Pergamon

0305-0548(95)O0020-8

Computers Ops Res. Vol. 23, No. 1, pp. 1–12, 1996

0305-0548/96 $9.50 + 0.00

HOW TO AVOID STOCKOUTS WHEN PRODUCING SEVERAL ITEMS ON A SINGLE FACILITY? WHAT TO DO IF YOU CAN'T?

Guillermo Gallego[1]† and Ilkyeong Moon[2]‡§

[1]Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, U.S.A. and [2]Department of Industrial Engineering, Pusan National University, Pusan 609-735, Korea

(Received October 1994; in revised form March 1995)

Scope and Purpose—We 'dress both existence and optimization problems related to stockouts in a multi-item single facility settii hen initial inventories, setup times, production and demand rates are known. For a given planning hc on T, we ask: does there exist a schedule that avoids stockouts up to time T? Given stockout penalties, we want to find a schedule that minimizes weighted stockout cost. We characterize certain properties of the optimal schedules and develop a MIP formulation for both problems. We have also developed a method to phase into a target cyclic schedule for infinite horizon problems. These can be used as a practical scheduling tool for temporarily overloaded facilities or to reschedule production after a disruption.

Abstract—This paper considers the Multiple Product Single Facility Stockout Avoidance Problem (SAP). That is the problem of determining, given initial inventories, whether there is a multiple product single facility production schedule that avoids stockouts over a given time horizon. The optimization version of the SAP where stockouts are penalized linearly is also studied. We call this problem the Weighted Stockout Problem (WSP). Both problems are NP-hard in the strong sense. Mixed Integer Linear Programming (MIP) formulations for both the SAP and the WSP are developed. We show that there exist polynomial algorithms for some special cases of the SAP and the WSP. We have also developed heuristics and computational results.

1. INTRODUCTION

We consider the Multiple Product Single Facility Stockout Avoidance Problem (SAP), which was proven NP-hard for the finite horizon by Anderson [1]. The SAP is the problem of finding a schedule that avoids stockouts over a given finite horizon. We also consider an optimization version of the SAP, in which demands which are not met are lost, and where stockouts are penalized linearly in their duration. We call this the Weighted Stockout Problem (WSP). The WSP can be easily proven to be NP-hard in the strong sense by reduction to the Weighted Tardiness Problem. (See Baker [2] for the definition of the Weighted Tardiness Problem.)

Both the SAP and the WSP are closely related to the widely studied Economic Lot Scheduling Problem (ELSP). The ELSP is the problem of scheduling the production of several items in a single facility to minimize the long run average inventory carrying and setup costs. See Dobson [3],

†Guillermo Gallego is an Associate Professor of Industrial Engineering and Operations Research at Columbia University. He received his Ph.D. in Operations Research from Cornell University. He currently serves as an associate editor of *IIE Transactions, Management Science, Naval Research Logistics, and Operations Research.*

‡Ilkyeong Moon is an Assistant Professor of Industrial Engineering at Pusan National University in Korea. He received his B.S. and M.S. in Industrial Engineering from Seoul National University, Korea, and Ph.D. in Operations Research from Columbia University in 1991. His interests are in the areas of Production Management, Manufacturing Systems Analysis, and Production Economics. He has research papers published or forthcoming in journals such as *Computers & Industrial Engineering, Computers & Operations Research, International Journal of Production Research, Journal of the Operational Research Society, Management Science, Naval Research Logistics, Omega, Operations Research,* and *The Engineering Economist.*

§To whom all correspondence should be addressed.

Gallego [4], Gallego and Moon [5, 6], Gallego and Roundy [7], Moon [8], and Zipkin [9] for the recent studies on the ELSP.

Research on the ELSP has concentrated on finding cost effective *cyclic* schedules, i.e. schedules that repeat periodically. An implicit assumption, universally imposed, is that the inventories required to start a cyclic schedule can be acquired instantly and at no cost. This assumption is almost never satisfied in practice. In fact, if inventories are low, the primary concern of management is not cost effectiveness, but rather to avoid stockouts (as far as it is possible) until emergency relief becomes available, say at time T, through overtime, temporary use of an additional machine, or external procurement. The problem of avoiding stockouts also emerges after a schedule disruption, such as a machine breakdown, which leaves the inventory levels critically low.

This gives rise to the SAP. The WSP is an optimization version of the SAP, and occurs when a single facility feeds different production lines and the stockout costs are proportional to the length of the stockout period during which the lines are starved. The main reason for this is that with lost sales customers may not return and the future demand may be lower. One way out is by stating that we are not producing to satisfy final demand, but rather producing to feed production lines, and that during the time we run out of stock we have to procure from other sources at a linear cost. When stockouts are expensive and inventories are low, managers may want to minimize the cost of stockouts up to a time, say, T, when inventories can be replenished, for instance during a weekend, or when demand ceases to exist, e.g. at the end of a model year. The WSP is sometimes encountered in metal stamping in the automobile industry.

For a long time horizon, we are also concerned with minimizing the holding and setup costs. We show how the finite horizon WSP can be modified to produce an initial schedule that phases into an efficient (in terms of holding and setup costs) cyclic schedule. In this way the modified finite horizon WSP can be used as a scheduling tool for temporarily overloaded facilities until the normal load condition is recovered. For example, if there is a sudden increase in demand or if a facility temporarily takes the burden of another facility while the latter is being repaired.

We develop a Mixed Integer Linear Programming (MIP) formulation of the WSP. The output of the MIP is a production schedule consisting of the sequence, say $\mathbf{f}$, in which the products are to be setup and produced in the machine, and of the length of the production runs, as well as any scheduled idle times. A slightly different formulation works for the SAP after some necessary conditions are verified. We show that some special cases of the SAP and the WSP can be solved in polynomial time. For instance, given a production sequence, both problems can be solved by Linear Programming (LP). When the production sequence is restricted to product permutations, a production sequence where each product is produced at most once, the SAP can be solved in polynomial time. But the WSP remains NP-hard.

