NSF grant number: DMI-0092854
NSF Program Name: Operations Research

CAREER: Scheduling of Large-Scale Systems: Probabilistic Analysis and Practical Algorithms for Due Date Quotation

Philip Kaminsky
University of California, Berkeley
Onur Kaya
University of California, Berkeley

Abstract: We present a novel single facility due-date quotation model and algorithm, and characterize the effectiveness of this algorithm employing the tools of probabilistic analysis. This model is intended to be a building block for the models that we will use to analyze due date quotation in make-to-order supply chains, and we present a summary of these models.

1. Introduction: The vast majority of recent supply chain research focuses on make-to-stock systems, and performance measures built around service and inventory levels. On the other hand, an increasing number of supply chains are better characterized as make-to-order systems. This is particularly true as more and more supply chains move to a mass customization-based approach to satisfying customers (see Simchi-Levi, Kaminsky, and Simchi-Levi [31]). Mass customization implies that at least the final details of project manufacturing must occur after specific orders have been received, and must thus be completed quickly and efficiently.

Clearly, make-to-order supply chains face many unique issues. One of the most significant is that since customers place orders and wait for products to arrive, due dates or delivery dates must typically

be quoted when a product is ordered. Recently, researchers have introduced a variety of models in an attempt to understand effective due date quotation. In some models, the assumption is made that orders will be placed independent of the length of the quoted lead time (see [1] for a survey), where the quoted lead time is typically defined to be the time span from the arrival of a job until its quoted start-processing time (or quoted due date). The objective in this case is frequently to minimize average quoted lead time subject to some constraint on the number or amount of tardy jobs. The majority of papers based on these models have been simulation based. For instance, Eilon and Chowdhury [2], Weeks [3], Miyazaki [4], Baker and Bertrand [5], and Bertrand [6] consider various due date assignment and sequencing policies, and in general demonstrate that policies which use estimates of shop congestion and job content information lead to better shop performance than policies based solely on job content.

Some analytical results do exist for limited versions of these models. Primarily, these consist of deterministic, common due date models, where a single due date must be assigned for all jobs, and static models, where all jobs are available at time 0. For these simplified models, a variety of polynomial algorithms have been developed (see Brucker[7], Kahlbacher[8], Panwalkar et al.[9], Hall and Posner [10], Seidmann et al.[11], and Chand and Chhajed [12]); however, these results don't extend in an obvious way to more complex models.

In other models, not all potential jobs are processed. In some of these models, a maximum lead time is associated with each customer. The firm decides whether or not to accept a particular offer, and if the offer is accepted, a perfectly reliable lead time (which must be less than the customers maximum lead time) is quoted. In other words, an accepted job has to start processing within its quoted lead time. The objective typically involves a revenue function which decreases with increasing quoted lead times. This is known as the Lead Time Quotation (LTQ) problem (see Keskinocak, Ravi, and Tayur [13] and the references therein). Again, effective algorithms exist for special cases of this problem, but not for the most general cases.

Some researchers approach LTQ models with a queuing theoretic framework. For example, Wein [14] considers a multiclass M/G/1 queuing system under the objective of minimizing the weighted average lead time subject to the constraints of the maximum fraction of tardy jobs and the maximum average tardiness. Spearman and Zhang [17] further characterize heuristic performance for these types of systems. In order to capture the impact of quoted lead times on demand, some models assume that given a quoted lead time, the customer decides whether or not to place an order. The probability that a customer places an order decreases with increasing lead time. Duenyas and Hopp [16] consider such a system using a queuing model, and provide effective heuristics under various problem characteristics. In their model, there is a single class of customers; the net revenue per customer is constant, customers have the same preferences for lead times, and the arrival and processing times follow the same distribution. Duenyas [15] extends their results to multiple customer classes, with different net revenues and lead time preferences.

Kaminsky and Hochbaum [30] survey due date quotation models in more detail.

In Kaminsky and Lee [29], we introduce a new due date quotation model (DDQP) which captures some of the key elements of the models and approaches above. In this model, jobs or orders arrive to a single server, representing a manufacturing organization, over time. All jobs are accepted, and due dates must be quoted immediately upon job arrival. The objective is to minimize average quoted lead time (or quoted due date), and all due dates must be met. Based on this 100% reliable due date quotation model, they develop an on-line due date quotation algorithm (DDQ) with several variations, characterize the asymptotic performance of this algorithm, and then analyze asymptotic probabilistic bounds on its performance. In Kaminsky and Lee [32], we extend the DDQP model to a more complex environment, the flow shop. In that paper, we developed a flow shop on-line due date quotation algorithm, analyze asymptotic bounds on its performance under some probabilistic assumptions, and then presented computational results which demonstrate the effectiveness of the algorithm.

In our current work, we are exploring due date quotation in supply chains. In particular, we will focus on a simple 2 party supply chain, in which a manufacturer works to satisfy customer orders. Customers arrive at the manufacturer over time, and the manufacturer produces to order. In order to complete production, the manufacturer needs to receive a component from a supplier. Each order takes a different amount of time to process at the manufacturer, and at the supplier. The manufacturer's objective is to determine a schedule and quote due dates in order to minimize a function of due date quoted, and lateness.

Our primary objective is develop effective approaches for scheduling both the centralized and decentralized versions of this model, so that we can investigate the relative advantage of a centralized system under various conditions. To do this, we will focus on three models. In the first model, the centralized model, both facilities are controlled by the same agent, who quotes a due date to the arriving customer, and then schedules jobs in both facilities in order to achieve the end objective. In the second model, the simple decentralized model, the manufacturer makes certain assumptions about the supplier in order to estimate a due date, and the supplier follows a simple scheduling priority rule. In this model, each facility is working to achieve its own goals. In the third model, the decentralized model with additional information exchange, both manufacturer and supplier quote due dates, the supplier to the manufacturer, and the manufacturer to the end customer based on the due date quoted to the manufacturer from the supplier.

We will model the situation in the centralized model as a flowshop. As pointed out in Kaminsky and Simchi-Levi [21] and Pinedo [22], most of the flow shop related research has focused on minimizing the makespan since the other objectives such as minimizing total completion time is very difficult to analyze. However, we analyze a due-date based flowshop model, and our analysis of the flowshop depends on novel analysis of a single machine model. In this model, jobs arrive at a single server over time. Due dates must be quoted to arriving customers, and jobs must be sequenced on the server to minimize a function of quoted due dates, and lateness. Our analysis of both centralized models will also rely on our analysis of the same model. In this paper, we present our analysis of this single machine model, and then introduce our supply chain models.

