Journal of the Korean
OR/ MS Society
Vol. 17, No. 3, December 1992

Multiple Product Single Facility Stockout Avoidance Problem (SAP) and Weighted Stockout Problem (WSP)

Ilkyeong Moon*

Abstract

We study 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 linerarly is also studied. We call this problem the Weighted Stockout Problem (WSP). Both problems are NP-hard in the strong sense. We develop Mixed Integer Linear Programming (MIP) formulations for both the SAP and the WSP. In addition, several heuristic algorithms are presented and performances are tested using computational experiments. We show that there exist polynomial algorithms for some special cases of the SAP and the WSP. We also present 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.

1. Introduction

We study the Multiple Product Single Facility Stockout Avoidance Problem (SAP), which was proven NP-hard in the strong sense for both finite and infinite horizons by Arkin et. al [2]. See also Anderson [1]. We also study an optimization version of the finite horizon 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). Arkin et. al [2] gave a

* Department of Industrial Engineering, Pusan National University

simple proof that WSP is NP-hard in the strong sense by reduction to the Weighted Tardiness Problem. (See Baker [3] for the definition of the Weighted Tardiness Problem.)

Both the SAP and the WSP are closely related to the Economic Lot Scheduling Problem (ELSP). The ELSP is the problem of scheduling the production of several items in a single facility so that demands are met without stockouts or backorders and the long run average inventory carrying and setup costs are minimized. This problem occurs in many production situations (Boctor [4]), for example in

(a) Metal forming and plastic production lines (press lines and plastic and metal extrusion machines) where each product requires a different die to set up on the machine.
(b) Assembly lines which produce several products and/or different product models (electric appliances, motor cars, etc.)
(c) Blending and mixing facilities (paints, beverages, animal food, etc.) in which different products are poured into different containers.
(d) Weaving production lines (textiles, carpets, etc.) in which the main products are manufactured in different colors, widths, and grades.

The ELSP has been widely studied over 35 years. See Dobson [7], Gallego and Moon [9], Glass [11], Moon et. al [14], Moon and Hwang [15] for recent contributions. In the ELSP it is typically assumed that production and demand rates are known item dependent constants, and that setup times and setup costs are known item dependent, but sequence independent constants. In addition, research in the ELSP has focused on *cyclic schedules*, i.e. schedules that are repeated periodically. Moreover, almost all researchers have restricted attention to cyclic schedules that satisfy the Zero Switch Rule (ZSR).This rule states that a production run for any particular item can be started only if its physical inventory is zero. Counterexamples to the optimality of this rule have been found, but they tend to be pathological in nature. It is difficult to construct reasonable examples where the ZSR does not give a near-optimal solution.

There are two main approaches for heuristic algorithms. One is the basic period approach. In addition to the ZSR this approach requires every item to be produced at equally-spaced intervals of time that are multiples of a basic time period. (This together with the ZSR implies that every item is produced in equal lot sizes.) Most of the heuristic algorithms that follow this approach first select the ***frequency*** (i.e. number of production runs per cycle) with which each item is to be produced, and then search for a feasible schedule that implements these frequencies. See Elmaghraby [8] for an excellent review. Under this approach it is NP-hard to determine the existence of a feasible schedule (see Hsu [12]). These difficulties have led some researchers to reject the basic-period paradigm, in particular the requirement of equally spaced production lots.

The time-varying approach which relaxes the restriction of equally spaced producion runs was initiated by Maxwell [13] and Delporte and Thomas [5]. Dobson [6] shows that any *production sequence* (i.e. the order in which the items are produced in a cycle) can be converted into a feasible schedule in which the quantities and timing of production lots are not necessarily equal provided that the proportion of time available for setups is postive. He also developed a bin-packing heuristic to determine a sensible production sequence. Optimal or near optimal schedules can be obtained by combining Dobson's heuristic with Zipkin's [16] algorithm. Recently, Gallego and Xiao [10] showed that the ELSP is NP-hard under this approach even without the ZSR restriction.

Research on the ELSP has concentrated on *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 would be to avoid stockouts (as far as it is possible) until emergency relief becomes available 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, leaves the inventory levels critically low.

This gives rise to the SAP:the problem of finding a schedule, if such exists, that avoids stockouts over a given finite horizon from a given configuration of initial inventories. The WSP is an optimization version of the SAP. It 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. 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.

