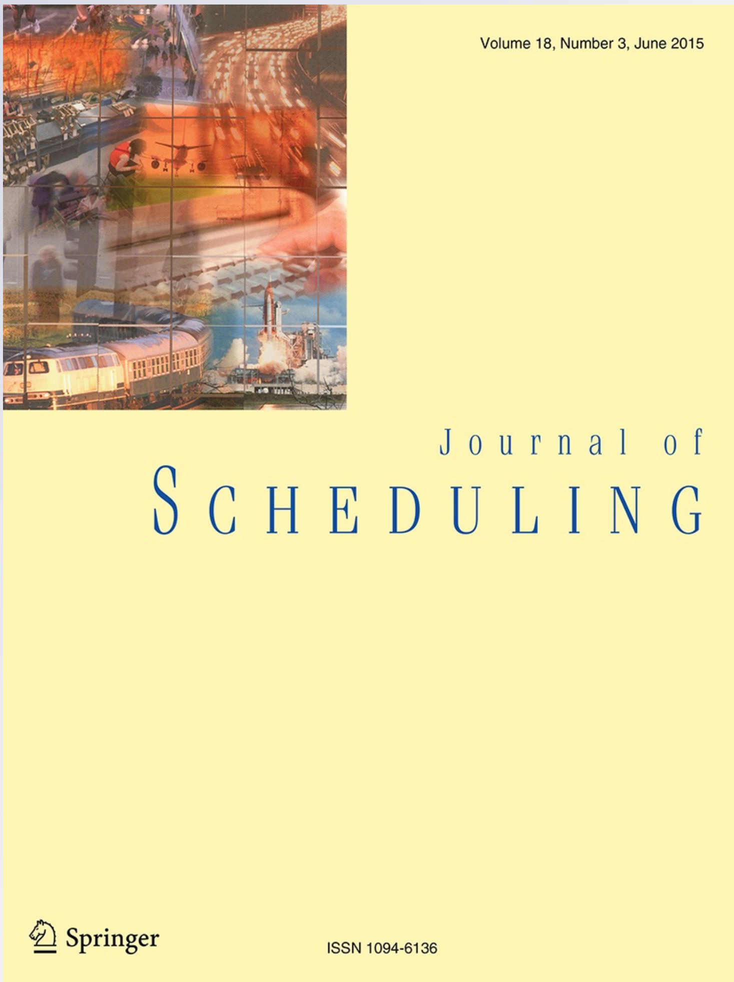
# *Trading off due-date tightness and job tardiness in a basic scheduling model*

**Kenneth R. Baker & Dan Trietsch**

**Journal of Scheduling**

ISSN 1094-6136
Volume 18
Number 3

J Sched (2015) 18:305-309
DOI 10.1007/s10951-014-0394-9




Springer

J Sched (2015) 18:305–309
DOI 10.1007/s10951-014-0394-9

# Trading off due-date tightness and job tardiness in a basic scheduling model

**Kenneth R. Baker · Dan Trietsch**

Received: 22 April 2014 / Accepted: 6 August 2014 / Published online: 23 August 2014


**Abstract** We consider a scheduling problem in which the criterion for assigning due dates is to make them as tight as possible, while the criterion for sequencing jobs is to minimize their tardiness. Because these two criteria conflict, we examine the trade-off between the tightness of the due dates and the tardiness of the jobs. We formulate a version of this trade-off in the case of a stochastic single-machine model with normally distributed processing times. Using lower bounds and dominance properties to curtail the enumeration, we develop a branch-and- bound procedure that is capable of solving large versions of the problem, and we report the results of computational experiments involving several hundred test problems.

**Keywords** Sequencing · Stochastic scheduling · Tardiness · Due-date assignment · Single machine

## 1 Introduction

Meeting due dates is one of the fundamental objectives in scheduling. In some cases, due dates are given, while in other cases due dates are decisions made by a scheduling agent. This latter case, sometimes called the due-date assignment model, has given rise to several research efforts, surveyed, for example, by Cheng and Gupta (1989) and by Gordon et al. (2002). When due dates are decisions, two considerations must be balanced. If due dates are very loose, it may possible to complete all required work on time, but the resulting schedule may be inefficient. On the other hand, if due dates are very tight, the schedule may be efficient, but due dates may be missed and jobs may be tardy. Thus, it is appropriate to seek a balance between tight due dates and job tardiness.

