SAFE SCHEDULING:
SETTING DUE DATES IN SINGLE-MACHINE PROBLEMS

by

Kenneth R. Baker
Tuck School, Dartmouth College, Hanover, NH 03755 USA

and

Dan Trietsch
College of Engineering, American University of Armenia, Yerevan, Armenia

ABSTRACT

We consider single-machine stochastic scheduling models with due dates as decisions. In addition to showing how to satisfy given service-level requirements, we examine variations of a model in which the tightness of due-dates conflicts with the desire to minimize tardiness. We show that a general form of the trade-off includes the stochastic E/T model and gives rise to a challenging scheduling problem. We present heuristic solution methods based on static and dynamic sorting procedures. Our computational evidence identifies a static heuristic that routinely produces good solutions and a dynamic rule that is nearly always optimal. The dynamic sorting procedure is also asymptotically optimal, meaning that it can be recommended for problems of any size.

Keywords: Scheduling, Stochastic Scheduling, Heuristics, Due-Date Setting

SAFE SCHEDULING: SETTING DUE DATES IN SINGLE-MACHINE PROBLEMS

1. INTRODUCTION

Safe scheduling is a relatively new approach to stochastic scheduling problems which explicitly acknowledges the role of safety time via the need to meet specified service levels. In this paper, we introduce several new and related problems in safe scheduling, and we provide full or partial solutions. These problems extend and generalize some of the familiar models of scheduling theory and thereby extend our knowledge about scheduling methods and solutions.

We build on the standard single-machine sequencing model containing n jobs (Baker 2005, Pinedo 2002). The key parameters in the model include the processing time for job j (p_j) and the due date (d_j). In the actual schedule, job j completes at time C_j, and the set of completion times is summarized in a measure of scheduling performance M that serves as the model's objective function.

There are broadly two types of models that involve performance measures related to due dates. In one scenario, due dates are given, and a performance measure, such as total tardiness, captures the effectiveness of the schedule at meeting the given due dates. In such a scenario, we may consider rejecting some jobs to enhance performance (Akker and Hoogeveen, to appear; Trietsch and Baker, to appear). In the other scenario, due dates are internal decisions (Soroush 1999, Portougal and Trietsch 2006). Such due dates are often negotiated with customers or chosen by production control (ERP) systems to guide or pace the progress of work, and the performance measure may include earliness costs and tardiness costs as well as measures of due-date tightness. In this paper, we describe several related scheduling problems that involve setting due dates; thus, we treat the d_j-values as decision variables.

Our models are also stochastic: the p_j-values are random variables, and we assume that their probability distributions are given. We also assume, unless we state otherwise, that these distributions are independent. As a consequence, job completion times are random variables. In traditional models of stochastic scheduling, the objective function is usually the expected value of the measure M. For example, whereas a deterministic model calls for the minimization of total tardiness, T, the traditional stochastic model would call for the minimization of E[T]. This value is the simplest stochastic analog of the deterministic

measure, and the model is referred to as the *stochastic counterpart* of the original deterministic model.

Safe scheduling models are not necessarily direct stochastic counterparts of deterministic models. A key element of safe scheduling models is the *service level*, defined for job j as $SL_j = \Pr\{C_j \le d_j\}$, the probability that job j completes by its due date. Let b_j denote a given target probability, $0 \le b_j \le 1$. Then the form of a service-level constraint for job j is $\Pr\{C_j \le d_j\} \ge b_j$. We say that job j is *stochastically on time* if its service-level constraint is satisfied; otherwise, the job is *stochastically tardy*. One approach, then, is to optimize some measure of the stochastic tardiness in a schedule. Another approach is to minimize the expected costs associated with missing the due date, which leads to optimal service levels. Both approaches involve safety time, which is defined as the difference between the expected completion time and the due date that achieves the service level.

Our paper discusses a collection of due date setting problems in safe scheduling with one machine. In section 2, we illustrate safe scheduling concepts by selecting due dates as tightly as possible, subject to service-level constraints. Next, we broaden this model to encompass a trade off between tight due dates and job tardiness. In section 3, we analyze this trade off for a case in which tardiness depends on the makespan for a batch of jobs and sequencing is not at issue. In section 4, we generalize to the case in which each job may contribute to tardiness, giving rise to a challenging sequencing problem. In section 5, we summarize computational experiments that compare heuristic solutions to this problem. In section 6, we generalize the model further and consider a weighted case. We show that this weighted case is also a generalization of the stochastic earliness-tardiness (E/T) problem analyzed by Soroush (1999) and by Portugal and Trietsch (2006). Finally, in section 7, we prove asymptotic optimality for two of the heuristics introduced in sections 5 and 6. Rather than begin with a literature review, we provide the main antecedents of our work as we proceed.

2. MEETING SERVICE-LEVEL TARGETS

To begin, we assume that a sequence is given, and we focus on the optimal determination of due dates. When we can set due dates, we are generally interested in setting them as tightly as possible—that is, we wish to minimize

$$D = \sum_{j=1}^{n} d_j$$

The simplest safe scheduling version of this problem is to minimize D subject to stochastic feasibility. (A schedule is *stochastically feasible* when all jobs in the schedule are stochastically on time.) The deterministic analog of this model was studied by Baker and Bertrand (1981), who treated a schedule as feasible if all job due dates are met. A review of the literature on setting due dates can be found in Cheng and Gupta (1989).

