Research Notes for Chapter 7*

Chapter 7 is our first chapter on safe scheduling, and indeed, to our knowledge, it is the first chapter on safe scheduling in any textbook. For this reason, our research notes for this chapter are extensive. We cover both historical background and advanced theoretical results (including some proofs that we omitted in the chapter). We mention some open research questions as we go and conclude with a brief list of additional open research questions. We also provide a simple expression for the minimum of $d + \gamma E(T)$ when processing time is lognormal (and $\gamma > 1$). Finally, the reference list is relatively extensive, but emphasizes early publications. Because this is not our last chapter on safe scheduling, we elaborate on some of these issues later, especially in Chapters 11 and 18 and their research notes.

HISTORICAL BACKGROUND

All safe scheduling models are stochastic. In its purest form, stochastic scheduling is based on the assumption that processing time distributions are known. This assumption is not as strong as one might think. First, it is often possible to use historical data to obtain such distributions. For instance, Trietsch, Mazmanyan, Gevorgyan and Baker (2010) demonstrated that the lognormal distribution fits project activity times obtained in a field study. Furthermore, they validated its use in representing such activity times based only on information available during the scheduling stage. Even if there is absolutely no historical data, decision makers must make decisions somehow, and their beliefs and estimates can be translated to distributions. Bayesian statistics is predicated on the ability of decision makers to assess such distributions at least implicitly.

Due Date Setting in Queueing Systems

A relatively early study that involves due date setting with safety time is Wein (1991). He analyzes sequencing and due date assignment rules within a multiclass M/G/1 queueing framework (i.e., with exponential time between arrivals, general processing time distribution and one machine/server). He also reports experimental results, but only for the more iconic M/M/1 system, where processing time is exponential too. The objective is to minimize weighted flowtime either subject to a prescribed service level or subject to a constraint on mean weighted tardiness. That is, he addresses our two main safe scheduling approaches. One of his main conclusions is that setting due dates correctly to match the desired service level or mean tardiness constraints is more important than selecting the best sequencing rule. Nonetheless, the findings also indicate

---

 The main purposes of the Research Notes series are to provide historical details about the development of sequencing and scheduling theory, expand the book's coverage for advanced readers, provide links to other relevant research, and identify important challenges and emerging research areas. Our coverage may be updated on an ongoing basis. We invite comments and corrections.

Citation details: Baker, K.R. and D. Trietsch (2010) Research Notes for Chapter 7 in *Principles of Sequencing and Scheduling* (Wiley, 2009). URL: http://mba.tuck.dartmouth.edu/pss/.

that MDD performs either best or close to best relative to other sequencing rules such as MST, EDD and SEPT. (We discuss experimental results of this sort much more extensively in Chapter 15.)

Models with Machine Breakdowns

Throughout the coverage of safe scheduling in the text, our focus is on the pure stochastic environment, with probabilistic processing times. But randomness in processing times could also arise because equipment breaks down and unscheduled maintenance must be carried out. Breakdowns can lead to similar problems as stochastic processing times, and in single-machine problems the two sources of randomness have comparable effects (e.g., see Zhou and Cai, 1997; Ng et al., 1999). Nonetheless, in more complex models, the situations are intrinsically different. For example, if we have parallel machines then a breakdown in one may lead to rescheduling jobs from that machine to other machines, but not necessarily a difference in processing times. Indeed, there is an extensive literature on *reactive scheduling*, which in most expositions, postulates that disruptions (typically machine breakdowns) will occur. In that research, the necessary probabilistic information describes the occurrence of breakdowns and the distribution of repair times, while processing times remain deterministic or at least sensitive to the effects of disruption in deterministic ways. In that situation, the task is to adapt the schedule to the effects of the disruption. Some of the key papers in this area are due to Leon, et al. (1994), Mehta and Uzsoy (1998), and McKay et al. (2000). Although this line of research can be considered as addressing stochastic problems, it is substantively different from pure stochastic scheduling and beyond the scope of our text. (Our implicit assumption is that a sequence that is determined with sufficient safety time is as likely to remain valid after a machine breakdown as it would be after a longer than expected processing time due to any other cause. To the extent this assumption is invalid, and indeed it may be problematic in multi-machine models with long repair times, research is required to address safety subject to machine breakdowns. The difficulty is in considering the interactions between safety time and reactive scheduling issues.)

Robust Scheduling

Another related line of work is *robust scheduling*. Whereas the phrase was used in seminal work by Leon et al. (1994) and by Daniels and Kouvelis (1995), robust scheduling does not have a standard definition. Some papers associate robust scheduling with "predictability" but then have difficulty quantifying what that means and often end up using surrogate measures (Leon et al., 1994; Mehta and Uzsoy, 1999). Daniels and Carillo (1997) extended the notion of robustness and defined *β-robustness* as maximizing the probability that the performance measure will be at least as good as a given target; that is akin to satisfying chance constraints. (Compare to Corollary 6.1, originally proved by Banerjee 1965.) Other papers associate robust scheduling with insulating the schedule from disruptions, so their work is perhaps more appropriately classified as belonging to the literature on reactive scheduling (Mehta and Uzsoy, 1998, and Bollapragada and Sadeh, 2004). Still other papers, such as those stimulated by Daniels and Kouvelis (1995), use a definition of robustness that adopts the *minimax regret* criterion from decision theory (i.e., minimizing the difference between the realized outcome and the best outcome that could have been achieved with hindsight). This definition does not use probability distributions, so it does not lead to stochastic scheduling problems of the type we address. Another common assumption we make is that all necessary decisions are made in advance. This process is sometimes called *off-line scheduling*, or *predictive scheduling*, as distinguished from reactive scheduling. At the risk of confusing off-line scheduling with problems where all jobs are released at time zero, it may also be called *static*, because the sequence itself is static rather than subject to dynamic changes. To keep our focus manageable, we limit our scope to predictive scheduling with stochastic processing times. In this approach, the complete schedule is determined at time zero, before the realizations of any processing times are known, but the complete schedule may call for releasing a job at some future date. Predictive scheduling is often preferred to dispatching in practice because it increases predictability in an uncertain environment. Finally, even if dynamic sequencing changes are desirable, it is usually beneficial to at least start with a good predictive schedule that serves as a basis for dynamic change. Having said that, recent surveys of scheduling-related research that lies beyond our definition of pure stochastic scheduling may be found in Aytug et al. (2005), Black et al. (2006), Herroelen and Leus (2005), and Kouvelis and Yu (1997).

Fuzzy Logic for Control and in Scheduling

Another alternative that has been proposed for stochastic scheduling is the use of fuzzy logic. Fuzzy logic has provided a major breakthrough for control problems involving dynamic continuous adjustment of parameters; that is, fuzzy logic is a very practical approach for controlling adjustable processes. Indeed, fuzzy controllers often provide amazing results, such as the ability to balance two or even three sticks on top of each other. Interestingly, one of the practical strengths of fuzzy controllers is that they don't require precise information about the system. Rather, they adjust by feedback and simple rules provided by experts. In that framework, the system state is represented by a fuzzy measurement: its degree of *membership* in various relevant sets. For example, the system may be assessed as 80% a member of the "hot" set and 30% "high speed." Combination sets such as "hot and high speed" can also be defined (because sometimes the best response to a combination is different from the sum of the best responses to the components). In our case, suppose the membership in this combination set is 10%. The controller then has to select a response from a set of available options, and it can do so with probabilities that reflect the relative membership. In our example, it can select the response fitting "hot" (one designed to reduce the temperature) with a probability of $0.8/(0.8 + 0.3 + 0.1) = 2/3$, the response of reducing speed is selected with a probability of $0.3/1.2 = 1/4$, and the response appropriate for "hot and fast" is selected with a probability of 1/12. This adjustment-selection process repeats frequently, so the controller is likely to switch between responses frequently, always based on current feedback. In this framework, *membership functions* are used to measure membership, on a scale between 0 and 100% for each set (80%, 30% and 10% in our example). The role of membership functions, again, is to guide the probability with which the controller will take a particular adjustment step. If the adjustment is wrong, the system is likely to slide towards an undesirable state, and its membership in the set that caused the wrong decision is reduced. At the same time, its membership in a set that requires reversing that adjustment increases. During such a slide, the membership profile changes, and the probabilities of the various responses change. Thus, in effect, membership functions can be used to guide effective feedback control.

Perhaps due to its spectacular success in continuous control, fuzzy logic has also been promoted for scheduling problems. Proponents of this approach claim that it is more capable of addressing practical needs. For example, the following quote: "instead of *optimising average behaviours* like in stochastic scheduling, fuzzy techniques rather aim at finding robust fault-tolerant schedules where all the constraints are satisfied to some extent, with a sufficient level of confidence." (Dubois et al. 2003; emphasis added). In effect, they propose to model the extent to which constraints are met by membership functions. There is no known way to construct such membership functions from data, however, and in the scheduling context they are inherently subjective. (By contrast, in control applications, the same experts who provide the adjustment rules can also help adjust the membership functions until the controller operates well.) The objective of maximizing membership is thus another example of using a surrogate measure and does not promote objectivity. Furthermore, sequencing problems, by nature, cannot be "dynamically adjusted" to optimality by a quick succession of potentially conflicting responses based on real-time feedback. Instead, they require discrete choices that cannot be changed in the short run. Nonetheless, we fully agree that solutions "where all the constraints are satisfied … with a sufficient level of confidence" are desirable. Our position is that safe scheduling models address this need directly and more objectively.

Safe Scheduling in the Context of the Alternative Approaches

In the text, rather than cover robust scheduling, or go into even more esoteric approaches such as fuzzy logic, we adopt the Bayesian approach and the use of models that include safety time (implicitly or explicitly). Our main justification for this decision is that robust scheduling models still require equally heroic assumptions about processing times and the nature of disruptions, but they yield results that are less powerful than those we obtain. Furthermore, in the pursuit of robustness, some robust scheduling models completely ignore the level of the primary performance measure. Other models consider solutions that are efficient in the sense that they trade off the primary performance measure with robustness. However, this approach is inherent in safe scheduling, because it treats total cost as a function of both the mean and the variance of the primary objective. In effect, small variance implies robustness. By assumption, robustness is important for the purpose of allowing managers to manage risk, and we address this problem head-on and more effectively with safe scheduling models. For example, we accommodate risk-averse decision makers by including the quadratic tardiness element in (RN5.1), which we repeat here:

$$f(S) = \sum_{j=1}^{n} [I(j)(w_j F_j + u_j \delta(T_j) + \alpha_j E_j + \beta_j T_j + \gamma_j T_j^2) + (1 - I(j))v_j] \qquad \text{(RN5.1)}$$

Stochastic counterparts of the models incorporated within (RN5.1) automatically balance the cost of safety with the primary performance measure.

