

The Effectiveness of the Longest Delivery Time Rule for the Flow Shop Delivery Time Problem

Philip Kaminsky

Department of Industrial Engineering and Operations Research, University of California, Berkeley, 4135 Etcheverry Hall, Berkeley, California 94720-1777

Received 11 January 2001; accepted 24 August 2002

DOI 10.1002/nav.10054

Abstract: In the flow shop delivery time problem, a set of jobs has to be processed on m machines. Every machine has to process each one of the jobs, and every job has the same routing through the machines. The objective is to determine a sequence of the jobs on the machines so as to minimize maximum delivery completion time over all the jobs, where the delivery completion time of a job is the sum of its completion time, and the delivery time associated with that job. In this paper, we prove the asymptotic optimality of the Longest Delivery Time algorithm for the static version of this problem, and the Longest Delivery Time among Available Jobs (LDTA) algorithm for the dynamic version of this problem. In addition, we present the result of computational testing of the effectiveness of these algorithms. © 2003 Wiley Periodicals, Inc. *Naval Research Logistics* 50: 257–272, 2003.

Keywords: scheduling; one-line algorithms

1. INTRODUCTION

In the *flow shop delivery time problem*, a set of jobs, each consisting of m operations, must be sequentially processed on m machines. Each machine can process at most one job at a time, and a job can only be processed on one machine at a time. The jobs have to be processed on each of the machines without preemption. In addition to a processing time on each machine, each job has two associated parameters: a release time and a delivery time. No job can be processed before its release time. The delivery-completion time of a job is its completion time on the final machine, machine m , plus its delivery time. We note that while each machine can handle at most one job at a time, any number of jobs can be *delivered* in parallel. The *flow shop delivery time problem* involves determining a sequence of the jobs on the machines which minimizes the maximum delivery-completion time over all jobs. This problem was shown to be NP-hard by Lenstra, Rinnooy Kan, and Brucker [8], even in the single machine case. If all release times are equal, the single machine version of the problem is trivially solved by sequencing the jobs in order of decreasing delivery time. However, if all release times are equal, even the two machine *flow shop* problem is NP-hard (see Pinedo [13]).

As we discuss in detail in Section 1 below, the flow shop delivery time completion problem is closely related to the flow shop maximum lateness problem. In the flow shop maximum

lateness problem, jobs are assigned a due date rather than a delivery time, and each job has an associated lateness, equal to its completion time minus its due date. In this case, the objective is to sequence the jobs in order to minimize the maximum lateness. For any *flow shop delivery time problem* instance, an instance of the *flow shop maximum lateness* problem can be constructed which has the same optimal sequence, and vice versa.

Although there has been little previous work on the *flow shop delivery time problem* or on the *flow shop maximum lateness problem*, there has been a fair amount of research on the corresponding single machine problems. As mentioned above, it is well known that the version of the single machine delivery time problem *without release dates*, i.e., when all jobs are available for processing at the same time, is trivially solved using the Longest Delivery Time (LDT) sequence, where jobs are scheduled in decreasing order of their due times. On the other hand, the single machine problem with release date was shown to be NP-Complete, by Lenstra, Rinnooy Kan, and Brucker [8], which motivated researchers to develop branch and bound based algorithms for the problem. Most of these papers, including the work of McMahon and Florian [10], Carlier [1], and Nowicki and Zdrzalka [11], are presented in the context of the maximum lateness problem.

In addition, researchers have developed, analyzed, and tested various *on-line* and *off-line* heuristics for scheduling problems. In this context, on-line scheduling algorithms sequence jobs at any time using only information pertaining to jobs which have been released by that time. In contrast, off-line algorithms may use information about jobs which will be released in the future. It is clear that in order to find the optimal solution to the delivery time (or maximum lateness) problem with release dates, it may be necessary to insert idle time into the schedule while waiting for certain jobs to arrive. Since off-line algorithms can utilize information about future arrivals in order to decide whether or not to insert idle time in the schedule, it seems likely that the most effective off-line algorithms will perform better than the most effective on-line algorithms for this model.

Kise, Ibaraki, and Mine [7] show that a simple on-line heuristic, LDTA, has a worse case bound of no more than two. In an LDTA sequence, when a job completes processing, the remaining *available* job with the longest delivery time is selected for processing. That is, if L^{LDTA} represents the objective obtained by sequencing jobs in LDTA order, and L^* is the optimal objective value, then $\frac{L^{\text{LDTA}}}{L^*} \leq 2$ for all instances. Potts [14] and Nowicki and Smutnicki [11] present iterative extensions of this approach with worst case bounds of $\frac{3}{2}$. Hall and Shmoys [3] present a set of polynomial approximation schemes for the problem with a worst case bound of $1 + \frac{1}{k}$ and running time of $O(n \log n + nk^{O(1)k^2})$, for $k = 1, 2, 3, \dots$. Finally, Hoogeveen and Vestjens [4] present an on-line algorithm with worst case bound $(\sqrt{5} + 1)/2$ and prove that no deterministic on-line algorithm can do better.

As mentioned above, it is well known that the LDTA heuristic performs quite well for the single machine version of the problem. This motivates the primary goal of this paper, which is to characterize the performance of this LDTA heuristic for the flow shop version of this problem. To do this, we will employ the tools of probabilistic analysis. Recently, researchers have had some success using probabilistic analysis to characterize both the structure of optimal solutions to certain flow shop scheduling problems, and to characterize the effectiveness of simple on-line heuristics for those problems. Ramudhin et al. [15] analyze the 2-machine flow shop model, when the objective is to minimize the makespan. They characterize the expected behavior of a variety of strategies including optimal and approximate algorithms. Kaminsky and Simchi-Levi [5] characterize the asymptotically optimal solution for the static flow shop weighted average completion time problem, and Kaminsky and Simchi-Levi [6] and Xia,

Shanthikumar, and Glynn [17] characterize the effectiveness of the well-known Shortest Processing Time rule for the flow shop average completion time problem. Liu and Simchi-Levi [9] extend these results to the weighted flow shop average completion time problem with release dates. In this paper, we utilize the tools of probabilistic analysis on a new class of scheduling problem—one in which the objective is *based on a single job* which completes somewhere in the sequence of jobs, rather than on the sum of characteristics associated with all jobs, as in the completion time case, or the completion time of the final job, as in the makespan case.