Of course, these are very difficult problems. Indeed, most versions of models related to those described above are NP hard. Thus, we focus on developing effective heuristics for deterministic versions of these models. Using the tools of probabilistic analysis, as well as computational testing, we characterize the performance of these heuristics under various conditions. In particular, we determine conditions under which these heuristics provide asymptotically optimal solutions for these models.

To put our work into perspective, we highlight other research relating to probabilistic analysis of scheduling problems. To the best of our knowledge, none of this work has focused on due date quotation models, and the vast majority has focused on relatively simple objectives, and the analysis of relatively simple algorithms. Much of this work has focused on parallel machine problems, including the work of Coffman, Frederickson and Lueker [23], Loulou [24] and Frenk and Rinnooy Kan [25], who analyze the parallel machine scheduling problem when the objective is to minimize the makespan, and Spaccamela et al.[26] and Webster [27], who analyze the parallel machine *weighted completion time* model. Ramudhin et al.[28] who analyze the 2-machine flow shop makespan model. Kaminsky and Simchi-

Levi(1999) [21], Kaminsky and Simchi-Levi [18] and Xia, Shanthikumar, and Glynn [20] analyze the flow shop average completion time problem. Finally, Kaminsky [33] considers the flow shop delivery time problem, which is closely related to the flow shop maximum lateness problem, a model which involves due dates, although they are given rather than determined by the model.

Also, we will focus on **on-line algorithms** for these models. 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. This models many real world problems, where job information is not known until a job arrives, and information about future arrivals is not known until these jobs arrive. 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 this model, it may be necessary to account for future arrivals when assigning due dates, and it may be necessary to insert idle time into the schedule while waiting for certain jobs to arrive, both of which suggest that off-line algorithms, which can utilize information about future arrivals to assign due dates and to decide whether or not to insert idle time in the schedule, will perform better than on-line algorithms which can't.

In the next section, we introduce the single facility model, and relevant heuristics and results. In section **3**, we introduce our supply chain models, which we intend to analyze in the near future.

2 The Single Machine Model and Main Results

2.1 The Model: We consider a single-machine online production system. Customers arrive at the system over time, and place an order (equivalently, jobs are released over time). The processing time of the order is known when the customer arrives, and the producer quotes a due date at the time of the arrival, with the goal of minimizing costs associated with the time until the due date, and costs associated with completing the processing of the job after its quoted due date.

In this model, a set of jobs needs to be processed non-preemptively on a single machine. Each job has an associated type $l, l = 1, 2, ...k$, and each type has an associated processing time p_l. Each job $i, i = 1, 2, ..., n$ also has an associated release time r_i. Also, the operator of the system quotes a due date for each job d_i. In particular, we focus on a system in which due dates are quoted without any knowledge of future arrivals - an online system. However, information about the current state of the system and previous arrivals can be used.

As mentioned above, we use the tools of probabilistic analysis, as well as computational testing, to characterize the performance of these heuristics under various conditions. In this type of analysis, we consider a sequence of randomly generated deterministic instances of the problem, and characterize the objective values resulting from applying a heuristic to these instances as the size of the instances (the number of jobs) grows to infinity. For this probabilistic analysis, we generate problem instances as follows. Each job has independent probability P_l of being job type l, where $\sum_{l=1}^{k} P_l = 1$ and job type l has known processing time p_l. Arrival times are determined by generating inter-arrival times drawn from identical independent distributions bounded above by some constant, with expected value ET.

The orders need to be processed on a single machine without preemption and no job can begin processing before its arrival time. Let C_i be the completion time of the order i. For each job, the producer incurs a cost of $c_i^d = c^d * d_i$, where where c^d is the unit cost of the quoted due date and a lateness cost of $c_i^L = c^L * [C_i - d_i]^+$ for order i if the order completes processing after its assigned due date, where c^L is the unit lateness cost. We assume that $c^L > c^d$; otherwise, setting all due dates equal to 0 will be optimal.

The objective of this problem is thus to determine a sequence of jobs and a set of due dates such that the total cost= $\sum_{i=1}^{n}(c_i^d + c_i^L)$ is minimized.

2.2 The Heuristic: Any algorithm for this problem has to do two things: quote due dates for each job, and sequence jobs for processing. The heuristic we propose, **SPTA-SL**, is based on sequencing the jobs according to the Shortest Processing Time Available (SPTA) rule. Under the SPTA heuristic, each time a job completes processing, the shortest available job which has yet not been processed is selected for processing. Note that this approach to sequencing does not take quoted due date into account, and is thus easily implemented.

Instead, the due date quotation rule takes the sequencing rule into account. To quote due dates, we maintain an ordered list of jobs that have been released and are waiting to be processed. In this list, jobs are sequenced in increasing order of processing time, so that the shortest job is at the head of the list. Since we are sequencing jobs SPTA, when a job completes processing, the first job on the list is processed, and each job moves up one position in the list. When a job i arrives at the system at its release time r_i with processing time p_i and the system is empty, it immediately begins processing and a due date equal to its release time plus its processing time is quoted:

$$d_i = r_i + p_i$$

However, if the system is not empty when a job i arrives, it is inserted into the appropriate place in the waiting list. Let r_i be the release time of job i, rt_i be the remaining time of the job in process at the time of arrival i, $pos[i]$ be the position of job i in the waiting list and $list[i]$ be the index of the i^{th} job in the waiting list. Then, a due date is quoted for this job i as follows:

$$d_i = r_i + rt_i + \sum_{j=1}^{pos[i]} (p_{list[j]}) + slack_i \qquad (0.1)$$

where $slack_i$ is some additional time added to the due date in order to account for future arrivals before this job is processed with processing times less than this job – these are the jobs that will be processed ahead of this job, and cause a delay in its completion time. The remainder of this subsection focuses on determining an appropriate value for $slack_i$. In the next section, we analytically demonstrate the effectiveness of the **SPTA-SL** approach.

Define the *minwait* time when a job arrives to be the remaining time of the job in process plus the total processing times of all of the jobs to be processed ahead of this arrival, so that

$$minwait_i = rt_i + \sum_{j=1}^{pos[i]-1} (p_{list[j]}).$$

Now, the slack for job i is calculated as follows, assuming a problem instance of size n.