Over an infinite horizon we are also concerned with holding and setup costs. For this reason we only consider the finite horizon versions of the SAP and WSP where these costs are ignored. We show how the WSP can be used to phase into target inventories from which an efficient (in terms of holding and setup costs) cyclic schedule can be followed. In this way the 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.

Anderson [1] considered the SAP over an infinite horizon and proposed a heuristic solution. His heuristic distregards cost information and attempts to find a cyclic schedule where the initial inventory levels are reached again at some future time. We have identified several weak points in Anderson's approach. First, the heuristic clearly fails if all the inventories are sufficiently low, and Anderson does not offer any advice on what to do in this case. Second, Anderson's approach is based on the idea of finding a cyclic schedule that recovers the initial inventories. Thus, the heuristic may call for an intricate sequence of production runs to get out of trouble only to go back into it. Third, Anderson tries to find a solution over the infinite horizon, where it may suffice to stay out of trouble until the end of the week or until another machine that is currently under repair is on working conditions. Fourth, the approach ignores cost and so allows production sequence that may be economically undesirable. We feel that it is more sensible to phase into a good target schedule where setup and inventory carryiong costs are also taken into account. The heuristic could not be tested for effectiveness since Anderson had no machinery to check whether there exists a feasible schedule.

We develop a Mixed Integer Linear Programming (MIP) formulation for the WSP. A similar formulation is applied to the SAP after some necessary conditions are verified. We show that some special cases of the SAP and 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, the SAP can be solved in polynomial time. But the WSP remains NP-hard! Here, permutation sequence means that the facility setups at most once for each product and **f** denotes a general sequence. Table 1 and 2 show the relationships between the SAP and the WSP.

The rest of this paper is organized as follows. In Section 2, we prove basic properties of optimal schedules and formulate the WSP. We formulate the WSP when the sequence is restricted to permutations and give the relationship between the solution procedures for the

Table 1. Complexity of the WSP

sequence restriction	complexity	solution method
none	NP-hard (Arkin et. al [2])	MIP (this paper)
permutation sequence	NP-hard (Arkin et. al [2])	MIP (this paper)
given **f**	polynomial (this paper)	LP (this paper)

Table 2. Complexity of the SAP

sequence restriction	complexity	solution method
none	NP-hard (Arkin et. al [2]) (Anderson [1])	MIP (this paper)
permutation sequence	O(mlogm) (this paper)	yes (this paper)
given **f**	polynomial (this paper)	LP (this paper)

WSP and 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 data for the WSP and SAP-finite are:

the index for the products		$i=1,\cdots,m$
constant demand rates	d_i	$i=1,\cdots,m$
constant production rates	p_i	$i=1,\cdots,m$
known setup times	s_i	$i=1,\cdots m$
known initial inventories	J_i	$i=1,\cdots m$
known stockout penalties	ρ_i	$i=1,\cdots m$
length of planning horizon	T	

The demand rates 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 $\mathbf{r}\equiv(\frac{1}{p_1},\cdots,\frac{1}{p_m})'$, $\mathbf{J}\equiv(J_1,\cdots,J_m)'$ and $\mathbf{e}\equiv(1,\cdots,1)'$. Define $\kappa\equiv1-\mathbf{e}'\mathbf{r}$.

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. 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:

$J_i<T \qquad i=1,\cdots,m$

$T>s_i>0 \qquad i=1,\cdots,m$

$p_i>1 \qquad i=1,\cdots,m$

If a product's initial inventory is enough to satisfy demand over the horizon, i.e. $J_i \geq T$, then we can avoid stockouts for the product without producing it. If $s_i=0$ for all i and $\kappa>0$, both WSP and 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.

2.1 Weighted Stockout Problem (WSP)

Here bold face represents a vector. 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 $\mathbf{f}$ (the sequence may contain repetitions) and a vector of production times $\mathbf{t}$ of dimension compatible with $\mathbf{f}$, so that the weighted stockout times over the planning horizon T are minimized.

We can easily show that there exists an optimal schedule that does not idle except possibly after the last production run. Consequently, we do not need to include idle times in the formulation.

2.1.1 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.

