STOCHASTIC SEQUENCING WITH JOB FAMILIES, SET-UP TIMES, AND DUE DATES

Mark P. Van Oyen
Dept of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208

Izak Duenyas and Chi-Yang Tsai
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 96-3

March 1996

Stochastic Sequencing with Job Families, Set-Up Times, and Due Dates

Mark P. Van Oyen[1]

Department of Industrial Engineering and Management Sciences

Northwestern University, Evanston, IL 60208-3119

E-mail: VanOyen@iems.nwu.edu

Izak Duenyas[2] and Chi-Yang Tsai

Department of Industrial and Operations Engineering

The University of Michigan, Ann Arbor, MI 48109-2117

March 1996

Abstract

We consider the problem of allocating a single server to families of jobs with random service times and due dates. Furthermore, a setup is required when the server switches from one job family to another. We address the problems of minimizing expected weighted flow time, expected maximum lateness, and total weighted tardiness with emphasis on problems for which it is assumed that each family can be setup only once (the Group Technology assumption). We derive conditions under which simple sequencing rules are optimal for each of these problems with group technology.

Keywords: Production/scheduling, stochastic sequencing, setup time, job families, weighted flow-time, maximum lateness, weighted tardiness

1. Introduction

We consider a system of N job families (alternately, parallel queues) and a single server attending them. There are no arrivals: we are concerned with the scheduling of the jobs initially present in

[1] Corresponding Author. The work of the first author was supported by the National Science Foundation under Grant No. DMI-9522795.

[2] The work of the second author was supported by NSF Grants No. DMI-9308290 and DMI-9424596.

the system. We assume random processing times and due dates, and focus on the problems of minimizing expected completion times, and expected maximum lateness and tardiness.

Although we consider stochastic sequencing problems, two of the formulations considered reduce to equivalent deterministic problems. This further supports the notion that some portion of the wealth of results on deterministic sequencing can be carried over to stochastic formulations. This paper offers an approach as well as specific insights for use in investigating particular problem formulations. We also note that new insights arise when stochastic formulations have solutions that are distinctly different (either simpler or more complex) than their deterministic counterparts. In particular, one may better appreciate the limitations and benefits of a deterministic model for a particular application.

The primary motivation for this class of problems with job families is that in many applications, sufficiently similar jobs can be grouped into families such that a job does not need a setup when following another job from the same family, but a setup time is required when switching from one family of jobs to another. The majority of the models to date have addressed these problems under completely deterministic assumptions or with only one job class per family. Webster and Baker [14] provide an excellent review of the deterministic literature on these problems. Duenyas and Van Oyen [4] provide a review of the literature on stochastic problems with one job class per family and with the objective of minimizing expected holding costs.

There is a large literature on stochastic scheduling of a single machine with random processing times and random due dates. An excellent review of this literature is provided in Chapters 9-11 of Pinedo [8], and in Righter [10]. However, neither of these reviews include any work on problems with job families. Very little work has been performed on scheduling of job families for a stochastic model. Van Oyen et al. [12] characterized the optimal solution for minimizing expected average completion time under the assumption that all jobs in a family have i.i.d. processing time distributions. With D denoting the job set-up time for any family, c_n denoting the holding cost rate per job of family n, μ_n^{-1} the mean service rate, and x_n the number of identical jobs of that family, the index of family n is given by $\overline{\nu}_n = c_n\mu_n(x_n\mu_n^{-1})/(x_n\mu_n^{-1} + E\{D\})$. The optimal policy is to serve the family with the highest index. This rule can be interpreted as the familiar $c\mu$ index multiplied by the fraction of time over which useful work is performed. These results are extended to queues interconnected by job feedback in Van Oyen and Teneketzis [13].

Except where noted, we address the scheduling of job families under the group technology assumption (GT). The GT assumption (see Potts and Van Wassenhove [9]) restricts the number of setups that can be performed to one per family and thus simplifies the scheduling problem to one of deciding in what sequence job families should be processed. As Webster and Baker [14] point out, the GT assumption "may reflect the fact that the setups are much longer than the job processing times, or it may result from a desire to minimize the time spent on setup in situations where capacity is scarce. It may also be imposed simply to make the problem more tractable." In fact, as we will show below, stochastic versions of the problems with due dates are significantly difficult even with the GT assumption.