Whereas the economic approach to safe scheduling promotes robustness, chance-constrained models can reduce robustness unless care is exercised. This is rarely an issue when service-level targets are high but there is no inherent requirement in the approach that forbids low targets. If the magnitude of tardiness counts, but we use service-level targets for convenience, we run the risk that a few jobs will be very tardy and incur large economic costs. For instance, in Chapter 15 we discuss classic simulation results for job shops that demonstrate this type of behavior when sequencing by SPT. By favoring short jobs we achieve schedules that tend to have fewer tardy jobs but tardy jobs are liable to be very tardy and thus such scheduling exhibits good performance in terms of satisfying chance constraints but bad performance in terms of minimizing economic costs. A particular danger exists when shop performance is measured by the fraction of tardy jobs, and yet the magnitude of tardiness counts. If so, once a job is counted as tardy there is no incentive to finish it at all, and tardiness thus increases without control (Spearman and Zhang, 1999). In our chance-constrained models with due dates as decisions, however, we assume jobs are performed in the correct order regardless of tardiness. Again, this is a concern in cases where the real economic damage increases with tardiness; e.g., it would not be a problem if tardy jobs are indeed useless (as in missing a boat). In this connection, recall that we address cases where jobs should only be performed if they are sufficiently likely to be on time by the stochastic *U*-problem.

Safe Scheduling and Stochastic Inventory Models

Both approaches to safe scheduling—stochastic feasibility under chance constraints and minimizing economic costs—have roots in inventory theory and both can be applied to time-setting with or without sequencing decisions. Historically, perhaps because inventory models are not analogous to sequencing models, time-setting models came first, whereas sequencing considerations were not addressed until much later. The seminal paper on stochastic inventory models is Arrow, Harris and Marschak (1951). In Section 3 of their paper they consider a rather general formulation of what they call the *static* problem, which involves ordering stock on a non-periodic basis. This formulation includes the case of piecewise linear penalties, where overage costs are associated with holding too much inventory and shortage costs are associated with preparing too little. In addition, they allow a fixed shortage element. For completeness, we transform their main result (Equation 3.8) into our terms. In their paper the amount stocked is denoted by S (or S^* if optimal), and we replace it by d_j (or d_j^*). They use the terms c and b_0 to denote components of α_j (i.e., $c + b_0 = \alpha_j$) and the terms B and a (where $a > c + b_0 = \alpha_j$) to denote components of $\alpha_j + \beta_j$ (i.e., $B + a = \alpha_j + \beta_j$). They also use some terms that are not applicable to scheduling (such as a term that reflects a quantity discount in purchasing). In our terms, their Equation 3.8 is then the following optimality condition,

$$\alpha_j - u_j f(d_j^*) - (\alpha_j + \beta_j)[1 - F(d_j^*)] = 0 \qquad \text{(RN7.1)}$$

In addition, the following second order condition is required,

$$-u_j f'(d_j^*) + (\alpha_j + \beta_j) f(d_j^*) > 0$$

where $f'(d_j*)$ is the derivative of the density function, $f(d_j*)$. (This second order condition is guaranteed for the critical fractile model due to its convexity.) Although the critical fractile result is clearly a special case obtained when $u_j = 0$, the authors do not present it explicitly. Instead, they show the optimal solution for the special case where $\beta_j = 0$. In that case, (RN7.1) yields $f(d_j*) = \alpha_j/u_j$, provided the density function is decreasing at d_j* (to satisfy the second order condition). This solution is not guaranteed to exist, however (in which case $d_j* = 0$). Another model they addressed involves dynamic reordering policies, including calculations involving maximization of net present value. (Safe scheduling models with net present value calculations may be developed, and indeed there are some project management results that take discounting into account, but to our knowledge, models that discount the costs and the benefits of safety time have not been developed yet.)

The critical fractile model, however, has earlier roots in the first Operations Research text by Morse and Kimball, originally published by the US Navy as Report OEG 54 in 1946.[*] When applied to stock levels (rather than to time), the critical fractile model is often referred to as the newsboy model; or, in more modern terms, as the newsvendor model. On page 32 of the report, Morse and Kimball solve the stocking problem of a newsboy who purchases newspapers for 2 cents each and sells them for 3 cents subject to random demand with a specific distribution (Poisson). The objective is to maximize the expected profit, which in this case is equivalent to minimizing the expected regret. The newspapers serve as an example for any single perishable stock item with possible overage and shortage costs. Morse and Kimball stressed that the newsvendor should not necessarily stock to meet the expected demand (in their example, a smaller stock is optimal), but they stopped short of presenting the critical fractile result explicitly.

Safe Scheduling and Stochastic Programming

In addition to the connection to stochastic inventory models, there is also a historical connection to the two main approaches to stochastic programming. The first paper about stochastic programming—Dantzig (1955)—took the economic approach. That approach also lies at the core of the utility approach to game theory (von Neumann and Morgenstern 1944), which preceded Dantzig's work. In this approach, reality is represented by a set of scenarios, also known as *states of nature,* with given probabilities. Indeed, our use of a stored sample is tantamount to listing several equally-likely states of nature, so one can trace our approach to these roots. Whereas von Neumann and Morgenstern focus on maximizing utility, our economic approach is based on minimizing disutility. Nonetheless, there is no significant difference between the two approaches. For example, Equation (RN5.1) can be viewed as a constant ($\sum_{j=1}^{n} v_j$) from which we subtract the minimal possible total cost, thus maximizing net utility. Dantzig's technical focus is on problems with two stages where the decisions in the first stage, together with the state of nature that is revealed later, form the basis of the decisions in the second stage. (The same structure can be extended to more than two stages, but it becomes much less tractable.) Dantzig's model—although originally presented as a version of linear

---

[*] Report OEG 54 had been a classified document. The unclassified text was published in 1951. The original report is currently available (gratis) on the web. We are indebted to Saul Gass for this reference.

programming—is known as *stochastic programming with recourse* because at the second stage, we have the ability to adjust to the now-revealed state of nature. Within the text, we use a very simple form of recourse, namely our second-stage decision is to begin the next job immediately or wait, and we can present it in advance in the form of a release date for each job (a decision variable). In other words, we can cast our approach as stochastic programming with recourse. In general, however, more complex recourse may be useful in stochastic scheduling models. For example, we may reschedule the remaining jobs as information about the state of nature is revealed during processing of the first few jobs. That is, while performing a schedule we can collect information about processing times, update our information for the remaining processing (potentially including updated processing time distributions), and use the updated information dynamically to reschedule the remaining jobs. Such dynamic models may constitute a fruitful area for future research, but not much has been done along these lines yet. The other main stochastic programming approach heralds the stochastic feasibility approach by employing arbitrary (or exogenously dictated) chance constraints. The earliest important publication in this area is Charnes and Cooper (1959). A related earlier paper is Charnes et al. (1958). However, in these seminal papers, neither Dantzig nor Charnes and his collaborators addressed scheduling specifically.

Safe Scheduling and Utility Functions

We now return to the issue of achieving robustness by an appropriately chosen economic cost function. von Neumann and Morgenstern (1944) introduce the first theoretical model that attempts to represent human choice in the face of uncertainty as the minimization (or maximization) of an expected value. It is useful to look at this model as determining the value of a lottery ticket that yields a random return. For small repetitive lotteries the expected dollar amount returned is a sufficient measure because over the long run, after many repetitions of the lottery, the average return will not differ by much from the expected value. But the authors recognized that when it comes to large returns—including large possible losses—the expected value no longer reflects typical human preferences. It is generally recognized that most people are *risk averse*: a large loss is more important to them than an equally large gain. In contrast, decision makers who are happy with the basic expected-value approach are *risk neutral* (and people may also actively seek risk—especially when the potential losses involved are tolerable whereas the potential gains are impressive). For example, buying insurance is rational for a risk-averse person who shuns large losses (although on average the insurance company pays out less than the premium), but the same person may also buy a lottery ticket for a small amount that produces great wealth with a very small probability, although the expected monetary gain is again negative. von Neumann and Morgenstern suggested the use of nonlinear *utility functions* to model such behavior. For example, if the utility function is concave—e.g., logarithmic with the return—then the marginal utility of the return is monotone decreasing (e.g., increasing a small return by one dollar has more utility than increasing a large return by one dollar). In our context, a risk-averse decision maker would want protection from very large tardiness at a rate that is proportionally higher than for small tardiness. But we chose to look at penalties rather than at positive returns, so a risk-averse decision maker will have a convex increasing penalty function such that the marginal penalty of an increase in tardiness will be increasing. In other words, if we use the expected value to compare schedules, we should find a way to incorporate risk-aversion into our loss functions. Adding a nonnegative quadratic tardiness cost element to our generic loss function achieves this end. The quadratic element, or more generally any strictly convex increasing function of tardiness, penalizes high tardiness at a proportionally higher rate and thus discourages large tardiness. Concentrating on the relevant part of (RN5.1), under the assumption that job *j* must be performed,

$$g_j(T_j) = u_j\delta(T_j) + \alpha j\ Ej + \beta jTj + \gamma_j T_j^2); \quad u_j, \alpha_j, \beta_j, \gamma_j > 0$$

the element $\gamma_j T_j^2$ serves this purpose. Emphatically, we do not include a quadratic penalty for lateness, but rather for tardiness only. In other words, if lateness is negative, the quadratic element is not in play. With this option in mind, we can see that minimizing expected penalties is a quite general approach. Although we will not use such a convex increasing element in most of our coverage, it is important from a theoretical point of view to understand that by optimizing expected penalties we are not automatically assuming risk neutrality. Furthermore, risk-averse decision makers can penalize tardiness at a higher rate than that selected by risk-neutral decision makers facing the same cost structure. To recap, because we can address risk aversion by the utility function approach, we do not need to address it by robust scheduling models.

Early Safe Scheduling Models for Given Sequences

Returning to the newsvendor model, Britney (1976) adopts it for project activities. Although he addresses a project with *n* activities—a much more complex environment than the single machine case, which we discuss further in Chapter 18—he essentially allocates to each activity its own safety time. Thus, in effect, he uses a single operation approach; i.e., his model is even simpler than the single-machine *n*-job model. The working assumption is that if the activity does not complete on time, it causes problems downstream that can be adequately modeled by the activity's individual tardiness cost, whereas if an activity is early the activities that must follow it wait for their originally scheduled release date. Thus, each activity acquires a Parkinson processing time distribution (Appendix A). This makes possible the use of the basic newsvendor model for each activity individually. Models that extend the newsvendor model to *n* jobs without ignoring the interactions between them did not emerge until the mid-1980s. One stream of research concerns models akin to those we discussed in the chapter: suppose we have to perform *n* activities in series (in a supply chain or project context). We may refer to each of the *n* activities as *stages*, but the model is essentially equivalent to processing *n* jobs on a single machine with a makespan objective. Assuming independent processing times, this model is analyzed independently by Yano (1987 and 1987a), Sarin and Das (1987) and Das and Sarin (1988). An alternative model involves parallel inputs that feed a single project or assembly, and we refer to it as the *assembly coordination model* (ACM). The ACM was introduced independently several times, including Ronen and Trietsch (1988), Kumar (1989) and Chu et al. (1993). Trietsch and Quiroga (2004) compiled and slightly extended these results. Hopp and Spearman (1993) addressed the ACM, but with a step tardiness cost. Yano (1987b and 1987c) studied simple combinations of parallel and serial operations, namely the case of a single activity following or succeeding two parallel activities. Trietsch (2006) extends the newsvendor model to projects with n activities and any network structure (including the serial and parallel structures as special cases and without requiring stochastic independence). We discuss more general cases in Chapters 11 (stochastic flow shops) and 18 (projects). Wilhelm and Wang (1986) also address the need for safety time in assembly operations without involving sequencing decisions. Because they do not involve sequencing decisions, we may refer to such results as *fixed-sequence safe scheduling models*.