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

Let

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

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

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

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

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

For convenience, define

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

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

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

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

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

Let S^k be the ***starting time*** of the k_{th} production run ***after*** the setup. Then

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

where for convenience, we let

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

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

$$x_i^k \geq 0, \qquad k=1,\cdots,N+1, \qquad i=1,\cdots m. \tag{6}$$

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

$$J_i^k \geq 0, \qquad k=0,\cdots,N+1, \qquad i=1,\cdots,m \tag{7}$$

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 Figure 1), we have

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

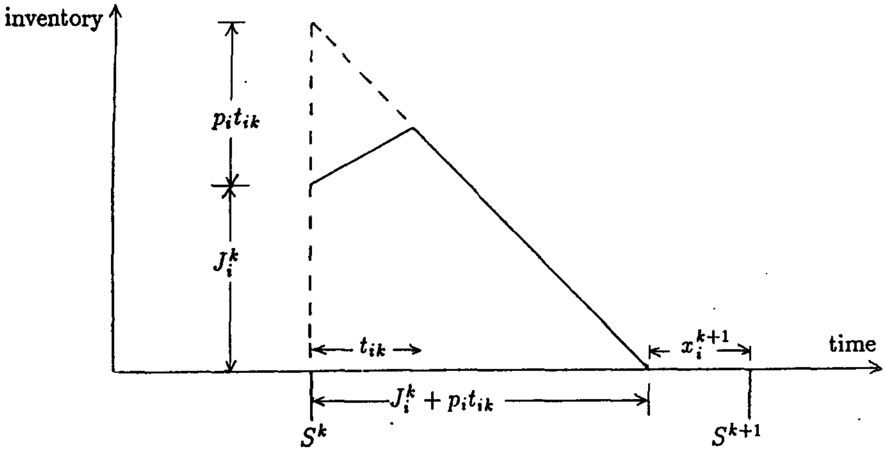

Figure 1. Relationship between inventory, stockout, production and demand

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

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

Proof. Suppose that $J_i^k>0$, $y_{ik}=1$, and that item i was produced last in position $l<k$. Let $\delta = min\ \{J_i^k/(p_i-1),\ t_{il}\}$, reduce t_{il} by δ and increase t_{ik} by δ. (See Figure 2). This has no effect on the production runs before l or after k, while the intervening production runs are scheduled δ units earlier with no increase in the stockout times. If $\delta=J_i^k/(p_i-1)$, then $J_i^k=0$ in the new schedule so the result holds. If $\delta=t_{il}$, we eliminate the setup and the production run of item i in position l. We can repeat the procedure until we either eliminate all previous setups of item i or obtain a new schedule satisfying the property. ■

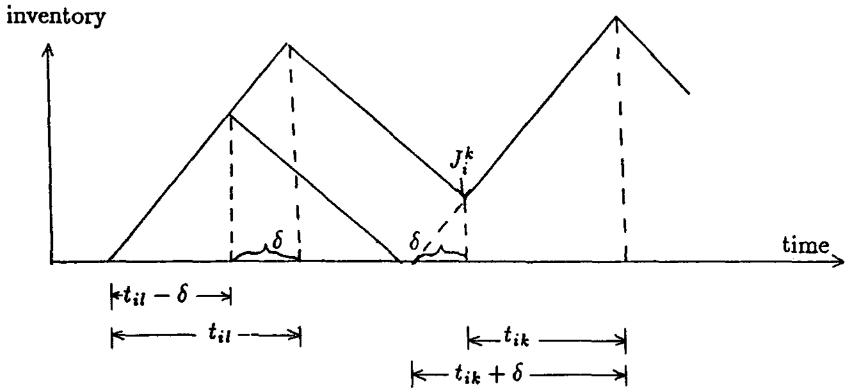

Figure 2. Production starts from zero inventory

Property 2. *There exists an optimal solution with final inventories equal to zero, i.e.,*

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

Proof. Suppose t^*, y^* constitutes an optimal solution and that $J_i^{N+1}>0$ for some i. Let $k>0$ be such that $y^*_{ik}=1$ and $y^*_{il}=0$ for $l>k$. Such a k must exist since $J_i^1<T$. Replace t^*_{ik} by $t'_{ik}=max(0,t^*_{ik}-\frac{J_i^{N+1}-T}{p_i-1})$. The resulting schedule reduces the final inventory of i retaining optimality. This process can be repeated until $J_i^{N+1}=0$ for all i. ■