In this paper, we investigate the balancing of due-date tightness and job tardiness in the context of the basic stochastic sequencing model. Our analysis builds on the basic single-machine scheduling model (see, for example, Baker and Trietsch 2009a). The key parameters in the model include the processing time for job $j$ ($p_j$); the due dates ($d_j$) are decision variables. In the actual schedule, job $j$ completes at time $C_j$, and its tardiness is defined by $T_j = \max\{0, C_j - d_j\}$. The trade-off between due-date tightness and job tardiness is captured by an objective function that combines a due-date component with a tardiness component:

$$G(d) = \sum\nolimits_{j=1}^{n} d_j + \gamma \sum\nolimits_{j=1}^{n} T_j \tag{1}$$

We write $G(d)$ as shorthand for $G(d_1, d_2, \ldots, d_n)$. We can also rewrite (1) as a total of job-by-job contributions to the overall objective:

$$G(d) = \sum\nolimits_{j=1}^{n} G_j(d_j) = \sum\nolimits_{j=1}^{n} (d_j + \gamma T_j) \tag{2}$$

In this expression, the term $G_j(d_j)$ represents the contribution of job $j$. The parameter $\gamma$ determines the weight given to total tardiness relative to the sum of the due dates. As we shall see, the choice of $\gamma$ determines the optimal service level for the jobs.

In the deterministic single-machine model, it is not difficult to minimize $G(d)$. Note that $d_j + \gamma T_j = \max\{d_j, (1 - \gamma)d_j + \gamma C_j\}$, so that, for a fixed sequence, if $\gamma \le 1$ then

K. R. Baker (✉)
Tuck School, Dartmouth College, Hanover, NH 03755, USA
e-mail: ken.baker@dartmouth.edu

D. Trietsch
College of Engineering, American University of Armenia, Yerevan, Armenia
e-mail: dan.trietsch@gmail.com

$G(d)$ is minimized with $d_j = 0$ (assuming due dates are constrained to be non-negative). On the other hand, if $\gamma > 1$ then $G(d)$ is minimized with $d_j = C_j$. In both cases $G(d)$ equals the sum of the completion times, which is minimized by sequencing jobs according to shortest processing time (SPT).

The stochastic version of this problem presents more of a challenge. We assume that the processing times are random variables, giving rise to a stochastic scheduling problem in which the objective is to minimize the expected value of the function in (2), which we can write as $\mathrm{E}[G(d)]$. As in the deterministic counterpart, we know that when $\gamma \leq 1$ the due dates should be set to zero, so in what follows we assume $\gamma > 1$. We also assume that the processing times $p_j$ are independent and follow a normal distribution with mean $\mu_j$ and standard deviation $\sigma_j$.

We use the normal because it is familiar and plausible for many scheduling applications. In addition, the use of the normal is convenient because it implies that completion times follow a normal distribution as well. The normal assumption is common in the literature, e.g., Balut (1973), Sarin et al. (1991), Soroush and Fredendall (1994), Seo, et al. (2005), Cai and Zhou (1997), Soroush (1999) , Jang (2002), Portougal and Trietsch (2006), and Wu et al. (2009). We note that, when used as a model for processing times, the normal distribution must yield positive values. For that reason, it is consistent to assume that the mean is at least 3-4 times the standard deviation so that negative samples would virtually never occur.

In place of the normal distribution, we could have assumed that processing times follow some other family of distributions. In fact, for any distribution of processing times, the completion times for jobs "later" in the schedule will be approximately normal, by the Central Limit Theorem. However, by assuming that the normal distribution applies, our analysis will be exact, and we do not have to address the closeness of the approximations.

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

$$H(d) = \mathrm{E}[G(d)] = \sum\nolimits_{j=1}^{n} (d_j + \gamma E[T_j]) \tag{3}$$

In this form, each job contributes $H_j(d_j) = d_j + \gamma \mathrm{E}[T_j]$ to the total. The problem consists of finding a set of due dates and a sequence of the jobs that produce the minimum value of $H(d)$ in (3).