The rest of this paper is organized as follows. In the next section, we give a formulation of the problem and the notation to be used. In Section 3, we show that minimizing expected weighted flow time over the class of dynamic scheduling policies reduces to an equivalent deterministic problem. In Section 4, we specify sequencing rules for minimizing expected maximum lateness and minimizing expected maximum tardiness. Finally, Section 5 focuses on sequencing to minimize total expected weighted tardiness.

2. Problem Formulation

A single server is to be allocated to an arbitrary number of jobs, all initially available for processing. Each job belongs to a job family, and the families are labeled $1, 2, \ldots, N$, where $N \in \mathbb{N}$ (where $\mathbb{N}$ denotes the naturals and $\mathbb{Z}^+$ the nonnegative integers). Family f has N_f jobs. We will name job s in family f to be of type (f, s) and this job will have service time $P(f, s)$ drawn from a distribution $B(f, s)$ with mean $\mu(f, s)^{-1} \in (0, \infty)$. Set-ups are required to switch production from a job of one *family* to another. There is no switching penalty for changes from one sub-type to another within the same family. The set-up time for family f is $S(f)$, which is drawn from the distribution $\Sigma(f)$ independently of all else and has mean $ES(f) \in [0, \infty)$.

In the expected weighted flowtime problem, the weight associated with a job of type (f, s) is $c(f, s) \in (0, \infty)$ per unit time it spends in the system. In the cases of maximum expected lateness and expected maximum lateness, we assume that the due date of a type (f, s) job is a random variable, $d(f, s)$, independent of all else with mean $Ed(f, s)$. We assume that the realization of the (random) due date does not occur until its completion (as in Pinedo [7] and Forst [5]). Finally, in the case of minimization of total weighted tardiness, there is in addition a tardiness penalty of $\beta(f)$ per unit of time tardy for any job of family f.

To define fully dynamic scheduling problems, we restrict attention to non-preemptive, non-idling, and non-anticipative policies. A policy specifies, at each job completion epoch, that the server either remain working in the present family on a particular subtype, or set-up another job family for service. At time zero, we assume that any family must be set up.

The special case of a GT problem makes the additional restriction that a set up of family f must be followed by the consecutive service of all jobs of family f.

3. Minimizing Expected Weighted Flow Time

In this section, we focus on minimizing expected weighted flow time, where flow times for jobs are weighted by their holding costs. That is, the weight for a job of type (f, s) is $c(f, s)$. We begin by showing that even the fully dynamic scheduling problem reduces to a deterministic problem. Although it is common to identify an optimal policy for a deterministic problem and then to show that it remains optimal in the stochastic version, the approach we take here does not require knowledge of an optimal policy to show that there exists a list policy (i.e., a static or open-loop policy) that is optimal. We begin by viewing the problem as a discrete-time stochastic control problem, using our assumption that set-ups and services cannot be interrupted. The key features that allow the system to be decomposed are (1) the linearity of the objective function, and (2) the fact that any non-randomized Markov policy dictates a deterministic sequence of states to be visited (although dwell times in each state are stochastic). We proceed with the details.

For simplicity of exposition, suppose $N_f = \bar{N}$ for $f = 1, \ldots, N$. Let the state space be

$$\mathcal{S} \triangleq \{(x_{f,s}, n) : x_{f,s} \in \{0, 1\}, f \in \{1, \ldots, N\}, s \in \{1, \ldots, \bar{N}\}, n \in \{1, \ldots, N\}\} . \tag{3.1}$$

At time t, x is an $N \times \bar{N}$ array in which $x_{f,s}$ equals 1 if a job of type (f, s) is available at t and equals 0 otherwise. The last component of the state, n, indicates that queue n is set up at t. We designate the control actions admissible when the system is in state (x, n) by the pair (f, s) , where $(f, s = 0)$ indicates the set-up of family f and $(f = n, s)$ indicates that job (n, s) will be served next provided $x_{n,s} = 1$. Let $\mathcal{A}\ (x, n)$ denote the above set of available actions in state (x, n). We have positive costs, finite state and action spaces, and no discounting. By Proposition 11 of