Suppose that we have a given sequence and a set of constraints of the form $SL_j \geq b_j$, and for convenience, assume that the jobs are sequenced in numbered order. An analytic solution is conceptually straightforward: for each job, set the due date d_j to the smallest possible value consistent with the service level constraint. In other words, choose d_j to be the smallest value for which $\Pr\{C_j \leq d_j\} \geq b_j$. Because the sequence is known, the completion time of the jth job is the sum of the first j processing times. As long as processing times are stochastically independent, the probability distribution for C_j is described by the convolution of the probability distributions for the first j processing times.

Normal distributions play a significant role in solving stochastic problems of this type. They are usually better models of empirical random behavior than exponential distributions (although the memoryless property of the exponential may lead to more elegant results). Furthermore, using the normal, we can obtain convolutions easily, and we can exploit the central limit theorem to approximate the distribution of the completion time of all but the first few jobs in the schedule. In a practical sense, then, we can assume that job processing times are independent normal distributions without much loss of generality.

An alternative approach would be to use simulation. Such an approach is useful not just because simulation can provide a good approximation with a sufficient sample size. We can also represent non-independent processing times with a simulation methodology. In contemporary applications, the massive data collection and data storage capabilities of ERP systems allow for the capture of empirical data that can be used to model dependent probability distributions using a large sample of observations.

Finally, we can address the sequencing problem. What job sequence will minimize the objective function, D? Although this appears to be a challenging problem in general, some special cases exist. In the deterministic counterpart, the optimal sequence is Shortest Processing Time (SPT). Therefore, in the stochastic case, it makes sense to examine the possibility that the optimal sequence is Shortest Expected Processing Time (SEPT). The following theorem provides such a result, and does not require the normality and the

independence assumptions, but it does require the notion of stochastic ordering. We say that one random variable, X, is *stochastically smaller* than another, Y, if $\Pr\{X \le t\} \ge \Pr\{Y \le t\}$ for any t. (We give the proof of the theorem later.)

Theorem 1: Suppose the objective is to minimize D subject to a common service-level constraint $\Pr\{C_j \le d_j\} \ge b$. If all processing times are in nondecreasing stochastic order, then SEPT is optimal.

3. OPTIMIZING THE BATCH DUE DATE

A more challenging problem involves the trade-off between tight due dates and tardiness performance. To illustrate that trade-off, we consider a special case involving one batch of jobs with a common due date. Equivalently, we can think of a set of jobs that are all delivered to the same customer, feeding an assembly operation. In this case, the makespan of the schedule determines performance, where the makespan is equivalent to the completion time, $C_{\max}$, of the last job in the batch. Internally, our scheduling system assigns a due date d to the batch, to help guide the progress of work in the system. We would like the due date to be tight, but we also want to avoid tardiness. If we set the due date at zero, we would incur substantial tardiness, but to avoid tardiness we would need a very large due date. An objective function that balances these two goals is the following:

$$H(d) = d + \gamma \mathrm{E}(\max\{0, C_{\max} - d\}) = d + \gamma \mathrm{E}(T)$$

where $\gamma > 0$ is a pre-specified weighting factor. To solve this problem, we compute an optimal service level (SL) for the batch and then set the due date so that the optimal service level is achieved (Baker and Trietsch 2007).

Theorem 2: Suppose the objective is to minimize $d + \gamma \mathrm{E}(T)$ for a given set of jobs. Then, if $\gamma \le 1$ we should set $d = 0$. Otherwise, it is optimal to set d equal to the smallest possible value that satisfies the condition

$$SL = \Pr\{C_{\max} \le d\} \ge (\gamma - 1)/\gamma$$

Proof:

»» The proof of Theorem 2 follows the *critical fractile* reasoning familiar to inventory analysts as the newsvendor model. Here, we show how to adapt the discrete form of that result to prove the theorem.

Suppose that we choose a due date of d, and then we observe a batch completion time of $C_{\max}$. In retrospect, we can ask whether it would be desirable to have increased d initially. The marginal effect of a unit increase in d would be to reduce tardiness if the job finished late—that is, if $C_{\max} > d$. Thus, the net marginal effect on the objective function would be $1 - \gamma(\Pr\{C_{\max} > d\})$ or equivalently, $1 - \gamma[1 - F(d)]$, where $F(d)$ denotes the cumulative distribution function (cdf) of the schedule length. It follows that we should increase the due date as long as this expected incremental cost drops; that is, while

$$1 - \gamma[1 - F(d)] < 0$$

or, after rearranging,

$$F(d) < (\gamma - 1)/\gamma$$

Note that this condition can never be met if $\gamma \leq 1$, so in that case, we should never increase the due date, and the optimal due date is zero. Otherwise, we should increase the due date until

$$F(d) \geq (\gamma - 1)/\gamma$$

In other words, we should set d equal to the smallest value for which this inequality holds. ««

Thus, to solve the problem, we need the probability distribution of the makespan, which is independent of the job sequence. The critical fractile of this distribution gives us the optimal due date. Nevertheless, the distribution of the makespan is an n-fold convolution and may not always be tractable. Again, the most relevant applications would rely on the normal distribution.

To summarize, a special case of the trade-off between tight due dates and job tardiness arises when all jobs are processed in a single batch for a common customer. Because the makespan dictates performance, job sequence is not at issue, just as in the deterministic makespan problem. The optimal choice of a due date is then determined according to the critical fractile result in Theorem 2.

4. TRADING OFF TIGHT DUE DATES AND JOB TARDINESS

A generalization of the batch due date model treats each job as having a due date and potentially incurring tardiness. The general objective is to minimize

$$\sum_{j=1}^{n} d_j + \gamma \sum_{j=1}^{n} \mathrm{E}(T_j) = D + \gamma \mathrm{E}(T)$$

As in the special case, we can determine the optimal values of the service levels from a critical ratio (Baker and Trietsch 2007).