Corollary 1. *Equation* (4) *is redundant.*

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

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

Since $x_i^{N+1}=(J_i^{N+1}+S^{N+1})-(J_i^N+S^N)-p_it_{iN}\geq 0$,

$$J_i^{N+1}+S^{N+1}\geq(J_i^N+S^N)+p_it_{iN}\geq S^N+p_it_{iN}$$
$$=\sum_{i=1}^{m}\sum_{k=1}^{N}(s_iy_{ik}+t_{ik})+(p_i-1)t_{iN}\geq\sum_{i=1}^{m}\sum_{k=1}(s_iy_{ik}+t_{ik})$$

By Property 1 and $S^{N+1}=T$, the above inequality reduces to equation (4). ◻

2.1.2 Objective Function

For the WSP, the objective is to minimize the weighted stockout times during the planning horizon $[0,T]$. So, giben N, the WSP becomes:

(MIP 1)

$$\text{minimize } \sum_{i=1}^{m}\rho_i\sum_{k=1}^{N+1}x_i^k$$
subject to (1)-(3), (5)-(9).

We now obtain an upper bound on N. Denote $[[x]]$ as the largest integer smaller than or equal to x.

Proposition 1. *Assume without loss of generality that* $s_1\leq s_2\leq\cdots\leq s_m$. *An upper bound on the total number of positions in the sequence for WSP is* $N=\max(N_1, N_2)$ *where*

$$N_1=2[[\frac{T}{(s_1+s_2)}]] \text{ and } N_2=2[[\frac{(T+s_2)}{(s_1+s_2)}]]-1.$$

Proof. 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. Since products cannot be produced consecutively, the worst case production pattern is the repetition of the two products with smallest setup times.

There are two possible cases.

(a) $\{s_1, t_1, s_2, t_2, s_1, t_3, \cdots, s_2, t_N\}$ which ends after setup for product 2. Let α_1 be the maximum number of repetitions of s_1 and s_2 which follows above pattern. Then, $\alpha_1(s_1+s_2)\le T$ and α_1 is an integer, consequently $\alpha_1=[[(\frac{T}{s_1+s_2})]]$. In this case, the total number of positions in the sequence is $N_1=2\alpha_1$.

(b) $\{s_1, t_1, s_2, t_2, s_1, t_3, \cdots, s_1, t_N\}$ which ends after setup for product 1. Let α_2 be the maximum number of repetitions of s_1 and s_2 which follows above pattern. Then, $\alpha_2 s_1+(\alpha_2-1)s_2\le T$ and α_2 is an integer, consequently, $\alpha_2=[[\frac{(T+s_2)}{(s_1+s_2)}]]$. In this case, the total number of positions in the sequence we need becomes $N_2=2\alpha_2-1$.

From (a),(b), we can get an upper bound on the total number of positions in the sequence we need. i.e. $N=\max(N_1, N_2)$. ■

2.2 WSP-permutation

We restrict to the schedules which allow at most one setup for each product. The problem remains NP-hard (Arkin et. al[2]). The MIP formulation is similar with the additional constraints

$$\sum_{k=1}^{m} y_{ik}\le 1 \qquad i=1,\cdots,m. \tag{12}$$

An upper bound on the total number of positions in the MIP under (12) is clearly m.

2.3 SAP

A slight modification of the MIP formulation MIP1 for the WSP can be used to solve the SAP problem. The idea is as follows. Use equal stockout penalties, say $\rho_i=1$ for all i. Solve the LP-relaxation of MIP1. If the objective value of the LP-relaxation is >0, stop, there is no feasible schedule, since the optimal value of the LP relaxation is a lower bound on the optimal value of MIP1. Else, solve MIP1. If the optimal objective value is 0, then we have a schedule that avoids stockout up to time T.

Proposition 2. *Assume without loss of generality that* $s_1\le s_2\le\cdots,\le s_m$. *An upper bound on the total number of positions in the sequence for SAP is* $N=\max(N_1,N_2)$ *where*

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

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

Proof. From Property 2, 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

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

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 set up 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) $\{s_1, s_2, s_1, s_2, \cdots, s_1, s_2, s_3, s_4, \cdots, s_m\}$. The order of the setups for the products $3, \cdots, m$ is arbitrary. Let α_1 be the maximum number of repetitions of s_1 in the pattern above. Then, $\alpha_1(s_1+s_2)+s_3+\cdots+s_m \le \kappa T+\mathbf{r}'\mathbf{J}$ and α_1 is an integer. Consequently, $\alpha_1=[[\frac{\{\kappa T+\mathbf{r}'\mathbf{J}-(s_3+\cdots+s_m)\}}{(s_1+s_2)}]]$. This pattern requires at most $N_1=2\alpha_1+m-2$ positions in the sequence.