In the next section we provide a detailed description of the model analyzed together with our main results.

2. THE MODEL AND THE MAIN RESULT

Consider a set of n jobs that have to be processed on m machines. Job i , $i = 1, 2, \dots, n$, has a processing time t_i^l on machine l , $l = 1, 2, \dots, m$. Associated with each job i , there is a delivery time $q_i \geq 0$, $i = 1, 2, \dots, n$ which represents additional delivery time that the job requires after it is processed.

Each job must be processed without preemption on each machine sequentially. That is, each job must be processed on machines 1 through m in that order. Jobs are available for processing at time zero, and there is unlimited intermediate storage between successive machines. The objective is to determine a *schedule*, or sequence of jobs, such that the maximum delivery-completion time over all of the jobs is minimized. Let c_i be the completion time of job i on the final machine in a particular schedule. The delivery-completion time of that job, L_i is defined as follows:

$$L_i = c_i + q_i.$$

The objective value of this problem is thus

$$L = \max_{1 \leq i \leq n} L_i.$$

We call this problem, the static flow shop completion time problem, Problem P and use L^* to denote its optimal objective function value. That is, L^* is the minimum possible maximum delivery-completion time of all jobs in Problem P . Similarly, given a heuristic H for this problem, we use L^H to denote the maximum delivery-completion time in the schedule produced by H . As pointed out in the Introduction, this problem is NP-hard, even in the 2-machine case.

For the probabilistic analysis below, we generate a sequence of problem instances with increasing numbers of jobs as follows. The processing times are drawn from an identical and bounded distribution with nonzero density $\phi(\cdot)$, defined on the interval $(0, 1]$. The delivery times are assumed to be nonzero random variables, with no additional restrictions.

Specifically, we consider the following heuristic: Sequence the jobs in nonincreasing order of delivery times. At the completion time of any job on the first machine, begin processing the next job in the nonincreasing delivery time sequence. On all other machines, process the jobs without preemption as soon as possible in the order that they arrive at that machine. Note that this means that all jobs will be processed in decreasing delivery time order on all machines. We call this heuristic the *Longest Delivery Time* (LDT) heuristic, and

denote the maximum delivery-completion time over all of the jobs when they are scheduled using the heuristic L^{LDT} .

As observed by Hall [2], this problem is equivalent to the problem of minimizing maximum lateness in a flow shop in the following sense. Given a problem equivalent to the one defined above but with due dates d_i associated with each job rather than delivery times, the lateness of job i is defined as follows:

$$L_i = c_i - d_i.$$

The objective of the problem of minimizing maximum lateness is thus to determine a sequence which minimizes the objective L , where in this case

$$L = \max_i L_i.$$

Observe that for any sequence of jobs, if some constant K is subtracted from the due dates of each of the jobs, the maximum lateness of the sequence increases by exactly K , and in particular the optimal sequence remains the same. If a large enough constant is subtracted so that all due dates are negative, then this problem is exactly equivalent to the delivery time problem where delivery times $q_i = -d_i$, since

$$L = \max_i \{c_i - d_i\} = \max_i \{c_i + q_i\}.$$

Note that in the delivery time model, the delivery-completion times of all jobs are positive, which facilitates our subsequent analysis.

In the analysis that follows, we relate the LDT objective to that of a lower bound on the problem. Note that in the original model, each job can only be processed on a particular machine l after it has been processed on machines 1 through $l - 1$. In this lower bound, we eliminate these precedence constraints (between operations of the same job) so that each job can be processed independently on each of the machines, without regard to when it is processed on the other machines. It is well known (see Pinedo [13]) that the single machine problem is optimally solved by sequencing jobs in LDT order.

Consider the solution to this lower bound (with relaxed precedence constraints) in which each machine is sequenced independently in LDT order. (We will in general indicate that a random variable or objective refers to this lower bound by using an underscore). Let c_j^l represent the completion time of job j on machine l for the case where precedence constraints are relaxed, let $\underline{L}_l^{\text{LDT}} = \max_{1 \leq j \leq n} \{c_j^l + q_j\}$, and let

$$\underline{L}^* = \max_i \underline{L}_i^{\text{LDT}}.$$

Clearly, \underline{L}^* is the optimal objective value for this lower bound, so that

$$\underline{L}^* \leq L^* \leq L^{\text{LDT}}.$$

In this paper, we prove the following result:

THEOREM 1: Consider the m -machine flow shop defined above. Let the processing times for job i , $t_i^1, t_i^2, \dots, t_i^m$, $i = 1, 2, \dots, n$, be independent random variables having the same continuous and bounded distribution $\phi(\cdot)$ with nonzero density defined on $(0, 1]$, and let the delivery times for job i , $i = 1, 2, \dots, n$, q_i , be nonzero random variables. Then for a series of randomly generated instances of Problem P of increasing size, with probability 1 we have

$$\lim_{n \rightarrow \infty} \frac{L^*}{n} = \lim_{n \rightarrow \infty} \frac{L^*}{n} = \lim_{n \rightarrow \infty} \frac{L^{\text{LDT}}}{n}.$$

Next, consider an extension of the model described above in which each job i , $i = 1, \dots, n$ has, in addition to the parameters listed above, a nonnegative release time r_i before which a job cannot begin processing on the first machine. We call this Problem P' . For the probabilistic analysis that follows, for each job i , the release times r_i and the delivery time q_i are required to be nonzero random variables with no further restrictions.