The rest of this paper is organized as follows. In Section 2, we prove some basic properties of optimal schedules and formulate the WSP. We formulate the WSP when the sequence is restricted to permutations and give a solution procedure for the SAP. In Section 3, we explore special cases where the problems are polynomially solvable. Heuristics for the WSP are developed in Section 4. In Section 5, we show how the WSP can be modified so that target inventory levels are achieved at a given future point in time. We also present a method to phase into a cyclic schedule after a finite horizon.

2. FORMULATION

The Weighted Stockout Problem (WSP) can be formally stated as follows. There is a single facility on which m distinct products are to be produced. We assume that demands which are not met are *lost*. The problem is that of finding a production sequence $\mathbf{f} = (f_1, f_2, \ldots, f_N)$ (the sequence may contain repetitions) and a vector of production times $\mathbf{t} = (t_1, t_2, \ldots, t_N)$, so that the weighted stockout times over the planning horizon T are minimized. Here $y_{if_n} = 1$ if $f_n = i$, and zero otherwise (y_{ik}'s are defined below).

The data for the WSP and the SAP are:

the index for the products		$i = 1, \ldots, m$
constant demand rates (units/day)	d_i	$i = 1, \ldots, m$
constant production rates (units/day)	p_i	$i = 1, \ldots, m$

known setup times (day)	s_i	$i = 1, \ldots, m$
known initial inventories (units)	J_i	$i = 1, \ldots, m$
known stockout penalties (\$/unit, day)	π_i	$i = 1, \ldots, m.$
length of planning horizon (day)	T	

The demand rates can be normalized to one by dividing the original p_i's, J_i's and d_i's by d_i for all i. This generates an equivalent problem, but simplifies the mathematical derivations. So, we assume all $d_i = 1$ from this point on

Let $\mathbf{r} \equiv (1/p_1, \ldots, 1/p_m)'$, $\mathbf{J} \equiv (J_1, \ldots, J_m)'$ and $\mathbf{e} \equiv (1, \ldots, 1)'$. Define $\kappa = 1 - \mathbf{e}'\mathbf{r}$. κ is the long run proportion of time available for setups. For infinite horizon problems $\kappa > 0$ is a necessary but not sufficient condition for the existence of a feasible schedule [3]. Over a finite horizon, however, we do not require $\kappa > 0$. Thus our procedures can be used when a facility is overloaded, for instance, when two parallel facilities are used to produce a given set of items, and the entire burden falls on one of them when the other suffers a breakdown.

We make three assumptions

$$\begin{aligned} &J_i < T && i = 1, \ldots, m \\ &T > s_i > 0 && i = 1, \ldots, m \\ &p_i > 1 && i = 1, \ldots, m. \end{aligned}$$

If a product's initial inventory is enough to satisfy demand over the horizon, i.e. $J_i \geqslant T$, then it need not be produced to avoid stockouts. If $s_i = 0$ for all i and $\kappa > 0$, both the WSP and the SAP are trivial. If $s_i > T$ we can not avoid stockouts for item i. If $p_i < 1$, then the facility cannot keep up with the demand of product i, much less with the demands of other products, thus it is natural to assume $p_i > 1$ for all i.

Throughout, we have followed the ELSP paradigm of a facility capable of producing only one item at a time. For this reason we have excluded from consideration the case where some of the setup times are zero. In fact, if two or more setup times are zero, the facility can effectively produce several items at a time by continuously switching production. Although this type of problem could be studied, the resulting schedules are likely to have an undesirably high setup cost because setups are likely to occur very frequently, in fact continuously, and because setup costs are ignored in the short run. One could, of course, add an upper bound on the number of setups for each item to prevent this from happening. We prefer, however, to keep things simple to facilitate the exposition.

2.1. Weighted Stockout Problem (WSP)

The following simple lemma, which we give without proof, states that we do not need to consider idle times except possibly after the last production run. So we do not need to include idle times in the formulation.

Lemma 1. There exists an optimal schedule that does not idle except possibly after the last production run.

Constraints

Let N be an upper bound on the number of positions in the sequence $\mathbf{f}$. Later we will show how to compute N.

Let y_{ik} be a binary decision variable to determine whether product i is produced in the kth position of the sequence

$$y_{ik} \in \{0, 1\} \qquad k = 1, \ldots, N,\ i = 1, \ldots, m. \tag{1}$$

Let

$$w_k = \sum_{i=1}^{m} y_{ik}, \qquad k = 1, \ldots, N. \tag{2a}$$

To ensure that a production sequence is well defined and uniquely represented, we impose

$$w_1 = 1, \qquad w_k \leqslant w_{k-1}, \qquad k = 2, \ldots, N. \tag{2b}$$

Let t_{ik} be the production *run time* of product i in the kth position of the sequence, then

$$t_{ik} \geqslant 0, \qquad k = 1, \ldots, N,\ i = 1, \ldots, m. \tag{3a}$$

For convenience, define

$$t_{i0} = 0, \qquad i = 1, \ldots, m. \tag{3b}$$

Since no production can take place without a setup, we impose

$$t_{ik} - T y_{ik} \leqslant 0, \qquad k = 1, \ldots, N. \tag{3c}$$

Also, the total time spent in setups and production runs must be at most equal to T

$$\sum_{k=1}^{N} \sum_{i=1}^{m} (s_i y_{ik} + t_{ik}) \leqslant T. \tag{4}$$

Let S^k be the *starting time* of the kth production run after the setup. Then

$$S^k = S^{k-1} + \sum_{i=1}^{m} (t_{i(k-1)} + s_i y_{ik}), \qquad k = 1, \ldots, N \tag{5a}$$

where for convenience, we let

$$S^0 = 0, \qquad S^{N+1} = T. \tag{5b}$$

Let x_i^k be the *stockout duration* of the product i in the time interval $[S^{k-1}, S^k]$; of course

$$x_i^k \geqslant 0, \qquad k = 1, \ldots, N+1, \; i = 1, \ldots, m. \tag{6}$$

Let J_i^k be the inventory on hand of the product i at time S^k; by definition

$$J_i^k \geqslant 0, \qquad k = 0, \ldots, N+1, \; i = 1, \ldots, m. \tag{7}$$

Clearly $J_i^0 = J_i$ for all i and J_i^{N+1} is the ending inventory of product i.