Early Safe Scheduling Models with Sequencing Decisions

To our knowledge, the first published safe scheduling model that involved sequencing decisions was due to Balut (1973), and it dealt with maximizing the number of stochastically feasible jobs with normal independent processing times; i.e., it used the chance constraint approach. Balut's model was later shown to be NP-hard so his proposed solution—a straightforward extension of Algorithm 2.1—is not guaranteed to produce the optimal solution (Kise and Ibaraki 1983). To date, the only known tractable cases of the U problem with chance constraints are those that involve stochastically ordered processing times: Akker and Hoogeveen (2008) identify several such instances that could all be solved by a straightforward extension of Algorithm 2.1; Trietsch and Baker (2008) show additional cases solvable by this algorithm. They also demonstrate that algorithm 7.1 applies in general when processing times are stochastically ordered. (We provide that proof later.) As mentioned in the Research Notes of Chapter 2, Algorithm 2.1—which is based on EDD as the initial order—is also known as the Moore-Hodgson algorithm, because Moore (1968) attributes it to Thom Hodgson. Moore's own algorithm is based on testing increasing subsets of the jobs in SPT order, sequencing them by EDD, and rejecting the last (longest) job upon any tardiness in the subset. This is essentially the structure of Algorithm 7.1 as well: the only difference is that Algorithm 7.1 must resort to the feasibility check instead of relying on EDD to test each subset. Algorithm 2.1 is more efficient (it takes $O(n \log n)$, whereas Moore's original algorithm requires $O(n^2)$), so the less efficient algorithm was essentially ignored for the last 40 years. But in the stochastic context it becomes valid again when due dates and service level targets are not agreeable.

Publications that use the economic approach to safe scheduling and involve sequencing as well as scheduling did not appear until the 1990s. Slightly earlier, Cheng (1987) studied setting due dates for independent processing times where setting a late due date incurs a convex increasing due-date cost (e.g., a linear earliness charge α_j) and an increasing function of quadratic E/T penalty (i.e., an increasing function of $E[L_j^2]$). One can argue, however, that this is not a proper E/T model but rather a model that includes earliness and squared lateness. In effect, it penalizes earliness twice whereas tardiness is only penalized once. As a result, the model actively discourages earliness and favors tardiness: the optimal service level cannot exceed 50%. For this model Cheng observed some cases are optimized by SEPT, but he did not present a generally applicable sequencing rule. We may note in passing that if we were to only penalize $E[L_j^2]$, then the optimal sequence would be in increasing variance order and the optimal due dates would match the expected completion times. Furthermore, it is straightforward to generalize this insight for the weighted case. Most other published models on E/T costs assume linear earliness and tardiness penalties or linear earliness and fixed tardiness penalties. Furthermore, with the exception of special cases (as discussed in the chapter), the state of the art in solving *all* these models is by heuristics, mainly based on adjacent pairwise interchanges or on dispatching with greedy selection of the next job. Within this group, Trietsch (1993) generalized the results of Ronen and Trietsch (1988) to the optimal scheduling of flights into and out of a hub airport. In this setting, the main "machine" is the airport (which imposes safety gaps between landings or takeoffs) and the jobs are flights. Flights may form blocks and within each block API can reduce the total cost. This model involves precedence constraints—outgoing flights cannot depart until their passengers arrive on incoming flights—and is generally more akin to project scheduling then to single-machine scheduling. Soroush and Fredendall (1994) presented three heuristics for sequencing n jobs on a single machine with independent normal processing times, given due dates and piecewise linear E/T penalties. However, they do so subject to a policy of continuous operation (i.e., no active release dates are utilized). As we saw in Chapter 5, in the deterministic version of this problem blocks are useful and indeed it can be shown that blocks may be useful in the context studied by Soroush and Fredendall, but further research is required for this case. Golenko-Ginzburg et al. (1995) use a dispatching approach to sequencing a job shop with chance constraints where the next job is selected from the available jobs greedily; i.e., such that API between the available jobs cannot lead to improvement. Soroush (1999) addressed a model similar to that of Soroush and Fredendall (1994), but where due dates are decisions (and idling is still not allowed). He found that sorting by $\sigma_j^2/(\alpha_j + \beta_j)\varphi(z_j^*)$—as we discussed in Section 7.6—is especially effective for this problem. Portougal and Trietsch (2006) showed that this particular heuristic is asymptotically optimal, and no fundamentally different sorting heuristic can be asymptotically optimal. We repeat these results below. They also identified tight bounds for the objective function. Baker and Trietsch (2009) extended these results to a case that combines the E/T cost with the flowtime cost. We repeat these results below as well. Laslo et al. (2008) use the approach of Golenko-Ginzburg et al. to select jobs in a job shop with normally distributed independent processing times with the objective of reducing $E[\alpha_j E_j + \beta_j T_j + u_j\delta(T_j)]$.

ADVANCED RESULTS

In this section we compile and slightly enhance results from Trietsch and Baker (2008), Portougal and Trietsch (2006) and Baker and Trietsch (2009). One purpose is to provide proofs missing in the chapter and discuss some additional results. Another objective is to illustrate some safe scheduling proof techniques, some of which go beyond those required for typical deterministic models. To date, several such approaches have been used to derive theoretical safe scheduling results. Here we discuss the ones that are most applicable to the basic single-machine model. These often involve asymptotic optimality of either the deterministic counterpart solution or of some function of both the mean and the variance of each job. Proofs may rely on stochastic dominance and the use of limiting assumptions (such as normality and independence) at least as a start. One way to weaken the independence assumption is to replace it by linear association. This essentially relies on Theorem 6.7 and other similar results given in Appendix A.4. In this section we also provide a new simple expression for the minimum of $d + \gamma\mathrm{E}(T)$ when processing time is lognormal (and $\gamma > 1$).

Algorithm 7.1: Proof of Optimality

We begin with the optimality of Algorithm 7.1. We rely on stochastic dominance and on Theorem 6.7. Recall that Algorithm 7.1 tests jobs in SEPT sequence by the feasibility check and rejects the longest job whenever the subset tested is not stochastically feasible. Before proving the main result, we require a lemma that helps determine which job to reject when necessary. Although we do not show that the extension of Algorithm 2.1 is optimal for agreeable due dates and service level targets, that result can be proved by using the same lemma (as shown by Trietsch and Baker, 2008).

Lemma RN7.1. Assume we are given a sequence of n jobs with stochastically-ordered and linearly-associated processing times, with fixed due dates. Suppose that we must reject exactly m out of the first k jobs (where $1 \le m \le k < n$) and that our objective is to minimize the number of stochastically tardy jobs among the last $(n - k)$ jobs in the sequence. Then it is optimal to reject the m stochastically largest jobs.

Proof.

»» Assume first that processing times are independent. Let X, Y, V, W, and S be independent random variables. If $X \ge_{st} Y$ and $V \ge_{st} W$, then $X + V \ge_{st} Y + W$ (Ross, 1996). Similarly, $S + X \ge_{st} S + Y$ and $S - X \le_{st} S - Y$. Therefore, the sum of the processing times of the m largest jobs is stochastically larger than the sum for any other m jobs, and the sum of the processing times of the remaining $(k - m)$ jobs is stochastically smallest among all possible such subsets. Accordingly, the completion time of each of the $(n - k)$ jobs that follow is stochastically minimized. Therefore, rejecting the m stochastically largest jobs maximizes the service levels of those $(n - k)$ jobs and thus minimizes the number of stochastically tardy jobs. Finally, by Theorem 6.7, the stochastic dominance relationships we obtained for the independent case still prevail for linearly associated processing times. ««

Theorem RN7.1. Algorithm 7.1 minimizes the number of stochastically-tardy jobs when processing times are stochastically-ordered and linearly associated.

Proof.

»» At stage k, the algorithm addresses the first k jobs in SEPT order, and we denote this set as $S[k]$. Let $B[k] = \{b[1], \ldots, b[m_k]\}$ be the subset of $S[k]$ accepted by the algorithm, where $m_k = |B[k]|$. The theorem is true if and only if $B[k]$ is optimal for all k. We proceed by induction on k; i.e., we first establish that the theorem is correct for $S[1]$; then, for $k \geq 2$, we assume it is correct for $S[k-1]$ and prove that it must be correct for $S[k]$.

For $k = 1$, the algorithm accepts Job 1 if and only if it is feasible, so $B[1]$ must be optimal for $S[1]$. For $k \geq 2$, the algorithm performs a feasibility check for $\{b[1], \ldots, b[m_{(k-1)}], k\}$. If this set is feasible, then $B[k] = \{b[1], \ldots, b[m_{(k-1)}], k\}$ and because $B[k-1]$ is optimal by assumption and $|B[k]| > |B[k-1]|$, $B[k]$ must be optimal. So assume the set $\{b[1], \ldots, b[m_{(k-1)}], k\}$ is not feasible and trace the feasibility check on $\{b[1], \ldots, b[m_{(k-1)}], k\}$. If any jobs are feasible in the last positions, m_k, $m_{(k-1)}$, …, we can ignore them because they cannot cause infeasibility in an earlier job. By the assumed infeasibility we know that there exists a set of $j \geq 1$ jobs, which includes job k, none of which is feasible in position j. By feasibility of $B[k-1]$ we know that it is sufficient to remove one of these jobs. Furthermore, jobs $(k+1)$, $(k+2)$ …, n that will subsequently be examined by the algorithm are stochastically larger than job k (and thus larger than each of the $j-1$ remaining jobs), so none of them will be feasible in one of the first $j-1$ positions without displacing at least one of the existing jobs from B as well. We now invoke Lemma RN7.1 to select job k, which is indeed the job that the algorithm will reject. Hence, as we start with an optimal $B[k-1]$ and take the optimal next step, the resulting $B[k]$ must also be optimal. ««

Using Algorithm 7.1 as a Heuristic

Notice that stochastic ordering is required because we rely on Lemma RN7.1. Therefore, if we use Algorithm 7.1 when processing times are not stochastically ordered, the algorithm becomes a heuristic. It is likely, however, that it is a more effective heuristic than the direct extension of Algorithm 2.1 unless due dates and service levels are agreeable (in which case the two algorithms always yield the same sequence but Algorithm 7.1 is less efficient). Whereas Moore's original algorithm requires $O(n^2)$ calculations, Algorithm 7.1 cannot rely on EDD within each subset so it requires $O(n^3)$ (the feasibility procedure is $O(n^2)$ and it has to be invoked $O(n)$ times). Yet when due dates and service level targets are agreeable, we can find a solution in $O(n \log n)$ time, exactly as in Algorithm 2.1. At the time of this writing, testing the effectiveness of this heuristic is an open research question. To resolve this question, it would be important to select examples that do not tend to be approximately stochastically ordered. For example, selecting processing times with similar coefficients of variation is likely to yield deceptively good results. By the same token, in environments with similar coefficients of variation the heuristic is likely to be effective.