(b) $\{s_1, s_2, s_1, \cdots, s_2, s_1, s_3, s_4, \cdots, s_m\}$. The order of the setups for the products $3, \cdots, m$ is arbitrary. Let α_2 be the maximum number of repetitions of s_1 in the pattern above. Then, $\alpha_2 s_1+(\alpha_2-1)s_2+s_3+\cdots+s_m \le \kappa T+\mathbf{r}'\mathbf{J}$ and α_2 is an integer. Consequently, $\alpha_2=[[\frac{\{\kappa T+\mathbf{r}'\mathbf{J}-(s_3+\cdots+s_m)+s_2\}}{(s_1+s_2)}]]$ This pattern requires at most $N_2=2\alpha_2+m-3$ positions in the sequence.

Form (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 cut for the SAP since the maximum time available for setups is explicitly calculated. The cut can be used to eliminate all sets of frequencies for which there is not enough time to avoid stockouts.

$$\sum_{i=1}^{m}(\sum_{k=1}^{N} y_{ik}) \le \kappa T+\mathbf{r}'\mathbf{J}$$

Example 1. We use Anderson's [1] example with $T=40$, The data are shown in the Table 3 Form Proposition 2, an upper bound of the total number of positions in the sequence is 8 Solving MIP1, we get an objective value of 0. The schedule is given by $\mathbf{f}=(1, 2, 3, 2)$ anc $\mathbf{t}=(4.333, 4.633, 1.300, 9.866)$. So, we have a schedule that avoids stockouts up to $T=40$. Fo $T=70$, a feasible schedule is $\mathbf{f}=(1, 2, 3, 1, 2, 3, 1, 2)$ and $\mathbf{t}=(2.451, 11.722, 2.536, 4.736, 12$ $911. 1.764. 2.147. 4.867)$.

Table 3. 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

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 formulate this as a Linear Program by assigning specific values to the zero-one variables in the MIP1. The LP can be used to test if a particular sequence is feasible or if its associated cost is within a reasonable bound.

Suppose $\mathbf{f}=(f_1, f_2,\cdots,f_N)$ is a given sequence. We can use the sequence information directly in the MIP1 to obtain an LP. t_k is the production time in position k. We obtain the following LP formulation.

WSP for a given sequence

(LP)

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

subject to

$$S^0=0,\ S^{N+1}=T$$

$$S^k=S^{k-1}+t_{k-1}+s_{f_k} \qquad k=1,\cdots,N$$

$$J_i^{k+1}-x_i^{k+1}=J_i^k+p_{f_k}t_k-(S^{k+1}-S^k) \qquad k=0,\cdots,N,\ \text{for all } i=f_k$$

$$J_i^{k+1}-x_i^{k+1}=J_i^k-(S^{k+1}-S^k) \qquad k=0,\cdots,N,\ \text{for all } i\neq f_k$$

$$t_0=0$$

$$J_i^0=J_i \qquad i=1,\cdots,m$$

$$t_k\geq 0 \qquad k=1,\cdots,N$$

$$x_i^k\geq 0 \qquad k=1,\cdots,N+1,\ i=1,\cdots,m$$

$$J_i^k\geq 0 \qquad k=1,\cdots,N+1,\ i=1,\cdots,m$$

The SAP with a given production sequence

If $C_f=0$, then the solution to the LP is a feasible schedule. Else, there exists no feasible schedule based on the sequence.

3.2 Solving the SAP when production sequences are restricted to permutations

From Property 2 in the precedings 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 $t_i \equiv \frac{(T-J_i)}{p_i}$ time units. Also, the sum of production and setup times must be at most T. Thus, a necessary condition for feasibility is $T \geq (\mathbf{e}'\mathbf{s}-\mathbf{r}'\mathbf{J}) / \kappa$. If this condition is satisfied, we show that the resulting problem is equivalent to the m/1/Tmax scheduling problem. This idea was used by Anderson [1] to develop a heuristic for SAP infinite. A similar transformation was used by Arkin et. al [2] to prove the NP-hardness of the WSP, by reduction to the Weighted Tardiness Problem.

