and the magnitude of E reflects the assembly structure. The number of each different part type in the A^+ class required by a subassembly dictates the density of constraints (7) and (8). The part delivery time distribution influences the kitting horizon D, affecting the number of constraints and variables in problem P_0.

Based on these factors, 24 test problems were defined as noted in Table 4. The pattern by which levels of the five factors were set for each problem is apparent so the interested reader can easily identify a pair of problems which differ by a change of one factor to assess its impact on runtime and solution quality.

Given levels of the five factors which define these 24 problems, Table 4 indicates the number of binary decision variables and constraints involved in each problem P_0 formulation. These problems are not large in terms of the parameters which describe their practical contexts, but they do represent rather large binary integer problems.

	Factors					Problem size	
Problem number	$\lvert J\rvert$	E	$\lvert P\rvert$	DS	PDP dist. number	NC	NV
1	6	3	56	1	1	890	588
2	10	3	56	1	1	897	910
3	6	3	56	3	1	890	588
4	10	3	56	3	1	897	910
5	6	3	112	2	1	1660	588
6	10	3	112	2	1	1625	910
7	6	3	112	4	1	1688	588
8	10	3	112	4	1	1625	910
9	13	2	56	1	1	681	429
10	20	2	56	1	1	727	660
11	13	2	56	3	1	703	429
12	20	2	56	3	1	738	660
13	13	2	112	2	1	1231	429
14	20	2	112	2	1	1332	660
15	13	2	112	4	1	1308	429
16	20	2	112	4	1	1343	660
17	6	3	56	1	2	1064	714
18	6	3	56	3	2	1064	714
19	6	3	112	2	2	1999	714
20	6	3	112	4	2	2033	714
21	13	2	56	1	2	849	546
22	13	2	56	3	2	877	546
23	13	2	112	2	2	1549	546
24	13	2	112	4	2	1647	546

DS, distribution set to generate the number of different A^+ part types required by each subassembly (Table 6); PDP dist. number, part delivery perturbation distribution number (Tables 7 and 8); NC, number of constraints in P_0; NV, number of binary variables in P_0.

Table 4. Problem parameters and sizes.

Problems with	Total number of A^+ parts	Echelon e		
		1	2	3
$E=2$	56	20	36	
$E=2$	112	36	76	
$E=3$	56	8	16	32
$E=3$	112	16	32	64

Table 5. Number of A^+ parts used at each echelon.

Problem numbers	Distribution set	Distribution for $i \in SE(E,j)$	Distribution for $i \notin SE(E,j)$
1, 2, 9, 10, 17, 21	DS 1	$U[5,6]$	$U[3,4]$
5, 6, 13, 14, 19, 23	DS 2	$U[10,12]$	$U[6,8]$
3, 4, 11, 12, 18, 22	DS 3	$U[6,8]$	$U[3,6]$
7, 8, 15, 16, 20, 24	DS 4	$U[12,16]$	$U[6,10]$

Table 6. The number of different part types from the A^+ class required by each subassembly.

So that each test could be easily duplicated, each involved a binary assembly tree in which each node (other than those in echelon E) had two immediate predecessors. This structure does not limit interpretation of test results, since any assembly tree can be recast into the binary form. Jobs are described either by an $E=2$ or an $E=3$ echelon binary network so that $|I(j)|$ is 3 or 7, respectively. For simplicity, we assumed that subassemblies in level e would be assembled in shop $s=e$ so that $S=E$.

The number of jobs was either 6 or 10 in the three-echelon problems and either 13 or 20 in the two-echelon problems. The number of parts in class A^+, $|P|$, was either 56 or 112; the number of A^+ parts assembled in each echelon is given in Table 5.

Four unique discrete uniform distributions were used to generate the number of different part types in class A^+ required by all subassemblies (factor 4) as indicated in Table 6. The fifth factor involved setting the promised delivery day for the parts required by each subassembly by perturbing the planned delivery time. Discrete uniform distributions used as the two levels of this factor are given in Tables 7 and 8.