Let pr_l be the probability that an arriving job has processing time less than p_l. For job types

l_i and l_j, assume without loss of generality that $p_{l_i} \leq p_{l_j}$ if and only if $l_i \leq l_j$. So, for a job type l,

$$pr_l = Pr\{p < p_l\} = \sum_{j=1}^{l-1} P_j \qquad l = 2, 3...k$$

and

$$pr_1 = 0.$$

Also, let EP_l the expected processing time of a job given that it is less than p_l,

$$EP_l = E[p|p < p_l] = \sum_{j=1}^{l-1} P_j * p_j.$$

and let $EA(t) = \frac{t}{ET}$ be the expected number of arrivals during a time interval of length t.

Since this is an n job instance, it may be that all of the jobs have arrived before job i is processed. So, we need to consider two cases according to the value of the total expected arrivals before job i is processed.

Let A_1^i be the expected number of all arrivals during the minwait time for job i of type l. So, $A_1^i = EA(minwait_i)$Then multiplying it with pr_l gives us the expected number of new arrivals that will be placed in front of job i. These arrivals will add an additional expected waiting time of

$$s_1^i = A_1^i * pr_l * EP_l.$$

Of course, during this time, some new additional jobs will also arrive; let $A_2^i = EA(s_1^i)$ represent the number of these additional jobs and $s_2^i = A_2^i * pr_l * EP_l$ will be the expected duration of the jobs among A_2^i that will be placed in front of job i.

More additional jobs may arrive while these jobs are processed. Thus, we need to determine the total number of jobs that are expected to arrive assuming that we have an infinite source of arriving jobs, instead of only n jobs, and compare it with the remaining number of jobs, $n - i$. Applying the same logic as above, we estimate the expected number of total arrivals :

$$TA_i = \sum_{j=1}^{\infty} (A_j^i)$$

where $A_j^i = EA(s_{j-1}^i)$, $s_j^i = A_j^i * pr_l * EP_l$ for $j = 1, 2, ...k$ and $s_0^i = minwait_i$.

Starting from A_1^i and s_1^i and writing the equations dynamically, we see that:

$$A_j^i = \frac{minwait_i (pr_l)^{j-1} (EP_l)^{j-1}}{ET^j}$$

and

$$TA_i = \sum_{j=1}^{\infty} \frac{minwait_i (pr_l)^{j-1} (EP_l)^{j-1}}{ET^j}.$$

If $\frac{pr_l * EP_l}{ET} \geq 1$ then this sum grows to infinity, which implies that all of the remaining jobs are expected to arrive to the system before job i goes into process. So, in this case, instead set $TA_i = n - i$.

If $\frac{pr_l EP_l}{ET} < 1$, then

$$TA_i = \sum_{j=1}^{\infty} \frac{minwait_i pr_l^{j-1} EP_l^{j-1}}{ET^j} = \frac{minwait_i}{ET - pr_l EP_l}$$

Of course, even in this case it is possible that $\frac{minwait_i}{ET - pr_l * EP_l}$ might be greater than the number of remaining jobs, $n - i$, again implying that the total expected number of arrivals $TA_i = n - i$.

Then, we quote the slack according to the following formula:

$$slack_i = TA_i * pr_l * EP_l$$

We summarize the due date quotation rule for job i of type l as follows:

If $\frac{pr_l * EP_l}{ET} \geq 1$

$$slack_i = (n-i) * pr_l * EP_l$$

else

$$slack_i = min\{\frac{minwait_i pr_l EP_l}{ET - pr_l EP_l}, (n-i) pr_l EP_l\} \quad (0.2)$$

and given this slack, we quote the following due date:

$$d_i = r_i + minwait_i + p_i + slack_i.$$

2.3 Analysis and Results For sets of randomly generated problem instances as described in preceding sections, let $Z_n^{SPTA-SL}$ represent the objective function value obtained by applying the **SPTA-SL** rule for an n, and let Z_n^* be the optimal objective function value for that instance. In this section, we prove

Theorem 1. *Consider a series of randomly generated problem instances meeting the requirements described above. Almost surely,*

$$\lim_{n \to \infty} \frac{Z_n^{SPTA-SL} - Z_n^*}{Z_n^*} = 0$$

In other words, **SPTA-SL** is asymptotically optimal for this problem.

To prove this theorem, we utilize a series of preliminary results and observations. First, we consider a feasible schedule for an instance of this problem, and define a *chain* to be a series of jobs processed consecutively without idle time between them, but with idle time before and after the series begins processing. Now consider the following:

Observation 1. *Consider any two non-delay schedules on a single machine. Number the chains consecutively in both of the schedules. There will be the same number of chains in both sequences and the k^{th} chain in either sequence will contain exactly the same jobs and will start and end at exactly the same time on the machine.*

We consider two cases, depending on the relationship between the expected processing and expected interarrival times, EP and ET.

Case 1: $EP < ET$

For this case, Gazmuri [19] proved the following lemma:

Lemma 1. *Consider a single machine problem. Let interarrival times $T_1, T_2, ..., T_{n-1}$ ($T_i = r_{i+1} - r_i$) be i.i.d. random variables, bounded above by some constant, where T is a generic random variable representing an interarrival time; the processing times $p_1, p_2, ..., p_n$ be i.i.d random variables, bounded above by some constant, where p is a generic random variable representing a processing time; the processing times and interarrival times be independent of each other, and let $EP < ET$.*

Then if M is a random variable representing the number of jobs in a chain, $E(M)$ and $E(M^2)$ are bounded by constants that are independent of n.

Now, let l_k be the chain index, and let $N(n)$ and M_k be the number of chains in an n job instance and the number of jobs in chain k, respectively. In the following discussion, we consider a single chain, and index 1,2...M_k in chain k. For each chain, let $Z_{l_k}^{SPTA-SL}$ and $Z_{l_k}^*$ be that portion of the objective function from jobs in the chain. Then

$$Z_{l_k}^{SPTA-SL} - Z_{l_k}^* = \sum_{i=1}^{M_k} (c^d d_i + c^L (C_i - d_i)^+) - Z_{l_k}^* \quad (0.3)$$

or summing over all chains, and reverting to original indexing:

$$Z_n^{SPTA-SL} - Z_n^* = \sum_{i=1}^{n} (c^d d_i + c^L (C_i - d_i)^+) - Z_n^* \quad (0.4)$$

We now bound both the "lateness" and the "due date" portions of this quantity.

