Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.

Volume 236, Issue 2, 16 July 2014
ISSN 0377-2217
ELSEVIER
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
Available online at www.sciencedirect.com
ScienceDirect
www.elsevier.com/locate/ejor
EURO

This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Author's personal copy

European Journal of Operational Research 236 (2014) 445–452

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Minimizing earliness and tardiness costs in stochastic scheduling

Kenneth R. Baker *

Tuck School of Business, Dartmouth College, Hanover, NH 03755, United States

ARTICLE INFO

Article history:
Received 5 July 2013
Accepted 5 December 2013
Available online 12 December 2013

Keywords:
Sequencing
Stochastic scheduling
Earliness/tardiness
Branch and bound
Neighborhood search
Due-date assignment

ABSTRACT

We address the single-machine stochastic scheduling problem with an objective of minimizing total expected earliness and tardiness costs, assuming that processing times follow normal distributions and due dates are decisions. We develop a branch and bound algorithm to find optimal solutions to this problem and report the results of computational experiments. We also test some heuristic procedures and find that surprisingly good performance can be achieved by a list schedule followed by an adjacent pairwise interchange procedure.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The single-machine sequencing model is the basic paradigm of scheduling theory. In its deterministic version, the model has received a great deal of attention from researchers, leading to problem formulations, solution methods, scheduling insights, and building blocks for more complicated models. Extending that model into the realm of stochastic scheduling is an attempt to make the theory more useful and practical. However, progress in analyzing stochastic models has been much slower to develop, and even today some of the basic problems remain virtually unsolved. One such case is the stochastic version of the earliness/tardiness (E/T) problem for a single machine.

This paper presents a branch and bound (B&B) algorithm for solving the stochastic E/T problem with normally-distributed processing times and due dates as decisions. This is the first appearance of a solution algorithm more efficient than complete enumeration for this problem, so we provide some experimental evidence on the algorithm's computational capability. Although B&B algorithms are not new, they have seldom been applied to problems in stochastic scheduling. Arguably, too little research has been done on the application of such optimization approaches in stochastic scheduling problems, so a broader goal of this paper is to demonstrate that methodologies common in deterministic scheduling can successfully be applied to problems in stochastic scheduling.

In addition, we explore heuristic methods for solving the problem, and we show that a relatively simple procedure can be remarkably successful at producing optimal or near-optimal solutions. These results reinforce and clarify observations made in earlier research and ultimately provide us with a practical method of solving the stochastic E/T problem with virtually any number of jobs.

In Section 2 we formulate the problem under consideration, and in Section 3 we review the relevant literature. In Section 4, we describe the elements of the optimization approach, and we report computational experience in Section 5. Section 6 deals with heuristic procedures and the corresponding computational tests, and the final section provides a summary and conclusions.

2. The problem

In this paper we study the stochastic version of the single-machine E/T problem with due dates as decisions. To start, we work with the basic single-machine sequencing model (Baker & Trietsch, 2009a). In the deterministic version of this model, n jobs are available for processing at time 0, and their parameters are known in advance. The key parameters in the model include the processing time for job j (p_j) and the due date (d_j). In the actual schedule, job j completes at time C_j, giving rise to either earliness or tardiness. The job's earliness is defined by $E_j = \max\{0, d_j - C_j\}$ and its tardiness by $T_j = \max\{0, C_j - d_j\}$. Because the economic implications of earliness and tardiness are not necessarily symmetric, the unit costs of earliness (denoted by α_j) and tardiness (denoted by β_j) may be different. We express the objective function, or total cost, as follows:

$$G(d_1, d_2, \dots, d_n) = \sum_{j=1}^{n} (\alpha_j E_j + \beta_j T_j) \tag{1}$$

* Tel.: +1 6036462064.
E-mail address: ken.baker@dartmouth.edu

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.12.011

Author's personal copy

The deterministic version of this problem has been studied for over 30 years, and several variations have been examined in the research literature. Some of these variations have been solved efficiently, but most are NP-Hard problems. In the stochastic E/T problem, we assume that the processing times are random variables, so the objective becomes the minimization of the expected value of the function in (1). The stochastic version of the E/T problem has not been solved.

To proceed with the analysis, we assume that the processing time p_j follows a normal distribution with mean μ_j and standard deviation σ_j and that the p_j values are independent random variables. We use the normal because it is familiar and plausible for many scheduling applications. Few results in stochastic scheduling apply for arbitrary choices of processing time distributions, so researchers have gravitated toward familiar cases that resonate with the distributions deemed to be most practical. Several papers have addressed stochastic scheduling problems and have used the normal distribution as an appropriate model for processing times. Examples include Anderson and Moodie (1969), Balut (1973), Cai and Zhou (2007), Jang (2002), Portougal and Trietsch (2006), Sarin, Erdel, and Steiner (1991), Seo, Klein, and Jang (2005), Soroush (1999), Soroush and Fredendall (1994), and Wu, Brown, and Beck (2009).

In our model, the due dates d_j are decisions and are not subject to randomness. The objective function for the stochastic problem may be written as

$$H(d_1, d_2, \ldots, d_n) = E[G(d_1, d_2, \ldots, d_n)] = \sum_{j=1}^{n}(\alpha_j E[E_j] + \beta_j E[T_j]) \qquad (2)$$

The problem consists of finding a set of due dates and a sequence of the jobs that produce the minimum value of the function in (2).