From the balance of inventory, stockouts, production, and demand (see Fig. 1), we have

$$J_i^{k+1} - x_i^{k+1} = J_i^k + p_i t_{ik} - (S^{k+1} - S^k), \qquad k = 0, \ldots, N, \; i = 1, \ldots, m. \tag{8}$$

We now present some structural results for this MIP. The following lemma states that there exists an optimal schedule in which every production run for an item, except perhaps the first, starts from zero inventory.

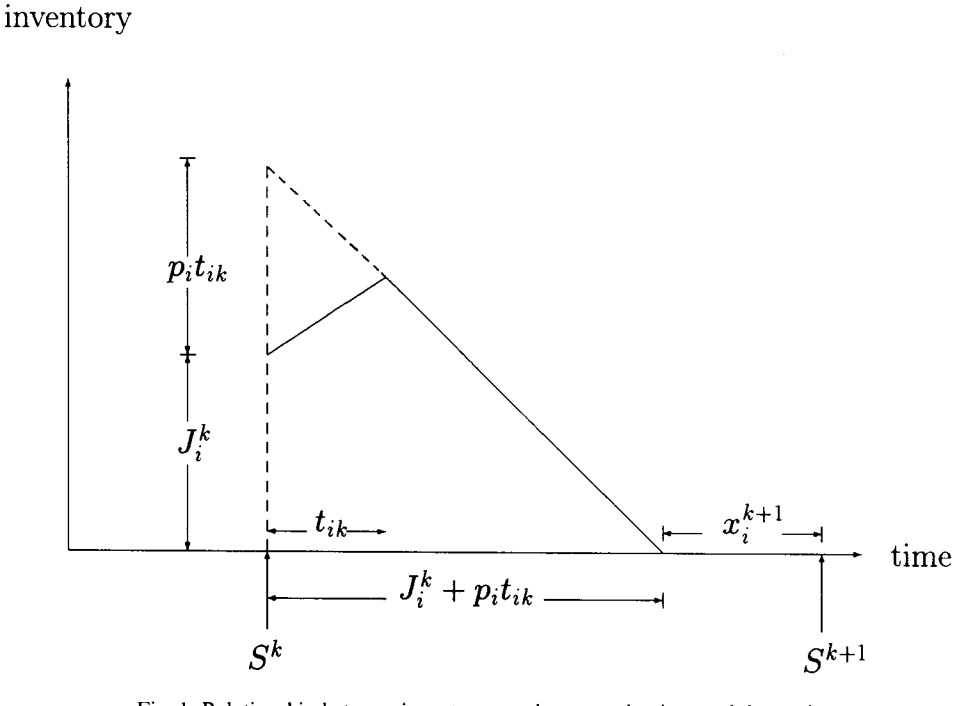

Fig. 1. Relationship between inventory, stockout, production, and demand.

Lemma 2. (Zero Switch Rule) There exists an optimal solution in which $J_i^k = 0$ whenever $y_{ik} = 1$ and $\sum_{l=1}^{k-1} y_{il} \geqslant 1$.

Lemma 3. There exists an optimal solution with final inventories equal to zero, i.e.

$$J_i^{N+1} = 0 \qquad i = 1, \ldots, m. \tag{9}$$

Corollary 1. Equation (4) is redundant.

Proof. Let i be the last item to be produced in the sequence. Since $J_i^N \geqslant 0$,

$$J_i^N + S^N \geqslant S^N = \sum_{i=1}^{m} \sum_{k=1}^{N} (s_i y_{ik} + t_{ik}) - t_{iN}.$$

Since $x_i^{N+1} = (J_i^{N+1} + S^{N+1}) - (J_i^N + S^N) - p_i t_{iN} \geqslant 0$,

$$J_i^{N+1} + S^{N+1} \geqslant (J_i^N + S^N) + p_i t_{iN} \geqslant S^N + p_i t_{iN}$$

$$= \sum_{i=1}^{m} \sum_{k=1}^{N} (s_i y_{ik} + t_{ik}) + (p_i - 1) t_{iN} \geqslant \sum_{i=1}^{m} \sum_{k=1}^{N} (s_i y_{ik} + t_{ik}).$$

By Lemma 3 and $S^{N+1} = T$, the above inequality reduces to equation (4) □

Objective function

For the WSP, the objective is to minimize the weighted stockout times during the planning horizon $[0, T]$

$$\text{minimize} \sum_{i=1}^{m} \pi_i \sum_{k=0}^{N} x_i^{k+1}$$

$$= \sum_{i=1}^{m} \pi_i \left[\sum_{k=0}^{N} \{(J_i^{k+1} + S^{k+1}) - (J_i^k + S^k)\} - \sum_{k=0}^{N} p_i t_{ik} \right]$$

$$= \sum_{i=1}^{m} \pi_i (T - J_i) - \sum_{i=1}^{m} \pi_i p_i \sum_{k=1}^{N} t_{ik}.$$

The last equality follows from Lemma 3 and the telescoping sum. So, given N, the WSP becomes

(MIP-WSP)

$$\text{maximize} \sum_{i=1}^{m} \pi_i p_i \sum_{k=1}^{N} t_{ik} \tag{10}$$

$$\text{subject to (1)–(3), (5)–(9).}$$

Perhaps, a simpler way of understanding (10) is to note that because of the lost sales assumption, the zero ending inventory property, and the unit demand rate transformation, the total stockout time for an item is equal to the cycle length T, minus its initial inventory J_i, minus its total production $p_i \sum_k t_{ik}$.