Observe that for each chain k, a job can't be late by more than the length of that chain since the release time of any job in a chain is at least as big as the start time of the chain, we set a due date greater than the release time of job, and the job completes by the end of the chain. Thus,

$$C_i - d_i \leq M_k p_{max}$$

and

$$\sum_{i=1}^{M_k} c^L (C_i - d_i)^+ \leq c^L M_k (M_k p_{max})$$

Summing this quantity over all chains, we get:

$$\sum_{i=1}^{n} c^L (C_i - d_i)^+ \leq \sum_{k=1}^{N(n)} c^L M_k^2 p_{max}$$

Next, we focus on the maximum difference between the optimal objective value and the due date cost related portion of the heuristic objective function value.

Clearly, a job's earliest possible due date in the optimal schedule is equal to that job's release date, plus its processing time. Also, recall that due dates are set according to the formula $d_i = r_i + minwait_i + p_i + slack_i$, so that an upper bound on the difference between the due date cost in the heuristic solution, and the total cost in the optimal solution, is:

$$\sum_{i=1}^{M_k} (c^d d_i) - Z_{l_k}^* \leq \sum_{i=1}^{M_k} c^d (slack_i + minwait_i) \quad (0.5)$$

Now, recall the quantity $minwait_i$, the minimum time that job i will wait before processing under algorithm **SPTA-SL** if there is no inserted slack time. Clearly, $minwait_i$ is also smaller than the length of its chain because when job i arrives, $minwait_i$ can be no longer than the processing time of all waiting jobs, which is no greater than the length of the chain. Thus,

$$minwait_i \leq M_k p_{max} \quad (0.6)$$

Summing over all the jobs in the chain, we get:

$$\sum_{i=1}^{M_k} minwait_i \leq M_k^2 p_{max} \quad (0.7)$$

Also, since since $EP < ET$ and $pr_i \leq 1$, from (0.2),

$$slack_i = min\{\frac{minwait_i pr_l EP_l}{ET - pr_l EP_l}, (n-i) pr_l EP_l\}$$

$$slack_i \leq \frac{minwait_i pr_l EP_l}{ET - pr_l EP_l} \leq minwait_i \frac{EP}{ET - EP}$$

or, if we define $U = \frac{EP}{ET-EP}$, and apply (0.6), then

$$slack_i \leq M_k p_{max} U \quad (0.8)$$

Summing over all the jobs in the chain, we get

$$\sum_{i=1}^{M_k} slack_i \leq M_k^2 * p_{max} * U \qquad (0.9)$$

Substituting (0.7) and (0.9) into (0.5), we get:

$$\begin{aligned}\sum_{i=1}^{M_k}(c^d d_i) - Z_{l_k}^* &\leq c^d(M_k^2 p_{max} U + M_k^2 p_{max}) \\ &\leq (c^d + Uc^d)M_k^2 p_{max} \quad (0.10)\end{aligned}$$

Summing over the $N(n)$ chains, and substituting into (0.4) ,

$$Z_n^{SPTA-SL} - Z_n^* = \sum_{i=1}^{N(n)} (c^d + Uc^d + c^L)M_k^2 p_{max}$$

Dividing by n^2 and multiplying and dividing by $N(n)$, we get:

$$\frac{Z_n^{SPTA-SL} - Z_n^*}{n^2} \leq (c^d + c^d U + c^L)p_{max} * \frac{N(n)}{n^2} * \frac{\sum_{k=1}^{N(n)} M_k^2}{N(n)}$$

Also, note that the following hold almost surely.

$$\lim_{n\to\infty} \frac{n}{N(n)} = E(M)$$

$$\lim_{n\to\infty} \frac{\sum_{k=1}^{N(n)}(M_k^2)}{N(n)} = E(M^2)$$

Taking the limit as the number of jobs goes to infinity, and substituting:

$$\lim_{n\to\infty} \frac{Z_n^{SPTA-SL} - Z_n^*}{n^2} \leq \lim_{n\to\infty} \frac{(c^d + c^d U + c^L)p_{max}E(M^2)}{E(M)n} = 0 \qquad (0.11)$$

Case 2: $EP \geq ET$

For this case, we consider the asymptotic difference between the heuristic and optimal solutions, and begin this part of the proof by adding and subtracting the value $\sum_{i=1}^{n} c^d * C_i^{SPTA}$, where C_i^{SPTA} denotes the completion time of job i within the SPTA sequence. Also, C_i^* denotes the completion time of job i with the optimal sequence.

$$\lim_{n\to\infty} \frac{Z_n^{SPTA-SL} - Z_n^*}{n^2} =$$

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} c^d d_i + c^L[C_i^{SPTA-SL} - d_i]^+ - c^d C_i^*}{n^2} =$$

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} c^d d_i - c^d \sum_{i=1}^{n} C_i^{SPTA} + c^d \sum_{i=1}^{n} C_i^{SPTA}}{n^2}$$

$$+ \lim_{n\to\infty} \frac{\sum_{i=1}^{n} c^L[C_i^{SPTA-SL} - d_i]^+ - \sum_{i=1}^{n} c^d C_i^*}{n^2} \qquad (0.12)$$

It is known (see Kaminsky and Simchi-Levi (2001)) that SPTA is asymptotically optimal for minimizing total completion time of a single machine problem under the conditions considered in this paper. Thus, we can write the following lemma.

Lemma 2. *For a series of randomly generated problem instances meeting the requirements described above, almost surely,*

$$\lim_{n\to\infty} \frac{c^d \sum_{i=1}^{n} C_i^{SPTA} - \sum_{i=1}^{n} c^d C_i^*}{n^2} = 0$$

Now, recall that the algorithm **SPTA-SL** sequences jobs in SPTA order, let S_n^{SPTA} be the sum of completion times of all jobs sequenced according to SPTA in an n job instance, and let $S_n^{SPTA-SL}$ be the total cost of all due dates, $\sum_{i=1}^{n} c_i^d = \sum_{i=1}^{n} c^d d_i$. To prove our final result, we employ the following Lemma:

Lemma 3. *Consider a series of randomly generated problem instances meeting the requirements described above. Almost surely,*

$$\begin{aligned} \lim_{n\to\infty} \frac{S_n^{SPTA-SL} - S_n^{SPTA}}{S_n^{SPTA}} &= \\ \lim_{n\to\infty} c^d \frac{\sum_{i=1}^{n} d_i - C_i^{SPTA}}{n^2} &= 0 \end{aligned}$$

Proof. For any job i, let $\tilde{A}_i^j$ be the number of jobs of type j that arrived after r_i but before job i of type l started processing. By **SPTA**, these jobs will be processed before job i. It is easy to see that in the SPTA sequence, for a job of type $l \geq 2$,

$$C_i = r_i + minwait_i + p_l + \sum_{j=1}^{l-1} \tilde{A}_i^j * p_j$$

and for a job of type 1

$$C_i = r_i + minwait_i + p_1.$$

Now we divide the job types into two sets. For an n job problem instance, all jobs of type l such that $ET - EP_l * pr_l \leq 0$ are in set B_1^n, and all jobs of type l such that $ET - EP_l * pr_l > 0$ are in B_2^n.