3. Literature review

The model considered in this paper brings together several strands of scheduling research – namely, earliness/tardiness criteria, due-date assignments, and stochastic processing times. We trace the highlights of these themes in the subsections that follow.

3.1. Earliness/tardiness criteria

Scheduling problems comprising both earliness costs and tardiness costs were first examined for the case in which processing times and due dates are given. This type of problem was first studied by Sidney (1977), who analyzed the minimization of maximum cost and by Kanet (1981), who analyzed the minimization of total absolute deviation from a common due date, under the assumption that the due date is late enough that it does not impose constraints on sequencing choices. This objective is equivalent to an E/T problem in which the unit costs of earliness and tardiness are symmetric and the same for all jobs. For this special case, Hall, Kubiak, and Sethi (1991) developed an optimization algorithm capable of solving problems with hundreds of jobs, even if the due date is restrictive. In addition, Hall and Posner (1991) solved the version of the problem with symmetric earliness and tardiness costs that vary among jobs. Their algorithm handles over a thousand jobs.

The case of distinct due dates is somewhat more challenging than the common due-date model. Garey, Tarjan, and Wilfong (1988) showed that the E/T problem with distinct due dates is NP-Hard, although for a given sequence, the scheduling of idle time can be determined by an efficient algorithm. Optimization approaches to the problem with distinct due dates were proposed and tested by Abdul-Razaq and Potts (1988), Ow and Morton (1989), Yano and Kim (1991), Azizoglu, Kondakci, and Kirca (1991), Kim and Yano (1994), Fry, Armstrong, Darby-Dowman, and Philipoom (1996), Li (1997), and Liaw (1999). Fry et al. addressed the special case in which earliness costs and tardiness costs are symmetric and common to all jobs. Their B&B algorithm was able to solve problems with as many as 25 jobs. Azizoglu et al. addressed the version in which earliness costs and tardiness costs are common, but not necessarily symmetric, and with inserted idle time prohibited. Their B&B algorithm solved problems with up to 20 jobs. Abdul-Razaq and Potts developed a B&B algorithm for the more general cost structure with distinct costs but prohibited inserted idle time. Their algorithm was able to solve problems up to about 25 jobs. Li proposed an alternative lower bound calculation for the same problem but still encountered computational difficulties in solving problems larger than about 25 jobs. Liaw's subsequent improvements extended this range to at least 30 jobs.

Because optimization methods have encountered lengthy computations times for problems larger than about 25–30 jobs, much of the computational emphasis has been on heuristic procedures. Ow and Morton were primarily interested in heuristic procedures for a version of the problem that prohibits inserted idle time, but they utilized a B&B method to obtain solutions (or at least good lower bounds) to serve as a basis for evaluating their heuristics. They reported difficulty in finding optimal solutions to problems containing 15 jobs. Yano and Kim compared several heuristics for the special case in which earliness and tardiness costs are proportional to processing times. The B&B algorithm they used as a benchmark solved most of their test problems up to about 16 jobs. Kim and Yano developed a B&B algorithm to solve the special case in which earliness costs and tardiness costs are symmetric and identical. Their B&B algorithm solved all of their test problems up to about 18 jobs. Lee and Choi (1995) reported improved heuristic performance from a genetic algorithm. To compare heuristic methods, they used lower bounds obtained from CPLEX runs that were often terminated after 2 hours of run time, sometimes even for problems containing 15 jobs. James and Buchanan (1997) studied variations on a tabu-search heuristic and used an integer program to produce optimal solutions for problems up to 15 jobs.

Detailed reviews of this literature have been provided by Kanet and Sridharan (2000), Hassin and Shani (2005) and M'Hallah (2007). The reason for emphasizing problem sizes in these studies, although they may be somewhat dated, is to contrast the limits on problem size encountered in studies of the distinct due-date problem with those encountered in the common due-date problem. This pattern suggests that stochastic versions of the problem may also be quite challenging when each job has its own due date.

3.2. Due-date assignments

The due-date assignment problem is familiar in the job shop context, in which due dates are sometimes assigned internally as progress targets for scheduling. However, for our purposes, we focus on single-machine cases. Perhaps the most extensively studied model involving due-date assignment is the E/T problem with a common due date. The justification for this model is that it applies to several jobs of a single customer, or alternatively, to several subassemblies of the same final assembly. The E/T problem still involves choosing a due date and sequencing the jobs, but the fact that only one due date exists makes the problem intrinsically different from the more general case involving a distinct due date assignment for each job. Moreover, flexibility in due-date assignment means that the choice of a due date can be made without imposing unnecessary constraints on the problem, so formulations of the due date assignment problem usually correspond to the common due-date problem with a given but nonrestrictive due date.

Panwalkar, Smith, and Seidmann (1982) introduced the due-date assignment decision in conjunction with the common

Author's personal copy

due-date model, augmenting the objective function with a cost component for the lead time. In their model, the unit earliness costs and unit tardiness costs are asymmetric but identical across jobs. The results include a simple algorithm for finding the optimal due date. Surveys of the common due-date assignment problem were later compiled by Baker and Scudder (1990) and by Gordon, Proth, and Chu (2002). As discussed below, however, very little of the work on due-date assignment has dealt with stochastic models.