Most of the existing research articles on due-date assignment rely on deterministic analysis (Gordon et al. 2002). Relatively little work addresses due-date assignment in a stochastic environment with tardiness as a criterion. Perhaps the closest related research involving stochastic analysis is due to Wein (1991), who considers due-date assignment in a queueing context and investigates the combined effect of due-date assignment rules and sequencing rules. However, in that work, sequencing decisions are restricted to priority sequencing policies rather than detailed sequencing decisions, and the trade-off is formulated by adopting an aggregate tardiness constraint and minimizing the average due date. Soroush (1999) and Portougal and Trietsch (2006) consider due-date assignment in a more traditional stochastic scheduling context but deal with heuristic procedures for different objective functions. In this paper, we investigate the optimal balancing of due-date tightness and job tardiness in a stochastic environment, a combination for which an optimizing procedure has not been offered in the scheduling literature.

## 2 Analysis of the due-date assignment problem

To analyze the model, 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. In formal terms, let $B_j$ denote the set of jobs preceding job $j$ in the schedule. Then $C_j$ follows a normal distribution with mean $\mathrm{E}(C_j) = \sum_{i \in B_j} \mu_i + \mu_j$ and variance var $(C_j) = \sum_{i \in B_j} \sigma_i^2 + \sigma_j^2$. To streamline the notation, we write $\mathrm{E}(C_j) = \mu_{B_j} + \mu_j$ and var $(C_j) = \sigma_{B_j}^2 + \sigma_j^2$. Once we know the properties of the random variable $C_j$, we can determine the optimal choice of $d_j$. Let $k = [d_j - \mathrm{E}(C_j)]/[\mathrm{var}(C_j)]^{1/2}$ be the standardized due date and use asterisks to denote optimal values.

**Theorem 1** *Given the mean $E(C_j)$ and the variance $var(C_j)$ of the normal distribution for $C_j$, the optimal choice of the due date $d_j$ is given in standardized form by*

$$\Phi(k^*) = \frac{\gamma - 1}{\gamma}$$

Here $\Phi(\cdot)$ denotes the standard normal cumulative distribution function (cdf), or equivalently, the probability that job $j$ completes on or before its due date. (This probability is also called the *service level* for job $j$.)

In other words, the optimal service level for job $j$ is given by the ratio $(\gamma - 1)/\gamma$. This result is a version of the well known *critical fractile rule*, sometimes also called the *newsvendor property* of inventory theory. Theorem 1 implies that the appropriate choice for the due date of job $j$ is

$$d_j = E(C_j) + k^*[var(C_j)]^{1/2} = \mu_{B_j} + \mu_j + k^*(\sigma_{B_j}^2 + \sigma_j^2)^{1/2} \tag{4}$$

In this expression, the due date $d_j$ depends on the previous jobs in sequence via the set $B_j$, and our objective is summarized in (3). From the algebra of newsvendor analysis, we can rewrite (3) by incorporating the optimal choice of $d_j$. The objective becomes

$$H(d^*) = \sum\nolimits_{j=1}^{n} \left[\mu_{B_j} + \mu_j + \gamma\varphi(k^*)(\sigma_{B_j}^2 + \sigma_j^2)^{1/2}\right], \tag{5}$$

where $\varphi(k_j^*)$ is the standard normal probability density function corresponding to the optimal service level of Theorem 1.

## 3 Analysis of the sequencing problem

Finding a job sequence to minimize the objective function in (5) would appear to be a challenging sequencing problem even in the case of normal processing times. To accelerate the computations, we can exploit various combinatorial techniques, as described in the following subsections.

### 3.1 Pairwise interchanges

One helpful technique is the evaluation of an adjacent pairwise interchange (API). Suppose that job $j$ appears immediately after job $i$ somewhere in the sequence and consider the conditions under which it would be better to interchange the two jobs. The completion times of jobs not involved in the interchange are unaffected by the swap, so the overall objective is improved if and only if the total contribution from jobs $i$ and $j$ is improved. The mean time to process the jobs preceding $i$ and $j$ can be denoted by $\mu_B$ and the variance of that time by $\sigma_B^2$. Also, for convenience, we introduce the notation $\theta = \gamma\varphi(k^*)$. For the sequence $i - j$, the total contribution is

$$z(i, j) = (\mu_B + \mu_i) + \theta(\sigma_B^2 + \sigma_i^2)^{1/2} + (\mu_B + \mu_i + \mu_j) + \theta(\sigma_B^2 + \sigma_i^2 + \sigma_j^2)^{1/2}]$$

The expression for $z(j, i)$, for the sequence $j - i$, is similar, and the change in the objective due to the interchange is

$$g_{ij}(\sigma_B^2) = z(j, i) - z(i, j) = \mu_j - \mu_i + \theta(\sigma_B^2 + \sigma_j^2)^{1/2} - \theta(\sigma_B^2 + \sigma_i^2)^{1/2} \tag{6}$$

Therefore, the interchange is undesirable (and the $i - j$ order is at least as good) as long as $g_{ij}(\sigma_B^2) \geq 0$, which we refer to as the *API condition*.