For all jobs in B_1^n, the quoted due date will be

$$d_i = r_i + minwait_i + p_l + (n-i) * pr_l * EP_l.$$

Similarly, for all jobs in B_2^n, the quoted due date will be:

$$\begin{aligned} d_i &= r_i + minwait_i + p_l \\ &\quad + min\{(n-i)pr_l EP_l, \frac{minwait_i pr_l EP_l}{ET - EP_l pr_l}\} \end{aligned}$$

Summing over all jobs and taking the limit as the number of jobs goes to infinity, a.s. we get:

$$\begin{aligned} & \lim_{n\to\infty} \frac{S_n^{SPTA-SL} - S_n^{SPTA}}{n^2} \\ = & \lim_{n\to\infty} \frac{c^d \sum_{i=1}^{n} d_i - C_i^{SPTA}}{n^2} \\ = & \lim_{n\to\infty} \frac{c^d \sum_{j=1}^{2} \sum_{i\in B_j^n} d_i - C_i^{SPTA}}{n^2} \end{aligned} \tag{0.13}$$

Now, we consider the B_1^n and B_2^n jobs separately. Let S_l denote the set of type l jobs in B_1^n, and thus, for the B_1^n jobs,

$$\begin{aligned} & \lim_{n\to\infty} \frac{c^d \sum_{i\in B_1^n} d_i - C_i^{SPTA}}{n^2} = \\ & \lim_{n\to\infty} \frac{c^d \sum_{i\in B_1^n} [(n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j]}{n^2} = \\ & \lim_{n\to\infty} \frac{c^d \sum_{l=1}^{k} \sum_{i\in S_l} [(n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j]}{n^2} \end{aligned}$$

So, for all type l jobs,

$$\begin{aligned} & \lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \\ & \lim_{n\to\infty} \frac{\sum_{i\in S_l} [(n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j]}{n^2} = \end{aligned}$$

$$\lim_{n\to\infty} \frac{|S_l|EP_l pr_l}{n} - \frac{\sum_{i\in S_l}(ipr_l EP_l - \sum_{j=1}^{l-1}\tilde{A}_i^j p_j)}{n^2}$$

$$= P_l EP_l pr_l - \lim_{n\to\infty} \frac{\sum_{i\in S_l}(ipr_l EP_l + \tilde{A}_i EP_l)}{n^2} \quad (0.14)$$

by using the facts that, $\lim_{n\to\infty} \frac{|S_l|}{n} = P_l$ almost surely and $\lim_{\tilde{A}_i\to\infty} \frac{\sum_{j=1}^{l-1}\tilde{A}_i^j p_j}{\tilde{A}_i} = EP_l$ a.s. where $\tilde{A}_i = \sum_{j=1}^{l-1}\tilde{A}_i^j$

We also use the following lemma to prove our result.

Lemma 4. *For the values described above, almost surely, the following holds.*

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} ipr_l + \tilde{A}_i}{n^2} = pr_l P_l$$

Proof. Since $i * pr_l$ is the expected number of jobs with processing time less than p_i that arrived before job i and $\tilde{A}_i$ is the number of jobs with processing time less than p_i that arrived after r_i while job i is waiting in the queue to be processed, $(i * pr_l + \tilde{A}_i)$ denotes the number of jobs with processing time less than p_i that arrived before job i goes into service. Since $ET - EP_l * pr_l \leq 0 \Rightarrow \frac{ET}{pr_l} \leq EP_l$, it is expected that all n jobs will arrive to the system before job i of type l is processed and $n * pr_l$ of them is expected to be located before job i, that is

$$\begin{aligned} \lim_{n\to\infty} \frac{\sum_{i\in S_l} i * pr_l + \tilde{A}_i}{n^2} &= \lim_{n\to\infty} \frac{\sum_{i\in S_l} n * pr_l}{n^2} \\ &= \lim_{n\to\infty} \frac{|S_l| * pr_l}{n} \\ &= pr_l * P_l \end{aligned}$$

∎

By employing this lemma, we can write equation 0.14 as:

$$\begin{aligned} &\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} \\ &= P_l EP_l pr_l - \lim_{n\to\infty} \frac{\sum_{i\in S_l}(ipr_l EP_l + \tilde{A}_i EP_l)}{n^2} \\ &= P_l * EP_l * pr_l - P_l * EP_l * pr_l = 0 \quad (0.15) \end{aligned}$$

Then, if we add all types of jobs and multiply with c^d, we get:

$$\begin{aligned} \lim_{n\to\infty} \frac{c^d \sum_{i\in B_1^n} d_i - C_i^{SPTA}}{n^2} &= \\ \lim_{n\to\infty} \frac{c^d \sum_{l=1}^{k}\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} &= 0 \end{aligned} \quad (0.16)$$

Now, for the jobs in B_2^n where $ET - EP_l * pr_l > 0$ for a job type l, the quoted slack will be $min\{(n-i) * pr_l * EP_l, \frac{minwait_i * pr_l * EP_l}{ET - EP_l * pr_l}\}$.

Let F denote the time of the last job arrival to the system, such that no more jobs will arrive after that point. Then we can divide the jobs in B_2^n into four groups according to the quoted due dates and actual completion times of the jobs. For an n job problem instance,

1. D_1^n denotes the set of jobs where the quoted slack for a job i of type l is $slack_i = (n - i) * pr_l * EP_l$ and the actual completion time for that job satisfies $C_i \geq F$.
2. D_2^n is the set of jobs where the quoted slack for a job i of type l is $slack_i = \frac{minwait_i * pr_l * EP_l}{ET - EP_l * pr_l}$ and $C_i \leq F$.
3. D_3^n is the set of jobs where the quoted slack for a job i of type l is $slack_i = \frac{minwait_i * pr_l * EP_l}{ET - EP_l * pr_l}$ and $C_i \geq F$.