Proposition 3. *Given J_i, s_i, p_i, t_i, $i=1,\cdots,m$, there exists a feasible permutation schedule that avoids stockouts if and only if the minimum value of the m/1/Tmax problem is 0, where the due dates are J_i+t_i, and the job lengths are s_i+t_i, $i=1,\cdots,m$.*

Proof. Assume, without loss of generality, that a feasible sequence is $\{1, 2,\cdots,m\}$ and let S_i be the start time of run i. Then from the feasibility of this cyclic schedule, we have $S_i \leq J_i$ for all i. Arrange the jobs according to the sequence $\{1,\cdots,m\}$. Obbiously $S_i+t_i \leq J_i+t_i$ for all i, implying that the jobs are completed before their due dates. Now assume, without loss of generality, that the sequence $\{1,\cdots,m\}$ has no tardy jobs. Let F_i be the finishing time of job i. Then, from the fact that tradiness is 0, we have $F_i \leq J_i+t_i$ for all i. Obviously $F_i-t_i \leq J_i$ 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 [3]). 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 $T \geq (\mathbf{e}'\mathbf{s}-\mathbf{r}'\mathbf{J})/\kappa$, go to *Step 2.* Else stop, there is no feasible schedule.

Step 2. Transform into $m/1/T$max scheduling problem via.
$d_i = J_i + t_i$, $l_i = s_i + t_i$, for all i where d_i and l_i are
the due date and the processing time, respectively, for job i.
Solve $m/1/T$max problem with d_i's and l_i' 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 is as in Example 1 with T=40. We restrict to permutation sequences. Step 1 is satisfied since $(\mathbf{e}'\mathbf{s}-\mathbf{r}'\mathbf{J})/\kappa \leq T$. The transformation to m/1/Tmax yields $\mathbf{d}=(18.\overline{3}, 25.5, 28.3)$ and $\mathbf{l}=(7.\overline{3}, 15.5, 6.3)$. Using the EDD rule, the optimum value is $0.86\overline{3}$, so at least one job is tardy. Consequently, there exists no feasible permutation schedule. ■

4. Heuristics for the WSP

Since the SAP is NP-Complete, 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 sequeeze 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-toal loss for other products)/(invested time). By saving 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.)

$J_i' \equiv J_i,\ T' \equiv T,\ i=1,\cdots,m, \qquad C \equiv \{1,\cdots,m\}$

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

$t_i \equiv \{T' - max(J_i',\ s_i)\}/p_i \qquad i=1,\cdots,m$

If $t_i \leq 0$, $C \leftarrow C \backslash \{i\}$.

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

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

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

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

$J_j' \leftarrow Max(J_j' - s_{i^*} - t_{i^*},\ 0) \qquad j \neq i$

$J_{i^*}' \leftarrow Max(J_{i^*}' - s_{i^*},\ 0) + (p_{i^*} - 1)t_{i^*}$

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

$C \leftarrow C \backslash \{i^*\}$

Step 4. (Stopping criteria.)

If $T' \leq 0$ *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 palnning horizon $[0, T]$. That is,

$$\text{minimize} \sum_{i=1}^{m} \rho_i \sum_{K=0}^{N} x_i^{k+1}$$
$$= \sum_{i=1}^{m} \rho_i \left[\sum_{K=0}^{N} \{ (J_i^{k+1} + S^{k+1}) - (J_i^k + S^k) \} - \sum_{K=0}^{N} p_i t_{ik} \right]$$
$$= \sum_{i=1}^{m} \rho_i (T - J_i) - \sum_{i=1}^{m} \rho_i p_i \sum_{k=1}^{N} t_{ik} .$$

The last equality follows from Property 2 and the telescoping sum. Consequently, the problem is equivalent to maximize $\sum_{i=1}^{m} \rho_i p_i \sum_{k=1}^{N} t_{ik}$. This heuristic can be called greedy, since we want to maximize $\sum_i \rho_i p_i t_i$ in the MIP formulation, and here we disregard 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.)
$i^* = argmax_{i \in C} \{ \rho_i p_i t_i / (s_i + t_i) \}$. 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, differenent distributions of s_i, p_i, J_i, and ρ_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 4 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 5 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 performances of the heuristics without adjacent pairwise interchanges (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. do 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.