We now obtain an upper bound on N.

Lemma 4. Assume without loss of generality that $s_1 \leqslant s_2 \leqslant \ldots \leqslant s_m$. An upper bound on the total number of positions in the sequence for the WSP is $N = \max(N_1, N_2)$ where $N_1 = 2\lfloor T/(s_1 + s_2) \rfloor$ and $N_2 = 2\lfloor (T + s_2)/(s_1 + s_2) \rfloor - 1$.

Proof. Similar to the proof of Lemma 5.

2.2. WSP-permutation

We restrict to the class of schedules which allow at most one setup for each product. The problem remains NP-hard. The MIP formulation is similar with the additional constraints

$$\sum_{k=1}^{m} y_{ik} \leqslant 1 \qquad i = 1, \ldots, m. \tag{11}$$

An upper bound on the total number of positions in the MIP under 11 is clearly m.

2.3. Stockout Avoidance Problem (SAP)

A slight modification of the MIP formulation MIP_WSP for the WSP can be used to solve the SAP. The idea is as follows. Use equal stockout penalties, say $\pi_i = 1$ for all i. Solve the LP-relaxation of MIP_WSP. If the objective value of the LP-relaxation is less than $\sum_{i=1}^{m}(T - J_i)$, stop, there is no feasible schedule, since the optimal value of the LP relaxation is an upper bound on the optimal value of MIP_WSP. Else, solve MIP_WSP. If the optimal objective value is $\sum_{i=1}^{m}(T - J_i)$, then we have a schedule that avoids stockout up to time T. Else, there exists no feasible schedule. An upper bound on N can be obtained using the following Lemma.

Lemma 5. Assume without loss of generality that $s_1 \leqslant s_2 \leqslant \ldots \leqslant s_m$. An upper bound on the total number of positions in the sequence for the SAP is $N = \max(N_1, N_2)$ where

$$N_1 = 2\left\lfloor \frac{\{\kappa T + \mathbf{r}'\mathbf{J} - (s_3 + \cdots + s_m)\}}{(s_1 + s_2)} \right\rfloor + m - 2,$$

$$N_2 = 2\left\lfloor \frac{\{\kappa T + \mathbf{r}'\mathbf{J} - (s_3 + \cdots + s_m) + s_2\}}{(s_1 + s_2)} \right\rfloor + m - 3.$$

Proof. From Lemma 3, we only need to seek a feasible schedule such that $J_i^{N+1} = 0$ for all i. Also the production time required to avoid stockouts is independent of the number of setups needed to avoid stockouts. Indeed, $t_i \equiv (T - J_i)/p_i$ is the production time required to avoid stockouts for item i. Consequently, the maximum time available for setups is

$$T - \sum_{i=1}^{m} t_i = T - \sum_{i=1}^{m} \frac{T - J_i}{p_i} = T\left(1 - \sum_{i=1}^{m} \frac{1}{p_i}\right) + \sum_{i=1}^{m} \frac{J_i}{p_i} = \kappa T + \mathbf{r}'\mathbf{J}.$$

We consider how many setups can fit into T. Let α_1 and α_2 be the maximum number of repetitions of s_1 and s_2 which follows a certain pattern. Each product must setup at least once. The largest possible number of setups is given by a pattern in which the two products with the smallest setups are alternatively produced, and all others are produced only once.

There are two possible cases.

(a) $\{s_1, s_2, s_1, s_2, \ldots, s_1, s_2, s_3, s_4, \ldots, s_m\}$. The order of the setups for the products $3, \ldots, m$ is arbitrary. Let α_1 be the maximum number of repetitions of s_1 in the above pattern. Then, $\alpha_1(s_1 + s_2) + s_3 + \cdots + s_m \leqslant \kappa T + \mathbf{r}'\mathbf{J}$ and α_1 is an integer. Consequently,

$$\alpha_1 = \left\lfloor \frac{[\kappa T + \mathbf{r}'\mathbf{J} - (s_3 + \cdots + s_m)]}{(s_1 + s_2)} \right\rfloor.$$

This pattern requires at most $N_1 = 2\alpha_1 + m - 2$ positions in the sequence.

(b) $\{s_1, s_2, s_1, \ldots, s_2, s_1, s_3, s_4, \ldots, s_m\}$. The order of the setups for the products $3, \ldots, m$ is arbitrary. Let α_2 be the maximum number of repetitions of s_1 in the above pattern. Then, $\alpha_2 s_1 + (\alpha_2 - 1)s_2 + s_3 + \cdots + s_m \leqslant \kappa T + \mathbf{r}'\mathbf{J}$ and α_2 is an integer. Consequently,

$$\alpha_2 = \left\lfloor \frac{[\kappa T + \mathbf{r}'\mathbf{J} - (s_3 + \cdots + s_m) + s_2]}{(s_1 + s_2)} \right\rfloor.$$

This pattern requires at most $N_2 = 2\alpha_2 + m - 3$ positions in the sequence.