4. D_4^n denotes the set of jobs where the quoted slack for a job i of type l is $slack_i = (n - i) * pr_l * EP_l$ and $C_i \leq F$.

   Then we can write:

$$\lim_{n\to\infty} \frac{c^d \sum_{i\in B_2^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{c^d \sum_{j=1}^{4} \sum_{i\in D_j^n} d_i - C_i^{SPTA}}{n^2} \quad (0.17)$$

For each of these cases, we can write the following:

1. For the jobs in D_1^n, let S_l denote the set of type l jobs in D_1^n. Then,

$$\lim_{n\to\infty} \frac{\sum_{i\in D_1^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} \quad (0.18)$$

   For each job type l, we can write,

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{i\in S_l} ((n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2} = \lim_{n\to\infty} \frac{|S_l| pr_l EP_l}{n} - \frac{\sum_{i\in S_l} (ipr_l + \tilde{A}_i) EP_l}{n^2} \quad (0.19)$$

   Since $C_i \geq F$ for all jobs i in D_1^n, almost surely

$$\lim_{n\to\infty} \frac{\tilde{A}_i}{n} = \lim_{n\to\infty} \frac{(n-i) * pr_l}{n}$$
$$\Rightarrow \lim_{n\to\infty} \frac{\tilde{A}_i + i * pr_l}{n} = \lim_{n\to\infty} \frac{n * pr_l}{n}$$
$$\Rightarrow \lim_{n\to\infty} \frac{\sum_{i\in S_l} (i * pr_l + \tilde{A}_i)}{n} = |S_l| * pr_l$$

   Substituting into equation 0.19, we get:

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{|S_l| * pr_l * EP_l}{n} - \frac{|S_l| * pr_l * EP_l}{n} = 0 \quad (0.20)$$

   Summing over all job types:

$$\lim_{n\to\infty} \frac{\sum_{i\in D_1^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = 0 \quad (0.21)$$

2. For the jobs in D_2^n, let S_l denote the set of type l jobs in D_2^n. Then,

$$\lim_{n\to\infty} \frac{\sum_{i\in D_2^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} \quad (0.22)$$

   Then, for each job type l, we have

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{i\in S_l} (\frac{minwait_i pr_l EP_l}{ET - EP_l pr_l} - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2}$$

$$
\begin{aligned}
&= \lim_{n\to\infty} \frac{\sum_{i\in S_l} minwait_i pr_l EP_l}{(ET - EP_l pr_l)n^2} - \\
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} (ET - EP_l pr_l) \sum_{j=1}^{l-1} \tilde{A}_i^j p_j}{(ET - EP_l pr_l)n^2} \\
&= \lim_{n\to\infty} \frac{\sum_{i\in S_l} (minwait_i + \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{(ET - EP_l pr_l)n^2} - \\
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} \frac{ET}{pr_l EP_l} \sum_{j=1}^{l-1} \tilde{A}_i^j p_j}{(ET - EP_l pr_l)n^2} \\
&= \lim_{n\to\infty} \frac{\sum_{i\in S_l} (minwait_i + \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{(ET - EP_l pr_l)n^2} - \\
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} \frac{ETEP_l \tilde{A}_i}{pr_l EP_l}}{(ET - EP_l pr_l)n^2} \qquad (0.23)
\end{aligned}
$$

Since $C_i = r_i + minwait_i + p_i + \sum_{j=1}^{l-1} \tilde{A}_i^j p_j$,

$$
C_i - r_i - p_i = minwait_i + \sum_{j=1}^{l-1} \tilde{A}_i^j p_j \qquad (0.24)
$$

where $C_i - r_i - p_i$ is the time interval between the release time and the service start time of job i.

Also, since $C_i \leq F$ and $\tilde{A}_i$ is the number of jobs that arrived during this time with processing time less than p_i, $\frac{\tilde{A}_i}{pr_l}$ denotes the expectation of total number of jobs that arrived during that time and $\frac{ET*\tilde{A}_i}{pr_l}$ denotes the expected length of that time interval. So, we can write almost surely that:

$$
\lim_{n\to\infty} \frac{\sum_{i\in S_l} C_i - r_i - p_i}{n^2} = \lim_{n\to\infty} \frac{\sum_{i\in S_l} \frac{ET\tilde{A}_i}{pr_l}}{n^2} \qquad (0.25)
$$

Then, if we combine equations 0.24 and 0.25 and substitute into equation 0.23, we get:

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \\
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} \frac{ET\tilde{A}_i}{pr_l}}{(ET - EP_l pr_l)n^2} - \frac{\sum_{i\in S_l} \frac{ET\tilde{A}_i}{pr_l}}{(ET - EP_l pr_l)n^2} = 0 \qquad (0.26)
\end{aligned}
$$

Summing over all job types:

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{\sum_{i\in D_2^n} d_i - C_i^{SPTA}}{n^2} = \\
&\lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = 0 \qquad (0.27)
\end{aligned}
$$

3. For the jobs in D_3^n, let S_l denote the set of type l jobs in D_3^n. Then,

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{\sum_{i\in D_3^n} d_i - C_i^{SPTA}}{n^2} = \\
&\lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} \qquad (0.28)
\end{aligned}
$$

For each job type l, we can write,

$$
\begin{aligned}
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \\
&\lim_{n\to\infty} \frac{\sum_{i\in S_l} (\frac{minwait_i pr_l EP_l}{ET - EP_l pr_l} - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2} \qquad (0.29)
\end{aligned}
$$

Since $slack_i = \frac{minwait_i pr_l EP_l}{ET - EP_l pr_l}$,

$$\frac{minwait_i pr_l}{ET - EP_l pr_l} < (n-i)pr_l$$
$$\Rightarrow \quad minwait_i + (n-i)pr_l EP_l < (n-i)ET$$
$$\Rightarrow \quad \lim_{n\to\infty} \frac{r_i + minwait_i + (n-i)pr_l EP_l}{n}$$
$$\leq \quad \lim_{n\to\infty} \frac{r_i + (n-i)ET}{n} \qquad (0.30)$$