Since holding costs incorporate the cost of A, B, and C (inventory) classes of parts, the number of parts required by each subassembly was generated randomly from discrete uniform distributions:

Echelon	A Parts	B Parts	C Parts
$i \in SE(E,j)$	$U[20,26]$	$U[30,36]$	$U[40,46]$
$i \notin SE(E,j)$	$U[6,8]$	$U[10,12]$	$U[14,16]$.

Customer order quantity, Q_j, was generated randomly from the discrete uniform distribution given in Table 9. Parameters R_{sd}, σ_{ij}, and γ_{ij} were defined as noted in

Table 10. H_{ij} values used are given in Table 11, and tardiness penalty $T_j = 5H_{ij}$ for all $j \in J$ (\$/day) was used.

5.2. *Test results*

Computational results are reported in this subsection; runtimes and solution values for heuristics H1 to H3 are given in Table 12. In general, runtime increases with the number of jobs (compare problems 3 and 4) and the delays of part deliveries (e.g. compare problems 1 and 17), since they extend the kitting horizon and thus increase the size of the problem. Hence, this indicates, as expected, that problems of large size are more difficult to solve. Runtime is also influenced by the number of A^+ parts required by subassemblies. Roughly speaking, the more A^+ parts required by subassemblies, the more CPU time is required to solve the problem (e.g. compare problems 5 and 6), indicating that problems with denser constraints are more 'difficult' to solve, as expected.

H1 obtained optimal solution values for 17 of the 24 test problems with all solution times $<0{\cdot}231$ s. H2 provided a better solution value than H1 in only one problem (i.e. problem 21), as did H3 (problem 22). In all test problems, H3 required less runtime than did H1 and H2. However, overall, H3 did not prescribe solutions of the quality of those prescribed by H1 and H2.

The reason that H1 did not perform best in problems 21 and 22 may be because parts that were supposed to be assigned to subassemblies of two (or more) jobs with small order quantities were used by subassemblies of a job with a large order quantity. According to step 2 of H1, a job with a large order quantity may be scheduled earlier than a job with a small order quantity, provided they have the same due day. Kitting subassemblies of a job with a large order quantity may result in kitting delays of the subassemblies of other jobs with small order quantities. Furthermore, this situation may result in higher tardiness costs. In a sense, H1 allows subassemblies of a job with a large order quantity to 'steal' parts from the subassemblies of other jobs with small order quantities. Since this situation results in higher tardiness costs, H1 could be refined to explicitly deal with it if the application warranted such changes.

Delivery distribution	1 day early	on time	1 day late	2 days late
Probability	0·3	0·4	0·2	0·1

Table 7. Part delivery perturbation distribution 1 for test problems 1–16.

Delivery distribution	Days early		Days tardy				
	1	0	1	2	3	4	5
Probability	0·1	0·4	0·1	0·1	0·1	0·1	0·1

Table 8. Part delivery perturbation distribution 2 for test problems 17–24.

Order quantity Q_j	20	30	40	50	60
Probability	0·2	0·2	0·2	0·2	0·2

Table 9. Customer order quantity distribution.

Shop capacity R_{sd}	$=[\|J\|/4]+3$ $=[\|J\|/2]+3$ $=\|J\|+3$	if $s=1$ if $s=2$ if $s=3$
Sojourn time σ_{ij}	$=4$ days $=3$ days $=2$ days	if $ij \in SE(E,j)$ if $ij \in SE(E-1,j)$ if $ij \in SE(E-2,j)$
Due day γ_{ij}	$=\max_{mj \in SE(E,j)} \{\Sigma_{i,j \in PATH(m,1,j)} \sigma_{i,j}\} - 1 +$ slack time generated from $U[0,4]$ $=$longest total sojourn time for operation series on job $j+$ slack time generated from $U[0,4]$	

Table 10. Parameters R_{sd}, σ_{ij}, and γ_j.