**Theorem 2** *A necessary condition for a sequence to be optimal is that every pair of adjacent jobs $i$ and $j$ (with $j$ following $i$) satisfies the API condition, $g_{ij}(\sigma_B^2) \geq 0$.*

Unfortunately, the API condition does not lead to a universal rule for determining whether $j$ should follow $i$ because the condition depends on $\sigma_B^2$, and therefore, on the jobs making up the partial sequence that precedes $i$ and $j$. Nevertheless, the API condition can be used to eliminate partial sequences, and thereby curtail an enumerative search.

### 3.2 Static dominance

We refer to the API condition as a *dynamic* property because its result depends on $\sigma_B^2$, and therefore, on the jobs making up the partial sequence that precedes $i$ and $j$. It is also possible to develop *static* dominance properties that hold for any partial sequence that precedes $i$ and $j$. Furthermore, the two properties stated below are global dominance properties, in the sense that they can show that job $i$ precedes job $j$ in an optimal solution even if some other jobs are scheduled between them.

**Corollary 3** *If $\mu_j - \mu_i \geq \theta\sigma_i$ then job $i$ dominates job $j$.*

*Proof* For two adjacent jobs, we revisit (6), and observe that $g_{ij}(\sigma_B^2)$ is as small as possible when $\sigma_j^2 = 0$, and in that case when $\sigma_B = 0$. Substituting these values into (6), we find that $g_{ij}(\sigma_B^2) = \mu_j - \mu_i - \theta\sigma_i$. Therefore, Theorem 2 holds (and $i$ dominates job $j$) if $\mu_j - \mu_i - \theta\sigma_i \geq 0$. To extend the result when $m$ jobs are scheduled between jobs $i$ and $j$, observe that by exchanging the jobs we increase the objective by $(\mu_j - \mu_i)$ a total of $(m + 1)$ times and reduce it by less than $\theta\sigma_i$ for each of the increases.

**Corollary 4** *For two jobs $i$ and $j$, if $\mu_i \leq \mu_j$ and $\sigma_i \leq \sigma_j$ then job $i$ dominates job $j$.*

*Proof* A pairwise interchange of jobs $i$ and $j$ justifies the result.

The static dominance conditions in Corollaries 3 and 4 can be evaluated once at the outset, and then applied as needed during the generation and evaluation of partial sequences. Thus, if we are augmenting a partial sequence and we find that job $i$ dominates job $j$, while neither appears in the partial sequence, then we need not consider the augmented partial sequence constructed by appending job $j$ next.

### 3.3 Lower bounds

As a general framework, we can build a branch-and-bound (B&B) approach, curtailing an enumerative search when a lower bound indicates that a partial sequence cannot lead to an optimum. Suppose we have a partial sequence of the jobs, denoted $\pi$, and we wish to compute a lower bound on the value of the objective function (5) that can be obtained by completing the sequence. The contribution to the objective function from the jobs in $\pi$ has presumably already been computed. Let $\pi'$ denote the set of unscheduled jobs. In the set $\pi'$, we take the set of means $\mu_j$ in smallest first order and, separately, the set of standard deviations $\sigma_j$ 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 means and standard deviations. Next we calculate each fictitious job's contribution to the objective and add it to the component for the partial sequence. This total is a lower bound on the value that could be achieved by completing the partial sequence in the best possible way. (A formal proof follows a standard pairwise interchange argument.) Thus, 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 known sequence, we know that the partial sequence can never lead to a solution better than the known sequence.

### 3.4 Initial solution

Given that we are using a B&B algorithm, it makes sense to begin by finding a good initial solution that can be effective at fathoming partial sequences with relatively few jobs. For this purpose, we implemented a greedy procedure. At any stage of this procedure, we have a partial sequence on hand, and we wish to determine which currently unscheduled job should come next in sequence. We test each unscheduled job to determine how much it increases the objective function, when appended to the current partial sequence, and we select the job that produces the smallest increase. Then, having appended that job to the partial sequence, we proceed to evaluate the next open position in the sequence, stopping when all positions are filled. The procedure, therefore, makes a single pass and fills the sequence positions from first to last.