Theorem 3: Suppose the objective is to minimize $D + \gamma \mathrm{E}(T)$ for a given sequence. Then, if $\gamma \leq 1$ we should set all due dates to 0. Otherwise, for job j, it is optimal to set d_j equal to the smallest value that satisfies the condition

$$SL_j = \Pr\{C_j \leq d_j\} \geq (\gamma - 1)/\gamma$$

Notice that Theorem 3 assumes that the job sequence is given. In the generalized problem, the ultimate objective is to find the sequence that minimizes $D + \gamma \mathrm{E}(T)$. The optimal sequence is not always easily found, but SEPT is known to be optimal in a special case in which neither the normality assumption nor stochastic independence is required.

Theorem 4: Suppose the objective is to minimize $D + \gamma \mathrm{E}(T)$. If all processing times are in nondecreasing stochastic order, then SEPT is optimal.

Proof:

»» We use an adjacent pairwise interchange argument. Consider a sequence containing adjacent jobs i and k, in that order, where job k is stochastically smaller than job i. We compare the sequence formed by interchanging jobs i and k. The later of the two jobs contributes the same amount to the objective function in either sequence. Let d_i^* and d_k^* be the respective due dates for which Theorem 3 is satisfied for the earlier job. Our task is to show that $d_k^* + \gamma \mathrm{E}(T_k) \leq d_i^* + \gamma \mathrm{E}(T_i)$. A key observation is that $C_k \leq_{st} C_i$ (because if $X \leq_{st} Y$ then $X + Z \leq_{st} Y + Z$ for any Z; here, Z represents the processing time of the jobs preceding i and k). By definition, the cdf of a stochastically smaller distribution yields the minimal d for any desired service level, so $d_k^* \leq d_i^*$. Then, $\mathrm{E}(T_k)$ and $\mathrm{E}(T_i)$ are given by the tail areas of the respective distributions above the cdfs $F_k(x)$ and $F_i(x)$, and below 1 to the right of d_k^* and d_i^*. Because $F_k(x)$ is monotone non-decreasing, if we replace the part of the tail representing

E(T_k) between d_k* and d_i* by a rectangle with width (d_i* − d_k*) and height 1/γ, we obtain an upper bound for E(T_k). By stochastic dominance, the remainder of the tail—to the right of d_i*—cannot exceed E(T_i), so E(T_k) ≤ E(T_i) + (d_i* − d_k*)/γ. Multiplying through by γ we obtain γE(T_k) ≤ γE(T_i) + d_i* − d_k*; i.e., d_k* + γE(T_k) ≤ d_i* + γE(T_i). ««

Notice that Theorem 1 is actually a corollary of Theorem 4. Furthermore, it can be shown for both objectives that if job *i* is stochastically smaller than job *j* then it must come earlier in the optimal sequence, thus yielding a slightly more general result.

The cdfs of any pair of normal random variables with different variances always intersect each other once. Thus we might think that Theorems 1 and 4 never apply to the normal distribution. However, that intersection can occur for a negative value, and we typically ignore negative processing times. Therefore, in a practical sense, cases do exist where the normal distribution yields stochastically-ordered processing times. One simple example is the case of a constant coefficient of variation, as in Example 4. Furthermore, for the purpose of Theorem 1, if the target service level is at least 0.5, it is sufficient if the stochastic dominance applies when the cdfs are truncated at their means; i.e., if E(X) ≤ E(Y) and we wish to know if it is safe to sequence X first, then instead of requiring $X \leq_{st} Y$ we require only that max{X, E(X)} $\leq_{st}$ max{Y, E(Y)}. If $\sigma_X \leq \sigma_Y$, the condition is satisfied. In other words, for the normal distribution, when the means and standard deviations are agreeable, SEPT is the optimal sequence. In such a case we say that *X dominates Y*. Such dominance is also sufficient for our current objective, as the next theorem establishes.

Theorem 5: Suppose the objective is to minimize $D + \gamma$E(T), with independent normal processing time distributions. For any pair of jobs *i* and *j* such that job *i* dominates job *j* (i.e., $\mu_i \leq \mu_j$ and $\sigma_i \leq \sigma_j$ with at least one inequality strict), job *i* must precede job *j* in an optimal sequence.

Proof:

»» Suppose an optimal sequence exists where job *i* appears in the sequence later than job *j*. By interchanging the jobs, the contribution of job *j* becomes identical to the former contribution of job *i* whereas the contribution of job *i* becomes smaller than the former contribution of job *j*. The contributions of any jobs sequenced between the two are also reduced. ««

5. COMPUTATIONAL RESULTS

As indicated in the previous section, finding the optimal sequence in the trade off between tight due dates and job tardiness is a challenging combinatorial problem. However, we know that in the deterministic counterpart, sequencing jobs according to Shortest Processing Time (SPT) provides an optimal sequence. We also know that in the case of stochastically-ordered processing time distributions, SEPT provides an optimal sequence. Thus, we might hypothesize that SEPT is an effective heuristic procedure for this problem.

After observing that SEPT does not provide optimal schedules in every case, we might next look for a slightly different heuristic procedure. Since processing time distributions are assumed to be normal, it makes sense to combine the mean and the variance in some way. A simple way to do so is to sum the mean and the standard deviation, ordering the jobs according to nondecreasing values of $(\mu_j + \sigma_j)$. We call this the MSD heuristic procedure. MSD is a static dispatching rule.