From (a) and (b), an upper bound on the total number of positions in the MIP formulation is $N = \max(N_1, N_2)$. □

Remark. We can add the following constraint for the SAP since the maximum time available for setups is explicitly calculated. The constraint can be used to eliminate all sets of frequencies for which there is not enough time to avoid stockouts

$$\sum_{i=1}^{m} \sum_{k=1}^{N} s_i y_{ik} \leqslant \kappa T + \mathbf{r}'\mathbf{J}.$$

Example 1. We use Anderson's [1] example with $T = 40$. The data are shown in Table 1. From Lemma 5, an upper bound on the total number of positions in the sequence is 8. Solving MIP_WSP, we get an objective value of 68 which corresponds to 0 for the minimization problem. The schedule

Table 1. Data for Anderson's example

Product number	Demand rate	Production rate	Setup time	Initial inventory
1	1.0	6.0	3.0	14.0
2	1.0	2.0	1.0	11.0
3	1.0	10.0	5.0	27.0

is given by $\mathbf{f} = (1, 2, 3, 2)$ and $\mathbf{t} = (4.333, 4.633, 1.300, 9.866)$. So, we have a schedule that avoids stockouts up to $T = 40$. For $T = 70$, a feasible schedule is $\mathbf{f} = (1, 2, 3, 1, 2, 3, 1, 2)$ and $\mathbf{t} = (2.451, 11.722, 2.536, 4.736, 12.911, 1.764, 2.147, 4.867)$.

3. SPECIAL CASES OF THE SAP AND THE WSP

3.1. Given a production sequence, both problems can be solved by LP

Given a production sequence, the problem is that of finding production run times. We can formulate this as a LP by assigning specific values to the zero-one variables in the MIP_WSP. The LP can be used to test if a particular sequence is feasible (SAP) or if its associated cost is within a reasonable bound (WSP).

3.2. Solving the SAP when production sequences are restricted to permutations

From Lemma 3 in the preceding section, we only need to seek a feasible schedule such that the inventory at time T is 0. Consequently, a necessary condition for the existence of a feasible schedule is to produce each product $t_i \equiv (T - J_i)/p_i$ time units. Also, the sum of production and setup times must be at most T. Thus, a necessary condition for feasibility is $T \geqslant (\mathbf{e}'\mathbf{s} - \mathbf{r}'\mathbf{J})/\kappa$. If this condition is satisfied, we show that the resulting problem is equivalent to the $m/1/T_{\max}$ scheduling problem [2]. The $m/1/T_{\max}$ scheduling problem is a single machine scheduling problem whose objective is to minimize the maximum tardiness of m jobs. This idea was used by Anderson [1] to develop a heuristic for the SAP. A similar transformation is used to prove the NP-hardness of the WSP, by reduction to the Weighted Tardiness Problem.

Proposition 1. Given $J_i, s_i, p_i, t_i, i = 1, \ldots, m$, there exists a feasible permutation schedule that avoids stockouts if and only if the minimum value of the $m/1/T_{\max}$ problem is 0, where the due dates are $J_i + t_i$ and the job lengths are $s_i + t_i$, $i = 1, \ldots, m$.

Proof. Assume, without loss of generality, that a feasible sequence is $\{1, 2, \ldots, m\}$ and let S_i be the start time for run i. Then from the feasibility of this cyclic schedule, we have $S_i \leqslant J_i$ for all i. Arrange the jobs according to the sequence $\{1, \ldots, m\}$. Obviously $S_i + t_i \leqslant J_i + t_i$ for all i, implying that the jobs are completed before their due dates. Now assume, without loss of generality, that the sequence $\{1, \ldots, m\}$ has no tardy jobs. Let F_i be the finishing time of job i. Then, from the fact that tardiness is 0, we have $F_i \leqslant J_i + t_i$ for all i. Obviously $F_i - t_i \leqslant J_i$ for all i which means that the initial inventories do not run out before the start of their production runs. □

To check the existence of a feasible cyclic schedule we solve an $m/1/T_{\max}$ problem with the appropriate transformation. This is done by sorting the products according to the Earliest Due Date (EDD) rule (see Baker [2]). It takes linear time to transform the data. Also, sorting using the EDD rule takes $O(m \log m)$, and checking the minimum value of maximum tardiness takes linear time. Consequently, the overall complexity is $O(m \log m)$. The overall algorithm is:

Algorithm SAP-Permutation

Step 1. If $T \geqslant (\mathbf{e}'\mathbf{s} - \mathbf{r}'\mathbf{J})/\kappa$, go to *Step 2*. Else stop, there is no feasible schedule.

Step 2. Transform into $m/1/T_{\max}$ scheduling problem via

$D_i(duedate) = J_i + t_i, L_i(job\ length) = s_i + t_i, i = 1, \ldots, m.$
Solve $m/1/T_{\max}$ problem by the EDD rule.
If optimum objective value of $m/1/T_{\max} = 0$,
there exists a feasible schedule.
Else, there is no feasible permutation schedule.

In practice, one should first run this algorithm. If it stops at *Step 1*, there is no feasible schedule. At this point, one may select stockout penalties and use a heuristic for the WSP. If in *Step 2*, the maximum tardiness is positive, then only a non-permutation sequence may be feasible.

Example 2. The data are as in Example 1 with $T = 40$. We restrict to permutation sequences. Step 1 is satisfied since $(\mathbf{e}'\mathbf{s} - \mathbf{r}'\mathbf{J})/\kappa \leqslant T$. The transformation to $m/1/T_{\max}$ yields $\mathbf{D} = (18.\bar{3}, 25.5, 28.3)$ and $\mathbf{L} = (7.\bar{3}, 15.5, 6.3)$. Using the EDD rule, the optimum value is $0.86\bar{3}$, so at least one job is tardy. Consequently, there exists no feasible permutation schedule. □

4. HEURISTICS FOR THE WSP

Since the WSP is NP-hard, the MIP cannot be used when the number of products is large. We present two heuristics for the WSP that generate permutation schedules and then extend the ideas to a general heuristic for the WSP. Note that the LP formulation in Section 3 could also be used to obtain optimal production runs for the resulting sequence.