Actually, in the deterministic case, if due dates are distinct, then the due-date assignment problem is trivial because earliness and tardiness can be avoided entirely. Baker and Bertrand (1981), who examined heuristic rules for assigning due dates, such as those based on constant, slack-based, or total-work leadtimes, also characterized the optimal due-date assignment when the objective is to make the due dates as tight as possible. Seidmann, Panwalkar, and Smith (1981) proposed a specialized variation of the single-machine model with unit earliness and tardiness costs common to all jobs, augmenting the objective function with a cost component that penalizes loose due dates if they exceed customers' reasonable and expected lead time. They provided an efficient solution to that version of the problem as well. Other augmented models were addressed by Shabtay (2008). Because the due-date assignment problem is easy to solve in the single-machine case when due dates are distinct, papers on the deterministic model with distinct due dates typically assume that due dates are given, and relatively few papers deal with distinct due dates as decisions. When processing times are stochastic, however, the due-date assignment problem becomes more difficult.

3.3. Stochastic processing times

The stochastic counterpart of a deterministic sequencing problem is defined by treating processing times as uncertain and then minimizing the expected value of the original deterministic performance measure. Occasionally, it is possible to substitute mean values for uncertain processing times and simply call on results from deterministic analysis. This approach works for the minimization of expected total weighted completion time, which is minimized by sequencing the jobs in order of shortest weighted expected processing time, or SWEPT (Rothkopf, 1966). For the minimization of expected maximum tardiness, it is optimal to sequence the jobs in order of earliest due date, or EDD (Crabill & Maxwell, 1969). However, replacing uncertain processing times by mean values and calculating the deterministic objective function under EDD may not produce the correct value for the stochastic objective function. In fact, suppressing uncertainty seldom leads to the optimal solution of stochastic sequencing problems; problems that are readily solvable in the deterministic case may be quite difficult to solve when it comes to their stochastic counterpart. An example is the minimization of the number of stochastically tardy jobs (i.e., those that fail to meet their prescribed service levels). Kise and Ibaraki (1983) showed that even this relatively basic problem is NP-Hard.

Stochastic scheduling problems involving earliness and tardiness have rarely been addressed in the literature. Cai and Zhou (2007) analyzed a stochastic version of the common due-date problem with earliness and tardiness costs (augmented by a completion-time cost), with the due date allowed to be probabilistic and the variance of each processing time distribution assumed to be proportional to its mean. Although the proportionality condition makes the problem a special case, it can at least be considered a stochastic counterpart of a deterministic model discussed by Baker and Scudder (1990). Xia, Chen, and Yue (2008) described a heuristic procedure to solve the stochastic E/T problem with common earliness costs and common tardiness costs, augmented by a cost component reflecting the tightness of the due dates. This formulation is the stochastic counterpart of a special case of the problem analyzed by Seidmann et al. (1981), which was solved by an efficient algorithm.

The more general stochastic version of the common due-date problem has not been solved, and only modest progress has been achieved on the stochastic E/T problem with distinct due dates. Soroush and Fredendall (1994) analyzed a version of that problem with due dates given. Soroush (1999) later proposed some heuristics for the version with due dates as decisions, and Portougal and Trietsch (2006) showed that one of those heuristics was asymptotically optimal. However, an optimization algorithm for that problem has not previously been developed and tested. Thus, this paper develops an optimization algorithm to solve a problem that heretofore has been attacked only with heuristic procedures. One hint that B&B techniques may be successful in attacking a stochastic sequencing problem with due-dates as decisions comes from the recent results in Baker (2014), who showed how to solve a basic safe-scheduling problem with branch and bound.

4. Analysis of the stochastic problem

To analyze the model in (2), we exploit the property that sums of normal random variables are also normal. Thus, in any sequence, the completion time of job j follows a normal distribution. Using notation, let B_j denote the set of jobs preceding job j in the schedule. Then C_j follows a normal distribution with mean $E[C_j] = \sum_{i\in B_j}\mu_i + \mu_j$ and variance $\mathrm{var}[C_j] = s_j^2 = \sum_{i\in B_j}\sigma_i^2 + \sigma_j^2$. To streamline the notation, we write $E[C_j] = \mu_{B_j} + \mu_j$ and $s_j^2 = \sigma_{B_j}^2 + \sigma_j^2$. Once we know the properties of the random variable C_j, we can determine the optimal choice of d_j.

Theorem 1. *Given the mean $E[C_j]$ and the standard deviation s_j of the normal distribution for C_j, the optimal choice of the due date d_j is given by:*

$$\Phi(k_j^*) = \frac{\beta_j}{\alpha_j + \beta_j}$$

where $k_j^* = (d_j - E[C_j])/s_j$ represents the standardized due date and where $\Phi(\cdot)$ denotes the standard normal cdf.

This result is originally due to Soroush (1999). For completeness and consistency in notation, it is derived in Appendix A.[1] The result is also familiar as the "newsvendor" property of inventory theory; it specifies the optimal service level (the probability that job j completes on time), thereby linking the model to basic notions of safe scheduling (Baker & Trietsch, 2009b).

The implication of Theorem 1 is that the appropriate choice for the due date of job j is

$$d_j = E[C_j] + k_j^* s_j = \mu_{B_j} + \mu_j + k_j^*(\sigma_{B_j}^2 + \sigma_j^2)^{1/2} \tag{3}$$

In this expression the due date d_j depends on the previous jobs in sequence via the set B_j, and the objective is summarized in (2). From the algebraic derivation given in Appendix A, we can rewrite (2) by incorporating the optimal choice of d_j. The objective becomes

$$H(d_1^*, d_2^*, \ldots, d_n^*) = \sum_{j=1}^{n} \varphi(k_j^*)(\alpha_j + \beta_j)s_j \tag{4}$$

where k_j^* is the standard normal variate corresponding to the optimal service level of Theorem 1. However, from the given values of α_j and β_j, we can compute the corresponding value of $\varphi(k_j^*)$ and substitute $c_j = \varphi(k_j^*)(\alpha_j + \beta_j)$, allowing us to rewrite the objective function more simply as

[1] The appendices can be found in an Online Supplement.