Table 4. Distributions for Data 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]

Table 5. Computational Results for Test Problems

	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

4.2 Heuristics for the WSP

The problem is more involved than 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 Heristic 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 $\delta \equiv [T(1-\kappa)-\mathbb{r}'\mathbb{J}]/\mathbb{e}'\mathbb{s}$. Consequently, the total complexity is determined by maximum of $O(\delta m^2)$ and the complexity of LP.

Step 0. (Initialization.)

$$J_i' \equiv J_i,\ T' \equiv T,\ i=1,\cdots,m, \qquad C \equiv \{1,\cdots,m\}$$

$$t_i \equiv (T - max(J_i,\ s_i))/p_i.$$

Do $l=[[(\kappa T+\mathbb{r}'\mathbb{J})/\mathbb{e}'\mathbb{s}]]+1$ *to* $[[T/\mathbb{e}'\mathbb{s}]]$

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

$t_i' \equiv Min\{[T'\text{-}max(J_i', s_i)]/p_i,\ t_i/l\} \quad i=1,\cdots,m$

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

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

Produce product i^* for t'_{i^*}.

Step 3. (Compress.)

Combine consecutive setups if they exist.

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

$J_j' \leftarrow Max(J_j' - s_{i^*} - t'_{i^*},\ 0) \quad j \neq i$

$J'_{i^*} \leftarrow Max(J'_{i^*} - s_{i^*},\ 0) + (p_{i^*} - 1)t'_{i^*}$

Compute cumulative stockout cost.

$T' \leftarrow T' - (s_{i^*} + t'_{i^*})$

If $max(J_i', s_i) \geq T'$, then $C \leftarrow C \backslash \{i\}$

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 6 shows the distributions for the data sets.

Table 7 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.

Table 6. Distributions for Data for Test Problems

Problem Data	Dense	Scattered
Setup time	[2, 4]	[1, 5]
Production rate	[4, 5]	[3, 6]
Initial inventory	[1, 3]	[0, 4]

Table 7. Computational Results for Test Problems

	Dense	Scattered
Mean ratio	1.117	1.105
Minimum ratio	1.001	1.025
Maximum ratio	1.284	1.175

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 Rotaion (Common Cycle) schedule or a heuristic schedule obtained by a time-varying lot size heuristic. In either case, we can compute the target inventories for the cyclic schedule. Let the target inventories be I_i for product i. Then we can find an optimal solution for the WSP problem by modifying the MIP1 slightly. Since the target inventories (ending inventories for the WSP) are specified, we can modify equation (10) to read

$$J_i^{N+1}=I_i \quad i=1,\cdots,m$$

5.2 Phasing into a rotation schedule with minimum aggregate inventory

Suppose we would like to follow a Rotation schedule (over the infinite horizon) after a finite horizon T. Then we can easily find an optimal (in the sense of minizing average setup and holding cost) Common Cycle. There are $m!$ different permutation schedules with this Common Cycle length. Among them we want one whose initial inventories I_i has minimum aggregate inventory $\sum_{i=1}^{m} r_i I_i$. This schedule reduces the burden of solving the WSP with these target inventories. The following propostion shows that SPR (Slowest Processing Rate) rule minimizes the aggregate inventory.

Proposition 4. **(SPR Rule)** *Aggregate inventory is minimized by the SPR sequence*:

$$s_{[1]}p_{[1]} \leq s_{[2]}p_{[2]} \leq \cdots \leq s_{[m]}p_{[m]}.$$

Proof. We use a pairwise interchange argument. Consider a sequence S that is not the SPR sequence. 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 same as in S. We temporarily adopt the notations $A(S)$ and $A(S')$ to represent the aggegate inventory under schedule S and S', respectively. We then show that $A(S')$ is smaller than $A(S)$.