The Stochastic E/T Problem

Consider the objective of minimizing the expected E/T cost with independent, normally-distributed processing times, as given by (7.4), which we now repeat,

$$\mathrm{E}[f(S)] = \sum_{j=1}^{n} [(\alpha_j + \beta_j) s_j \varphi(z_j^*)] \qquad (7.4)$$

Our next task is to prove the asymptotic optimality of sequencing by sorting by nondecreasing ratio of $\sigma_j^2/(\alpha_j + \beta_j)\varphi(z_j*)$, with ties broken in favor of the smallest σ_j. While doing that, we also develop a similar result for the [suboptimal] approach of not using safety time. In this case, instead of (7.4) we obtain

$$\mathrm{E}[f(S)] = \sum_{j=1}^{n} [(\alpha_j + \beta_j)s_j\varphi(0)] \qquad \text{(RN7.2)}$$

Here, instead of sorting by $\sigma_j^2/(\alpha_j + \beta_j)\varphi(z_j*)$ we should sort by $\sigma_j^2/(\alpha_j + \beta_j)\varphi(0)$. Define $b_j = (\alpha_j + \beta_j)\varphi(z_j*)$ or $(\alpha_j + \beta_j)\varphi(0)$, depending on which objective we address; the objective is then given by $\Sigma b_j s_j$ in both cases. We show that sorting by σ_j^2/b_j is asymptotically optimal for either definition of b_j. As noted in the chapter, optimizing for the two versions of b_j may yield different sequences. However, if all optimal service levels are equal (that is, when $\alpha_j/\beta_j = \alpha/\beta$ for all j), the two objectives are optimized by the same sequence because $z_j* = z*$ for all j, and the expressions in (7.4) and (RN7.2) are then proportional to each other. In the following development, until further notice, we require strictly positive variances. However, it is clear that to minimize (7.4) or (RN7.2), activities with zero variance should always be scheduled first, which would also be the sequence in which our sorting rule would place them, so practically this is not restrictive.

Within our context, b_j is effectively a *weight*, because (7.4) and (RN7.2) can be seen as weighted sums of s_j elements. This interpretation might lead us to adapt the SWEPT approach to this case by using $\sigma_j/(\alpha_j + \beta_j)\varphi(z_j*)$ or $\sigma_j/(\alpha_j + \beta_j)\varphi(0)$ (i.e., σ_j/b_j) to sort the jobs. Indeed, Soroush and Fredendall (1994) proposed this sorting rule (but without treating due dates as decisions) and later the $\sigma_j/(\alpha_j + \beta_j)\varphi(z_j*)$ version was one of two sorting rules tested by Soroush (1999), but it was often found inferior. Much of our coverage here focuses on showing why sorting by the other rule, σ_j^2/b_j, is better. In a nutshell, the advantage follows because as we add variance elements to the sequence, the marginal contribution to s_j becomes approximately proportional not to σ_j but to σ_j^2. Because it is the marginal contribution that counts, the advantage of sorting by σ_j^2/b_j increases as we add jobs. That is, the advantage is associated with the asymptotic optimality of this rule. Thus, what we need to show is not only that this rule is asymptotically optimal, but also that the other rule is *not* asymptotically optimal.

To recap from the chapter, let $f(S*)$ denote the objective function value with the optimal sequence, $S*$ (for any given objective), and let $f(S^H)$ be the value associated with a heuristic. We say that the heuristic is asymptotically optimal if, in the limit, as $n \to \infty$ $[f(S^H) - f(S*)]/f(S*) \to 0$. One of the fundamental techniques in analyzing safe scheduling with objectives like (7.4) under stochastic independence is to analyze in two steps: one involving variances and the other concerning standard deviations. As long as we assume stochastic independence, the first step of such analysis is often tractable because variances are additive. In our present context we use this approach as follows. In step 1, we look at $\Sigma b_j s_j^2$ or parts thereof. In step *2*, we consider our true objective function, $\Sigma b_j s_j$. At stage j, we can draw the contribution of job $[j]$ to the objective function as a rectangle with a basis of $b_{[j]}$ starting at $\Sigma_{k=1,\ldots,j-1} b_{[k]}$, and with a height of s_j^2 at step 1 or s_j at step 2. That is, the x-axis is used for $b_{[j]}$, cumulatively, and the y-axis for the variance or the standard deviation of the completion time, as the case may be. We refer to the domain of the variables of step 1 as the b-σ^2 *domain*, and to those of step 2 as the b-σ *domain*. Figure RN7.1 depicts the two domains for $b_{[1]}$=1.5, $b_{[2]}$=0.5, $b_{[3]}$=0.75 with σ^2=2.25, 1, and 2 respectively. The true (step 2) objective function is the area below the (lower) steps depicting the *b-σ* domain, and the step 1 objective function is the area below the higher steps. Lemma RN7.2 is instrumental for using the results of step 1 to draw conclusions for step 2.

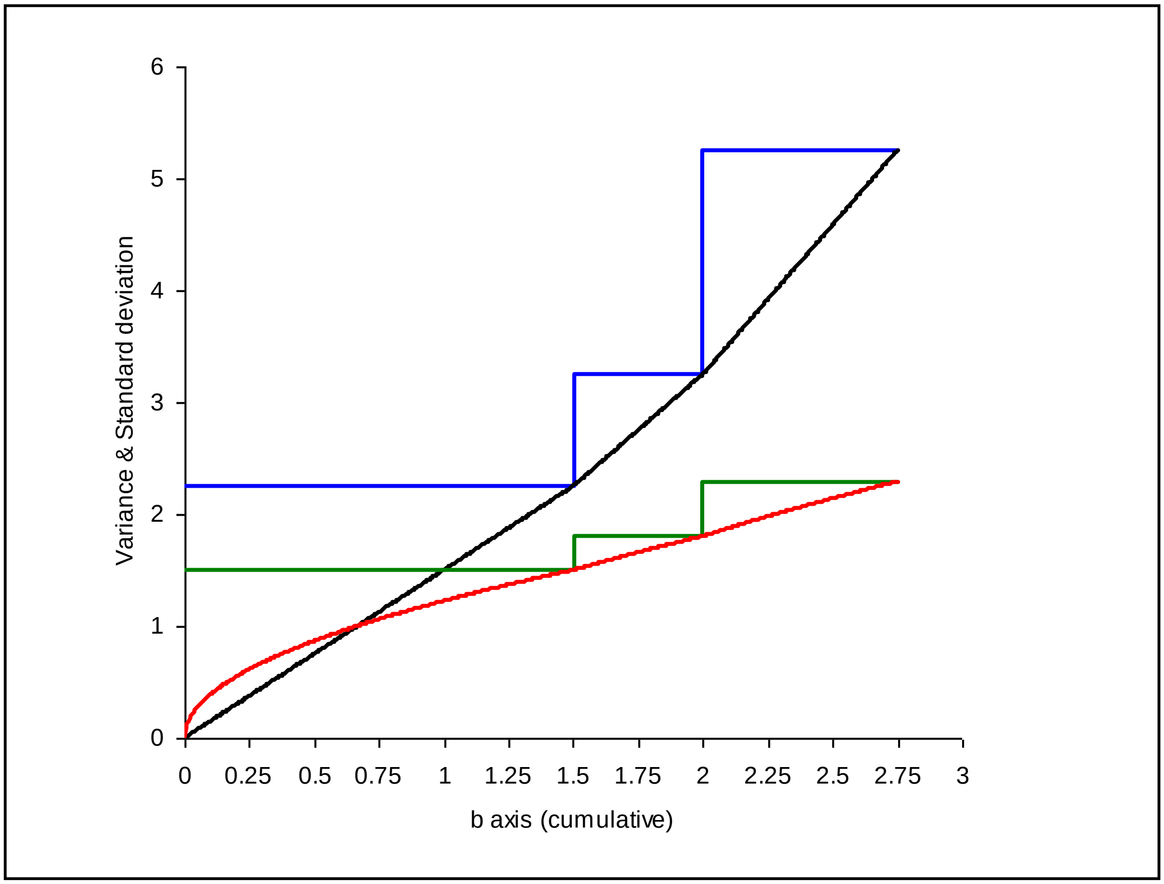

Figure RN7.1. *The Objective Function in the* b-σ *and* b-σ^2 *Domains*

Lemma RN7.2: Let $a, b, q, r > 0$ satisfy $a(q^2+r^2) \geq bq^2$, then for any $s \geq 0$

$$a\left(\sqrt{s^2+2q^2+r^2}-\sqrt{s^2+q^2}\right) > b\left(\sqrt{s^2+2q^2+r^2}-\sqrt{s^2+q^2+r^2}\right)$$

Proof.

»» Denote

$$\Delta_a = \sqrt{s^2+2q^2+r^2}-\sqrt{s^2+q^2}\;;\;\Delta_b = \sqrt{s^2+2q^2+r^2}-\sqrt{s^2+q^2+r^2}$$

Clearly, $\Delta_a > \Delta_b$, $a(q^2+r^2) \geq bq^2$ if and only if $a(q^2+r^2)/bq^2 \geq 1$, so if we can show $\Delta_a/\Delta_b > (q^2+r^2)/q^2$ the lemma will be proved (because $(q^2+r^2)/q^2 \geq b/a$). Multiplying Δ_a by

$$\sqrt{s^2+2q^2+r^2}+\sqrt{s^2+q^2}$$

leads to the following [circular] expression

$$\Delta_a = \frac{q^2+r^2}{\sqrt{s^2+2q^2+r^2}+\sqrt{s^2+q^2}} = \frac{q^2+r^2}{2\sqrt{s^2+2q^2+r^2}-\Delta_a}$$

and similarly

$$\Delta_b = \frac{q^2}{2\sqrt{s^2+2q^2+r^2}-\Delta_b}$$

Therefore,

$$\frac{\Delta_a}{\Delta_b} = \left(\frac{q^2+r^2}{q^2}\right)\frac{2\sqrt{s^2+2q^2+r^2}-\Delta_b}{2\sqrt{s^2+2q^2+r^2}-\Delta_a}$$