Author's personal copy

$$H(d_1^*, d_2^*, \ldots, d_n^*) = \sum_{j=1}^{n} c_j s_j \tag{5}$$

Having specified the objective function, we are interested in finding its optimal value. It is possible, of course, to find an optimum by enumerating all possible job sequences and then selecting the sequence with minimal objective function value. Until now, enumeration has been the only solution algorithm known for this problem, as in Soroush (1999). However, as a computational method, enumeration is ultimately limited by the size of the solution space. (Soroush reported obtaining solutions for 12-job problems in an average of more than 9 hours of cpu time.) We can compute optimal solutions more efficiently by employing dominance conditions and a B&B algorithm, which we describe next.

4.1. A dominance condition

Dominance conditions can accelerate the search for an optimal schedule. Job *j* is said to *dominate* job *k* if an optimal sequence exists in which job *j* precedes job *k*. If we confirm a dominance condition of this type, then in searching for an optimal sequence we need not pursue any partial sequence in which job *k* precedes job *j*. Dominance conditions can reduce the computational effort required to find an optimal schedule, but the extent to which they apply depends on the parameters in a given problem instance. For that reason, it is difficult to predict the extent of the improvement.

In our model, a relatively simple dominance condition holds.

Theorem 2. *For two jobs j and k, if $c_j \geqslant c_k$ and $\sigma_j \leqslant \sigma_k$ then job j dominates job k.*

Theorem 2, noted by Portougal and Trietsch (2006), can be proven by means of a pairwise interchange argument, and the details appear in Appendix B for consistency in notation and level of detail. Thus, if we are augmenting a partial sequence and we notice that job *j* dominates job *k* while neither appears in the partial sequence, then we need not consider the augmented partial sequence constructed by appending job *k* next. Although we may encounter quite a few dominance properties in randomly-generated instances, it is also possible that no dominance conditions hold. That would be the case if the costs and standard deviations were all ordered in the same direction. In such a situation, dominance properties would not help reduce the computational burden, and the computational effort involved in testing dominance conditions would be counterproductive. Nevertheless, in our randomly-generated test problems, dominance properties reduced the computational burden considerably, as compared to complete enumeration.

4.2. A lower bound

Suppose we have a partial sequence of the jobs, denoted by π, and we wish to compute a lower bound on the value of the objective function (5) that can be obtained by completing the sequence. From the partial sequence we can calculate the portion of the objective function contributed by the jobs in π. Now, let π' denote the set of unscheduled jobs. In the set π', we take the set of coefficients c_j in largest-first order and separately, the set of standard deviations σ_j in smallest-first order, and we treat these values as if they were paired in the set of unscheduled jobs. These are fictitious jobs due to the rearrangement of coefficients and standard deviations. Next we calculate each fictitious job's contribution to the objective and add it to the portion for the partial sequence π. This total provides a lower bound (denoted by LB1) on the value that could be achieved by completing the partial sequence in the best possible way. The justification is based on the following two (new) results.

Theorem 3. *For any sequence of coefficients c_j, the expression $\sum_{j=1}^{n} c_j s_j$ is minimized by sequencing the σ-values in nondecreasing order. (See Appendix B for a proof.)*

Theorem 4. *For any sequence of σ-values, the expression $\sum_{j=1}^{n} c_j s_j$ is minimized by sequencing the c-values in nonincreasing order. (See Appendix B for a proof.)*

Thus, in the course of an enumerative search, if we encounter a partial sequence for which the lower bound is greater than or equal to the value of the objective function for a full sequence, we know that the partial sequence can never lead to a solution better than the full sequence. We can also combine lower bounds along with dominance conditions in the search. This combination involves the most testing of conditions, but it can eliminate the most partial sequences, and the combination leads to reduced computation times in all but the smallest problems.

To illustrate the calculations involved in the calculation of the lower bound, consider the five-job example in Table 1, and assume that job 1 has been placed first in sequence so that $\pi = \{1\}$, with the other four jobs remaining to be scheduled.

We first consider the four unscheduled jobs. If we order the jobs according to Theorem 2, with σ-values in nondecreasing sequence, the job sequence is 4-2-3-5. If we order the jobs according to Theorem 3, with *c*-values in nonincreasing sequence, the job sequence is 5-4-3-2. Thus, the sequence starting with job 1 followed by four fictitious jobs (f1–f4) exhibits the parameters obtained from these two sequences, as shown in Table 2.

The lower bound LB1 is computed as $\sum_{j=1}^{n} c_j s_j = 91.560$.