Chapter 5 of Bertsekas [2] (or see Ross [11]) the following optimality equation is a necessary and sufficient for a stationary, nonrandomized policy to be optimal:

$$\begin{aligned} V(x,n) &= \min_{(f,s)\in\mathcal{A}\ (x,n)} E\{\sum_{i=1}^{N}\sum_{j=1}^{\bar{N}} c(i,j)x_{ij}[P(f,s)\mathbb{1}\{s\neq 0\}+S(f)\mathbb{1}\{s=0\}+ \\ &\qquad V(x-e_{f,s}\mathbb{1}\{x_{f,s}=1,s\neq 0\},f)\}, \end{aligned} \tag{3.2}$$

where $e_{f,s}$ is a unit vector with element f,s equal to 1. By stationary, we mean that the actions depend only on the state, rather than time. Nonrandomized means that for each state, one and only one action can be taken. For this problem, each action leads deterministically to the next state at the next decision epoch. By these properties, there exists an optimal policy under which the sequence of states visited is deterministic. Thus, it suffices to consider only open loop, or list policies. Moreover, the optimality equation (3.2) can be rewritten as

$$\begin{aligned} V(x,n) &= \min_{(f,s)\in\mathcal{A}\ (x,n)} \sum_{i=1}^{N}\sum_{j=1}^{\bar{N}} c(i,j)x_{ij}[\mu(f,s)^{-1}\mathbb{1}\{s\neq 0\}+ES(f)\mathbb{1}\{s=0\}]+ \\ &\qquad V(x-e_{f,s}\mathbb{1}\{x_{f,s}=1,s\neq 0\},f)\ . \end{aligned} \tag{3.3}$$

Thus, we observe that the solution of this stochastic scheduling problem is identical to the deterministic scheduling problem with constant service times $\mu(f,s)^{-1}$ and set-up times $ES(f)$.

The optimal policy for the deterministic scheduling problem with constant service times $\mu(f,s)^{-1}$ and setup times $ES(f)$ has been characterized by Theorem 7.5 of Ham, Hitomi and Yoshida [6] under the GT assumption. Therefore, their result directly carries over to the stochastic case.

Theorem 1: *For the problem of minimizing expected weighted flowtime under the GT assumption, an optimal schedule will sequence families in nondecreasing order of* $(ES(f)+\mu(f)^{-1})/c(f)$ *where* $\mu(f)^{-1} = \sum_s \mu(f,s)^{-1}$ *and* $c(f) = \sum_s c(f,s)$ *. Furthermore, jobs types within each family are sequenced in nondecreasing order of* $\mu(f,s)^{-1}/c(f,s)$.

For a partial characterization of the solution to the fully dynamic scheduling problem, see also Webster and Baker.

4. Minimizing Maximum Expected Lateness and Expected Maximum Lateness

In this section, we focus on determining optimal list policies that minimize both the maximum expected lateness and the expected maximum lateness when processing times as well as due dates are stochastic. If we let $F_g(i)$ denote the (random) flowtime of the i^{th} job served under g, and let $d_g(i)$ denote the (random) due date of the i^{th} job served under policy g, minimizing maximum expected lateness corresponds to minimizing $J(g) = \max_i E[F_g(i) - d_g(i)]$ and minimizing expected maximum lateness corresponds to minimizing $J(g) = E \max_i [F_g(i) - d_g(i)]$.

It is well known that under stochastic processing times, when due dates are deterministic, and setup times are zero, the Earliest Due Date (EDD) rule is optimal under either of the two objectives (e.g., see Chapter 9 of Pinedo (1995)). However, in problems with family setup times under the GT assumption, the two objectives result in different optimal policies even when the due dates are deterministic. Consider the following simple example with two families:

Example 1: Family 1 has only one job, and family 2 has two jobs. The processing time of the job in family 1 is 20 with probability 0.5, and 10 with probability 0.5 and its due date is 29. The processing time of job 1 in family 2 is 25 with probability 0.5 and 15 with probability 0.5 and its due date is 5. Similarly, the processing time of job 2 in family 2 is 30 with probability 0.5 and 20 with probability 0.5 and its due date is 30. Setup times are zero. Under both objectives, under the optimal policy, once family 2 is set up, job 1 in family 2 is processed before job 2 in family 2. However, under group technology, it is optimal to process family 1 first, and family 2 second when the objective is to minimize the maximum expected lateness and family 2 first and family 1 second when the objective is to minimize the expected maximum lateness. □