Because $\Delta_b < \Delta_a$,

$$\frac{2\sqrt{s^2+2q^2+r^2}-\Delta_b}{2\sqrt{s^2+2q^2+r^2}-\Delta_a} > 1$$

and the lemma follows. ««

To demonstrate the relevance of the lemma to our problem, we now show that it leads to a sufficient condition for the sorting rule to resolve correctly the order of adjacent jobs. It is intuitively clear (and we also show formally later) that if $b_i \geq b_j$ and $\sigma_i \leq \sigma_j$, with at least one inequality strict, and jobs i and j are adjacent, then job i should precede job j. In such case we say the weights and standard deviations (variances) are agreeable. But when weights and variances are not agreeable, it is less clear which should come first. The lemma shows that if the sorting rule places a job with a larger variance after a job with a smaller variance, then this order is correct (as long as the jobs are adjacent). Because the condition is not necessary, however, the lemma does not guarantee the correct order when the sorting rule places the job with the larger variance first. To cast the lemma as that sufficient condition, we first rewrite it in the following equivalent form:

$$\text{If } q^2/a \le (q^2+r^2)/b \text{ then } a\Delta_a > b\Delta_b$$

Next, interpret a as $\min\{b_i, b_j\}$, b as $\max\{b_i, b_j\}$ (so $a \le b$), q^2 as $\min\{\sigma_i^2, \sigma_j^2\}$, q^2+r^2 as $\max\{\sigma_i^2, \sigma_j^2\}$, and s as the standard deviation of the completion time of the preceding jobs. Temporarily, combine the two jobs to one job with a weight given by the sum of the weights (i.e., $a + b$) and a variance given by the sum of the variances (i.e., $2q^2 + r^2$). The contribution of the combined job to the objective function (7.4) or (RN7.2) is

$$(a+b)\sqrt{s^2+2q^2+r^2}$$

If we sequence the job with the lower variance first, the true contribution of the two jobs is obtained by subtracting $a\Delta_a$ from this expression, whereas if we place the job with the higher variance first we should subtract $b\Delta_b$. The rewritten lemma states that the former gain is larger if $q^2/a \le (q^2+r^2)/b$. But if $q^2/a \le (q^2+r^2)/b$ then the sequence suggested by our rule is indeed to place the low variance job first (and in case $q^2/a = (q^2+r^2)/b$, our tie-breaker still places the job associated with q^2/a first). That demonstrates the sufficient condition. To show that the heuristic may place a large job too early, consider a 2-job example with standard deviations of 1 and 2 and weights 1 and 5. The heuristic places the second job first because 4/5 < 1/1, but the optimal sequence is the reverse, because $s = 0$. (The heuristic sequence would be correct, however, for $s > 0.716$.)

If we define Δ_j as the increment of the standard deviation when we add job j to a set of preceding jobs with standard deviation s, the proof also shows a connection between Δ_j, which belongs to the b-σ domain, and σ_j^2, which belongs to the b-σ^2 domain. For large enough s, we obtain the following approximation for Δ_j (and a similar expression applies for Δ_i).

$$\Delta_j = \frac{\sigma_j^2}{2\sqrt{s^2+\sigma_j^2}-\Delta_j} \approx \frac{\sigma_j^2}{2\sqrt{s^2+\sigma_j^2}} \approx \frac{\sigma_j^2}{2s}$$

This approximation explains the advantage of using σ^2 for sorting instead of using σ.

There are two basic cases where the heuristic produces optimal solutions. One is when all σ_j are equal to each other and the sequence calls for non-increasing $b_{[j]}$. The other is when all b_j are equal to each other, and the sequence calls for non-decreasing $\sigma_{[j]}$. The proof of the latter result is incorporated within the proof of a more general sufficient condition (Theorem RN7.3), which generalizes these two basic cases. We prove the former now.

Lemma RN7.3. When $\sigma_j^2 = \sigma^2$ for all j, the optimal sequence is by non-increasing b_j (non-decreasing σ^2/b_j).

Proof.

»» Without loss of generality, let $\sigma^2 = 1$ ($\forall\ j$), so the variance of the completion time of job [j] is j. The contribution of job [j] to the objective function is proportional to $b_{[j]}\ j^{0.5}$. By allocating the larger (smaller) b_j to the smaller (larger) standard deviation we minimize the sum. ««

Next, consider a relaxation of the problem by allowing preemption. Specifically suppose we can partition each job to fractions and sequence all fractions of all jobs in any order, letting each part have its own due date. Any such partition must preserve the total weight of the job, b_j, and the total variance, σ_j^2. Furthermore, the parts must be statistically independent of each other, or the partitioned problem will not be an instance of the original problem. To meet these conditions, a fraction f of job j is allocated a weight of fb_j (that is, E/T cost rates of $f\alpha_j$ and $f\beta_j$) and a variance of $f\sigma_j^2$. Thus the total variance of the parts job j is σ_j^2 and the total weight is b_j.

Lemma RN7.4. The relaxed problem is solved optimally without utilizing any preemption by the sequence $\sigma_{[1]}^2/b_{[1]} \leq \sigma_{[2]}^2/b_{[2]} \leq \ldots \leq \sigma_{[n]}^2/b_{[n]}$.

Proof.

»» To any required degree of accuracy, it is possible to partition all jobs to fractions with the same variance across the board. After the partition, by Lemma RN7.3, we should place the fractions with the smallest σ_j^2/b_j first. Since every fraction of job j has the same ratio σ_j^2/b_j as the job itself, this can be done without preemption, and the sequence is established. ««

On the one hand, the objective function value of the relaxed problem, when using the job fractions instead of the original jobs for the calculation, is strictly lower than the true objective function. Therefore, it can serve as a lower bound. To see this, consider that only the last fraction of each job attracts the correct expected early-tardy cost, whereas the first fraction of a job incurs a lower cost (because it has a lower standard deviation). On the other hand, if we calculate the objective function using the original jobs in the optimal relaxed sequence, we obtain a feasible value (which is not guaranteed to be optimal) for the un-relaxed problem. Therefore, it provides an upper bound for the optimal solution. To prove that the heuristic is asymptotically optimal we show that as the number of jobs increases the difference between the lower and upper bounds becomes negligible in comparison to the lower bound. But we also want to show that other heuristics are *not* asymptotically optimal. For this purpose, we first derive a more general convergence property.

Consider the relaxed problem but without actually allowing preemption. That is, each job is partitioned to many subjobs with the same variance everywhere, but the parts of each job—although they acquire individual due dates—are kept together in the schedule as strings. (A *string* is a set of jobs that must be adjacent to each other in the sequence.) In general, we cannot tell in advance how finely we must partition each job to achieve equal variance in all fractions of all jobs. So, we must assume the worst case and use infinitesimal fractions. Therefore, for any relaxed solution the b-σ^2 depiction of the objective function is not a set of adjacent rectangles with increasing heights, as the true objective function. Instead, it is a set of adjacent trapezoids, with their upper boundaries constituting a continuous piece-wise linear function (see Figure RN7.1). We refer to the latter as the *relaxed function*. In contrast, the true function is a step function. To draw the relaxed function, start at the origin and connect it to the point ($b_{[1]}$, $\sigma_{[1]}{}^2$) by a straight segment. At stage j (j = 2, ..., n) connect the points ($b_{[j-1]}$, $s_{[j-1]}{}^2$) and ($b_{[j]}$, $s_{[j]}{}^2$) by a straight segment. The result is a piecewise linear function (and it is convex for step 1 of the optimal relaxed solution because the slope of the segment drawn in stage j, $\sigma_{[j]}{}^2/b_{[j]}$, is monotone non-decreasing). The triangles captured between the step function and the relaxed function represent the difference between the lower bound and the associated feasible solution (in the b-σ^2 domain). As Figure RN7.1 demonstrates, the transformation of the relaxed function to the b-σ domain is neither piecewise linear nor convex, but the curvature decreases with σ. In that domain, the objective is measured by the area below the graph, so the area captured between the functions is the difference between the relaxed and the true objectives.)

Lemma RN7.5. Let all job parameters be sampled independently from a multivariate distribution with a finite covariance matrix and such that $0 < \delta < \sigma_j < \infty$ for all j. Denote the objective function of the relaxed problem with n jobs (partitioned to small parts) in some given sequence by f_n and let F_n be the true objective function. Then, as $n \to \infty$, $(F_n - f_n)/F_n \to 0$ (i.e., $f_n/F_n \to 1$).

Proof.

»» Let $s_B{}^2$ denote the variance of the set of jobs preceding job j (and if job j is the first in the sequence then $s_B{}^2 = 0$). Let $s_j = [s_B{}^2+\sigma_j{}^2]^{0.5}$ be the standard deviation of the same set augmented by job j. Finally, let $\Delta_j = s_j - s_B$. As we developed within the proof of Lemma RN7.2, $\Delta_j = \sigma_j{}^2/(2s_B + \Delta_j)$ (there, we have used $2s_j - \Delta_j$ in the denominator, but $2s_j - \Delta_j = 2s_B + \Delta_j$). The true contribution of job j to the objective function is $b_j s_j$. The difference between the true and the relaxed contribution of job j to the objective function is given by

$$\frac{b_j \Delta_j}{3}\left(1+\frac{s_B}{2s_B+\Delta_i}\right)$$

To see this begin with

$$s_j b_j - \int_0^{b_j} \sqrt{s_B^2 + x\sigma_j^2/b_j}\,dx = s_j b_j - \frac{2b_j}{\sigma_j^2}[s_j^3 - s_B^3]$$

and apply some algebra starting with $\sigma_j{}^2 = s_j{}^2 - s_B{}^2$. Except for the first job, this is bounded from above by

$$\frac{b_j \Delta_j}{3}\left(1+\frac{s_B}{2s_B+\Delta_j}\right) < \frac{b_j \sigma_j^2}{2s_B}$$

Dividing the upper bound by the contribution of job j, $b_j s_j$, we obtain

$$\frac{\sigma_j^2}{2s_j s_B} < \frac{\sigma_j^2}{2s_B^2}$$

For the first job, which is not covered by this expression (unless we interpret division by zero as $+\infty$), the exact ratio is 1/3. By assumption job distribution parameters are drawn independently from some multivariate distribution with a finite covariance matrix. It follows that the variance series is distributed independently and the variance of the variance is finite. Therefore, for large n, the expected value of this bound approaches $1/2n$ almost surely. Let ε be any [small] positive value, then, because $\sigma_j > \delta$ there exists some m such that for any $j \geq m$, $E(\sigma_j^2/2s_B^2) < \varepsilon$ almost surely. An upper bound on F_n is given by adding $\Sigma_{j=1,\ldots,m} b_{[j]}\sigma_{[j]}^2/2s_{[j-1]}$ (where $s_{[j-1]}$ is the standard deviation of the completion time of the first $j-1$ jobs) to f_m. Both F_m and f_m are finite, and we can write,

$$\frac{F_n - f_n}{F_n} < \frac{bs_m(n-m)\varepsilon + F_m - f_m}{bs_m(n-m) + F_m}$$

where s_m is the standard deviation of the completion time of the first m jobs and b is the average of all b_j. By the arguments above, the numerator of the right hand side is an upper bound on the numerator of the left hand side, and the denominator of the right hand side is a lower bound on that of the left hand side. Therefore the inequality is correct. Nonetheless, the limit of the right hand side as $n \to \infty$ is ε, and ε is as small as we wish. ««

Theorem RN7.2. A sorting heuristic is almost surely asymptotically optimal (i.e., it yields an F_n such that as $n \to \infty$ $(F_n - F_n^*)/F_n^* \to 0$ with probability one) for minimizing (7.4) or (RN7.2) *if and only if* it yields $\sigma_{[1]}^2/b_{[1]} \leq \sigma_{[2]}^2/b_{[2]} \leq \ldots \leq \sigma_{[n]}^2/b_{[n]}$.

Proof.