4.3. Another lower bound

A different lower bound was described by Portougal and Trietsch (2006), although they did not test their bound in a B&B algorithm. Again, suppose we have a partial sequence of the jobs, denoted by π, and we wish to compute a lower bound on the value of the objective function (5) that can be obtained by completing the sequence. From the partial sequence we can calculate

$$s_0 = \left(\sum_{i \in \pi} \sigma_i^2\right)^{1/2}$$

Again, let π' denote the set of unscheduled jobs and reorder them by nondecreasing ratios of σ_i^2/c_i. Temporarily, we also renumber the jobs in π' according to this ordering, and we let:

$$s_j^2 = \sum_{i \in \pi} \sigma_i^2 + \sum_{i \leqslant j} \sigma_i^2$$

For job *j*, a lower bound on its contribution (in this sequence) to the objective function is calculated as[2]

$$g_j = \frac{2c_j}{3\sigma_j^2}\left(s_j^3 - s_{j-1}^3\right)^2.$$

Therefore, a lower bound (denoted by LB2) on the best objective that can be obtained by completing the partial schedule π takes the form $\sum_{j \in \pi} c_j s_j + \sum_{j \in \pi'} g_j$. In this paper, we test for the first time the computational implications of using this bound.

To illustrate the calculations involved in the calculation of LB2, we return to the example in Table 1, with job 1 placed first in sequence. For this partial sequence, $s_0 = 5$.

The unscheduled jobs are considered in nondecreasing order of their weighted variance, and the calculations of g_j follow. Finally, the lower bound is calculated as $\text{LB2} = \sum_{j \in \pi} c_j s_j + \sum_{j \in \pi'} g_j = 8.775 + \sum_{j \in \pi'} g_j = 93.878$ (see Table 3).

[2] An erroneous formula appeared in the original article due to a misprint.

Author's personal copy

Table 1
A five-job example.

Job j	1	2	3	4	5
σ_j	5	2	3	1	4
α_j	1	1	2	2	3
β_j	9	12	16	25	27
c_j	1.755	1.876	3.409	3.786	5.265

Table 2
Parameters of fictitious jobs when $\pi = \{1\}$.

Job j	1	$f1$	$f2$	$f3$	$f4$
σ_j	5	1	2	3	4
c_j	1.755	5.265	3.786	3.409	1.876
s_j	5	5.099	5.477	6.245	7.416

4.4. Special cases

Before describing the computational experiments, we pause to reflect on two special cases related to the development of the deterministic E/T problem. First, consider the special case in which earliness costs are the same for all jobs ($\alpha_j = \alpha$) and tardiness costs are the same for all jobs ($\beta_j = \beta$). In that case, the ratio $\beta_j/(\alpha_j + \beta_j)$ is the same for all jobs. Thus, $\varphi(k_j^*) = \varphi(k^*)$ and $c_j = c$. The objective function in (5) becomes

$$H(d_1^*, d_2^*, \ldots, d_n^*) = \sum_{j=1}^{n} cs_j = c\sum_{j=1}^{n} s_j$$

Therefore, by Theorem 3, the optimal sequence in the stochastic case is obtained by ordering the jobs by shortest variance (equivalently, shortest standard deviation). Thus, when the unit earliness and tardiness costs are common to all jobs, the optimal solution can be obtained efficiently. In the stochastic E/T problem, then, the problem becomes much more challenging when the unit costs are distinct. This property echoes the result of Seidmann et al. (1981) for the deterministic counterpart.

A further special case corresponds to the minimization of total expected absolute deviation (of completion time from due date), which corresponds to $\alpha_j = \beta_j = 1$. In this case, the optimal sequence is again obtained by ordering the jobs by shortest variance. In addition, $k_j^* = 0$, so each job's due date is optimally assigned to the job's expected completion time in the sequence. This property echoes the result of Baker and Bertrand (1981) for the deterministic counterpart.

5. Computational results for optimization methods

For experimental purposes, we generated a set of test problems that would permit comparisons of the various algorithms. The mean processing times μ_j were sampled from a uniform distribution between 10 and 100. Then, for each job j, the standard deviation was sampled from a uniform distribution between $0.10\mu_j$ and $0.25\mu_j$. In other words, the mean was between 4 and 10 times the standard deviation, so the chances of encountering a negative value for a processing time would be negligible. (Very few precedents can be found in the literature for designing test problems for stochastic scheduling. However, this design resembles previous experimental work with the normal distribution: Xia et al. (2008) also generated mean values as small as 4 times the standard deviation, while Soroush (1999) and Cai and Zhou (2007) allowed for means as small as 3.4 times the standard deviation.) In addition, the unit costs α_j and β_j were sampled independently from a uniform distribution between 1 and 10 on a grid of 0.1. For each value of n, a sample of 100 randomly-generated problem instances was created.

Table 3
Parameters for calculating LB2.

Job j	1	4	2	3	5
σ_j	5	1	2	3	4
c_j	1.755	3.786	1.876	3.409	5.265
s_j	5.000	5.099	5.477	6.245	7.416
s_j^3	125.000	132.575	164.317	243.555	407.891
$s_j^3 - s_{j-1}^3$		7.575	31.742	79.238	164.336
σ_j^2/c_j		0.264	2.132	2.640	3.039
g_j		19.118	9.925	20.009	36.051

The B&B algorithm was coded in VBA to maintain an easy interface with Excel and was implemented on a laptop computer using a 2.9 gigahertz processor. Table 4 summarizes the computational experience in these test problems, using lower bounds LB1 and LB2, as compared to complete enumeration. Performance is measured by the average cpu time required and the average number of nodes encountered in the search. Each average was taken over 100 randomly-generated problem instances with the same parametric features.