We first focus on the objective of minimizing *maximum expected lateness.* Define the list policy g as a string (list)

$$g = g_1 \; g_2 \; g_3 \; \cdots \; g_{k(g)}, \tag{4.1}$$

where $k(g)$ denotes the sum of the number of jobs initially in the system plus set-ups, and where g_i designates the job type vector $(f(i), s(i))$ served on the ith stage. We take $(f(i), 0)$ to represent a set-up of family $f(i)$ and we avoid applying a due date to stage l at which a set-up occurs by defining $\mathcal{L} \triangleq \{l \in \{1, \dots, k(g) : s(l) \neq 0\}$. We can now express the cost of policy g as

$$\begin{aligned} J(g) &= \max_{l \in \mathcal{L}} E\{\sum_{k=1}^{l} [P(f(k), s(k)) \mathbb{1}\{s(k) \neq 0\} + S(f(k)) \mathbb{1}\{s(k) = 0\}] - d(f(l), s(l))\} \\ &= \max_{l \in \mathcal{L}} \{\sum_{k=1}^{l} [\mu^{-1}(f(k), s(k)) \mathbb{1}\{s(k) \neq 0\} + ES(f(k)) \mathbb{1}\{s(k) = 0\}] - Ed(f(l), s(l))\}. \quad (4.2) \end{aligned}$$

Since a policy is optimal if it minimizes $J(g)$ as defined in (4.2), we observe once again that the solution of this stochastic scheduling problem is identical to the deterministic scheduling problem with constant service times $\mu^{-1}(f,s)$, setup times ES_f and due dates $Ed(f,s)$. This simplifies the problem considerably as the optimal policy for the deterministic problem has been characterized (see Potts and Van Wassenhove [9] for family-dependent due dates). We cite the result of Webster and Baker [14] in the following.

Theorem 2: *For the problem of minimizing maximum expected lateness, the optimal list policy under group technology, GT, will serve job j within each family i in nondecreasing order of $Ed(i,j)$. Reorder jobs in each family in nondecreasing order of their expected due dates. For family i, define the variable $q(i,j) = \sum_{k=j+1}^{N_i} \mu^{-1}(i,j)$, and $d'(i,j) = Ed(i,j) + q(i,j)$ and let $d'(i) = \min_j d'(i,j)$. The optimal policy will process families in nondecreasing order of $d'(i)$.*

We note that the result of Theorem 2 is interesting because the optimal sequence does not depend on $E(S_i)$ or on t. Since $EL_{i,j} = t + ES(i) - d'(i,j)$ with $q(i,0)$ denoting the total processing time for family i, we see that it suffices to minimize $d'(i,j)$ to identify the family with the maximum expected lateness. Since this minimization is unaffected by the starting time, t, we conclude that Theorem 2 remains valid under the GT assumption even when we allow dynamic scheduling policies.

We next focus our attention on finding optimal list policies that minimize *expected maximum lateness*. We begin our analysis without the GT assumption. Thus, we view a list policy as a series of *runs*, each beginning with a set-up of a family, say i, and containing only jobs of that family. We first characterize the ordering of jobs within a run and then the ordering of runs.

Theorem 3: *For the problem of minimizing expected maximum lateness, if the due dates of jobs within the same family and run are stochastically ordered, an optimal policy will schedule these jobs within a family in stochastically nondecreasing order of their due dates.*

Proof: Consider a sequence $\sigma = A,(i,k),B,(i,j),C$, where A, B and C are partial sequences of jobs with (i,k) and (i,j) in the same run, where $d(i,j) \leq_{st} d(i,k)$. Without loss of generality, assume that $d(i,j)$ is also stochastically less than or equal to the due dates of all the jobs in B. (Otherwise, there exists at least one job in B that should be ahead of (i,j), and the same interchange argument could be applied to that job first). Now, consider a sequence $\sigma' = A,(i,j),B,(i,k),C$, where σ' is obtained from σ by interchanging jobs (i,k) and (i,j). Since all jobs in B must be from family i, the interchange results in no additional setups. Let P_A, P_B and P_C be random variables denoting the sum of processing times and setup times in partial sequences A, B and C. We define $L_{i,j}$ as the lateness of job (i,j). Clearly, the lateness of all jobs in A and C are the same under both schedules. Also note that $L_{i,k} \geq_{st} L_{i,b} \geq_{st} L_{i,k}$ (where job (i,b) is the b^{th} job, for arbitrary b, in B with due date $d(i,b)$). Thus, it suffices to compare the lateness of jobs (i,j), (i,k) and (i,b) under g' with the lateness of job (i,j) under g. We also let P_b be the (random) sum of the processing times of the first b jobs in partial sequence B.