»» By Lemma RN7.4, for this sequence, f_n is a lower bound on the optimal solution. Therefore, the "if" part is assured by Lemma RN7.5. The "only if" part is by contradiction, as follows. Assume an asymptotically optimal heuristic sorting exists that does not satisfy the condition, say Heuristic 1, and let Heuristic 0 be any heuristic that satisfies the condition. Then there must exist at least two possible jobs, say jobs 1 and 2, such that Heuristic 0 yields the sequence (1, 2) but Heuristic 1 sorts them in the sequence (2, 1). If no such two jobs exist, then Heuristic 1 must always yield the same sequence as Heuristic 0, and therefore it must satisfy the condition that Heuristic 0 satisfies. Denote σ_1^2/b_1 by C_1 and similarly let $C_2 = \sigma_2^2/b_2$, where $C_2 > C_1$. Now consider a set of n jobs where job parameters are generated by tossing a coin and selecting a copy of job 1 upon head, and of job 2 upon tail. For mathematical convenience we set the probability of head to $b_2/(b_1 + b_2)$ [Otherwise, the proof will become more tedious, but any probability except 0 or 1 will do]. As a result, the expected total weight of jobs of type 1 equals that of type 2. As we add jobs to the set, Heuristic 0 sequences all the job 1 copies first, followed by all the job 2 copies. Heuristic 1 will do the opposite. We may measure the size of the set of jobs by the total weight, and let $2b$ be the size in question. Therefore, we expect a total weight of b to be composed of type 1 jobs, and the same weight of type 2 jobs. By Lemma RN7.5, for large enough job sets, it is enough to compare the lower bound values associated with the relaxed functions of the two sequences. In the b-σ^2 domain the relaxed function of Heuristic 0 starts at the origin and reaches the argument b at a constant slope of C_1. It then continues at a slope of C_2 until it reaches the total weight $2b$. The relaxed function of Heuristic 1 starts with the higher slope of C_2 until b, and then continues with slope C_1 until it meets the other function at the point $(b, b(C_1 + C_2))$. The relaxed objective function value of Heuristic 0 is given by,

$$\int_0^b \sqrt{C_1 x}\,dx + \int_b^{2b} \sqrt{C_1 b + C_2(x-b)}\,dx$$

That of Heuristic 1 is given by,

$$\int_0^b \sqrt{C_2 x}\,dx + \int_b^{2b} \sqrt{C_2 b + C_1(x-b)}\,dx < \int_0^{2b} \sqrt{C_2 x}\,dx$$

The difference between the two is,

$$\int_0^b \left(\sqrt{C_2} - \sqrt{C_1}\right)\sqrt{x}\,dx + \int_b^{2b} \left(\sqrt{C_2 b + C_1(x-b)} - \sqrt{C_1 b + C_2(x-b)}\right)dx > \int_0^b \left(\sqrt{C_2} - \sqrt{C_1}\right)\sqrt{x}\,dx$$

It is possible to calculate the exact ratio between the difference and the optimal value (of the relaxed solution), and the result is not a function of b. But for simplicity we note instead that the ratio between the lower bound of the difference and the upper bound of the Heuristic 1 result must be a lower bound on the exact result. This ratio,

$$\frac{\int_0^b \left(\sqrt{C_2} - \sqrt{C_1}\right)\sqrt{x}\,dx}{\int_0^{2b} \sqrt{C_2 x}\,dx} = \sqrt{\frac{1}{8}}\left(1 - \sqrt{\frac{C_1}{C_2}}\right) > 0 \ \forall\, b$$

is positive and constant for any b, thus contradicting the assumption that Heuristic 1 is asymptotically optimal. ««

Example: Let $b_1 = 2$, $b_2 = 3$, $\sigma_1 = 3$, $\sigma_2 = 4$. For this example, the two elementary dispatching heuristics—by weighted standard deviation or by weighted variance—yield different sequences (because 3/2 > 4/3, sequencing job 2 first, but 9/2 < 16/3, sequencing job 1 first). If we now generate many copies of each job, with a proportion of 60% type 1 and 40% type 2 (to make the total weights equal), our bound yields 0.0288. An exact calculation yields 0.050965. Comparing the true objective function values exactly for $n = 5$, 10, 15, and 100 yields 0.0546, 0.0544, 0.0539, and 0.0517, which confirms the asymptotic applicability of the calculation based on the relaxed function to the true one.

Asymptotic Optimality for General Distributions

Suppose now that jobs are not necessarily distributed normally, but by the regularity conditions we imposed no subset of jobs dominates any other subset of the same size and all variances are finite. Therefore, for a large enough (but finite) m, the completion time of job $m + k$, C_{m+k}, approaches the normal distribution as accurately as we may wish. In the proof of asymptotic optimality above, assume now that we make m large enough to ensure this as well as the previous requirement. So the completion times of jobs m through n are all approximately normal for any $n > m$ even if the distributions of the jobs are not normal. Therefore, our proof will still hold and our heuristic is asymptotically optimal even without the normality assumption.

In conclusion, sorting by σ_i^2/b_i, or by a one-to-one function of it, must be part of any asymptotically optimal sorting heuristic when processing times are independent, with any distributions. In the next section we present a sufficient condition for normally distributed processing times that our tie-breaker identifies when it is satisfied, so one can say that our heuristic cannot be dominated by any other sorting heuristic. (Of course, we can devise polynomial complexity heuristics that may achieve better results, e.g., by employing API to obtain a local optimum—which our heuristic does not guarantee unless the sufficient condition holds. Nonetheless, such heuristics are not elementary—they depend on the relationships between jobs in their current positions—and using our heuristic does not exclude their subsequent use. In fact we recommend API among the first few jobs wherever the sufficient condition is not satisfied.)

A Sufficient Condition

Theorem RN7.3. For independent normal processing times, a sequence that satisfies $\sigma_{[1]}^2/b_{[1]} \le \sigma_{[2]}^2/b_{[2]} \le \ldots \le \sigma_{[n]}^2/b_{[n]}$ and $\sigma_{[1]}^2 \le \sigma_{[2]}^2 \le \ldots \le \sigma_{[n]}^2$ is optimal.

Proof.

»» By contradiction, suppose such a sequence, say S, exists but is not optimal. Then there is an optimal solution, S^*, with at least one pair of adjacent jobs, i and j, such that either $\sigma_i^2 > \sigma_j^2$ or $\sigma_i^2/b_i > \sigma_j^2/b_j$ or both. The existence of S rules out the possibility that either $\sigma_i^2/b_i < \sigma_j^2/b_j$ and $\sigma_i^2 > \sigma_j^2$ or $\sigma_i^2 < \sigma_j^2$ and $\sigma_i^2/b_i > \sigma_j^2/b_j$. Therefore, it is enough to consider $\sigma_i^2 \ge \sigma_j^2$ and $\sigma_i^2/b_i \ge \sigma_j^2/b_j$ with at least one strict inequality. If $\sigma_i^2 = \sigma_j^2$, Lemma RN7.3

applies, so only the case with $\sigma_i^2 > \sigma_j^2$ and $\sigma_i^2/b_i \geq \sigma_j^2/b_j$ remains. But, in our discussion directly following Lemma RN7.2, we already showed that if the sorting heuristic places the job with the smaller variance first then this is strictly advantageous, thus contradicting the assumption that S^* is optimal when such an S exists. ««

Formally, Theorem RN7.3 includes as special cases both Lemma RN7.3 and the use of non-decreasing $\sigma_{[j]}^2$ when b_j is constant. Therefore, henceforth we will refer to it as *the sufficient condition.* If the sufficient condition can be satisfied, then, due to the tie-breaking rule, our sequence is guaranteed to do so. Portougal and Trietsch (2006) provide additional bounds and precedence relations that can support a branch and bound solution, but because the heuristic is very effective, finding the optimal solution in this case is of secondary importance.* If we wish to employ API to improve the sorting heuristic solution, however, then it is sufficient to consider pairs where the heuristic places a high-variance job ahead of a low-variance one. This is especially worthwhile if this job is scheduled early (because as s grows large the sorting heuristic is progressively likely to provide the correct order).

Trading Off Due-Date Tightness and Tardiness: The Weighted Case

We now shift our attention to more general cases, which include accounting for flowtime as well as E/T costs. We already encountered some results of this genre in the chapter: Theorem 7.4 and Corollary 7.1 demonstrated that stochastic ordering is sufficient for the optimality of SEPT both for minimizing D subject to stochastic feasibility with a common service level or minimizing $D + \gamma E(T)$. The requirement for stochastic ordering in this case (and in context of Algorithm 7.1) is perhaps less onerous than it might seem. First, as a rule, we recommend the use of the lognormal distribution, and in that case we obtain stochastic dominance if the coefficient of variation is constant. As we discuss in our Research Notes for Chapter 18, however, that case is likely in practice. As a more familiar instance, take the normal distribution. The cdfs of any pair of normal random variables with different variances always intersect each other once. Thus we might think that Theorems 7.4 and Corollary 7.1 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, again, the case of a constant coefficient of variation. Furthermore, for the purpose of Corollary 7.1 (and Algorithm 7.1), 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 medians; i.e., in the case of the normal distribution, if $E(X) \leq E(Y)$ 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 Theorem 7.4, as the next theorem establishes.

---

* If B&B were successful here—as we strongly suspect it should be—it would constitute an example of a stochastic problem that can be solved without resorting to stored samples.

Theorem RN7.4: Suppose the objective is to minimize $D + \gamma\mathrm{E}(T)$, with independent normal processing time distributions. For any pair of jobs i and j such that job i dominates job j (i.e., $\mu_i \le \mu_j$ and $\sigma_i \le \sigma_j$ 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. ««

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

$$\sum_{j=1}^{n} \alpha_j[d_j + \gamma\mathrm{E}(T_j)] \qquad \text{(RN7.3)}$$

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 (RN7.3) should still satisfy Theorem 7.3 (i.e., satisfy the critical fractile result). 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:

$$d_j = C_j + \max\{0, d_j - C_j\} - \max\{0, C_j - d_j\} \qquad \text{(RN7.4)}$$

Here, $\max\{0, d_j - C_j\}$ 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 (RN7.4), 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 (RN7.3) 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.

$$\sum_{j=1}^{n} \alpha_j\mathrm{E}(C_j) + \sum_{j=1}^{n} [\alpha_j\mathrm{E}(E_j) + \beta_j\mathrm{E}(T_j)] \qquad \text{(RN7.5)}$$

In this 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 (i.e., letting $\beta_j = \alpha_j(\gamma_j - 1)$), and (RN7.5) would be obtained without a proportionality restriction. Trietsch (1993) presents a model where such generalized terms arise. In that case the earliness and tardiness rates reflect the time value of passengers, aircraft and crews on different flights. In Chapter 18 (pp. 429-433) we present a similar example where passenger time value dictates earliness and tardiness costs that are not necessarily proportional. Here, however, we continue our analysis subject to the proportionality assumption.