Since $C_i \geq F$, all of the jobs will have arrived before job i completes processing. So, we can write, almost surely, $\lim_{n\to\infty} \frac{\tilde{A}_i}{n} = \lim_{n\to\infty} \frac{(n-i)*pr_l}{n}$. Then, the completion time of a job i of type l is a.s.:

$$\lim_{n\to\infty} \frac{C_i}{n} =$$
$$\lim_{n\to\infty} \frac{r_i + minwait_i + p_i + \sum_{j=1}^{l-1} \tilde{A}_i^j p_j}{n} =$$
$$\lim_{n\to\infty} \frac{r_i + minwait_i + (n-i)pr_l EP_l}{n} \qquad (0.31)$$

Also, observe the following about the value of F with respect to the release time of job i:

$$\lim_{n\to\infty} \frac{F}{n} = \lim_{n\to\infty} \frac{r_i + (n-i)ET}{n} \qquad (0.32)$$

Substituting equation 0.31 and 0.32 into equation 0.30, we get:

$$\lim_{n\to\infty} \frac{C_i}{n} \leq \lim_{n\to\infty} \frac{F}{n}$$

However, we also know that, for job i in D_3^n, $C_i \geq F$. Thus, we have the following lemma:

Lemma 5. *The number of jobs in set S_l, that satisfies the above relations, almost surely satisfies the following:*

$$\lim_{n\to\infty} \frac{|S_l|}{n} = 0$$

Proof. Since F is a single time point, if $C_i \geq F$ and $\lim_{n\to\infty} \frac{C_i}{n} \leq \lim_{n\to\infty} \frac{F}{n}$ holds at the same time, then C_i should satisfy the following relation $C_i = F + K$ where K is a constant independent of n. So, the number of such jobs i, can not depend on n which gives us the result, $\lim_{n\to\infty} \frac{|S_l|}{n} = 0$. ■

Also, we know that for any job i of type l, $minwait_i \leq n * p_{max}$ and $\sum_{j=1}^{l-1} \tilde{A}_i^j * p_j \leq n * p_{max}$. Then we can write the following inequalities:

$$\lim_{n\to\infty} \frac{\sum_{i\in S_j} \frac{minwait_i pr_l EP_l}{ET - EP_l pr_l}}{n^2} \leq$$
$$\lim_{n\to\infty} \frac{|S_j| n p_{max} pr_l EP_l}{(ET - EP_l pr_l)n^2} = 0 \qquad (0.33)$$

and

$$\lim_{n\to\infty} \frac{\sum_{i\in S_j} \sum_{j=1}^{l-1} A_i^j p_j}{n^2} \leq \lim_{n\to\infty} \frac{n|S_j|p_{max}}{n^2} = 0 \qquad (0.34)$$

Combining these results with equation 0.29:

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} =$$
$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} (\frac{minwait_i pr_l EP_l}{ET - EP_l pr_l} - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2}$$
$$= 0 \qquad (0.35)$$

Summing over all job types:

$$\lim_{n\to\infty} \frac{\sum_{i\in D_3^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = 0 \quad (0.36)$$

4. For the jobs in D_4^n, let S_l denote the set of type l jobs in D_4^n. Then,

$$\lim_{n\to\infty} \frac{\sum_{i\in D_4^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} \quad (0.37)$$

For each job type l, we can write,

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{i\in S_l} ((n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2} \quad (0.38)$$

Since $slack_i = (n-i)pr_l EP_l$, we have that

$$\begin{aligned} \frac{minwait_i pr_l}{ET - EP_l pr_l} &> (n-i)pr_l \\ \Rightarrow minwait_i &> (n-i)(ET - pr_l EP_l) \end{aligned} \quad (0.39)$$

However, since $ET - EP_l * pr_l > 0$, we can say that almost surely $minwait_i < K$ for some constant $K < \infty$, that is, the queue length in front of job i will almost surely be finite.

Then, we can write the following lemma about the number of jobs in D_4^n.

Lemma 6. *The number of jobs in set S_l that satisfies the above relations, almost surely satisfies the following:*

$$\lim_{n\to\infty} \frac{|S_l|}{n} = 0$$

Proof. Since there exists $K < \infty$ s.t. $minwait_i < K$ and also $minwait_i > (n-i)(ET - pr_l EP_l)$, $(n-i)$ will be finite, the number of such jobs i is finite. ∎

Also, we know that for any job i of type l, $\sum_{j=1}^{l-1} \tilde{A}_i^j p_j \leq np_{max}$. Then we can write the following equations

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} (n-i)pr_l EP_l}{n^2} \leq \lim_{n\to\infty} \frac{|S_l| npr_l EP_l}{n^2} = 0 \quad (0.40)$$

and

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} \sum_{j=1}^{j-1} \tilde{A}_i^j p_j}{n^2} \leq \lim_{n\to\infty} \frac{n|S_l|p_{max}}{n^2} = 0 \quad (0.41)$$

So, substituting into our original equation 0.38:

$$\lim_{n\to\infty} \frac{\sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{i\in S_l} ((n-i)pr_l EP_l - \sum_{j=1}^{l-1} \tilde{A}_i^j p_j)}{n^2} = 0$$

Summing over all job types:

$$\lim_{n\to\infty} \frac{\sum_{i\in D_4^n} d_i - C_i^{SPTA}}{n^2} = \lim_{n\to\infty} \frac{\sum_{l=1}^{k} \sum_{i\in S_l} d_i - C_i^{SPTA}}{n^2} = 0 \quad (0.42)$$

Substituting all the results 0.21, 0.27, 0.36 and 0.42 into equation 0.17 and combining it with equation 0.16, we get

$$\begin{aligned}
& \lim_{n\to\infty} \frac{S_n^{SPTA-SL} - S_n^{SPTA}}{S_n^{SPTA}} \\
= & \lim_{n\to\infty} \frac{c^d \sum_{j=1}^{2} \sum_{i\in B_j^n} d_i - C_i^{SPTA}}{n^2} \\
= & \lim_{n\to\infty} \frac{c^d \sum_{i\in B_1^n} d_i - C_i^{SPTA}}{n^2} + \\
& \lim_{n\to\infty} \frac{c^d \sum_{j=1}^{4} \sum_{i\in D_j^n} d_i - C_i^{SPTA}}{n^2} = 0
\end{aligned} \tag{0.43}$$

To complete this part, we need to consider the tardiness portion of equation 0.12. When we consider only the tardy jobs, we can write the following Lemma:

Lemma 7. *For a series of randomly generated problem instances meeting the requirements described above, almost surely*