4.1. Heuristics for the WSP under permutation schedules

Since each product is setup at most once, the key role of the heuristic is the choice of the production sequence. We give two heuristics which are similar except in the choice of the product to be produced next. After we obtain a sequence, we perform adjacent pairwise interchanges to squeeze out better schedules.

Heuristic 1 (Savings Heuristic)

This heuristic chooses the next product by selecting the one with largest value of (savings for the product − total loss for other products)/(invested time). By savings we mean the weighted stockout if the product is not setup immediately. The total loss for the other products is the sum of their weighted stockouts. The invested time is the setup plus the production time for the selected product. The complexity of this heuristic is $O(m^2)$ before the adjacent pairwise interchanges and $O(m^3)$ after the adjacent pairwise interchanges.

Step 0. (Initialization.)

$$J_i' \equiv J_i, \quad T' \equiv T, \quad C \equiv \{1, \ldots, m\}, \quad \forall i.$$

Step 1. (Compute the production time for each product.)

$$t_i \equiv \{T' - \max(J_i', s_i)\}/p_i, \quad \forall i. \qquad \text{If } t_i \leqslant 0,\ C \leftarrow C \setminus \{i\}.$$

Step 2. (Choose the product to be produced.)

$$i^* = argmax_{i \in C}\{(\pi_i p_i t_i - \textstyle\sum_{j \in C, j \neq i} \pi_j \max[0, s_i + t_i + s_j - J_j])/(s_i + t_i)\}.$$

Produce product i^* for t_{i^*}.

Step 3. (Update the inventories, the time horizon, and the set of unscheduled products.)

$$J_j' \leftarrow \text{Max}(J_j' - s_{i^*} - t_{i^*}, 0), \quad j \neq i. \quad J_{i^*}' \leftarrow \text{Max}(J_{i^*}' - s_{i^*}, 0) + (p_{i^*} - 1)t_{i^*}.$$

$$T' \leftarrow T' - (s_{i^*} + t_{i^*}). \qquad C \leftarrow C \setminus \{i^*\}.$$

Step 4. (Stopping criteria.)

If $T' \leqslant 0$ or $C = \phi$, then go to *Step 5*. Else go to *Step 1*.

Step 5. (Adjacent pairwise interchanges.)

Perform adjacent pairwise interchanges and find a better solution.

Heuristic 2 (Greedy Heuristic)

This heuristic is the same as Heuristic 1 except for *Step 2*. This heuristic chooses the product which has the largest value of (savings/invested time). The meanings of savings and invested time are as those in Heuristic 1.

For the WSP, the objective is to minimize the weighted stockout times during the planning horizon $[0, T]$. The problem is equivalent to maximizing $\sum_{i=1}^{m} \pi_i p_i \sum_{k=1}^{N} t_{ik}$. This heuristic can be called greedy, since our objective is to maximize $\sum_i \pi_i p_i t_i$ in the MIP formulation without

Table 2. Distributions for data and computational results for test problems

Problem data	Dense	Scattered
Setup time	[1, 2]	[0, 3]
Production rate	[4, 5]	[3, 6]
Initial inventory	[1, 3]	[0, 4]
Stockout penalty	[10, 20]	[10, 50]

Computational results	Heuristic 1		Heuristic 2	
	before a.p.i.	after a.p.i.	before a.p.i.	after a.p.i.
Mean ratio	1.006	1.001	1.008	1.001
Maximum ratio	1.059	1.057	1.085	1.031
Number of problems with ratio = 1 among 80 problems	53	72	48	75

considering the effects of the constraints. The complexity of this heuristic is $O(m)$ before the adjacent pairwise interchanges and $O(m^3)$ after the adjacent pairwise interchanges.

Step 2. (Choose the product to be produced.)

$i^* = argmax_{i \in C}\{\pi_i p_i t_i/(s_i + t_i)\}$. Produce product i^* for t_{i^*}.

We used the above two heuristics on 16 sets of problems (5 problems in each set). As in a factorial design, different distributions of s_i, p_i, J_i, and π_i were used. The number of products in the test problems was 4. The length of the planning horizon was generated from uniform distribution on [10, 30]. The data sets were generated randomly from uniform distributions on the given intervals. Table 2 shows distributions for the data sets.

The different factor distributions were mixed in all possible combinations to yield 16 problem types. An example of one problem type is: 4 products, each with dense setup time distribution, production rate distribution, and initial inventory distribution, and with scattered stockout penalty distribution.

Table 2 also shows the results for these runs. Several tables would be required to display the 16 different distributions that were used in the study. For the sake of brevity the detailed results are omitted here. The ratio in the table is (total weighted stockout costs using heuristic/minimum weighted stockout costs using MIP). In order to see the performance of the heuristics without adjacent pairwise interchange (a.p.i.), we also show the results before we apply the a.p.i. The last row in the table tells the number of problems (among 80 test problems) in which the heuristic solution is optimal.

The results are fairly good even before we apply the a.p.i. Even though the a.p.i. does not improve performances significantly in the 4 product case, its impact is expected to be greater when the number of products is large. Moreover, the LP can be run on the resulting sequence to obtain the minimum weighted stockout.

4.2. Heuristics for the WSP

The problem is more involved than the WSP-permutation because of the need to compute production frequencies. We vary the frequencies within a range and compute the cost of a schedule based on each set of frequencies. We then use the LP formulation developed in Section 3 on the production sequence of the least cost schedule to obtain optimal run times for that sequence.

Heuristic 3

This heuristic is a generalization of Heuristic 1 for the WSP. The complexity from *Step 1* to *Step 5* is $O(m^2)$. An upper bound on the repetition of l is $\delta \equiv [T(1-\kappa) - \mathbf{r}'\mathbf{J}]/\mathbf{e}'\mathbf{s}$. Consequently, the total complexity is determined by maximum of $O(\delta m^2)$ and the complexity of LP.