As Table 4 shows, a 10-job problem took more than 50 seconds on average to find a solution using complete enumeration. The other algorithms took less than 0.02 second. Where enumeration is concerned, computation times become prohibitive for larger problem sizes. In particular, 12-job problems took roughly 1.7 hours with complete enumeration. (This figure compares to the nine-hour solution times reported by Soroush; the improvement probably reflects advances in hardware since the time of his experiments.)

The results in Table 4 also show that lower bound LB1 yields slightly better performance than LB2, although this advantage disappeared for larger problem sizes. (As discussed below, additional experiments revealed better performance of LB2 compared to LB1 at problem sizes of $n \geqslant 15$.)

Next, we report computational experience on larger problem sizes, but with an enhancement to the B&B algorithm. The enhancement is a preliminary step which finds a good solution using a heuristic method. In our case, as discussed in the next section, we found that an excellent heuristic procedure is represented by implementing a neighborhood search based on adjacent pairwise interchanges (API).

Table 5 summarizes results for problem sizes of $12 \leqslant n \leqslant 20$. As in Table 4, each entry is based on 100 randomly-generated problem instances generated under the same parametric conditions. For each algorithm, the table shows the average cpu time and the maximum cpu time in the 100 instances. Shown in parentheses is the number of times that the solution could not be confirmed in an hour of cpu time. For the larger problem sizes, the best results came from using LB2. The B&B algorithm based on LB2 was able to solve all 18-job problems in under an hour. By comparison, the algorithm based on LB1, was not quite able to do so, even with the API heuristic solution computed first. For the 20-job problems, the algorithm based on LB2 was able to solve 94 of the 100 test problems in less than an hour, and this figure remained the same when the API heuristic was incorporated. However, the average time required to find a solution was somewhat lower with the API heuristic.

6. Computational results for heuristic methods

Because the stochastic E/T problem is difficult to solve optimally, it is relevant to explore heuristic procedures that do

Author's personal copy

Table 4
Computation effort for modest problem sizes.

Size (n)	Average Time (s)			Average Nodes		
	Enumeration	B&B/LB1	B&B/LB2	Enumeration	B&B/LB1	B&B/LB2
6	0.015	0.001	0.001	2676	106.8	143.5
8	0.566	0.002	0.003	149920	460.1	681.4
10	50.572	0.013	0.018	13492900	3181.8	4380.5

Table 5
Computation effort for larger problem sizes.

Size (n)	Average Time (s)				Maximum Time (s)			
	LB1	LB1/API	LB2	LB2/API	LB1	LB1/API	LB2	LB2/API
12	0.09	0.06	0.13	0.08	0.55	0.46	0.68	0.55
15	2.53	1.63	2.53	1.39	19.10	14.37	19.45	11.92
18	255.89	174.25	137.01	83.17	(2)	(1)	3042.20	1866.91
20			1780.20	1187.20			(6)	(6)

not require extensive computing effort. In this section, we study the performance of some heuristic procedures.

A simple and straightforward heuristic procedure is to create a list schedule. In other words, the list of jobs is sorted in some way and then the schedule is created in one pass by sequencing the jobs in their sorted order. In some stochastic scheduling models, sorting by expected processing time can be effective, but in this problem, the optimal choice of the due dates adjusts for differences in the jobs' processing times. Instead, it makes sense to focus on the standard deviations or variances of the processing times as a means of distinguishing the jobs. The simplest way to do so is to sort the jobs by smallest standard deviation or by smallest variance. Because the jobs are also distinguished by unit costs, it also makes sense to investigate cost-weighted versions of those orderings, such as smallest weighted standard deviation (SWSD), which takes the jobs in nondecreasing order of σ_j/c_j, and smallest weighted variance (SWV) which takes the jobs in nondecreasing order of σ_j^2/c_j. Soroush (1999) tested list schedules for these two rules and found that, at least on smaller problem sizes, they often produced solutions within 1% of optimality.

A standard improvement procedure for sequencing problems is a neighborhood search. In this case, we can use a sorting rule (SWV) to find an initial job sequence and then test adjacent pairwise interchanges (API) in the schedule to seek an improvement. If an improvement is found, the API neighborhood of the improved sequence is tested, and the process iterates until no further improvement is possible. API methods have proven effective in solving deterministic versions of the E/T problem, a finding that dates back to Yano and Kim (1991). In our tests, API methods were remarkably effective.

The quality of the heuristic solutions is summarized in Table 6 for the same set of test problems described earlier. Three summary measures are of interest: (1) the number of optima produced, (2) the average (relative) suboptimality, and (3) the maximum suboptimality. As the table shows, the performances of the heuristic procedures were quite different. The SWSD list schedule produced solutions that averaged about 2% above optimal, and the SWV list schedule produced solutions that were two orders of magnitude better. The SWV procedure also produced optimal solutions in about 60% of the test problems. However, perhaps the most surprising result was that the API heuristic generated optimal solutions in every one of the 700 test problems. (The API heuristic cannot guarantee optimality, however. This point is discussed in Appendix C.)

To some extent, these results are consistent with previous work on the stochastic E/T problem. Soroush (1999) provided computational results demonstrating the effectiveness of the SWV list schedule. His testbed was slightly different, but his results showed that SWV performed better than SWSD and frequently produced optimal sequences. Portougal and Trietsch (2006) showed that SWV is asymptotically optimal. In other words, the difference between the objective function produced by SWV and the optimal value becomes negligible (relative to the optimal value) as n grows large. Portugal and Trietsch discussed the fact that other rules, such as SWSD, do not possess this property. The difference between a rule that exhibits asymptotic optimality and a rule that does not is illustrated in our comparisons of SWV and SWSD. Neither of those earlier studies, however, tested the effectiveness of the API rule.