We will employ the same notation as in the static case described above—it will be clear from the context of the paper which model we are referring to—and focus on permutation schedules for this problem. In a permutation schedule, jobs are processed on each of the machines in the same sequence. This implies that once the first machine is scheduled, jobs are processed on each of the other machines in a first-come, first-served manner. We consider the following heuristic for this problem. Each time a job has completed processing on machine 1, we consider the set of jobs which have been released by that time and have not yet been processed on machine 1, and process the job with the longest delivery time. If no jobs are available, machine 1 remains idle until a job arrives. As in the optimal solution, jobs are processed on subsequent machines in the same sequence as on machine 1. We call this heuristic the *Longest Delivery Time among Available Jobs* (LDTA) heuristic.

In our analysis, we will relate the LDTA solution to the solution of a lower bound constructed exactly the same way as in the no-release-date case, except that a job cannot start processing on any machine before its release date. We note that although this single machine problem is itself NP-hard, it is well known (see Potts [14]) that the LDTA solution to this single machine problem can be no more than a constant bigger than the optimal solution, where the constant is equal to the processing time of a particular job in the sequence. Thus, finding a “good” solution to this problem is fairly easy. Once again, we denote the optimal solution to this lower bound \underline{L}^* .

Below, we use some of the insight generated by our analysis of the static problem, and show that in the dynamic problem, the following result holds:

THEOREM 2: Consider the m -machine flow shop defined above. Let the processing times for job i , $t_i^1, t_i^2, \dots, t_i^m$, $i = 1, 2, \dots, n$, be independent random variables having the same continuous and bounded distribution $\phi(\cdot)$ with nonzero density defined on $(0, 1]$. In addition, let delivery times q_i and release times r_i be nonnegative random variables. Then for a series of randomly generated instances of Problem P' of increasing size, with probability 1 we have

$$\lim_{n \rightarrow \infty} \frac{\underline{L}^*}{n} = \lim_{n \rightarrow \infty} \frac{L^*}{n} = \lim_{n \rightarrow \infty} \frac{L^{\text{LDTA}}}{n}.$$

We note that Theorem 1 is a special case of Theorem 2. However, as the proof of Theorem 1 is simpler and more intuitive than the proof of Theorem 2, in the next section, Section 2, we

first present the proof of Theorem 1 and then follow this with the proof of Theorem 2, and associated results. We follow this theoretical analysis with the results of a computational analysis of the performance of the LDTA heuristic in Section 3. We observe that LDTA is not only theoretically effective, but also computationally quite effective for this problem.

3. PROOF OF THE MAIN THEOREMS

3.1. Proof of the Static Theorem, Theorem 1

This proof proceeds by bounding the difference between the lower bound on the optimal objective value \underline{L}^* , and the upper bound L^{LDT} . First, we observe that in this case (with all release dates equal), the sequence of jobs will be the same on all machines in the LDT solution, and on each machine in the lower bound. Now, we define I_j^a , $a = 2, \dots, m$, $j = 2, \dots, n$, to be the total idle time incurred in the LDT solution on machine a , between the time the first job starts on that machine and the time job j departs from that machine. Since the sequence of jobs on all machines are the same, if c_j is the completion time of job j in the LDT solution, clearly

$$c_j \geq \underline{c}_j^l, \quad l = 1, 2, \dots, m.$$

In particular, let j^{LDT} represent the job which achieves the maximum delivery-completion time in the LDT solution, let \underline{j}^* represent the job which achieves the maximum delivery-completion time in the lower bound, and let $\underline{c}_{\underline{j}^*}$ represent the completion time of that job on the machine on which it achieves its maximum:

$$\underline{c}_{\underline{j}^*} = \max_{1 \leq l \leq m} \underline{c}_{\underline{j}^*}^l,$$

noting that this is not necessarily machine m .

Recall that $c_{\underline{j}^*}$ is the completion time of job \underline{j}^* in the (upper bound) LDT sequence, and that $\underline{c}_{\underline{j}^*}^m$ represent the completion time of job \underline{j}^* on machine m , the final machine, in the lower bound solution.

Observe that

$$c_{\underline{j}^*} = \underline{c}_{\underline{j}^*}^m + I_{\underline{j}^*}^m,$$

since the sequence is the same on the final machine in both the upper bound and the lower bound. Now, since \underline{j}^* is the job which achieves the maximum in the lower bound, we have

$$\underline{c}_{\underline{j}^*}^m + q_{\underline{j}^*} \leq \underline{c}_{\underline{j}^*} + \underline{q}_{\underline{j}^*}$$

and

$$\underline{c}_{\underline{j}^*} + \underline{q}_{\underline{j}^*} \leq c_{\underline{j}^*} + \underline{q}_{\underline{j}^*}.$$

Thus,

$$c_{\underline{j}^*} + q_{\underline{j}^*} \leq \underline{c}_{\underline{j}^*}^m + q_{\underline{j}^*} + I_n^m \leq c_{\underline{j}^*} + \underline{q}_{\underline{j}^*} + I_n^m \leq c_{\underline{j}^*} + q_{\underline{j}^*} + I_n^m.$$

Thus, we conclude that:

$$L^{\text{LDT}} \leq \underline{L}^* + I_n^m \leq L^* + I_n^m \leq L^{\text{LDT}} + I_n^m. \tag{1}$$

Next, we observe that Kaminsky and Simchi-Levi [6] have shown that:

LEMMA 1: For every $j, j \geq 2$, and $a = 1, 2, \dots, m$, we have

$$I_j^a = \max \left\{ 0, \max_{l=2,3,\dots,j} \left(I_l^{a-1} + \sum_{k=2}^l (t_k^{a-1} - t_{k-1}^a) \right) \right\}.$$

Xia, Shanthikumar, and Glynn [16] have shown (their Theorem 6.2) that:

LEMMA 2 (Xia, Shanthikumar, and Glynn [17]): For every $a = 2, 3, \dots, m$, under the conditions stated previously (and, in fact, under somewhat weaker conditions), almost surely,

$$\lim_{n \rightarrow \infty} \frac{\max_{1 \leq l \leq n} \left| \sum_{k=1}^l (t_k^a - t_k^l) \right|}{n} = 0.$$

We observe that in our analysis the processing times are independent and identically distributed across machines as well as across jobs, and that $\sum_{k=2}^l (t_k^{a-1} - t_{k-1}^a) = \sum_{k=2}^{l-1} (t_k^{a-1} - t_k^a) + t_l^{a-1} - t_1^a \leq \sum_{k=2}^{l-1} (t_k^{a-1} - t_k^a) + 1$, and thus by Lemma 2, we can conclude that the rightmost term in the equation in Lemma 1 grows as $o(n)$. Observing also that there is no idle time on the first machine, and thus

$$I_j^2 = \max \left\{ 0, \max_{l=2,3,\dots,j} \left(\sum_{k=2}^l (t_k^{a-1} - t_{k-1}^a) \right) \right\}.$$

makes it clear that in general, $I_j^a = o(n)$ for any fixed number of machines. Thus, dividing inequality (1) by n and taking the limit as n goes to infinity completes the proof of Theorem 1. \square

Finally, we note that both L^{LDT} and L^* grow as $\Omega(n)$ since they are both bounded from below by the completion time of the final job in the sequence.

2.2. Proof of the Dynamic Theorem, Theorem 2

For this proof, we demonstrate a series of lemmas which together bound the difference between lower bound described above (with relaxed precedence constraints) and the LDTA solution. We note that this is significantly harder than the static case described above for two reasons: First, the sequence on each of the single machine lower bounds may be different, and thus different than the LDTA solution (except for the first machine), and second, the single machine problem in the lower bound is NP-hard. However, if we consider the single machine

version of this problem, it has been shown (see Potts [14]) that, for any LDTA sequence, there is a specific job I called the “interference job” such that:

LEMMA 3:

$$L^{\text{LDTA}} < L^* + t_I < L^* + \bar{P},$$

where L^{LDTA} and L^* refer to the single machine LDTA and optimal objective values, respectively, and \bar{P} is the maximum processing time of jobs in the sequence.

We start by bounding the difference between the completion time of each job j in the flow shop LDTA sequence on the first machine, c_j^1 , and its completion time on any other machine i in the flow shop, c_j^i . Recall that this is a permutation schedule, so the sequence will be the same on all machines. Number the jobs in LDTA order and note that the completion time of job j on machine 1 is equal to the release time of one of the jobs preceding j plus the processing time of all the jobs from that job until job j :

$$c_j^1 = \max_{1 \leq k \leq j} \left\{ r_k + \sum_{l=k}^j t_l^1 \right\}.$$

Similarly, the completion time of job j on machine i , c_j^i , is:

$$c_j^i = \max_{1 \leq k_1 \leq k_2 \leq \dots \leq k_i \leq j} \left\{ r_{k_1} + \sum_{a=k_1}^{k_2} t_a^1 + \sum_{a=k_2}^{k_3} t_a^2 + \dots + \sum_{a=k_i}^j t_a^i \right\}.$$

(We note that the same approach to calculating flow shop completion times is well known for static problems—see, for example, Pinedo [13]—and was used by Lui and Simchi-Levi [9] for a dynamic problem.)

Thus, we can bound the difference between the completion time of job j on machine i , $i = 2, 3, \dots, m$ and on machine 1 as follows:

$$\begin{aligned} c_j^i - c_j^1 &= \max_{1 \leq k_1 \leq k_2 \leq \dots \leq k_i \leq j} \left\{ r_{k_1} + \sum_{a=k_1}^{k_2} t_a^1 + \sum_{a=k_2}^{k_3} t_a^2 + \dots + \sum_{a=k_i}^j t_a^i \right\} - \max_{1 \leq k \leq j} \left\{ r_k + \sum_{l=k}^j t_l^1 \right\} \\ &\leq \max_{1 \leq k_1 \leq k_2 \leq \dots \leq k_i \leq j} \left\{ r_{k_1} + \sum_{a=k_1}^{k_2} t_a^1 + \sum_{a=k_2}^{k_3} t_a^2 + \dots + \sum_{a=k_i}^j t_a^i - \left[r_{k_1} + \sum_{l=k_1}^j t_l^1 \right] \right\} \\ &= \max_{1 \leq k_1 \leq k_2 \leq \dots \leq k_i \leq j} \left\{ \sum_{a=k_1}^{k_2} (t_a^1 - t_a^1) + \sum_{a=k_2}^{k_3} (t_a^2 - t_a^1) + \dots + \sum_{a=k_i}^j (t_a^i - t_a^1) \right\}. \end{aligned} \tag{2}$$

Next, for any instance of this problem P' , consider a related single machine instance in which each job has the same release and delivery times as in the flow shop problem, and

processing times equal to the processing times on the first machine of the flow shop problem. We call this the first machine problem, P'_1 . Sequence this single machine instance in LDTA order, let the resulting objective function be $\underline{L}_1^{\text{LDTA}1}$, and let the job which achieves the maximum delivery completion time be job $j_1^{\text{LDTA}1}$. We note that this is *exactly the same sequence*, with exactly the same completion times, as on the first machine of the flow shop problem. Also, recall that the LDTA sequence is not necessarily optimal for this problem, and let \underline{L}_1^* be the optimal objective value for this single machine problem. Clearly, the optimal solution to this single machine problem is a lower bound on the optimal solution to the flow shop problem.