The first sum in (RN7.5) 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 (RN7.5) 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 above. Our model would thus generalize the stochastic E/T problem if we allowed distinct γ_j values, in which case (RN7.5) would remain unchanged but feature independent α_j and β_j. In our special case, where α_j and β_j are proportional, the sorting rule we discussed for the E/T problem calls for sequencing the jobs according to nondecreasing order of σ_j^2/α_j. Thus, we are faced in (RN7.5) with an objective consisting of one term that drives sequencing toward shortest weighted mean processing times and another that drives sequencing toward smallest 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 actually requires an explicit no-idling restriction. However, for (RN7.5), and thus also for (RN7.3), the no-idling assumption is unnecessary. We next prove this property for the general case, without requiring α_j and β_j to be proportional for all jobs and without requiring normality or stochastic independence.

Theorem RN7.5. 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 $C_{[n]}$, $d_{[n]}*$ (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. ««

Numerical experience we reported in Baker and Trietsch (2009) for normally distributed processing times (or when there are enough jobs to justify using the central limit theorem at least for most jobs) suggests that good sequencing heuristics for this problem should take into account the variance as well as the mean of each job. The simplest heuristic for this objective sequences the jobs according to nondecreasing values of the ratio $(\mu_j + \sigma_j)/\alpha_j$. This heuristic, the *weighted mean*-standard *deviation* (WMSD) rule is very simple and even though our analysis above suggests that we should prefer using some function of σ_j^2 rather than σ_j, this simpler heuristic is still useful. Nonetheless, a heuristic that gives priority to the job with the smallest value of $[\mu_j + \gamma\varphi(z)\Delta_j]/\alpha_j$ is even better. It is based on the observation that for any pair of adjacent jobs this sequence is optimal when all other jobs are in the same positions. However, that is a dynamic priority rule because, as we showed above, if s_B is the standard deviation of the completion time of all jobs processed ahead of job j then $\Delta_j \approx \sigma_j/2s_B$. But s_B increases dynamically as we schedule additional jobs. Thus, a job with high σ_j may be discouraged in the early positions but become attractive later. We refer to this procedure as the *weighted pair interchange* (WPI) Rule. We conducted a set of computational experiments with test problems containing weights and the results are summarized in Table RN7.1, reflecting a set of 150 test problems. In the table, R reflects random results, SWEPT ignores variance, and the last column shows the percentage of cases where the WPI rule achieves optimal results.

Table RN7.1

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

In the table, we see that the randomly-generated sequence fares poorly but the other heuristic rules all perform well. The static priority WMSD rule improves on SWEPT by roughly an order of magnitude. The dynamic priority WPI rule improves by nearly another order of magnitude and produces optimal solutions in most test problems. Furthermore, two of these rules, SWEPT and WPI are asymptotically optimal: the former because as $n \to \infty$ variance becomes less important and the latter because except for the first few jobs it is likely to coincide with SWEPT, and the difference induced by the first jobs can be shown to become negligible as n grows large. In this connection, WMSD is *not* asymptotically optimal because it does not give variance a decreasing weight.

We adapt the technique of allowing preemption for our derivations. Here, 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.) Above, we showed that using such strings yields a lower bound on the expected E/T penalty. We also showed that 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 (RN7.5), it can be shown that using strings in this manner leads to the correct value minus a constant.

Theorem RN7.6. Suppose the objective is to minimize $\sum_{j=1}^{n} \alpha_j \mathrm{E}(C_j) + \sum_{j=1}^{n} [\alpha_j \mathrm{E}(E_j) + \beta_j \mathrm{E}(T_j)]$ 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 $n \to \infty$, $(\eta/\lambda)(1 + \delta)\varphi(0)/n^{0.5} \to 0$. ««

Theorem RN7.6 implies that SEPT is asymptotically optimal for the objective of minimizing $D + \gamma \mathrm{E}(T)$. In the weighted case, where we just showed that 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 RN7.6 implies that in (RN7.5), 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 (RN7.5). This property is not obvious because, relative to SWEPT, 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 for *all* subsequent jobs. 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 again 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. 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 $k + 1, k + 2, \ldots, n$. 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 RN7.7. Suppose the objective is to minimize $\sum_{j=1}^{n} \alpha_j \mathrm{E}(C_j) + \sum_{j=1}^{n} [\alpha_j(\mathrm{E}(E_j) + (\gamma - 1)\mathrm{E}(T_j))]$ 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 $\mathrm{E}(C_B^{PI})$ and $\mathrm{E}(C_B^*)$ 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_ε + 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. ««

WPI with Linearly Associated Processing Times

To our knowledge, there are no published results about the effectiveness of various heuristics for the case of linearly associated processing times (let alone general processing time distributions without stochastic independence). By studying Section A.3 in Appendix A, and Theorem A.6 in particular, it can be shown that in the limit the individual variance of the jobs matters less than the mean for sequencing decisions, and yet the expected E/T penalty does not become negligible. The reason is that jobs are subject to a completion time coefficient of variation (cv) that is bounded from below by the cv of the common bias element. Thus, we can conclude that SWEPT should remain asymptotically optimal for this case. Adapting WPI to this case remains an open research challenge. One approach to this problem is to assume all processing times are distributed lognormal, use the lognormal central limit theorem and use a basic lognormal common factor. In this scheme, the use of a lognormal distribution to present the sum of the independent positive processing times is supported by the lognormal central limit theorem (Appendix A), but the crux is whether the common bias element can be approximated by a lognormal variable too. If we assume that there are multiple small causes of bias then their combined effect is indeed approximately lognormal. This is true because by the regular central limit theorem the sum of the logarithms of these small causes is approximately normal and therefore their product is approximately lognormal.

The Minimal Value of d + γ*E(*T*) with Lognormal Processing Time*

Let μ and s be the mean of the lognormal distribution and the standard deviation of its core normal (see Appendix A for the relationship between these parameters). For a given due date, d, the service level is $\Phi(z)$, where $z = \ln(d/\mu)/s + s/2$. As we mentioned in our Research Notes for Chapter 6, $E(T) = \mu\Phi(s - z) - d\Phi(-z)$. By the discussion below Equation (RN7.4), $d = \mu + E(T) - E(E)$. So $E(E) = \mu + E(T) - d$. Using this result, if $d*$ is the optimal due date for minimizing $d + \gamma E(T)$, then, for $\gamma > 1$, some algebra reveals that,

$$d* + \gamma E(T) = \mu\gamma\Phi(s - z*)$$

Because the optimal service level does not depend on the distribution, and because the service level of a lognormal is determined by its core normal, $z^* = \Phi^{-1}((\gamma-1)/\gamma)$. We also obtain $d^* = \exp(m + sz^*)$, where m is the mean of the core normal.

When Are Active Release Dates Useful?

In Theorem RN7.5 we showed a particular case where no idling is needed. In other words, we did not require the use of release dates because all jobs start as soon as the machine is ready for them. In general, however, active release dates are often justified. For example, the problem studied by Trietsch (1993) requires setting release dates for outgoing flights and due dates for incoming ones. In such analysis, much depends on the true earliness costs. For instance, an underlying assumption behind Theorem RN7.5 is that all jobs are available at time zero so we have to pay for their flow time starting at time zero. But if we have the option of postponing the induction of job j, we only have to pay for its flow time from its release date until its completion. In such case, it is not necessary for the flow time cost to be identical to the earliness cost. Thus we obtain a more general model. In more detail, if jobs require inputs whose acquisition can be postponed and thus reduce costs (e.g., when scheduling a service, customers may be able to utilize the time prior to their job start outside the system), then the earliness cost should include the time value of these inputs and active release dates become desirable. In such cases the release date is used to schedule the arrival of jobs and other inputs for the jobs and there is a clear marginal saving associated with postponing them. To clarify this distinction, in this section, we study the usefulness of active release dates when the objective is given by (RN5.1) and includes flowtime cost. The relevant part of (RN5.1) is,

$$f(S) = \sum_{j=1}^{n} [w_j F_j + \alpha_j E_j + \beta_j T_j] \qquad \text{(RN7.6)}$$

Recall that in Chapter 2 we defined flow time as the time a job spends in the system and we defined r_j as the time the job becomes available, so we obtained $F_j = C_j - r_j$. Until now, however, we assumed that all jobs are available at time zero. Therefore, $F_j = C_j$, and the relevant part of (RN5.1) is,

$$f(S) = \sum_{j=1}^{n} [w_j C_j + \alpha_j E_j + \beta_j T_j]$$

Comparing to (RN7.5), the only difference is that completion time is weighted by w_j instead of α_j. By the proof of Theorem RN7.5, if $w_j \geq \alpha_j$ then release dates are unnecessary. (It is often reasonable to assume that $w_j \geq \alpha_j$ because the holding costs start when jobs become available, i.e., from time zero, whereas w_j may also include the economic time value associated with being able to quote an earlier (and yet reliable) due date to the customer.) We may also conclude that solving properly for $w_j < \alpha_j$ can require setting active release dates *and* due dates. However, our objective does not account directly for the value of machine time. In practice, machine time has positive economic value, often addressed as an opportunity cost. In addition, machine time may have to be booked in advance for the jobs at hand. Using the index 0 for the machine, if we book it from time zero (i.e., $r_0 = 0$) for a period of d_0, there is a charge of $w_0 d_0$ reflecting the expected opportunity cost. In other words, $w_0 d_0$ is the expected alternative profit that we forfeit by the booking. If we book too much time, we may be able to salvage the excess time later, but we should expect to recoup less than the full booking cost. If we denote the difference between w_0 and the expected salvage time value by α_0, then earliness relative to d_0 implies a loss of $\alpha_0(d_0 - C_0)^+$, where C_0 equals $C_{\max}$ (or $C_{[n]}$). However, if we are not ready to release the machine at the end of the booking period, we must forfeit w_0 for each time unit of tardiness and assess an additional tardiness penalty of $\beta_0(C_0 - d_0)^+$, to reflect the cost of disruption. For example, because the machine could otherwise have been booked, the disruption may cause tardiness later or the need for overtime. Whereas the economic cost of a time unit during tardiness, w_0, cannot be avoided by changing the booking time, the expected disruption penalty *is* a function of the booking time. Hence we obtain an E/T component for the booking with earliness cost of α_0 and tardiness cost of β_0. When we add these costs to our objective we obtain a more general objective,

$$f(S) = \sum_{j=0}^{n} [w_j \mathrm{E}(F_j) + \alpha_j \mathrm{E}(E_j) + \beta_j \mathrm{E}(T_j)] \qquad \text{(RN7.7)}$$

Technically, the only difference between (RN7.7) and (RN7.6) can be described as the addition of "job 0," which represents the machine. Conceptually, however, in this case we know that $w_0 > \alpha_0$, and there is therefore a stronger disincentive to include active release dates anywhere in the schedule. For any given set of release dates, if we use stored sample analysis, d_0 is solved by $d_0 = C_{\max}(\lceil r\beta_0/(\alpha_0 + \beta_0)\rceil)$, where the effect of active release dates, if any, is incorporated into the C_{max} column. (Similarly, but not identically, $d_{[n]} = C_{\max}(\lceil r\beta_{[n]}/(\alpha_{[n]} + \beta_{[n]})\rceil)$.) The presence of the machine time element in the objective, and the observation that $w_0 > \alpha_0$, give rise to a stronger version of Theorem RN7.5 (with an essentially identical proof).