$$\begin{aligned}
L'_{i,j} &= P_A + P(i,j) - d(i,j) \\
L'_{i,k} &= P_A + P(i,j) + P_B + P(i,k) - d(i,k) \\
L'_{i,b} &= P_A + P(i,j) + P_b - d(i,b) \\
L_{i,j} &= P_A + P(i,k) + P_B + P(i,j) - d(i,j)
\end{aligned}$$

We see that $L'_{i,j} \leq_{st} L_{i,j}$. Since all processing times and set-up times are mutually independent, $L'_{i,k} \leq_{st} L_{i,j}$ follows from the fact that $d(i,j) \leq_{st} d(i,k)$. Similarly, $L'_{i,b} \leq_{st} L_{i,j}$ follows from the fact that $P_B >_{st} P_b$ and $d(i,j) \leq_{st} d(i,b)$. Therefore, we have that $\max_{i,j} L'_{i,j} \leq_{st} \max_{i,j} L_{i,j}$, which implies that $E[max_{i,j}L'_{i,j}] \leq E[max_{i,j}L_{i,j}]$. □

The previous theorem together with the GT assumption indicates that if stochastic order between the due dates of jobs within a family exists, there is an optimal sequence in which the jobs in a family are scheduled in stochastically nondecreasing order of their due dates. The problem is simplified because there is only one run per family. Note that within a family, the sequence is equivalent to earliest expected due date (EEDD). In the following, we will assume the stochastic ordering of due dates for jobs within each family, as such an ordering will exist in many situations (e.g., when due dates are deterministic or exponentially distributed).

Suppose that family i, with its jobs numbered in stochastically nondecreasing order of $d(i,j)$ begins its setup at time t. Then the lateness of job (i,j) is

$$L_{ij} = t + S_i + \sum_{k=1}^{j} P(i,k) - d(i,j) \tag{4.3}$$

Define $d''(i,j) = d(i,j) + q'(i,j)$ where $q'(i,j) = \sum_{k=j+1}^{N_i} P(i,k)$, and define the random variable

$d''(i) = \min_j d''(i,j)$. Then, we can rewrite (4.3) as

$$L_{ij} = t + S_i + \sum_{k=1}^{N_i} P(i,k) - d''(i,j) \tag{4.4}$$

Therefore, we have $\max_j L_{ij} = t + S_i + \sum_{k=1}^{N_i} P(i,k) - \min_j d''(i,j)$ and by the definition of $d''(i)$, we have

$$\max_j L_{ij} = t + S_i + \sum_{k=1}^{N_i} P(i,k) - d''(i) \tag{4.5}$$

We can now state our result for the optimal list policy for minimizing expected maximum lateness.

Theorem 4: *Under the GT assumption, if a stochastic ordering of $d''(i)$ exists for families of jobs, an optimal list policy will sequence job families in stochastically nondecreasing order of $d''(i)$.*

Proof: The argument is an interchange argument exactly the same as in the proof of Theorem 3, with a run taken as a composite job. We omit it. □

We note that a stochastic ordering of $d''(i)$ for job families may not exist. Thus, Theorems 3 and 4 provide a sufficient condition for an optimal policy and not a necessary one. They can also be used to provide a partial ordering between job families. We demonstrate the use of Theorems 3 and 4 on Example 1 from the previous section. Since the due dates are deterministic, a stochastic ordering between them trivially exists and by Theorem 3, we can conclude that when processing family 2, it is optimal to process job 1 first and job 2 second. Computing the d'' values for the families, we obtain the result that $d''(1) = 29$, and that $d''(2)$ equals 25 with probability 0.5 and 30 with probability 0.5. There is no stochastic ordering between these relationships, so Theorems 3 and 4 do not specify an optimal solution. However, we *can* conclude that if the due date of the job in family 1 is greater than or equal to 30 (all else being equal), then an optimal schedule will process family 2 ahead of family 1. Similarly, if the due date of job 1 in family 2 is 9 or above (everything else being equal), an optimal schedule will process family 2 first and family 1 second.