A more sophisticated heuristic procedure derives from considering a pairwise interchange of adjacent jobs. Suppose we have already scheduled a subset of jobs, B, such that their completion time, C_B, has a normal distribution with mean m_B and standard deviation s_B. Let $\varphi(z)$ denote the density function of the standard normal distribution with z corresponding to a cdf value of $(\gamma - 1)/\gamma$. Suppose also that the next two jobs in sequence are jobs i and j, in that order. We investigate whether it would be desirable to interchange jobs i and j. In the original sequence, the contributions to the objective function from the two jobs are:

Original: i: $C_B + \mu_i + \gamma\varphi(z)(s_B^2 + \sigma_i^2)^{0.5}$
j: $C_B + \mu_i + \mu_j + \gamma\varphi(z)(s_B^2 + \sigma_i^2 + \sigma_j^2)^{0.5}$

Interchanged: j: $C_B + \mu_j + \gamma\varphi(z)(s_B^2 + \sigma_j^2)^{0.5}$
i: $C_B + \mu_j + \mu_i + \gamma\varphi(z)(s_B^2 + \sigma_j^2 + \sigma_i^2)^{0.5}$

The original sequence is preferable if

$$\mu_i + \gamma\varphi(z)(s_B^2 + \sigma_i^2)^{0.5} \le \mu_j + \gamma\varphi(z)(s_B^2 + \sigma_j^2)^{0.5}$$

Define $\Delta_k = (s_B^2 + \sigma_k^2)^{0.5} - s_B$ and notice that the condition remains valid if we subtract $\gamma\varphi(z)s_B$ from both sides. This leads to an equivalent condition,

$$\mu_i + \gamma\varphi(z)\Delta_i \le \mu_j + \gamma\varphi(z)\Delta_j$$

In other words, it is desirable to give priority to the job with the smallest value of $\mu_j + \gamma\varphi(z)[(s_B^2 + \sigma_j^2)^{0.5} - s_B]$. This rule is dynamic, rather than static, because the choice between two jobs, which depends on s_B, may be different depending on which jobs have already been scheduled. However, when dominance is present, the rule satisfies Theorem 5. A simple implementation of the pairwise interchange (PI) rule works as follows:

1. Initialize $s_B = 0$.
2. Find job j with minimum $\mu_j + \gamma\varphi(z)\Delta_j$ among unscheduled jobs.
3. Schedule job j next.
4. Update s_B and return to Step 2 until all jobs are scheduled.

As a benchmark for testing the heuristic ordering procedures, we also include a randomly-selected sequence (R) in our comparisons. Note that the evaluation of such sequences presumes that the optimal due dates will be selected (using Theorem 3), so the suboptimality in the randomly-selected sequence is due solely to the sequence.

We next created a set of test problems. For each test problem, we first generated a value of γ, drawn randomly from the interval between 2 and 20. Then, for the processing time of each job j in the problem, we generated a pair of values (μ_j and σ_j) to specify its normal distribution. The mean value was drawn randomly from the integers between ($m - a$) and ($m + a$), and the standard deviation was drawn randomly from the integers between 1 and b.

The parameters m, a, and b define an experimental combination, and for each combination, we generated ten random problem instances. By varying the parameter m, we can scale the processing times to be larger or smaller. We expect that when the processing times are relatively large in scale, stochastic variation plays a minor role, and SEPT should perform well. By varying the parameter a, we can create relatively more or less diversity among the processing times, for a given scale. We expect that when diversity is greater, SEPT should also perform well. By varying the parameter b, we can create relatively more or less stochastic variation. We expect that when stochastic variation is small, SEPT should perform well.

We generated test problems of various sizes up to $n = 20$, but we did not see patterns that differed significantly according to problem size. We report data observed for test problems containing $n = 10$ jobs. For each test problem, we compared the objective function generated by the heuristic procedure to the optimal value.

In the first series of results, we examine the effect of m, which dictates the relative scale of the processing times. For each of the heuristic procedures, we report the percent

suboptimality, averaged over ten replications. (These percentages are small, but remember that we are examining only the sequencing decision; due dates are optimized by applying Theorem 3 to every sequence.) For the PI Rule, we also report the number of problem instances in which optimal solutions were produced. As shown in Table 5, we varied *m* from 75 to 300, keeping the range of mean processing times at ±10 and the standard deviations between 1 and 20.

Table 5

m	*a*	*b*	R	SEPT	MSD	PI Rule	PI opt
75	10	20	5.26%	0.96%	0.09%	0.012%	9
100	10	20	3.10%	0.82%	0.06%	0.001%	9
125	10	20	2.77%	0.76%	0.06%	0.008%	9
150	10	20	2.76%	0.49%	0.03%	0.001%	9
300	10	20	1.80%	0.44%	0.02%	0.005%	9

As expected, the relative performance of SEPT improves as the scale of the processing times increases, and this is also true for the randomly-selected sequence. However, the MSD sequence was better than SEPT by an order of magnitude, and well within 1% of the optimum in most instances. The PI Rule, which requires more computation, obtained an optimal solution in 90% of the test problems.

In the next series of results, we examine the effect of *a*, which dictates the level of diversity among the mean processing times. As shown in Table 6, we varied *a* from 0 to 20, keeping the mean processing times at 100 and the standard deviations between 1 and 20.

Table 6

m	*a*	*b*	R	SEPT	MSD	PI Rule	PI opt
100	20	20	7.35%	0.57%	0.04%	0.000%	10
100	15	20	5.22%	0.61%	0.06%	0.000%	10
100	10	20	2.66%	0.70%	0.09%	0.002%	8
100	5	20	2.66%	1.43%	0.05%	0.000%	10
100	0	20	2.06%	2.06%	0.00%	0.000%	10

As expected, greater diversity leads to improved performance by SEPT. When $a = 0$, there is no diversity, and all jobs have the same mean processing time. In that case, all sequences are SEPT sequences, and the average performance of SEPT (with ties broken arbitrarily) is no different from that of a random sequence. However, MSD produces much better results, and the PI Rule is better still. (For the case $a = 0$, both MSD and PI yield the optimal sequence per Theorem 5.)