Table 13 shows the relative performance of the three heuristics. On average, H1 provided solutions that were within 0·6% of the optimal solutions. The value of a solution from H1 was 3·4% better than that from H2 and 10·7% better than that from H3 on average.

H1 performed extremely well for the part delivery time distribution in Table 7, prescribing optimal solutions for 13 of 16 problems (i.e. problems 1–16), but did not perform quite as well for the part delivery time distribution in Table 8. In this case, H1 prescribed optimal solution values for four of eight problems (i.e. problems 17–24). According to Table 8, part deliveries can be up to 5 days late compared with 3 days for the distribution in Table 7. Part delivery delays may create more competition for available parts. Hence, these tests may indicate that optimal solution values are more difficult to obtain by H1 when problems involve more competition for parts. Although H1 did not perform as well when the part delivery time distribution in Table 8 was used, it still performed better than did H2 and H3 in this case, on average.

6. Summary and conclusions

This paper deals with the kitting problem encountered in multi-echelon, multi-product assembly systems. This problem is prevalent in industry but has not been researched extensively.

A new heuristic procedure, which is not based on LP approximation, is presented in this paper. The average performance of the heuristic is compared empirically with two heuristics commonly applied in industry, namely, 'dedicate' (H2) and 'compete' (H3). Since H1 is shown to run in polynomial time in the worst case and since it outperforms heuristics H2 and H3 on average, it could be used by any optimizing approach which requires initial upper bounds on the horizon, D, and on the optimal value of the objective function of problem P_0, Z_0. In addition, it appears to be well suited for direct application to large-scale industrial problems.

A set of 24 test problems was devised to evaluate the computational characteristics of the heuristics. Heuristic H1 obtained optimal solution values for 17 of the 24 test problems with solution times under 0·231 s (on average). H2 provided a better solution value than did H1 in only one problem, as did H3. On average, H1

Holding cost	Subassembly i 1	2	3	4	5	6	7
H_{ij} \$/day for problems with $E=3$	0·44	0·16	0·16	0·055	0·055	0·055	0·055
H_{ij} \$/day for problems with $E=2$	0·16	0·055	0·055				

Table 11. Cost parameters H_{ij}.

Problem number	Optimal solution value v(OPT)	Heuristic H1 Solution value v(H1)	Heuristic H1 Runtime	Heuristic H2 Solution value v(H2)	Heuristic H2 Runtime	Heuristic H3 Solution value v(H3)	Heuristic H3 Runtime
1	894·5	*894·5	0·038	929·7	0·050	1198·0	0·023
2	1567·0	*1567·0	0·079	1617·3	0·078	1954·3	0·030
3	979·0	*979·0	0·043	1001·7	0·052	1004·2	0·023
4	1567·0	*1567·0	0·093	1621·6	0·079	1771·7	0·031
5	979·0	*979·0	0·060	986·4	0·085	1232·7	0·026
6	1482·5	*1482·5	0·121	1501·4	0·132	1719·9	0·036
7	979·0	*979·0	0·070	996·5	0·092	998·6	0·028
8	1401·0	*1401·0	0·145	1453·5	0·144	1653·1	0·040
9	690·0	696·0	0·060	713·9	0·049	714·9	0·022
10	984·0	*984·0	0·120	1014·2	0·070	1015·2	0·028
11	567·0	573·0	0·066	634·9	0·052	634·9	0·023
12	981·0	1011·0	0·147	1078·0	0·076	1085·8	0·030
13	676·5	*676·5	0·102	692·8	0·083	692·8	0·027
14	1143·0	*1143·0	0·215	1166·4	0·129	1169·6	0·036
15	696·0	*696·0	0·120	725·8	0·093	752·8	0·028
16	1047·0	*1047·0	0·231	1092·8	0·141	1070·4	0·039
17	2278·5	*2378·5	0·056	2379·5	0·077	2639·3	0·028
18	2319·5	*2319·5	0·065	2485·7	0·095	2671·9	0·031
19	2447·5	*2447·5	0·100	2468·2	0·158	2720·8	0·042
20	2404·0	2447·5	0·117	2466·6	0·180	2920·3	0·043
21	1740·0	1791·0	0·076	1786·2†	0·053	2009·8	0·024
22	1548·0	1590·0	0·084	1698·0	0·059	1588·8‡	0·025
23	1684·5	*1684·5	0·132	1742·2	0·097	1758·1	0·031
24	1746·0	1789·5	0·156	1790·8	0·104	1810·0	0·033