5. Minimizing Expected Total Weighted Tardiness

We now turn to the objective of obtaining a list policy that minimizes the total expected weighted job tardiness. We restrict attention to the following class of problems. We require all job processing times in the same family, f, to be drawn from the same distribution $(B(f,s) = B(f,1)$

for all s) and to all share *the same* random due date $d(f)$ (so $d(f,s) \equiv d(f)$ for all s), the realization of which is obtained only after processing the last job of family f. Let Δ_f denote the cumulative distribution function (c.d.f.) governing the due date of family f.

Group technology is applied. Since the processing times of jobs in the same family are independent and identically distributed random variables and group technology is applied, the job sequence within a family does not affect the result. Therefore, the objective is equivalent to obtaining a family sequence that minimizes the total weighted job tardiness in expectation.

Let the random variable $T(g)$ denote the total weighted tardiness of sequence g. Our main result rests on the following definition.

Definition 1: Job family processing time distributions are said to be *compatible* with the family due dates, set-up times, tardiness penalties, and numbers of jobs if, for families labeled such that $P(1,1) \leq_{st} P(2,1) \leq_{st} \ldots \leq_{st} P(N,1)$, and for all $f < N$, we have $d(f) \leq_{st} d(f+1)$, $S(f) \leq_{st} S(f+1)$, $\beta(f) \geq \beta(f+1)$, and $N_f \geq N_{f+1}$.

We now present a theorem which indicates conditions under which processing the jobs in increasing stochastic order of their processing times is optimal. This result can be viewed as an extension of those found in Forst [5]) and Chang and Yao [3]).

Theorem 5: *Provided job family processing time distributions are* compatible *with the family due dates, set-up times, tardiness penalties, and numbers of jobs, $E\{T(g)\}$ is minimized by sequencing the families in increasing stochastic order of their job processing times.*

Proof: The argument is an interchange argument. Suppose the hypothesis of the theorem is satisfied, but policy g sequences families in the order $1,2,\ldots,i-1,j,i,\ell,\ldots$ for some i,j such that $j > i$. We show that sequence $g' \triangleq 1,2,\ldots,i-1,i,j,\ell,\ldots$, which simply interchanges families i and j, is at least as good as g. To contrast the two sequences g,g', we have only to contrast the expected total weighted tardiness of families i,j. Let $F(f,s)$ $(F'(f,s)$) denote the flowtime of job (f,s) and T_f (T'_f) be the total weighted tardiness of family f under sequence g (g'). Sequence g is better than g' if and only if $E\{T(g)\} - E\{T(g')\} \geq 0$. We have

$$
\begin{aligned}
& E\{T(g)\} - E\{T(g')\} \\
= \;& E\{T_i + T_j\} - E\{T_i' + T_j'\} \\
= \;& E\{\sum_{s=1}^{N_j} \beta(j)[(\max(F(j,s)\ ,d(j)) - d(j)) - (\max(F'(j,s)\ ,d(j)) - d(j))] + \\
& \sum_{s=1}^{N_i} \beta(i)[(\max(F(i,s)\ ,d(i)) - d(i)) - (\max(F'(i,s)\ ,d(i)) - d(i))\} \qquad (5.1) \\
= \;& \sum_{s=1}^{N_j} \{\beta(j)[E\{\max(F(j,s)\ ,d(j))\} - E\{\max(F'(j,s)\ ,d(j))\}] + \\
& \beta(i)[E\{\max(F(i,s+N_i-N_j),d(i))\} - E\{\max(F'(i,s),d(i))\}]\} + \\
& \sum_{s=N_j+1}^{N_i} \beta(i)[E\{\max(F(i,s-N_j),d(i))\} - E\{\max(F'(i,s),d(i))\}]\ , \qquad (5.2)
\end{aligned}
$$

where the last line follows from $N_i \geq N_j$. We proceed to prove that the two sums of (5.2) are both nonnegative quantities. First, we show that the second summation of (5.2) is nonnegative because $F(i, s - N_j) \geq_{st} F'(i, s)$ for $N_j + 1 \leq s \leq N_i$. To see this, note that

$$
\begin{aligned}
& F(i, s - N_j) - F'(i, s) \\
= \;& F(i-1, N_{i-1}) + S(j) + \sum_{k=1}^{N_j} P(j,k) + S(i) + \sum_{k=1}^{s-N_j} P(i,k) \\
& -[F(i-1, N_{i-1}) + S(i) + \sum_{k=1}^{s} P(i,k)] \\
= \;& S(j) + \sum_{k=1}^{N_j} [P(j,k) - P(i, k+s-N_j)]\ . \qquad (5.3)
\end{aligned}
$$