The feature of asymptotic optimality is important in two ways. First, although it is a limiting property, we can see in Table 6 that asymptotic behavior is approached in the range of problem sizes covered: the worst-case suboptimality drops to 0.1% by the time n reaches 20. We can expect that schedules produced by SWV are even closer to optimality for larger problem sizes. Second, the computational limits of the B&B algorithm – that is, the difficulty of finding optimal solutions for $n > 20$ – are not particularly worrisome. A practical approach to solving larger problems would be to

Table 6
Performance of the heuristic methods.

	Optima			Average			Maximum		
n	SWSD	SWV	API	SWSD (%)	SWV (%)	API (%)	SWSD (%)	SWV (%)	API (%)
6	34	84	100	1.51	0.03	0.00	11.98	0.55	0.00
8	3	75	100	2.03	0.03	0.00	6.84	0.30	0.00
10	4	69	100	2.25	0.03	0.00	9.03	0.26	0.00
12	0	64	100	2.44	0.02	0.00	10.73	0.34	0.00
15	0	46	100	2.52	0.02	0.00	7.28	0.11	0.00
18	0	43	100	2.39	0.01	0.00	7.19	0.07	0.00
20	0	40	100	2.40	0.01	0.00	5.60	0.09	0.00

Author's personal copy

sort the jobs by SWV (exploiting its asymptotic optimality property) and then optimize the sequence for the first 12 or 15 jobs using B&B. This approach gives us a reliable way of solving the stochastic E/T problem of virtually any size, with a strong likelihood that our solution will be within 0.1% of the optimum.

7. Summary and conclusions

We have analyzed the stochastic E/T problem with distinct unit costs among the jobs and distinct due dates as decisions. In this problem, we seek to assign due dates and sequence the jobs so that the expected cost due to earliness and tardiness is as small as possible. We first noted that the optimal assignment of due dates translates into a critical fractile rule specifying the optimal service level for each job. We then described a B&B approach to this problem, incorporating lower bounds and dominance conditions to reduce the search effort. Our computational experiments indicated that we can solve problems of up to around 20 jobs within an hour of cpu time. Although these problem sizes might not seem large, our literature review reminds us that computational experience for the deterministic counterpart was seldom much better. In addition, the 20-job problem is about twice the size of a problem that could be solved by enumeration in an hour of cpu time. Nevertheless, a possible avenue for further research would be the development of more sophisticated dominance conditions and lower bounds.

We pointed out that a special case of this problem – when all jobs have identical (but asymmetric) unit costs of earliness and tardiness – can be solved quite efficiently, by sorting the jobs from smallest to largest variance. A cost-weighted version of this procedure is not optimal in the general problem but appears to produce near-optimal solutions reliably when used as a heuristic rule. Moreover, when that solution is followed with an Adjacent Pairwise Interchange neighborhood search for improvements, the resulting algorithm produced optimal solutions in all of our test problems.

Our analysis was based on the assumption that processing times follow normal distributions. The normal distribution is convenient because it implies that completion times follow normal distributions as well. As Portougal and Trietsch (2006) observed, the role of completion times in the objective function leads theoretically to the use of convolutions in the analysis. Among standard probability distributions that could be used for processing times, only the normal gives us the opportunity to rely on closed-form results. In place of the normal distribution, we could assume that processing times follow lognormal distributions. The lognormal is sometimes offered as a more practical representation of uncertain processing times; indeed, it may be the most useful standard distribution for that purpose. However, sums of lognormal distributions are not lognormal, implying that it would be difficult to model completion times exactly. Nevertheless, the lognormal is associated with a specialized central limit theorem which resembles the familiar one that applies to the normal distribution (Mazmanyan, Ohanian, & Trietsch, 2008). In other words, our analysis for the normal could be adapted, at least approximately, for the lognormal as well. However, by focusing here on the normal distribution, our analysis has been exact, and no approximations have been necessary.

Looking back to the research done on the deterministic counterpart, we note that the E/T model was sometimes augmented with an objective function component designed to capture the tightness of due dates or to motivate short turnaround times. Augmenting the stochastic E/T problem in such ways, as suggested in the safe-scheduling overview in Baker and Trietsch (2009b), would appear to be another fruitful area in which to build on this research.

Acknowledgements

The author wishes to thank the referees for their comments and suggestions on an earlier draft, which have altered and improved the paper considerably.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.ejor.2013.12.011.

References

Abdul-Razaq, T., & Potts, C. (1988). Dynamic programming state-space relaxation for single-machine scheduling. *Journal of the Operational Research Society, 39*, 141–152.

Anderson, D., & Moodie, C. (1969). Optimal buffer storage capacity in production line systems. *International Journal of Production Research, 7*, 233–240.

Azizoglu, M., Kondakci, S., & Kirca, O. (1991). Bicriteria scheduling problem involving total tardiness and total earliness penalties. *International Journal of Production Economics, 23*, 17–24.

Baker, K. (2014). Setting optimal due dates in a basic safe scheduling model. *Computers & Operations Research, 41*, 109–114.

Baker, K., & Bertrand, W. (1981). A comparison of due-date selection rules. *AIIE Transactions, 13*, 123–131.

Baker, K., & Scudder, G. (1990). Sequencing with earliness and tardiness penalties: A review. *Operations Research, 38*, 22–36.