Finally, we examine the effect of *b*, which dictates the level of stochastic variation among the processing times. As shown in Table 7, we varied *b* from 5 to 25, keeping the mean processing times at 100 and the range of mean processing times at ±10.

Table 7

m	*a*	*b*	R	SEPT	MSD	PI Rule	PI opt
100	10	25	4.18%	1.64%	0.07%	0.002%	8
100	10	20	3.66%	0.94%	0.08%	0.003%	9
100	10	15	4.12%	0.46%	0.06%	0.008%	9
100	10	10	2.71%	0.16%	0.03%	0.005%	8
100	10	5	2.65%	0.05%	0.00%	0.000%	10
		Overall	**3.53%**	**0.81%**	**0.05%**	**0.003%**	**91.3%**

As expected, the performance of SEPT improves with smaller variances. In the limit, when there is no stochastic variation, MSD and the PI Rule would become identical to SEPT, and all would produce optimal solutions.

Note that one parametric case ($m = 100$, $a = 10$, $b = 20$) was replicated in each of the three comparisons and thus helps us gauge the extent of sampling error. However, the overall average (i.e., over 150 observations) is probably insulated from much of the sampling error and is therefore a good indicator of performance. The last row of Table 7 shows the overall averages.

In these test problems, the selection of γ dictated positive *z*-values and optimal service levels between 50% and 95%. We repeated the experiments for γ values between 1 and 2, corresponding to negative *z*-values. The results were somewhat similar, with SEPT producing better solutions and MSD producing worse solutions. The PI Rule produced optimal solutions in more than 95% of the 150 test problems.

In summary, a solution that is very close to optimal can be constructed by using a static dispatching rule that sorts the jobs according to $(\mu_j + \sigma_j)$ and then calculates the due dates from Theorem 3. Our computational results suggest that solutions that are well within 1% of optimum can be constructed in this way when the underlying processing times follow normal distributions. However, we can improve on that performance with the PI Rule, which is a dynamic heuristic procedure. The PI Rule improves suboptimality performance by an order of magnitude and generates optimal solutions in the vast majority of cases.

6. THE WEIGHTED PROBLEM

In this section, we consider a weighted version of the tightness-tardiness trade off, and we show how it relates to the stochastic E/T problem. We introduce weighting factors $\alpha_j > 0$, and accordingly our objective becomes:

$$\sum_{j=1}^{n} \alpha_j[d_j + \gamma \mathrm{E}(T_j)] \qquad (1)$$

The coefficients α_j weight the contributions of one job as compared to another, but the coefficient γ applies to all jobs because in typical applications, the trade-off of tightness for tardiness applies to the entire job set. We assume that $\gamma > 1$ and thus deal with nonzero due dates. For a given sequence, the optimal due dates for (1) should still satisfy Theorem 3. The challenge, again, is in sequencing.

We can transform this problem to an equivalent form. For any realization of the completion time C_j, we can write:

$$d_j = C_j + \max\{0, d_j - C_j\} - \max\{0, C_j - d_j\} \qquad (2)$$

Next, notice that $\max\{0, d_j - C_j\}$ is the earliness of job j, denoted E_j, whereas $\max\{0, C_j - d_j\}$ is the job's tardiness, T_j. Taking the expectations of both sides of (2), we obtain

$$\mathrm{E}(d_j) = d_j = \mathrm{E}(C_j) + \mathrm{E}(E_j) - \mathrm{E}(T_j).$$

Substituting for d_j in the objective function (1) yields

$$\sum_{j=1}^{n} \alpha_j[\mathrm{E}(C_j) + \mathrm{E}(E_j) + (\gamma - 1)\mathrm{E}(T_j)].$$

If we define $\beta_j = \alpha_j(\gamma - 1)$, then we can write the objective function as follows.

$$\sum_{j=1}^{n} \alpha_j \mathrm{E}(C_j) + \sum_{j=1}^{n} [\alpha_j \mathrm{E}(E_j) + \beta_j \mathrm{E}(T_j)] \qquad (3)$$

Thus, in our formulation, the earliness and tardiness penalties, α_j and β_j, are proportional. Mathematically, however, the objective function could be generalized by replacing γ with γ_j, and (3) would be obtained without a proportionality restriction. Trietsch (1993) discusses a model where such generalized terms arise.

The first sum in (3) is equivalent to the expected weighted completion time. This sum is therefore the objective of the stochastic weighted completion time problem, which is minimized by sequencing the jobs according to shortest weighted expected processing time (SWEPT). That is, the optimal sequence for that sum would be nondecreasing order of μ_j/α_j.

The second sum in (3) is equivalent to the expected earliness/tardiness cost, but with proportional unit penalty costs α_j and β_j. The second sum is therefore a special case of the objective of the stochastic E/T problem, as studied by Soroush (1999) and by Portugal and Trietsch (2006). Our model would thus generalize the stochastic E/T problem if we allowed distinct γ_j values, in which case (3) contains independent α_j and β_j. There is no known polynomial time optimizing algorithm for the stochastic E/T problem, although Soroush provided an effective sorting rule. In our special case, this rule calls for sequencing the jobs according to nondecreasing order of σ_j^2/α_j. Thus, we are faced in (3) with an objective consisting of one term that drives sequencing toward shortest weighted mean processing times and another that drives sequencing toward shortest weighted variances.