Step 0. (Initialization)

$$J_i' \equiv J_i, \quad T' \equiv T, \quad C \equiv \{1, \ldots, m\}, \quad t_i \equiv (T - \max(J_i, s_i))/p_i, \quad \forall i.$$

Do $l = \lfloor(\kappa T + \mathbf{r}'\mathbf{J})/\mathbf{e}'\mathbf{s}\rfloor + 1$ to $\lfloor T/\mathbf{e}'\mathbf{s}\rfloor$

Table 3. Distributions for data and computational results for test problems

	Dense	Scattered
Problem data		
Setup time	[2, 4]	[1, 5]
Production rate	[4, 5]	[3, 6]
Initial inventory	[1, 3]	[0, 4]
Computational results		
Mean ratio	1.117	1.105
Minimum ratio	1.001	1.025
Maximum ratio	1.284	1.175

Step 1. (Compute the production time for each product.)

$$t_i' = \text{Min}\{[T' - \max(J_i', s_i)]/p_i, t_i/l\}, \quad \forall i.$$

Step 2. (Choose the product to be produced.)

$$i^* = argmax_{i \in C}\{(\pi_i p_i t_i' - \textstyle\sum_{j \in C, j \neq i} \pi_j \max[0, s_i + t_i' + s_j - J_j'])/(s_i + t_i')\}.$$

Produce product i^* for t_{i^*}'.

Step 3. (Compress.)
Combine consecutive setups if they exist.

Step 4. (Update the inventories, compute the cumulative stockout cost, the time horizon, and the set of unscheduled products.)

$$J_j' \leftarrow \text{Max}(J_j' - s_{i^*} - t_{i^*}', 0), \quad j \neq i. \quad J_{i^*}' \leftarrow \text{Max}(J_{i^*}' - s_{i^*}, 0) + (p_{i^*} - 1)t_{i^*}'.$$

Compute cumulative stockout costs.

$$T' \leftarrow T' - (s_{i^*} + t_{i^*}'). \qquad \text{If } \max(J_i', s_i) \geqslant T', \text{ then } C \leftarrow C \setminus \{i\}.$$

Step 5. (Stopping criteria.)
If $T' > 0$ and $C \neq \phi$, then go to *Step 1.*

end

Step 6. (Optimization for the given sequence.)
Choose the sequence with smallest cost. Solve LP using the sequence to squeeze out a best solution for the sequence.

We used the above heuristic on 2 sets of problems (10 problems in each set). The number of products in the test problems was 3. The length of the planning horizon and the stockout penalty were generated from uniform distributions on [10, 30] and [10, 20], respectively. The reason that we do not use a spread distribution for stockout penalty such as [10, 50] is to avoid trivial situations which result in permutation schedules. The data sets were generated randomly from uniform distributions on the given intervals. Table 3 shows the distributions for the data sets.

Table 3 also shows the results for these runs. The ratio in the table is (total weighted stockout costs using heuristic/minimum weighted stockout costs using MIP). The mean and minimum ratios are similar for both the dense and the scattered cases. The maximum ratio is larger for the dense case suggesting that those problems are harder.

5. PHASING INTO A CYCLIC SCHEDULE

5.1. WSP when the target inventories are given

Suppose that we want to phase into a target cyclic schedule by time T. The cyclic schedule can be either an optimal rotation (Common Cycle) schedule or any heuristic schedule, for instance a schedule obtained by a time-varying lot size heuristic [3]. In either case, we can compute the target inventories, say I_i's, for the cyclic schedule. Then MIP formulation for this problem is nearly identical to MIP_WSP, except that equation (9) is modified to read

$$J_i^{N+1} = I_i \qquad i = 1, \ldots, m.$$

5.2. Phasing into a rotation schedule with minimum aggregate inventory

When the facility suffers a disruption that results in inventory levels that are lower than desirable, our main concern over the short run is to avoid stockouts. Over a longer time horizon, we are also concerned about the holding and setup costs. In this section we address the problem of scheduling the facility to minimize stockout penalties over a transition period, say $[0, T]$, and thereafter follow an optimal (with respect to holding and setup costs) rotation schedule. There are $m!$ different average cost optimal rotation schedules, one for every order in which the products are setup for production in the facility. During the transition period, we need to bring the product's inventory levels up to levels from which one of these $m!$ optimal rotation schedules can be followed. Phasing into different rotation schedules will impose different production requirements over the transition period. If stockouts are to be avoided, and we are to effect a transition from the initial inventories J_i's, to the target inventories, say I_i's, of a rotation schedule, then the facility must produce the set of products for $\sum_{i=1}^{m} r_i(T - J_i + I_i)$ units of time during the transition period. Since r_i's, J_i's and T are data, minimizing the required production times is equivalent to minimizing $\sum_{i=1}^{m} r_i I_i$. This is accomplished by sorting the items in ascending order of the production rates multiplied by the setup times; see Proposition 2 below. We call this the Slowest Processing Rate (SPR) rule since it schedules the slowest production first when everything else is equal. One may argue that this choice may not be adequate if by chance the J_i's are already in the inventory path traced by an optimal rotation schedule. However, this is unlikely if the J_i's are the inventories that result from a disruption that throws the inventories off from such a cyclic schedule. Moreover, even if this were not the case, checking whether the J_i's lie in the path of an optimal rotation schedule can be very difficult. In any case, what we propose here is a sensible selection of one of the $m!$ rotation schedules. This selection is based on minimizing the burden on the facility over the transition period. Furthermore, such selection need only be done once, since sorting of the items is independent of T and of J.