$$\mathrm{A}(\mathrm{S}) = \sum_{k=1}^{m} r_k \mathrm{I}_k = \sum_{k<i} r_k \mathrm{I}_k + \sum_{k>j} r_k \mathrm{I}_k + r_i\Big(\sum_{k\leq i} s_k + \mathrm{T}\sum_{k<i} r_k\Big) + r_j\Big(\sum_{k\leq j} s_k + \mathrm{T}\sum_{k<j} r_k\Big),$$

$$\mathrm{A}(\mathrm{S}') = \sum_{l=1}^{m} r_l \mathrm{I}_l = \sum_{l<j} r_l \mathrm{I}_l + \sum_{l>i} r_l \mathrm{I}_l + r_j\Big(\sum_{l\leq j} s_l + \mathrm{T}\sum_{l<j} r_l\Big) + r_i\Big(\sum_{l\leq i} s_l + \mathrm{T}\sum_{l<i} r_l\Big).$$

Therefore,

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

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 Interger 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 in Setp 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: $m=3$, $\mathbb{p}=(6, 2, 10)$, $\mathbf{s}=(3, 1, 5)$, $\mathbf{J}=(8, 20, 17)$, $\rho=(1, 1, 1)$, $\mathbb{a}$($\equiv$ setup cost)$=(20, 10, 100)$, $\mathbb{h}$($\equiv$ 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 and recovers to the target inventory levels. The sequence and production times are: $\mathbf{f}=(1, 3, 2, 1)$, $\mathbf{t}=(2.7543, 4.2713, 5.5000, 4.7932)$. ■

6. Conclusions

This paper presented a MIP formulation for the WSP and applied the formulation to the SAP. The MIP formulation can be directly applied to solve the SAP or the WSP with moderate problem sizes. However, if the problem size is big, computational cost for using the MIP approach will be very high. Several heuristics are developed and effectiveness are tested. 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. Extensions to the multiple family case might be an interesting research problem.

Reference

[1] Anderson, E., "Testing Feasibility in the Lot Scheduling Problem," *Operations Research*, Vol.38(1990), pp.1079-1088.

[2] Arkin,E., G.Gallego. and R.Roundy, "Complexity of the Multiple Product Single Facility Stockout Avoidance Problem," Technical Report No 822. School of OR & IE, Cornell University, 1988.

[3] Baker,K., *Introduction to Sequencing and Scheduling*, John Wiley and Sons, New York, 1974.

[4] Boctor,F., "The G-group Heuristic for Single Machine Lot Scheduling," *International Journal of Production Research*, Vol.25(1987), pp.363-379.

[5] Delporte,C. and L.Thomas, "Lot Sizing and Sequencing for N Products on One Facility," *Management Science*, Vol.23(1978), pp.1070-1079.

[6] Dobson,G., "The Economic Lot-Scheduling Problem:Achieving Feasibility Using Time-Varying Lot Sizes," *Operations Research*, Vol.35(1987), pp.764-771.

[7] Dobson, G., "The Cyclic Lot Scheduling Problem with Sequence-Dependent Setups," *Operations Research*, Vol.40(1992), pp.736-749.

[8] Elmaghraby,S., "The Economic Lot Scheduling Problem (ELSP):Review and Extensions," *Management Science*, Vol.24(1978), pp.587-598.

[9] Gallego,G. and I.Moon, "The Effect of Externalizing Setups in the Economic Lot Scheduling Problem," *Operations Research*, Vol.40(1992), pp.614-619.

[10] Gallego,G. and D.Xiao, "Complexity and Heuristics for some Economic Lot Scheduling Problems," Technical Report, Department of IE & OR, Columbia University, 1990.

[11] Glass,C., "Feasibility of Scheduling Lot Sizes of Three Products on One Machine," *Management Science*, Vol.38(1992), pp.1482-1494

[12] Hsu, W., "On the General Feasibility Test for Scheduling Lot Sizes for Several Products on One Machine," *Management Science*, Vol.29(1983), pp.93-105.

[13] Maxwell,W., "The Scheduling of Economic Lot Sizes," *Naval Research Logistics Quarterly*, Vol.11(1964), pp.89-124.

[14] Moon,I., G.Gallego. and D.Simchi-Levi, "Controllable Production Rates in a Family Production Context," *Internation Journal of Production Research*, Vol.29(1991) pp.2459-2470.

[15] Moon, D. and H.Hwang, "Integrated Production-Inventory Model for Multi-Products on a Single Facility," *International Journal of Systems Science*, Vol.23(1992), pp.273-281.

[16] Zipkin, P., "Computing Optimal Lot Sizes in the Economic Lot Scheduling Problem," *Operations Research*, Vol.39(1991), pp.56-63.