## 4 Computational experience

The solution algorithm was programmed in VBA and tested on a variety of problem instances. For a given problem size $n$, we created a problem instance by sampling the $\mu$-values from a discrete uniform distribution between 10 and 100. Once $\mu_j$ was obtained, $\sigma_j$ was sampled from a uniform distribution between 0.10 and 0.25 $\mu_j$. (Thus, the mean was between 4 and 10 times the standard deviation, so the chances of encountering a negative value for a processing time were negligible.) In the basic testbed, we took $\gamma = 10$, which corresponds to a critical fractile of 90 %. For each value of $n$, we generated 100 problem instances and found solutions using the curtailed enumeration algorithm with the components described in the previous section.

The algorithm proved to be quite effective. For example, using a 2.9 GHz processor, it is capable of solving 200 job problems in less than half a minute of cpu time. Table 1 summarizes the solution times, in seconds, for various problem sizes.

**Table 1** Mean, median, and maximum solution times for various problem sizes

| Problem Size | Mean Time (s) | Median Time (s) | Maximum Time (s) |
|---|---|---|---|
| 50 | 0.02 | 0.02 | 0.05 |
| 100 | 0.34 | 0.32 | 0.93 |
| 150 | 3.03 | 2.64 | 8.39 |
| 200 | 18.09 | 14.95 | 56.22 |
| 250 | 81.82 | 107.71 | 245.17 |
| 300 | 289.20 | 254.05 | 6233.59 |

**Table 2** Effect of $\gamma$ on average run times for 100 job problems

| $\gamma$ | 10 | 20 | 50 | 100 | 160 | 200 |
|---|---|---|---|---|---|---|
| Service level | 0.900 | 0.950 | 0.980 | 0.990 | 0.994 | 0.995 |
| Seconds | 0.343 | 0.424 | 0.499 | 0.558 | 0.555 | 0.557 |

The main conclusion from the data in Table 1 is that problems with up to a few hundred jobs can be solved in relatively modest run times. Relatively few sequencing problems of 200–300 jobs can be solved to optimality in a matter of seconds or minutes, making this problem (or perhaps this solution algorithm) notable in the literature on sequencing. A second observation is that median times do not differ a great deal from the mean times for a given problem size. This property is atypical for B&B algorithms, which often exhibit variations in solution times that span two orders of magnitude.

An additional experiment was designed to investigate the effect of varying $\gamma$ on solution times. From the base case of $\gamma = 10$, which corresponds to a service level of 90 %, we revisited the 100 job instances and ran the algorithm for several larger values of $\gamma$, as summarized in Table 2. The results suggest that solution times increase with $\gamma$ and eventually level off. In the cases shown, the run times level off around $\gamma = 100$, with solution times that are roughly 60 % longer than those in the base case.

## 5 Summary and conclusions

We posed a single-machine, stochastic scheduling problem in which the objective is to find an optimal trade-off between due-date tightness and job tardiness. At the heart of the optimal trade-off is a decision rule that sets a target service level for each job based on the factor ($\gamma$) that balances tightness and tardiness. In addition, the jobs must be sequenced optimally. We described an algorithm for this problem that is capable of solving problems with as many as 300 jobs in just minutes of cpu time.

The key role played by a service level target identifies this problem as aligned with others in a class of *safe scheduling* problems (Baker and Trietsch 2009b). In fact, this trade-off was identified in the initial paper on safe scheduling, but an optimizing algorithm has not been developed until now. As stochastic scheduling receives increasing attention from researchers, safe-scheduling analysis, such as highlighted in this research, may provide both practical and theoretical insights. The analysis in this paper is intended to further that line of research and expand our knowledge about stochastic scheduling. In particular, we showed that the problem can be addressed by approaches usually reserved for deterministic models, such as adjacent pairwise interchanges, dominance properties, and lower bounds. Thus, safe-scheduling models should be of interest to researchers who may not have addressed stochastic models in the past.

## References


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.

Cheng, T., & Gupta, M. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research*, *38*, 156–166.

Cai, X., & Zhou, S. (1997). Scheduling stochastic jobs with asymmetric earliness and tardiness penalties. *Naval Research Logistics*, *44*, 531–557.

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.

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

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

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.

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.

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.

Wein, L. M. (1991). Due-date setting and priority sequencing in a multiclass M/G/1 queue. *Management Science*, *37*, 834–850.

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