Baker, K., & Trietsch, D. (2009a). *Principles of Sequencing and Scheduling*. New York: John Wiley & Sons.

Baker, K., & Trietsch, D. (2009b). Safe scheduling: setting due dates in single-machine problems. *European Journal of Operational Research, 196*, 69–77.

Balut, S. (1973). Scheduling to minimize the number of late jobs when set-up and processing times are uncertain. *Management Science, 19*, 1283–1288.

Cai, X., & Zhou, S. (2007). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics, 44*, 531–557.

Crabill, T., & Maxwell, W. (1969). Single machine sequencing with random processing times and random due-dates. *Naval Research Logistics Quarterly, 16*, 549–555.

Fry, T., Armstrong, R., Darby-Dowman, K., & Philipoom, P. (1996). A branch and bound procedure to minimize mean absolute lateness on a single processor. *Computers & Operations Research, 23*, 171–182.

Garey, M., Tarjan, R., & Wilfong, G. (1988). One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research, 13*, 330–348.

Gordon, V., Proth, J.-M., & Chu, C. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research, 139*, 1–25.

Hall, N., Kubiak, W., & Sethi, S. (1991). Earliness–tardiness scheduling problems. II: Deviation of completion times about a restrictive common due date. *Operations Research, 39*, 847–856.

Hall, N., & Posner, M. (1991). Earliness–tardiness scheduling problems. I: Weighted deviation of completion times about a common due date. *Operations Research, 39*, 836–846.

Hassin, R., & Shani, M. (2005). Machine scheduling with earliness, tardiness and non-execution penalties. *Computers & Operations Research, 32*, 683–705.

James, R., & Buchanan, J. (1997). A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem. *European Journal of Operational Research, 102*, 513–527.

Jang, W. (2002). Dynamic scheduling of stochastic jobs on a single machine. *European Journal of Operational Research, 138*, 518–530.

Kanet, J. (1981). Minimizing the average deviation of job completion times about a common due date. *Naval Research Logistics Quarterly, 28*, 643–651.

Kanet, J., & Sridharan, V. (2000). Scheduling with inserted idle time: problem taxonomy and literature review. *Operations Research, 48*, 99–110.

Kim, H., & Yano, C. (1994). Minimizing mean tardiness and earliness in single-machine scheduling problems with unequal due dates. *Naval Research Logistics, 41*, 913–933.

Kise, H., & Ibaraki, T. (1983). On Balut's algorithm and NP-completeness for a chance-constrained scheduling problem. *Management Science, 29*, 384–388.

Lee, C., & Choi, J. (1995). A genetic algorithm for job sequencing problems with distinct due dates and general early–tardy penalty weights. *Computers & Operations Research, 22*, 857–869.

Li, G. (1997). Single machine earliness and tardiness scheduling. *European Journal of Operational Research, 96*, 546–558.

Liaw, C. (1999). A branch and bound algorithm for the single machine earliness and tardiness scheduling problem. *Computers & Operations Research, 26*, 679–693.

Author's personal copy

Mazmanyan, L., Ohanian, V. & Trietsch, D. (2008). The lognormal central limit theorem for positive random variables. Working Paper. Reproduced within the Research Notes for Appendix A. <http://mba.tuck.dartmouth>.

M'Hallah, R. (2007). Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. *Computers & Operations Research, 34*, 3126–3142.

Ow, P., & Morton, T. (1989). The single machine early/tardy problem. *Management Science, 35*, 177–191.

Panwalkar, S., Smith, M., & Seidmann, A. (1982). Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research, 30*, 391–399.

Portougal, V., & Trietsch, D. (2006). Setting due dates in a stochastic single machine environment. *Computers & Operations Research, 33*, 1681–1694.

Rothkopf, M. (1966). Scheduling with random service times. *Management Science, 12*, 707–713.

Sarin, S., Erdel, E., & Steiner, G. (1991). Sequencing jobs on a single machine with a common due date and stochastic processing times. *European Journal of Operational Research, 27*, 188–198.

Seidmann, A., Panwalkar, S., & Smith, M. (1981). Optimal assignment of due-dates for a single processor scheduling problem. *International Journal of Production Research, 19*, 393–399.

Seo, D., Klein, C., & Jang, W. (2005). Single machine stochastic scheduling to minimize the expected number of tardy jobs using mathematical programming models. *Computers & Industrial Engineering, 48*, 153–161.

Shabtay, D. (2008). Due date assignments and scheduling a single machine with a general earliness/tardiness cost function. *Computers & Operations Research, 35*, 1539–1545.

Sidney, J. (1977). Optimal single-machine scheduling with earliness and tardiness penalties. *Operations Research, 25*, 62–69.

Soroush, H. (1999). Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operations Research, 113*, 450–468.

Soroush, H., & Fredendall, L. (1994). The stochastic single machine scheduling problem with earliness and tardiness costs. *European Journal of Operational Research, 77*, 287–302.

Wu, C., Brown, K., & Beck, J. (2009). Scheduling with uncertain durations: Modeling β-robust scheduling with constraints. *Computers & Operations Research, 36*, 2348–2356.

Xia, Y., Chen, B., & Yue, J. (2008). Job sequencing and due date assignment in a single machine shop with uncertain processing times. *European Journal of Operational Research, 184*, 63–75.

Yano, C., & Kim, Y. (1991). Algorithms for a class of single-machine tardiness and earliness problems. *European Journal of Operational Research, 52*, 167–178.