Also, let j^{LDTA} be the job which achieves the maximum delivery-completion time in the flow shop LDTA solution. Applying Lemma 3 and substituting, we get

$$\begin{aligned}
 L^{\text{LDTA}} - \underline{L}_1^* &\leq \underline{L}^{\text{LDTA}} - (L_1^{\text{LDTA}1} - \bar{P}) \\
 &= c_{j^{\text{LDTA}}}^m + q_{j^{\text{LDTA}}} - c_{j_1^{\text{LDTA}1}}^1 - q_{j_1^{\text{LDTA}1}} - \bar{P} \\
 &\leq c_{j^{\text{LDTA}}}^m + q_{j^{\text{LDTA}}} - c_{j^{\text{LDTA}}}^1 - q_{j^{\text{LDTA}}} - \bar{P} \\
 &= c_{j^{\text{LDTA}}}^m - c_{j^{\text{LDTA}}}^1 - \bar{P} \\
 &\leq \max_{1 \leq k_1 \leq k_2 \leq \dots \leq k_i \leq j^{\text{LDTA}}} \left\{ \sum_{a=k_1}^{k_2} (t_a^1 - t_a^1) + \sum_{a=k_2}^{k_3} (t_a^2 - t_a^1) + \dots + \sum_{a=k_i}^{j^{\text{LDTA}}} (t_a^i - t_a^1) \right\} - \bar{P}. \quad (3)
 \end{aligned}$$

Noting that $\underline{L}_1^* \leq L^* \leq L^{\text{LDTA}}$, taking the limit as n goes to infinity, and noting Lemma 2, we get

$$\lim_{n \rightarrow \infty} \frac{\underline{L}_1^*}{n} \leq \lim_{n \rightarrow \infty} \frac{L^*}{n} \leq \lim_{n \rightarrow \infty} \frac{L^{\text{LDTA}}}{n} \leq \lim_{n \rightarrow \infty} \frac{\underline{L}_1^*}{n},$$

which completes the proof of Theorem 2. \square

We note that, surprisingly, this result indicates that the objective of P'_1 is asymptotically equal to the optimal flow shop objective value. We can generalize this first machine problem to any other machine i , by setting the processing times in the single machine problem to be equal to the processing times on machine i of the flow shop problem, creating problem P'_i . However, the proof above relies on the fact that the sequence is the same on the first machine problem when it is sequenced in LDTA order, and in the LDTA flow shop sequence. Since the lower bound defined in Section 1 is the maximum of all of the single machine lower bounds, it seems reasonable to expect that all of the single machine lower bounds are asymptotically equal to the optimal flow shop objective value.

Indeed, in the following theorem, we show the asymptotic equivalence of the LDTA objective for each of the single machine problems, P'_1 to P'_m . From Lemma 3, it is clear that as long as processing times are bounded, LDTA is asymptotically optimal for the single machine problem. Let $\underline{L}_i^{\text{LDTA}i}$ be the objective value of the LDTA sequence of the single machine problem based on the processing times of machine i in the flow shop problem:

THEOREM 3: Under the conditions described above, for any machine i , $i = 2, 3, \dots, m$, almost surely,

$$\lim_{n \rightarrow \infty} \frac{\underline{L}_1^{\text{LDTA}1}}{n} = \lim_{n \rightarrow \infty} \frac{\underline{L}_i^{\text{LDTA}i}}{n}.$$

PROOF: We begin with an expression for the difference in completion times of a particular job in any two single-machine sequences, as long as the machines *have the same sequence*. Recall that the job can have different processing times on the two machines, but has the same release time and completion time on each machine. First, we observe that the completion time of job j on machine i , c_j^i , can be characterized as follows:

$$c_j^i = \max_k \left\{ r_k + \sum_{l=k}^j t_l^i \right\}$$

Thus,

$$\begin{aligned} c_j^a - c_j^b &= \max_k \left\{ r_k + \sum_{l=k}^j t_l^a \right\} - \max_k \left\{ r_k + \sum_{l=k}^j t_l^b \right\} \\ &\leq \max_k \left\{ r_k + \sum_{l=k}^j t_l^a - r_k - \sum_{l=k}^j t_l^b \right\} \\ &= \max_k \left\{ \sum_{l=k}^j t_l^a - \sum_{l=k}^j t_l^b \right\} \\ &= \max_k \left\{ \sum_{l=k}^j (t_l^a - t_l^b) \right\}. \end{aligned} \tag{4}$$

Next, we use Lemma 3 to characterize the following relationships between the objective value of the LDTA sequence on machine one single-machine problem, and the objective value of the LDTA sequence on any other single-machine problem. In the proof, we will sequence some machine i using the LDTA sequence of some other machine l . In this case, we call this sequence the LDTA l sequence. Finally, we let $j_l^{\text{LDTA}i}$ be the job which achieves the maximum LDTA i processing time on machine l , and recall that $\underline{L}_j^{\text{LDTA}i}$ is the objective obtained by the LDTA i sequence on machine j .

Clearly,

$$\begin{aligned}
\underline{L}_1^{\text{LDTA}1} - \underline{L}_i^{\text{LDTA}i} &\leq (\underline{L}_1^* + \bar{P}) - \underline{L}_i^{\text{LDTA}i} \\
&\leq \underline{L}_1^{\text{LDTA}i} - \underline{L}_i^{\text{LDTA}i} + \bar{P} \\
&= c_{J_1^{\text{LDTA}i}}^1 + q_{J_1^{\text{LDTA}i}} - c_{J_i^{\text{LDTA}i}}^i - q_{J_i^{\text{LDTA}i}} + \bar{P} \\
&\leq c_{J_1^{\text{LDTA}i}}^1 + q_{J_1^{\text{LDTA}i}} - c_{J_1^{\text{LDTA}i}}^i - q_{J_1^{\text{LDTA}i}} + \bar{P} \\
&\leq \max_k \left\{ \sum_{l=k}^{J_1^{\text{LDTA}i}} (t_l^1 - t_l^i) \right\} + \bar{P} \tag{5}
\end{aligned}$$