Proposition 2. (SPR Rule) Aggregate inventory is minimized by the SPR sequence

$$s_{[1]} p_{[1]} \leqslant s_{[2]} p_{[2]} \leqslant \ldots \leqslant s_{[m]} p_{[m]}.$$

Proof. We use a pairwise interchange argument. Without loss of generality, let $S = \{1, 2, \ldots, m\}$ be a sequence. Suppose S is not following the SPR rule. That is, somewhere in S there exists a pair of adjacent products, i and j, with j following i, such that $s_{[i]} p_{[i]} > s_{[j]} p_{[j]}$. Now construct a new sequence, S', in which jobs i and j are interchanged in sequence and all other jobs are the same as in S. We adopt the notations $A(S)$ and $A(S')$ to represent the aggregate inventory under schedule S and S', respectively. We then show that $A(S')$ is smaller than $A(S)$

$$A(S) = \sum_{k=1}^{m} r_k I_k = \sum_{k<i} r_k I_k + \sum_{k>j} r_k I_k + r_i \left(\sum_{k \leqslant i} s_k + T \sum_{k<i} r_k \right) + r_j \left(\sum_{k \leqslant j} s_k + T \sum_{k<j} r_k \right).$$

$$A(S') = \sum_{l=1}^{m} r_l I_l = \sum_{l<j} r_l I_l + \sum_{l>i} r_l I_l + r_j \left(\sum_{l \leqslant j} s_l + T \sum_{l<j} r_l \right) + r_i \left(\sum_{l \leqslant i} s_l + T \sum_{l<i} r_l \right).$$

Therefore

$$A(S) - A(S') = r_j s_i + T r_j r_i - r_i s_j - T r_i r_j = r_j s_i - r_i s_j > 0.$$

In other words, the interchange of products i and j reduces the value of A. Therefore any sequence that is not the SPR sequence can be improved with respect to A by such an interchange of an adjacent pair of products. It follows that the SPR sequence itself must be optimal. □

After we sequence the items, the target inventories I_i's are easily computed. Then we can apply a modified Mixed Integer Program to solve the WSP for the finite horizon and we can follow a rotation schedule thereafter. We summarize the procedure as follows:

Algorithm Phase

Step 1. Sort the items by the SPR rule.
Step 2. Find an optimal rotation schedule based on the sequence of step 1.
Step 3. Compute the initial inventories of the rotation schedule found in Step 2.
Step 4. Solve the WSP using the modified Mixed Integer Program.

Example 3. We want to find a schedule which minimizes stockout penalties (if any) during 30 days. After that we will follow a rotation schedule. The data are as follows: $m = 3$, $\mathbf{p} = (6, 2, 10)$, $\mathbf{s} = (3, 1, 5)$, $\mathbf{J} = (8, 20, 17)$, $\pi = (1, 1, 1)$, $\mathbf{a}$ ($\equiv$ setup cost) $= (20, 10, 100)$, $\mathbf{h}$ ($\equiv$ holding cost) $= (0.1, 0.2, 0.3)$.

Using the SPR rule, we obtain a rotation schedule with sequence $(2, 1, 3)$. The target inventories are easily computed using the common cycle length and rotation schedule: $\mathbf{I} = (23.285, 1, 29.713)$. Solving the modified MIP, we obtain a schedule that avoids stockout during 30 days starting from given initial inventories $\mathbf{J}$ and recovers to the target inventory levels $\mathbf{I}$. The sequence and production times are: $\mathbf{f} = (1, 3, 2, 1)$, $\mathbf{t} = (2.7543, 4.2713, 5.5000, 4.7932)$. □

6. CONCLUDING REMARKS

This paper presented a MIP formulation for the WSP and applied the formulation to the SAP. We showed that some special cases of the WSP and the SAP can be solved in polynomial time. We also demonstrated how the WSP can be coupled with a cyclic schedule after a finite horizon. These models can be used as a practical scheduling tool for temporarily overloaded facilities until a normal load condition is recovered or to phase into a target cyclic schedule after a disruption.

In some flexible manufacturing systems, group technology principles divide items naturally into families (groups) so that substantial setups occur only when switching production between families. That is, if production is switched from one item to another in the same family, only a minor intra-family setup is required. If, however, production is switched to an item outside the family, then a major inter-family setup is required. We can extend the results in this paper to the multi-family case.

Acknowledgement—The authors are grateful to the useful suggestions made by Dr Edward Anderson, Sangjin Choi, and two anonymous referees on an earlier version of the paper.

REFERENCES

1. E. Anderson, Testing feasibility in the lot scheduling problem. *Ops Res.* **38,** 1079–1088 (1990).
2. K. Baker, *Introduction to Sequencing and Scheduling.* Wiley, New York (1974).
3. G. Dobson, The economic lot-scheduling problem: achieving feasibility using time-varying lot sizes. *Ops Res.* **35,** 764–771 (1987).
4. G. Gallego, Scheduling the production of several items with random demands in a single facility. *Mgmt Sci.* **36,** 1579–1592 (1990).
5. G. Gallego and I. Moon, The effect of externalizing setups in the economic lot scheduling problem. *Ops Res.* **40,** 614–619 (1992).
6. G. Gallego and I. Moon, Strategic investments to reduce setup times in the economic lot scheduling problem. *Nav. Res. Logist.* **42,** 773–790 (1995).
7. G. Gallego and R. Roundy, The economic lot scheduling problem with finite backorder costs. *Nav. Res. Logist.* **39,** 729–739 (1992).
8. I. Moon, Multiproduct economic lot size models with investment costs for setup reduction and quality improvement: review and extensions. *Int. J. prod. Res.* **32,** 2795–2801 (1994).
9. P. Zipkin, Computational optimal lot sizes in the economic lot scheduling problem. *Ops Res.* **39,** 56–63 (1991).