One characteristic of the stochastic E/T problem is that the objective function can often be reduced by inserting idle time between jobs. For this reason, the stochastic E/T problem requires an explicit no-idling restriction. However, for (3), and thus also for (1), the no-idling assumption is unnecessary. We next prove this property for the general case and without requiring normality or stochastic independence.

Theorem 6: Suppose the objective is to minimize $\sum_{j=1}^{n} \alpha_j \mathrm{E}(C_j) + \sum_{j=1}^{n} [\alpha_j \mathrm{E}(E_j) + \beta_j \mathrm{E}(T_j)]$. There exists an optimal solution without inserted idle time.

Proof:

»» We prove by contradiction and induction. Assume that some idle time ("delay") of $A > 0$ must precede job $[n]$ in the optimal solution. Let $C_{[n]}$ denote the completion time including the effect of the delay, and let $C'_{[n]}$ is the completion time when no delay is imposed. Then $C'_{[n]} \leq_{\mathrm{st}} C_{[n]}$, because the completion time cannot be earlier when the start time is delayed. Therefore, if we draw the cdfs of $C'_{[n]}$ and $C_{[n]}$, denoted $F'_{[n]}(x)$ and $F_{[n]}(x)$, there must be a gap between them with an area of A (although this gap is not necessarily contiguous). Now draw a perpendicular line through the optimal due date for $C_{[n]}$, as $d_{[n]}$* (determined by Theorem 3). This line partitions the gap between $F'_{[n]}(x)$ and $F_{[n]}(x)$ into two non-negative areas, B_1, and B_2, to the left and right of $d_{[n]}$*, such that $B_1 + B_2 = A$. Removing the delay leads to a direct benefit of $\alpha_{[n]}A$ (by reducing $\alpha_{[n]}\mathrm{E}(C_{[n]})$) plus $\beta_{[n]}B_2$ (by reducing $\beta_{[n]}\mathrm{E}(T_{[n]})$). The cost of removing the delay is an increase in earliness penalty of $\alpha_{[n]}B_1$. But $\alpha_{[n]}B_1 \leq \alpha_{[n]}A \leq \alpha_{[n]}A + \beta_{[n]}B_2$. Thus, the total cost of removing the delay cannot exceed the benefit, and the

delay cannot be necessary for optimality. Furthermore, we may be able to achieve an additional benefit by adjusting the due date to its new optimal value. This argument completes the proof for the last job; for preceding jobs, we use induction for jobs $[n-1]$, $[n-2]$ etc., noting that removing any delay reduces not only the completion time of the imminent job but also that of all subsequent jobs. ««

From the numerical experience reported in Section 5, we should not expect SWEPT to be a good heuristic for the weighted case because it does not account for variance. Instead, we can adapt each heuristic rule. The simplest weighted version of MSD sequences the jobs according to nondecreasing values of the ratio $(\mu_j + \sigma_j)/\alpha_j$. We use the acronym WMSD for this rule. In the same way, a heuristic adaptation of the PI Rule gives priority to the job with the smallest value of $[\mu_j + \gamma\varphi(z)\Delta_j]/\alpha_j$. We refer to this procedure as the WPI Rule. To test these heuristic adaptations, we conducted another set of computational experiments with test problems containing weights and found results that were similar to the unweighted case. Table 8 shows the overall summary, averaged over another set of 150 test problems.

Table 8

	R	SWEPT	WMSD	WPI Rule	WPI opt
Overall	**32.92%**	**0.23%**	**0.03%**	**0.007%**	**87.3%**

In Table 8, we see that the randomly-generated sequence fares worse than in the unweighted case, but the heuristic rules still perform well. The WMSD rule, which again is a static rule, improves on SWEPT by roughly an order of magnitude. The WPI Rule, a dynamic priority rule, improves by nearly another order of magnitude and again produces optimal solutions in the vast majority of test problems.

7. ASYMPTOTIC OPTIMALITY

In the stochastic E/T problem, Soroush (1999) demonstrated the effectiveness of a static sorting procedure that sequences the jobs in nondecreasing order of $\sigma_j^2/(\alpha_j + \beta_j)\varphi(z_j)$, where z_j denotes the critical fractile of the normal distribution for the value $\beta_j/(\alpha_j + \beta_j)$, in analogy to Theorem 3. Portougal and Trietsch (2006) showed that this sorting procedure is asymptotically optimal. A heuristic rule is said to be *asymptotically optimal* if its suboptimality percentage becomes negligible as n grows large. In this section, we explore this same property as it applies to the weighted version of our problem, with the objective function in (3). A *string* is defined as a set of jobs that must appear contiguously in a fixed

order. For our purpose, we treat a job with weight α_j, mean μ_j and variance σ_j^2 as a string of α_j unweighted jobs, each with mean μ_j/α_j and variance σ_j^2/α_j. (This representation is most convenient when weights are integers but we can always rescale noninteger weights approximately as integers without changing sequencing decisions in any important way.) Portougal and Trietsch showed that using such strings yields a lower bound on the expected E/T penalty. This lower bound is asymptotically equal to the correct value in the sense that the difference becomes relatively negligible as n grows. As for the completion time component of (3), it can be shown that using strings in this manner leads to the correct value minus a constant.

Theorem 7: Suppose the objective is to minimize $\sum_{j=1}^{n} \alpha_j E(C_j) + \sum_{j=1}^{n} [\alpha_j E(E_j) + \beta_j E(T_j)]$ and that $\beta_j/\alpha_j \le \delta < \infty$, $\sigma_j^2/\alpha_j \le \eta^2 < \infty$, and $\mu_j/\alpha_j \ge \lambda > 0$ for all j. Then, sorting the jobs by μ_j/α_j (SWEPT) is asymptotically optimal.