Theorem RN7.8: For any given sequence and any job $[j]$ in the jth position, if $w_0 - \alpha_0 + \sum_{k=j}^{n} [w_{[k]} - \alpha_{[k]}] > 0$, then no active release date $r_{[j]}$ can improve the performance measure in (RN7.7).

Although Theorem RN7.8 gives sufficient and not necessary conditions, it is possible that release dates could be desirable when these conditions are not satisfied. Given any set of release dates, we already know how to set due dates, but the combined problem of setting release dates and due dates requires searching for the release dates and adjusting the due dates based on the release dates in such a manner that the service level targets are satisfied and the total cost is minimized. The good news is that this problem is still convex (for any given sequence), so it is amenable to solution by numerical search. Notice, however, that if $w_j = 0$ (for $j = 0, 1, \ldots, n$), then the optimal release dates should be very large. Theoretically, depending on the processing time distributions, the optimal release dates and due dates may not even be bounded. But if $\max\{w_0, w_{[n]}\} > 0$, all optimal release dates and due dates are finite. So this is not a problem in practice.

It is interesting to compare this solution to the Britney (1976) approach, because it is quite popular in practice. Under this approach we allocate enough safety time for each job individually, without considering its interactions with other jobs, and then set subsequent release dates after the allowed processing time for the previous job. Thus each job has a Parkinson distribution, and the scheduling is done as if the Parkinson tails do not exist. This approach not only wastes machine capacity and delays jobs unnecessarily but also fails to deliver the desired service levels: tardiness in an early job may delay subsequent jobs in a domino effect. This creates the well-known practical phenomenon that earliness is wasted but tardiness accumulates. In other words, using the Parkinson distribution recklessly just because it has lower variance exacerbates the waste associated with Parkinson's Law. We discuss this issue further in Chapter 18.

SOME ADDITIONAL OPEN RESEARCH QUESTIONS

Whereas we have provided a partial solution to the problem of maximizing the number of jobs processed subject to stochastic feasibility, there are many open questions around the stochastic weighted and un-weighted *U*-problem. Within the framework of stochastic feasibility, Akker and Hoogeveen (2008) discuss a version of the problem with several parallel machines. But even for a single machine, practically no results exist for the economic version of the problem. Both dynamic programming and branch and bound are likely candidates, but both require testing. Good heuristics are also in short supply at the moment.

Shifting our attention to models with due dates or release dates as decisions, most of the published results to date assume either a fixed tardiness cost or a proportional tardiness rate, but rarely both. Exceptions do exist: the original paper by Arrow, Harris and Marschak considers both and see also Laslo et al. (2007). Such models require further research, however. Studies of models with convex increasing penalty functions that can be used to model risk-averseness are also needed. One potential approach is to use piecewise linear penalty functions but adjust the rates to obtain the approximately correct result for the convex function. This is akin to the iterative use of linear programming to solve general convex programming models.

REFERENCES

Akker, van den J.M. and J.A. Hoogeveen (2008) "Minimizing the Number of Late Jobs in Case of Stochastic Processing Times with Minimum Success Probabilities, *Journal of Scheduling* 11, 59-69.

Arrow, K.J., T. Harris and J. Marschak (1951), "Optimal Inventory Policy," *Econometrica* 19(3), 250-272.

Baker, K.R. and D. Trietsch (2009) "Safe Scheduling: Setting Due Dates in Single-Machine Problems," *European Journal of Operational Research* 196, 69-77.

Baker, K.R. and D. Trietsch (2007) Safe Scheduling, Chapter 5 in *Tutorials in Operations Research* INFORMS, November (T. Klastorin, ed.), 79-101.

Balut, S.J. (1973) "Scheduling to Minimize the Number of Late jobs when Set-Up and Processing Times are Uncertain," *Management Science* 19(11), 1283-88.

Banerjee, B.P. (1965) "Single facility sequencing with random execution times," *Operations Research* 13, 358-364.

Black, G., K.N. McKay, and T.E. Morton (2006) "Aversion Scheduling in the Presence of Risky Jobs," *European Journal of Operational Research*, 175, 338-361.

Bollopragada, R. and N. Sadeh (2004) "Proactive Release Procedures for Just-in-Time Job Shop Environments, subject to Machine Failures," *Naval Research Logistics* 51, 1018-1044.

Britney, R.R. (1976) "Bayesian Point Estimation and the PERT Scheduling of Stochastic Activities," *Management Science* 22(9), 938-948.

Charnes, A., W. Cooper and G.H. Symonds (1958) "Cost Horizons and Certainty Equivalents: An Approach to Stochastic Programming of Heating Oil," *Management Science* 4, 235-263.

Charnes, A. and W. Cooper (1959) "Chance-Constrained Programming," *Management Science* 6, 73-79.

Cheng, T.E.C. (1987) "Optimal Due-Dates Determination and Sequencing with Random Processing Times," *Mathematical Modeling* 9, 573-576.

Chu, C., J.-M. Proth and X. Xie (1993) "Supply Management in Assembly Systems," *Naval Research Logistics* 40, 933−949.

Daniels, R.L., and J. Carillo (1997) "β-Robust Scheduling for Single-Machine Systems with Uncertain Processing Times," *IIE Transactions* 29, 977-985.

Daniels, R.L., and P. Kouvelis (1995) "Robust Scheduling to Hedge Against Processing Time Uncertainty in Single-Stage Production," *Management Science* 41, 363–376.

Dantzig, G.B. (1955) "Linear Programming under Uncertainty," *Management Science* 1(3), 197-206.

S.K. Das and S.C. Sarin (1988) "Selection of a Set of Part Delivery Dates in a Multi-Job Stochastic Assembly System," *IIE Transactions* 20(1), 4-11.

Dubois, D., H. Fargier and P. Fortemps (2003) "Fuzzy Scheduling: Modeling Flexible Constraints vs. Coping with Incomplete Knowledge," *European Journal of Operational Research* 147, 231-252.

Golenko-Ginzburg, D., S. Kesler and Z. Landsman (1995) "Industrial Job-Shop Scheduling with Random Operations and Different Priorities," *International Journal of Production Economics* 40, 185-195.

Herroelen, W. and R. Leus (2005) "Project Scheduling under Uncertainty: Survey and Research Potentials," *European Journal of Operational Research* 165, 289–306.

Hopp, W.J and M.L. Spearman (1993) "Setting Safety Leadtimes for Purchased Components in Assembly Systems," *IIE Transactions* 25, 2−11.

Kise, H. and T. Ibaraki (1983) "On Balut's Algorithm and NP-Completeness for a Chance Constrained Scheduling Problem," *Management Science* 29, 384-388.

Kouvelis, P and G. Yu (1997) *Robust Discrete Optimization and Its Applications*, Kluwer Academic Publishers, Boston.

Kumar, A. (1989) "Component Inventory Costs in an Assembly Problem with Uncertain Supplier Lead-Times," *IIE Transactions* 21, 112−121.

Laslo, Z., D. Golenko-Ginzburg and B. Keren (2007), "Optimal Booking of Machines in a Virtual Job Shop with Stochastic Processing Times to Minimize Total Machine Rental and Job Tardiness Costs," *International Journal of Production Economics* 111(2), 812-821.

Leon, V.J., S.D. Wu, and R.H. Storer (1994) "Robustness Measures and Robust Scheduling for Job Shops," *IIE Transactions* 26, 32–43.

McKay, K., T. Morton, P. Ramnath, and J. Wang (2000) "'Aversion Dynamics' Scheduling when the System Changes," *Journal of Scheduling* 3, 71-88.

Mehta, V. and R. Uzsoy (1998) "Predictable Scheduling of a Job Shop subject to Breakdowns," *IEEE Transactions on Robotics and Optimization* 14, 365-378.

Mehta, V. and R. Uzsoy (1999) "Predictable Scheduling of a Single Machine subject to Breakdowns," *International Journal of Computer Integrated Manufacturing* 12, 15-38.

Moore, J.M. (1968) "An *n* Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," *Management Science* 15, 102-109.

Morse, P.M. and G.F. Kimball (1951) *Methods of Operations Research*, MIT Press (see http://www.cna.org/about/history/ for the 1946 version, Report OEG 54).

von Neumann, J. and O. Morgenstern (1944) *Theory of Games and Economic Behavior*, Princeton University Press, NJ. [Republished in 2004 as a 60th anniversary edition.]

Ng C.T., X. Cai X. and T.C.E. Cheng (1999) "Scheduling Jobs with Random Processing Times on a Single Machine subject to Stochastic Breakdowns to Minimize Early-Tardy Penalties," *Naval Research Logistics* 46(4), 373-398.

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

Ronen, B. and D. Trietsch (1988) "A Decision Support System for Planning Large Projects," *Operations Research* 36, 882−890.

Sarin, S.C. and S.K. Das (1987) "Determination of Optimal Part Delivery Dates in a Stochastic Assembly Line," *International Journal of Production Research* 25, 1013−1028.

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

Soroush, H.M. (1999) "Sequencing and Due-Date Determination in the Stochastic Single Machine Problem with Earliness and Tardiness Costs," *European Journal of Operational Research* 13, 450-468.

Spearman, M.L. and R.Q. Zhang (1999), "Optimal Lead Time Policies," *Management Science* 45 (2), 290-295.

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

Trietsch, D. (2006) "Optimal Feeding Buffers for Projects or Batch Supply Chains by an Exact Generalization of the Newsvendor Model," *International Journal of Production Research* 44(4), 627-637.

Trietsch, D. and K. Baker (2008) "Minimizing the Number of Tardy Jobs with Chance Constraints and Stochastically Ordered Processing Times," *Journal of Scheduling* 11, 71-73.

Trietsch, D., L. Mazmanyan, L. Gevorgyan and K.R. Baker (2010), "A New Stochastic Engine for PERT," Working Paper.

Trietsch, D. and F. Quiroga (2004) "Coordinating *n* Parallel Stochastic Activities by an Exact Generalization of the Newsvendor Model," August 2004, ISOM Working Paper No. 282 (Revised July 2005). Available at: http://ac.aua.am/Trietsch/Web/Trietsch&Quiroga.pdf.

Wein, L.M. (1991) "Due-Date Setting and Priority Sequencing in a Multiclass M/G/1 Queue," *Management Science* 37, 834-850.

Weiss, G. (1992) "Turnpike Optimality of Smith's Rule in Parallel Machines Stochastic Scheduling," *Mathematics of Operations Research* 17, 255-270.

Wilhelm, W. E. and L. Wang (1986) "Management of Component Accumulation in Small Lot Assembly Systems," *Journal of Manufacturing Systems* 5, 27-39.

Yano, C.A. (1987) "Setting Planned Leadtimes in Serial Production Systems with Tardiness Costs," *Management Science* 33, 95−106.

Yano, C.A. (1987a) "Stochastic Leadtimes in Two-Level Assembly Systems," *IIE Transactions* 19, 371−378.

Zhou, X. and X. Cai (1997) "General Stochastic Single-Machine Scheduling with Regular Cost Functions," *Mathematical Computer Modelling* 26, 95-108.