and, similarly,

$$\begin{aligned}
\underline{L}_i^{\text{LDTA}i} - \underline{L}_1^{\text{LDTA}1} &\leq (\underline{L}_i^* + \bar{P}) - \underline{L}_1^{\text{LDTA}1} \\
&\leq \underline{L}_i^{\text{LDTA}1} - \underline{L}_1^{\text{LDTA}1} + \bar{P} \\
&= c_{J_i^{\text{LDTA}1}}^i + q_{J_i^{\text{LDTA}1}} - c_{J_1^{\text{LDTA}1}}^1 - q_{J_1^{\text{LDTA}1}} + \bar{P} \\
&\leq c_{J_i^{\text{LDTA}1}}^i + q_{J_i^{\text{LDTA}1}} - c_{J_i^{\text{LDTA}1}}^1 - q_{J_i^{\text{LDTA}1}} + \bar{P} \\
&\leq \max_k \left\{ \sum_{l=k}^{J_i^{\text{LDTA}1}} (t_l^i - t_l^1) \right\} + \bar{P}. \tag{6}
\end{aligned}$$

Combining Eqs. (5) and (6), dividing by n , taking n to infinity, and applying Lemma 2 prove the theorem. \square

4. COMPUTATIONAL TESTING OF BOUNDS AND HEURISTICS

The analysis described above indicates that, as the number of jobs goes to infinity, LDTA will be a very effective heuristic for the flow shop delivery time problem. In order to determine how effective this heuristic is for smaller numbers of jobs, and how the other parameters of the problem effect this heuristic, we undertook limited computational testing of the heuristic. In the computational testing, we varied the following parameters:

1. **Machines:** two, five, and 10 machine examples were tested.
2. **Jobs:** The number of jobs varied between 10, 25, 50, 100, 500, and 1000.
3. **Processing times:** Processing times were randomly generated, either from a uniform distribution over (0, 1], or from a normal distribution with mean .5 and standard deviation .25.

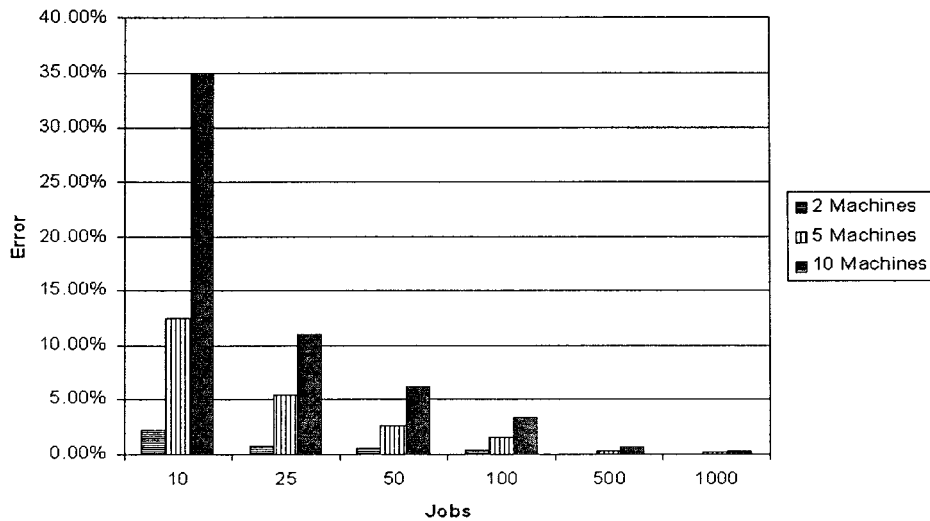


Figure 1. Error for varying jobs and machines.

4. **Release time:** Release times were drawn from a uniform distribution over $(0, r_t n]$, where n is the number of jobs, and r_t is a multiplier with the values .1, .25, .5, 1, 2, and 5.
5. **Delivery time:** Delivery times were drawn from a uniform distribution over $(0, d_t n]$, where n is the number of jobs, and d_t is a multiplier with the values .1, .25, .5, 1, 2, and 5.
6. **Permutation:** In addition, tests were done both with a permutation LDTA schedule as described in the analysis above, and a nonpermutation schedule, in which each machine was scheduled using LDTA.

All of the possible permutations of the parameters listed above were tested. Ten random tests of each parameter combination were completed, and averages were recorded. Below, we describe some of our observations—the complete data set is available from the author. We report the relative error $(L^{\text{LDTA}} - \underline{L}^{\text{LDTA}})/\underline{L}^{\text{LDTA}}$ for a variety of different parameters. Note that this number does not distinguish between the weakness of the lower bound and the weakness of the heuristic solution. In general, LDTA performs well except for small problem instances. We observed the largest relative errors for small numbers of jobs, and for small release data multipliers.

4.1. Jobs and Machines

Figure 1 is a plot of the relative error for different numbers of jobs and machines. For these data, the uniform distribution was used, $r_t = 1$, $d_t = 1$, and a permutation schedule is used. In this plot we observe that the relative error decreases with number of jobs and number of machines. We also observed this trend for other distributions and release time and delivery time multipliers. Although this relative error can be quite large for fewer than 50 jobs and more than two machines (in some instances approaching 130%), in all of our tests with greater than 25 jobs or less than 10 machines, the relative error was less than 15%, and typically much less than 15%.