Proof:

»» Rather than provide a complete formal proof, we just show that as n grows large the E/T contribution to the objective function becomes relatively negligible. The theorem follows because SWEPT optimizes the remaining part of the objective. For convenience, we use strings and thus transform SWEPT to SEPT. For any $n > 1$, assume $n - 1$ jobs have already been sequenced with a mean $m_{(n-1)} \ge (n-1)\lambda$ and a standard deviation $s_{(n-1)} \le (n-1)^{0.5}\eta$. Now consider the contribution of job [n] to the objective function. The flow time contribution is $\alpha_n(m_{(n-1)} + p_j) \ge \alpha_n n\lambda$. For any distribution, it can be shown that the contribution of job n to the expected E/T penalty is proportional to s_n. In particular if we assume that the processing time distributions are normal, or that n is large enough to invoke the central limit theorem, then this contribution is given by $(\alpha_n + \beta_n)\varphi(z^*)s_n$. In our case, $(\alpha_n + \beta_n)\varphi(z^*)s_n \le \alpha_n(1 + \delta)\varphi(0)\eta(n)^{0.5}$. Taking the ratio of the E/T contribution to the flow time contribution we obtain at most $\alpha_n(1 + \delta)\varphi(0)\eta(n)^{0.5}/\alpha_n n\lambda = (\eta/\lambda)(1 + \delta)\varphi(0)/n^{0.5}$. But for any admissible δ, η and λ, as $n \to \infty$, $(\eta/\lambda)(1 + \delta)\varphi(0)/n^{0.5} \to 0$. ««

Notice that Theorem 7 applies to the unweighted case covered earlier. In other words, SEPT is asymptotically optimal for the objective of minimizing $D + \gamma E(T)$. However, as we saw in the computational results, better heuristic procedures than SEPT exist when there are a limited number of jobs. In the weighted case, where SWEPT is asymptotically optimal, we

also were able to identify better heuristic procedures for a limited number of jobs. The best of both worlds, in a sense, is represented by the WPI Rule. Not only is this procedure capable of producing an optimal solution in most of the cases with few jobs, but it is also asymptotically optimal, as we demonstrate next. To this end, note that Theorem 7 implies that in (3), the expected E/T penalty becomes negligible relative to the weighted completion time as n grows large (and this remains true for any sequence). So our main task is to show that the WPI Rule is asymptotically optimal with respect to the completion time component of (3). This property is not obvious because relative to the optimal sequence, the WPI Rule tends to postpone jobs with high variance even if their weighted means are small, and thus it may lead to a larger completion time component. In our proof, we conservatively assume that this is the case; otherwise, asymptotic convergence would occur even faster. That is, we show that although the weighted completion time obtained under the WPI Rule may be higher than under the optimal sequence, the relative difference is driven to zero asymptotically as n grows large.

Recall that the WPI Rule sorts jobs by $[\mu_j + \gamma\varphi(z)\Delta_j]/\alpha_j$, and therefore, we can say that we partition each job into a string of α_j unweighted jobs each with the same Δ_j value (given s_B, as defined by the preceding string). Strictly speaking, this is not equivalent to allocating the variance to the jobs equally, but the difference is asymptotically negligible. (It can be shown that as s_B grows large, Δ_j is given by $\sigma_j^2/2s_B$, and allocating it equally to the jobs making up the strings is practically equivalent to allocating the variance equally.) Furthermore, we don't need the assumption that the variance is allocated equally because the sequencing rule remains intact. Let S^{WPI} denote the sequence obtained by WPI, let S^* be the optimal sequence. Without loss of generality, index the jobs according to S^{WPI}. For some $k < n$ and a sequence S, let $S[\leq k]$ denote the subsequence of S from job [1] to job [k]. Similarly, let $S[>k]$ denote the subsequence of S from job [k + 1] to job [n], to which we refer as the *tail*; e.g., the tail of S^{WPI} comprises jobs k + 1, k + 2, . . ., n. Let $f(S)$ be the objective function value of sequence S, and if the argument is a subsequence—e.g., $f(S[>k])$—then we interpret f as the contribution of the jobs in the subsequence to the objective function (i.e., $f(S) = f(S[\leq k]) + f(S[>k])$). A lower bound on $f(S)$ may be obtained by considering only the completion time component of the objective function. Let C_B^{WPI} and C_B^* denote the completion times of the batches consisting of the first k jobs under sequences S^{WPI} and S^*, and similarly let s_B^{WPI} and s_B^* denote the standard deviations of C_B^{WPI} and C_B^*. With this background we are ready to prove our result more formally.

Theorem 8: Suppose the objective is to minimize $\sum_{j=1}^{n} \alpha_j E(C_j) + \sum_{j=1}^{n} [\alpha_j(E(E_j) + (\gamma - 1)E(T_j))]$ and that $\gamma < \infty$, $0 < \delta^2 < \sigma_j^2/\alpha_j \leq \eta^2 < \infty$, and $\mu_j/\alpha_j \geq \lambda > 0$ for all j. Then, sorting the jobs by $[\mu_j + \gamma\varphi(z)\Delta_j]/\alpha_j$ is asymptotically optimal.