*H1 obtained the optimal solution value.
†H2 obtained better solution value than H1.
‡H3 obtained better solution than value H1.
v(*) Objective function value for solution procedure *.

Table 12. Comparisons of heuristic procedures.

Problem number	$v(H1)/v(OPT)$	$v(H2)/v(OPT)$	$v(H3)/v(OPT)$
1	1·000	1·039	1·339
2	1·000	1·032	1·247
3	1·000	1·023	1·026
4	1·000	1·035	1·131
5	1·000	1·008	1·259
6	1·000	1·013	1·160
7	1·000	1·018	1·020
8	1·000	1·037	1·180
9	1·009	1·035	1·036
10	1·000	1·031	1·032
11	1·011	1·120	1·120
12	1·031	1·100	1·107
13	1·000	1·024	1·024
14	1·000	1·020	1·023
15	1·000	1·043	1·082
16	1·000	1·044	1·022
17	1·000	1·044	1·158
18	1·000	1·072	1·152
19	1·000	1·008	1·112
20	1·018	1·026	1·215
21	1·029	1·027	1·155
22	1·027	1·097	1·026
23	1·000	1·034	1·044
24	1·025	1·026	1·037
Average	1·006	1·040	1·113

$v(*)$ Objective function value for solution procedure *.

Table 13. Relative performance of heuristics.

prescribed solutions that were within 0·6% of the optimal solutions. The value of a solution from H1 was 3·4% better than that from H2 and 10·7% better than that of H3 (on average). These percentages are not large, but they would represent significant cost savings in actual industrial applications.

The new heuristic would promote material flow co-ordination throughout a multi-echelon assembly system, providing a balance between the cost of in-process inventory and schedule performance, and prescribing the allocation of parts to achieve cost-effective schedule performance while promoting shop efficiency. Time-managed flow control would result, co-ordinating material flow throughout the multi-echelon system to achieve cost-effective due date performance.

Acknowledgment

This material is based on work supported by the National Science Foundation on Grant number DMC-8896303.

Appendix. Notation

Indices

d	$=1, \ldots, D$ index for days in the planning horizon
i	$=1, \ldots, I(j)$ index for subassemblies composing job j
j	$\in J$ a job (customer order)

e	$=1, \ldots, E_j$ index of echelons in the structure of job j
s	$=1, \ldots, S$ index for shops
p	$\in P$ an A^+ part

Cost parameters

$C_{ij}(d)$	=total cost incurred by kitting subassembly ij on day d
H_{ij}	=holding cost/day for subassembly ij [$i\in I(j)$] in lot size Q_j
T_j	=tardiness cost/day for job j

Decision variables

X_{ijd}	=1 if subassembly ij is kitted on day d; 0 otherwise

Job information

$I(j)$	=set of subassemblies comprising job j
J	=set of jobs
Q_j	=lot size of job j
$SE(e,j)$	=set of subassemblies of job j in echelon e
γ_j	=due day of job j

Part information

n_{p0}	=number of part p on hand when the kitting policy is set
n_{pd}	=number of part p with delivery promised on day $d=1, \ldots, D-1$
N_{pd}	=cumulative number of part p available through day

$$d=\sum_{t=0}^{d-1} n_{pt} \qquad \text{for } d=1, \ldots, D$$

P	=set of A^+ parts

Shop information

R_{sd}	=capacity of shop s on day d (i.e. the number of kits that can be launched into shop s on day d)