Since the terms of (5.3) are independent and the processing times of family j are stochastically larger than those of family i, it follows that $\max(F(i, s - N_j), d(i)) \geq_{st} \max(F'(i, s), d(i))$ (see example 8.2a of Ross) and thus the second summation of (5.2) is nonnegative.

To conclude that the first summation of (5.2) is nonnegative, it is helpful to observe that for $1 \leq s \leq N_j$,

$$
F(i, s + N_i - N_j) \geq_{st} F'(j, s) \geq_{st} F(j, s) \geq_{st} F'(i, s) \qquad (5.4)
$$

To see this, note that

$$
F(i, s + N_i - N_j) - F'(j, s)
$$

$$\begin{aligned}
&= F(i-1,N_{i-1})+S(j)+\sum_{k=1}^{N_j}P(j,k)+S(i)+\sum_{k=1}^{s+N_i-N_j}P(i,k)\\
&\quad -[F(i-1,N_{i-1})+S(i)+\sum_{k=1}^{N_i}P(i,k)+S(j)+\sum_{k=1}^{s}P(j,k)]\\
&= \sum_{k=s+1}^{N_j}P(j,k)-\sum_{k=s+N_i-N_j+1}^{N_i}P(i,k)\ , \qquad (5.5)
\end{aligned}$$

and the first inequality of of (5.4) follows from the mutual independence of the processing times and from the fact that the processing times the processing times of family j are stochastically larger than those of family i, and that $N_j \geq N_i$. Similarly, the rest of of (5.4) follows from

$$\begin{aligned}
F'(j,s)-F(j,s) &= S(i)+\sum_{k=1}^{N_i}P(i,k)\\
F(j,s)-F'(i,s) &= S(j)-S(i)+\sum_{k=1}^{s}[P(j,k)-P(i,k)]
\end{aligned}$$

Next, observe that because $\max(F(j,s),d(j))$ is a nonnegative random variable, its expectation can be expressed as $\int_0^\infty 1-P(\max(F(j,s),d(j))\leq t)dt = \int_0^\infty 1-\Delta_j(t)P(F(j,s)\leq t)dt$. Thus, it suffices to show that for $1\leq s\leq N_j$, the following is nonnegative:

$$\begin{aligned}
&\beta(i)[E\{\max(F(i,s+N_i-N_j),d(i))\}-E\{\max(F'(i,s),d(i))\}]-\\
&\beta(j)[E\{\max(F'(j,s)\ ,d(j))\}-E\{\max(F(j,s)\ ,d(j))\}]\\
=\ &\beta(i)\int_0^\infty(1-\Delta_i(t)P(F(i,s+N_i-N_j)\leq t))-(1-\Delta_i(t)P(F'(i,s)\leq t))dt-\\
&\beta(j)\int_0^\infty(1-\Delta_j(t)P(F'(j,s)\leq t))-(1-\Delta_j(t)P(F(j,s)\leq t))dt\\
=\ &\int_0^\infty\beta(i)\Delta_i(t)[P(F'(i,s)\leq t)-P(F(i,s+N_i-N_j)\leq t)]dt-\\
&\beta(j)\Delta_j(t)[P(F(j,s)\leq t)-P(F'(j,s)\leq t)]dt\ . \qquad (5.6)
\end{aligned}$$

The above is nonnegative because $\beta(i)\geq\beta(j).\Delta_i(t)\geq\Delta_j(t)$ (by hypothesis) and because (5.4) gives

$$P(F'(i,s)\leq t)-P(F(i,s+N_i-N_j)\leq t)\geq P(F(j,s)\leq t)-P(F'(j,s)\leq t)\ . \qquad (5.7)$$