Proof:

»» Using strings, we may assume that all jobs have equal weights; so the adapted conditions are $0 < \delta < \sigma_j \leq \eta < \infty$, and $\mu_j \geq \lambda$. For an arbitrarily small but positive ε, we have to show that there exists a value n_ε such that for any $n > n_\varepsilon$, $(f(S^{PI}) - f(S^*))/f(S^*) < \varepsilon$. We start by producing a finite k value (namely k_ε) for which $f(S^{PI}[>k]) < f_L(S^*[>k])(1 + \varepsilon/2)$. Notice that $E(C_B^{PI})$ and $E(C_B^*)$ must both exceed λk, whereas both s_B^{PI} and s_B^* are in the range $[k^{0.5}\delta, k^{0.5}\eta]$. Using η as the upper bound on $\sigma_{(k+1)}$ and $k^{0.5}\delta$ as the lower bound on s_B^{PI}, it can also be shown that $\Delta_{(k+1)} < \eta^2/2k^{0.5}\delta$. Now select the integer k_ε such that in job $(k_\varepsilon + 1)$, the marginal contribution of the variance to the objective function will be at most $\varepsilon/2$ as large as the marginal contribution to the total completion time, $\mu_{(k+1)}$. This condition implies $\Delta_{(k+1)}\gamma\varphi(z^*) \leq \mu_{(k+1)}\varepsilon/2$, and if we use the upper bound for $\Delta_{(k+1)}$ and the lower bound for $\mu_{(k+1)}$ it yields $k_\varepsilon = [\gamma\varphi(z^*)\eta^2/\delta\varepsilon]^2$. This choice guarantees that the relative marginal contribution of the variances of subsequent jobs will also be below $\varepsilon/2$ times the marginal contribution of the mean processing time to the total completion time. We make the conservative assumption that $E(C_B^{PI}) > E(C_B^*)$. On the one hand, if we measure the completion time cost of the tail starting at C_B^{PI}, it will not be larger than that associated with $(1 + \varepsilon/2)$ times the value obtained by applying SEPT to the tail of S^*. This yields an upper bound on the deviation from the optimal expected completion time value: we must be within $\varepsilon/2$ of the optimal value. On the other hand, under our conservative assumption, for each of the jobs in the tail, there is an additional nonnegative contribution to the completion time, $(C_B^{PI} - C_B^*)$, which may yield a difference that tends to grow linearly with n. However, the expected total completion time is not smaller than $n^2\lambda/2$, so that difference can also be driven below $\varepsilon/2$ in the relative sense, yielding the required convergence within ε when considering both the variance and the completion time together. ««

8. SUMMARY

We have considered a collection of stochastic scheduling models in which due dates are decisions. These models give rise to problems in safe scheduling because job tardiness is interpreted in stochastic terms: a job is stochastically tardy if it fails to meet its service-level

requirement. We began with a simple model in which the only task is to set due dates as tightly as possible, consistent with given service level requirements. This model illustrates how service-level performance can be recognized in scheduling.

We next examined a model in which the tightness of due-dates conflicts with the desire for little or no tardiness. We solved a special case of this model in which all jobs are delivered to the same customer, exploiting the critical fractile result that is a familiar building block in stochastic inventory theory. We then generalized this model so that each job faces its own trade off between due-date tightness and tardiness. Although this model gives rise to a challenging combinatorial problem, we presented a simple heuristic procedure that sorts the jobs and then computes optimal due dates. Our computational evidence indicates that this procedure can routinely produce solutions that are within 1% of optimality. A dynamic sorting procedure—the PI Rule—can obtain optimal solutions most of the time.

Finally, we generalized our model with weighting factors and showed that it also provides a generalization of the stochastic E/T model. We then proved that SWEPT is an asymptotically optimal heuristic procedure for the problem. However, in smaller versions of the problem, we can employ better sorting procedures than SWEPT. The WPI Rule is also asymptotically optimal, and it generates optimal solutions most of the time when the problem contains a limited number of jobs. Thus it can be recommended for problems of any size.

Future research on this topic should address more complex environments, such as flow shops, job shops, and projects. Another important generalization would be to relax the assumption of stochastic independence. In practice, processing times are often subject to common causes that render them stochastically dependent, so this would be an important generalization. As we have noted, some of our results already apply without stochastic independence, but a comprehensive treatment of stochastic dependencies would be a welcome contribution.

REFERENCES

Akker, van den J.M. and Hoogeveen, J.A. Minimizing the number of late jobs in case of stochastic processing times with minimum success probabilities. *Journal of Scheduling*; 2007.

Baker, K.R. *Elements of Sequencing and Scheduling,* Dartmouth College, Hanover, NH; 2005.

Baker, K.R. and Bertrand, J.W.M. A comparison of due-date selection rules. *AIIE Transactions* 1981; 13, 123-131.

Baker, K.R. and Trietsch, D. A tutorial on safe scheduling. *Tutorials in Operations Research* INFORMS; 2007.

Cheng, T.C.E. and Gupta, M.C. Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research* 1989; 38, 156-66.

Pinedo, M. *Scheduling: Theory, Algorithms, and Systems*, 2nd edition, Prentice Hall; 2002.

Portougal, V. and Trietsch, D. Setting Due Dates in a Stochastic Single Machine Environment, *Computers & Operations Research* 2006; 33, 1681-1694.

Soroush, H.M. and Fredendall, L.D. The Stochastic Single Machine Scheduling Problem with Earliness and Tardiness Costs. *European Journal of Operations Research* 1994 ; 77, 287-302.

Soroush, H.M. Sequencing and Due-Date Determination in the Stochastic Single Machine Problem with Earliness and Tardiness Costs. *European Journal of Operations Research* 1999; 113, 450-468.

Trietsch, D. and Baker, K.R. Minimizing the number of tardy jobs with stochastically-ordered processing times. *Journal of* Scheduling; 2007.

Trietsch, D. Scheduling Flights at Hub Airports. *Transportation Research, Part B* (*Methodology*) 1993; 27B, 133−150.