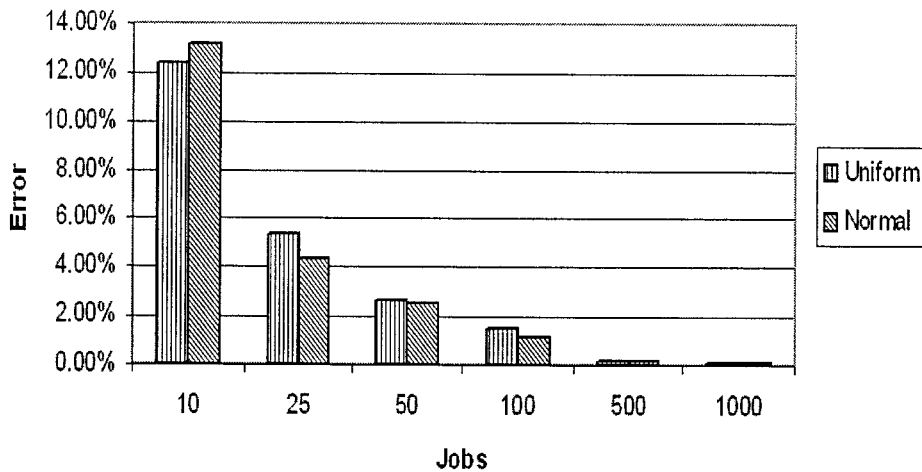


Figure 2. Error for varying jobs and distributions.

This suggests that it may be valuable to use a more advanced scheduling approach for small scale problems. Either branch and bound or local search based approaches may have more value for small problem instances.

4.2. Normal and Uniform Distribution

Figure 2 is a plot of the relative error for different numbers of jobs when processing times are generated using either the normal or the uniform distribution. For this data, $r_i = 1$, $d_i = 1$, five machines are considered, and a permutation schedule is used. LDTA appears to perform about the same, regardless of which distribution is used to generate processing times. Once again, we also observed this trend for different numbers of machines, and for other release time and delivery time multipliers.

4.3. Delivery Time and Release Time Multipliers

Figure 3 is a plot of the relative error for different release time and delivery time multipliers. For these data, 500 jobs and five machines are considered, the processing times are uniformly distributed, and a permutation schedule is used. LDTA performs well in all cases, although the relative error is significantly worse for release time multipliers of .5 or less. This is not surprising given that relative errors were for the most part highest when all jobs had the same release dates. This can in all likelihood be attributed to the lower bound. If all release times are relatively small, completion times for the single machine lower bound will typically be much less than completions times in the LDTA upper bound. In addition, performance seems to improve slightly with increasing delivery time multiplier. This may also be due to the lower bound, since we would expect the performance of LDTA to be good when delivery times are small, because if delivery times are small, the optimal solution will be very close to the completion time of the last job in the sequence. We would expect this completion time to be very similar in both the optimal solution to a large problem instance, and the LDTA solution to the same instance. Indeed, we ran a number of trials with a delivery time multiplier of .01, and in all cases with a release time multiplier of 1 or greater, and 50 or more jobs, the relative error was less than 10%.

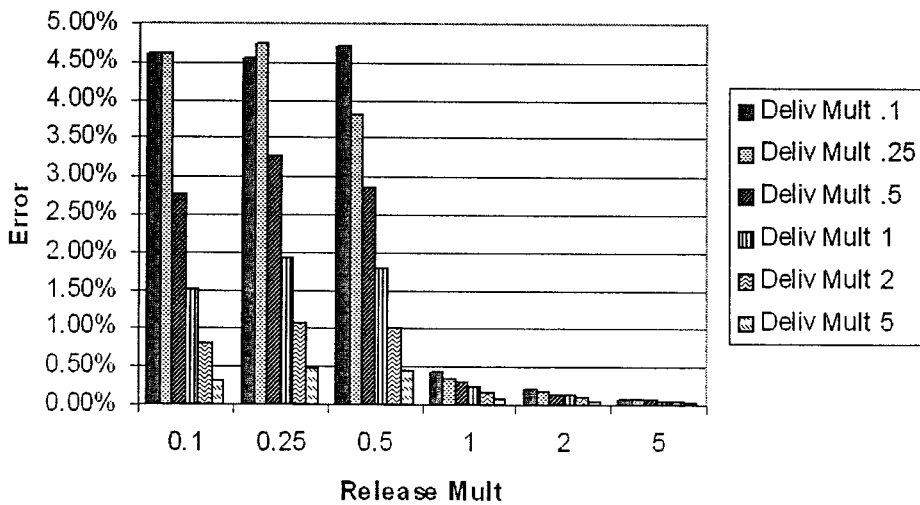


Figure 3. Error for varying release and delivery time multipliers.

We also observed similar behavior for different numbers of machines and jobs, and different distributions.

4.4. Jobs and Permutation Schedules

Figure 4 compares the relative error for different numbers of jobs when either a permutation or a non-permutation schedule is used. For these data, $r_t = 1$, $d_t = 1$, five machines are considered, and a uniform distribution is used to generate processing times. As expected, the nonpermutation schedule has slightly smaller relative errors.

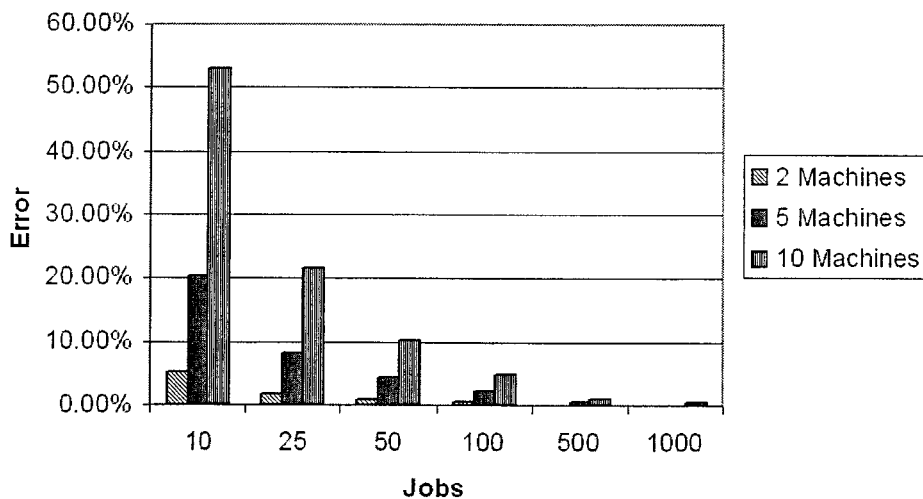


Figure 4. Error for permutation and nonpermutation schedules.