Thus (5.2) is nonnegative and the result follows. □

We conclude this section by noting that violating even a single one of the conditions in Definition 1 results in the conclusion of Theorem 5 not to be valid. Consider the following example in which the conditions of Definition 1 are violated because $n_2 > n_1$. Let family 1 have 1 job while family

2 has 2 jobs and set $\beta_1 = \beta_2 = 1$. The setup times are 4 for family 1, and 5 for family 2. The processing times are 20 for family 1 and 21 for family 2. The due date is 8 for family 1 and either 10 (with probability 0.6) or 12 (with probability 0.4) for family 2. It is easy to see in this example that $S_2 \geq S_1$, $P(2,1) \geq P(1,1)$, $\beta_1 \geq \beta_2$ and $d_2 \geq d_1$. It is easy to check that even though this is the only violation, it is sufficient to make it optimal to process family 2 first. Beginning with family 1, the total lateness is 117 with probability .6 and with probability .4 this lateness is reduced by 4 giving an average of 115.4; however, an average cost of 114.4 is obtained by starting with family 2. Numerical examples which show that violating any one of the other conditions invalidates the conclusion of Theorem 5 can be similarly obtained.

6. Conclusions

In this paper, we have explored a variety of stochastic sequencing problems. We examined the problem of minimizing expected weighted flow time, for which an optimal GT sequence can be explicitly characterized. When the GT assumption is removed, the optimal policy is complex even in the deterministic case. Although a complete characterization of an optimal policy is not presently known, we were able to formulate the dynamic scheduling problem as a Markov decision process and to prove that there exists an open-loop or list policy that is optimal within the class of dynamic control policies.

For the problem of minimizing maximum expected lateness, we showed that the stochastic sequencing problem can be solved by analyzing its deterministic counterpart. This is not the case in general for the objective of minimizing expected maximum lateness, for which we defined sufficient conditions under which a relatively simple policy is optimal.

Finally, for the objective of minimizing expected total weighted tardiness, we identified a strong sufficient condition under which the shortest expected processing time sequence is optimal with group technology. In addition, we provided an example demonstrating that the result no longer holds under relaxed assumptions.

Bibliography

[1] K.R. Baker, *Elements of Sequencing and Scheduling,* (Dartmouth College, Hanover, NH, 1993).

[2] D.P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models,* (Prentice-Hall, Englewood Cliffs, 1987).

[3] C.-S. Chang and D.D. Yao, Rearrangement, majorization, and stochastic scheduling, IBM Technical Report RC-16250.

[4] I. Duenyas and M.P. Van Oyen, Heuristic scheduling of parallel heterogeneous queues with set-ups, forthcoming in *Management Science* (1996).

[5] F.G. Forst, Stochastic Sequencing on One Machine with Earliness and Tardiness Penalties, *Probability in the Engineering and Informational Sciences,* **7**, (1993), 291-300.

[6] I. Ham, K. Hitomi, and T. Yoshida, *Group Technology: Application to Production Management,* (Kluwer-Nijhoff Publishing, Boston, 1985).

[7] Pinedo, M., "Stochastic Scheduling with release dates and due dates," *Operations Research,* **30** (1983) 559-572.

[8] Pinedo, M., *Scheduling: Theory, Algorithms and Systems,*, Prentice Hall, New Jersey, (1995).

[9] C.N. Potts and L.W. Van Wassenhove, Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *J. of the Operational Research Society* **43** (1992) 395–406.

[10] Righter, R., "Scheduling," in *Stochastic Orders and Their Applications,* Shaked M., and J.G. Shanthikumar editors, Academic Press, Boston, (1994).

[11] S.M. Ross, *Introduction to Stochastic Dynamic Programming,* Academic Press, New York, (1983).

[12] M.P. Van Oyen, D.G. Pandelis, and D. Teneketzis, Optimality of index policies for stochastic scheduling with switching penalties, *J. of Appl. Prob.* **29** (1992) 957–966.

[13] M.P. Van Oyen, and D. Teneketzis, Optimal Stochastic Scheduling of Forest Networks with Switching Penalties. *Adv. Appl. Prob.* **26** (1994) 474–497.

[14] S. Webster and K.R. Baker, Scheduling Groups of Jobs on a Single Machine, *Operations Research* **43** (1994) 692–703.