Subassembly information

$A(i,j)$	=set of all predecessors of subassembly ij
e_{ij}	=echelon of subassembly ij
s_{ij}	=the shop in which subassembly ij is processed
$K(i,j)$	=set comprised of the immediate successor of subassembly ij
$M(i,j)$	=set of immediate predecessors of subassembly ij
$PATH(m,i,j)$	=the set of nodes on the path from node mj to node ij (inclusive)
q_{ijp}	=quantity of part p required by subassembly ij to assemble lot size Q_j
$SS(s,j)$	$=\{i \mid s_{ij}=s\}$: set of subassemblies of job j processed in shop s
σ_{ij}	=planned lead time to produce subassembly ij, an integer number of days

Notation used by H1

$FINAL$	=0 for the first $\lvert J \rvert$ passes, then 1 for the last pass which prescribes the kitting schedule
$IDUE(i,j)$	$=\gamma_j+1-\Sigma_{aj\in PATH(ij,1j)} s_{aj}$ =due day for kitting subassembly ij

$KIT(i,j)$	= the kitting day for subassembly ij scheduled by heuristic H1
$POSITION$	= a counter identifying how many sequence positions remain to be considered $= \lvert J \rvert, \ldots, 1$
SLA_j	$= \gamma_j + 1 - \max_{mj \in SE(E_j,j)} \Sigma_{aj \in PAT\ H(mj,1j)} s_{aj}$ = slack of job j
TC_j	= tardiness cost of sequencing job j to complete on day D $= T_j \max[0, D + \sigma_{1j} - \gamma_j - 1]$
Π_1 and Π_2	= subsets of J

References

Fisher, M. L., 1972, Optimal solution of scheduling problems using Lagrangian multipliers: II. *Proceedings of Symposium on Theory of Scheduling and Its Applications*, North Carolina State University, Raleigh, NC, May 15–17, pp. 294–318.

Fisher, M. L., 1973, Optimal solution of scheduling problems using Lagrangian multipliers: I. *Operations Research*, **21,** 1114–1127.

Funk, J. L., 1986, The potential market for robot assembly. *International Journal of Production Research*, **24**(3), 663–686.

Gershwin, S. B., Akella, R., and Choong, Y. C., 1984, Short term production scheduling of an automated manufacturing facility. Report No. LIDS-FR-1356, Laboratory for Information and Decision Systems, MIT.

Goldradt, E., 1980, Optimized production timetable: a revolutionary program for industry. *Proceedings, APICS 23rd Annual Conference.*

Han, M., and McGinnis, L. F., 1980, Throughput maximization in flexible manufacturing cells. *AIIE Transactions*, **20,** 409–417.

Hax, A. C., and Candea, D., 1983, *Production and Inventory Management* (Englewood Cliffs, NJ: Prentice-Hall).

Irastorza, J. C., and Deane, R. H., 1974, A loading and balancing methodology for job shop control. *IIE Transactions*, **6,** 302–307.

Mazzola, J. B., and Neebe, A. W., 1986, Resource-constrained assignment scheduling. *Operations Research*, **34**(4), 560–572.

Roderick, L. M., 1990, A comparison of order release strategies in production control systems. PhD Dissertation, Department of Industrial Engineering, Texas A&M University.

Slowinski, R., 1981, Multiobjective network scheduling with efficient use of renewable and non-renewable resources. *European Journal of Operational Research*, **7,** 265–273.

US Department of Commerce, 1990, *Statistical Abstract of the United States*, p. 426.

Wilhelm, W. E., and Chen, J. F., 1990, An approach for optimizing the allocation of components to kits in small-lot, multi-echelon assembly systems. Working Paper, Department of Industrial Engineering, Texas A&M University.