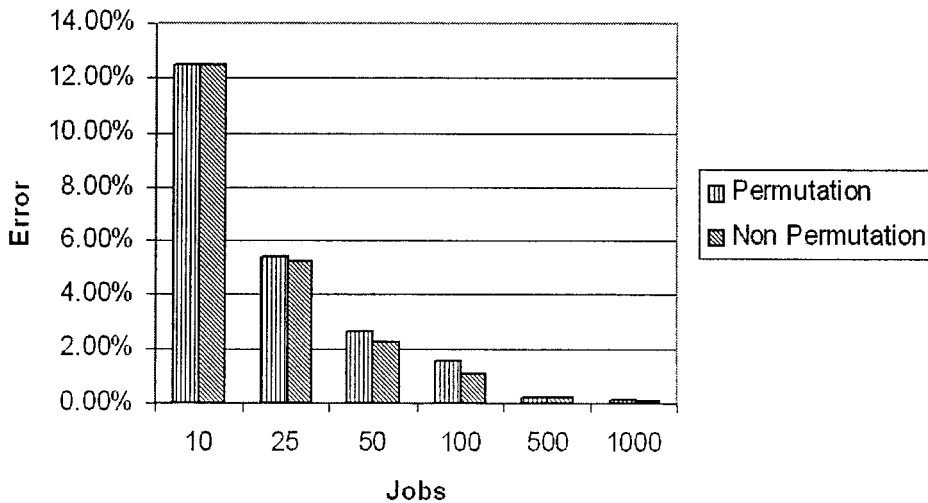


Figure 5. Error for varying jobs and machines with equal release times.

4.5. No Release Times

We also tested LDT scheduling for the static problem. Note that Figures 5 and 1 both present results for which uniform distribution and permutation scheduling were used, and for which $d_i = 1$. By comparing these figures, it can be observed that in general the error is larger for the static problem. As we discussed above, this is likely due to the weakness of the lower bound.

5. CONCLUSIONS

In this paper, we analytically and computationally demonstrated the effectiveness of a simple, on-line dispatch rule for the flow shop delivery time problem, both static and dynamic versions. The flow shop delivery time problem is strongly related to another important scheduling problem, the flow shop maximum lateness problem. To prove our results, we extended the tools of probabilistic analysis to a class of flow shop scheduling problems for which they had not, to the best of our knowledge, been used before—problems in which the objective is determined by a single job which can fall anywhere in the sequence. Of course, there are many more complex scheduling problems to which the tools of probabilistic analysis have not yet been applied. In the future, we hope to extend the types of approaches demonstrated in this paper to more complex scheduling models.

ACKNOWLEDGMENTS

We would like to thank the associate editor and referees for their valuable feedback and comments on earlier drafts of this paper. This research was supported in part by NSF Contract DMI-9732795 and by NSF CAREER Award DMI-0092854.

REFERENCES

- [1] J. Carlier, The one machine sequencing problem, *Eur J Oper Res* 11 (1982), 42–47.
- [2] L.A. Hall, “Approximation algorithms for scheduling,” *Approximation algorithms for NP-hard problems*, D.S. Hochbaum (Editor), PWS, Boston, 1997.

- [3] L.A. Hall and D. Shmoys, Jackson's rule for single-machine scheduling: Making a good heuristic better, *Math Oper Res* 17 (1992), 22–35.
- [4] J.A. Hoogeveen and A.P.A. Vestjens, A best possible deterministic on-line algorithm for minimizing maximum delivery time on a single machine, *SIAM J Discrete Math* 13 (2000), 56–63.
- [5] P. Kaminsky and D. Simchi-Levi, Probabilistic analysis and practical algorithms for the flow shop weighted completion time problem, *Oper Res* 46 (1998), 872–882.
- [6] P. Kaminsky and D. Simchi-Levi, The asymptotic optimality of the SPT rule for the flow shop mean completion time problem, *Oper Res* 49 (2001), 293–304.
- [7] H. Kise, T. Ibaraki, and H. Mine, Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times, *J Oper Res Soc Jpn* 22 (1979), 205–224.
- [8] J.K. Lenstra, A.H.G. Rinnooy Kan, and P. Brucker, Complexity of machine scheduling problems, *Ann Discrete Math* 1 (1977), 343–362.
- [9] H. Lui and D. Simchi-Levi, On the asymptotic optimality of on-line algorithms for the flow shop problem with release dates, Working paper, Northwestern University, Evanston, IL, 1999.
- [10] G.B. McMahon and M. Florian, On scheduling with ready times and due dates to minimize maximum lateness, *Oper Res* 23 (1975), 475–482.
- [11] E. Nowicki and Smutnicki, An approximation algorithm for a single machine scheduling problem with release times and delivery dates. *Discrete Appl Math* 48 (1994), 69–79.
- [12] E. Nowicki and S. Zdrzalka, A note on minimizing maximum lateness in a one-machine sequencing problem with release dates, *Eur J Oper Res* 23 (1986), 266–267.
- [13] M. Pinedo, *Scheduling: Theory, algorithms and systems*, Prentice Hall, Englewood Cliffs, NJ, 1995.
- [14] C.N. Potts, Analysis of a heuristic for one machine sequencing with release dates and delivery times, *Oper Res* 10 (1980), 1436–1441.
- [15] A. Ramudhin, J.J. Bartholdi, J. Calvin, J.H. Vande Vate, and G. Weiss, A probabilistic analysis of 2-machine flowshops, *Oper Res* 44 (1996), 899–908.
- [16] C.H. Xia, J.G. Shanthikumar, and P.W. Glynn, On the asymptotic optimality of the SPT rule for the static flow shop completion time problem, *Oper Res* 48 (2000), 615–622.