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} c^L [C_i^{SPTA-SL} - d_i]^+}{n^2} = 0$$

Proof. The proof follows the proof of Lemma 3. For Lemma 3, we have already proven that

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} d_i - C_i^{SPTA}}{n^2} = 0$$

Similarly, we can follow similar steps but sum exclusively over tardy jobs, rather than over all jobs. Then one can show that

$$\lim_{n\to\infty} \frac{\sum_{i\in L_j} c^L * (C_i^{SPTA-SL} - d_i)}{n^2} = 0$$

where L_j is the set of late jobs. ■

Finally, by combining Lemmas 2, 3 and 7, we prove Theorem 1.

$$\lim_{n\to\infty} \frac{Z_n^{SPTA-SL} - Z_n^*}{n^2} = 0 \tag{0.44}$$

3. The Supply Chain Models Utilizing our analysis of the single stage model discussed in the previous section, in subsequent work we will consider two parties, a supplier and a manufacturer, working to satisfy customer orders. Customer order $i, i = 1, 2,$ come to the manufacturer at arrival time(release time) r_i and the manufacturer quotes each customer a due date d_i at the arrival of the order. The release times are not known in advance and they are drawn from a certain probabilistic distribution. To complete the processing and to deliver the product to the customer, the manufacturer requires materials from the supplier, and orders these materials from the supplier at the time of customer order arrival. The processing time of an order i at the supplier is denoted by p_i^1. Then, in a flow shop setting, as soon as an order is completed at the supplier, it is moved to the manufacturer, where it requires p_i^2 time units to complete processing. The processing times p_i^1 and p_i^2 are known at the arrival of the order. The orders are processed both at the supplier and the manufacturer on a single machine without preemption, and no job can begin processing before its arrival time. The materials are sent to the manufacturer as soon as they are completed at the supplier and the orders are delivered to the customers as soon as they complete processing.

The objective of this research will be to determine a simple and asymptotically optimal online schedule and due date quotation heuristic for the manufacturer and the supplier to minimize the total cost function $\sum_{i=1}^{n}(c^d * d_i + c^L * L_i)$ where $L_i = (C_i - d_i)^+$ is the tardiness of job i and c^d and c^L are the unit due date and tardiness costs where $c^l > c^d$. This research will utilize the results developed in the previous sections of this paper.

Our goal is to develop effective approaches for scheduling and due date quotation for both the centralized and decentralized versions of this model, so that we can investigate the relative advantage of a centralized system under various conditions. To do this, we intend to focus on three models.

- In the **Centralized Model**, we model the system as a two facility flow shop model. It is assumed that the manufacturer and the supplier works as a single entity and they are both controlled by the same agent. The decisions about the scheduling of the jobs at both facilities and due date setting for the customer are made by this agent. The objective in this model is to quote the due dates so that the cost function $\sum_{i=1}^{n}(c^d * d_i + c^L * L_i)$ is minimized for the whole system. It is assumed that in the centralized case, the mean of the distribution of order interarrival times and the mean processing times at both the manufacturer and the supplier are known by the agent. However, the actual interarrival and processing times are not known until the arrival of that order to the system.

- In the **Simple Decentralized Model**, we intend to consider the possibility that the manufacturer and the supplier work independently, and that they both try to minimize their own costs. When the customer comes to the manufacturer and places the order, the manufacturer immediately quotes a due date, but the manufacturer has no knowledge of the suppliers operation. He doesn't know the process time of jobs at the supplier, or the schedule of the supplier. We assume that the manufacturer only knows the mean interarrival time of orders and mean processing time of the jobs at the supplier and at his own facility. Also, he knows *the number of jobs at the supplier*, since this quantity is equal to the number of orders arrived to the manufacturer minus the number of orders the supplier finished and sent to the manufacturer. The manufacturer therefore has to quote the due dates to the customers without knowing the schedule at the supplier and thus without knowing when the raw materials for that order will arrive to him from the supplier.

  In this decentralized case, we will consider the problem from the manufacturer side, since he quotes due dates to the customer, and we will attempt to find an effective scheduling rule and a due date quotation heuristic for the manufacturer to minimize the total cost.

- In the **Decentralized Model with Additional Information Exchange**, we will consider another version of the decentralized case in which the manufacturer has more information about the supplier, and thus can quote more effective due dates. In this case, we will assume that the supplier also sets a due date for the completion time of jobs at his site. When a customer order arrives, the manufacturer asks when the supplier will deliver the raw materials to him and the manufacturer uses this information to set due dates for his customers. However, the manufacturer doesn't know what schedule the supplier uses, and thus the manufacturer can only use the due date quoted by the supplier as additional information.

For each of these models, we will focus on developing effective heuristics for scheduling and due date quotation for these models, and analytically characterizing the effectiveness of these heuristics. In particular, building on the results described in section , we will use the tools of probabilistic analysis, as well as computational testing, to characterize performance of these heuristics under various conditions. Finally, we will analytically and computationally compare the various models, to assess the benefit of cooperative due date quotation in make-to-order supply chains.

References

[1] Cheng, T.C.E. and M.C. Gupta (1989), Survey of Scheduling Research Involving Due Date Determination Decisions. *European Journal of Operational Research* **38**, pp. 156-166.

[2] Eilon, S. and I.G. Chowdhury (1976), Due Dates in Job Shop Scheduling. *International Journal of Production Research* **14**, pp. 223-237.

[3] Weeks, J.K. (1979), A Simulation Study of Predictable Due-Dates. *Management Science* **25**, pp. 363-373.

[4] Miyazaki, S. (1981), Combined Scheduling System for Reducing Job Tardiness in a Job Shop. *International Journal of Production Research* **19**, pp. 201-211.

[5] K.R. Baker and J.W.M. Bertrand (1981), A Comparison of Due-Date Selection Rules. *AIIE Transactions* **13**, pp. 123-131.

[6] Bertrand, J.W.M. (1983), The Effect of Workload Dependent Due-Dates on Job Shop Performance. *Management Science* **29**, pp. 799-816.

[7] Brucker, P. (1998), *Scheduling Algorithms*. Springer, New York.

[8] Kahlbacher, H. (1992), Termin-und Ablaufplanung - ein analytischer Zugang, *Ph.D. thesis*, University of Kaiserslautern, cited in [7]

[9] Panwalkar, S.S., M.L. Smith, and A. Seidmann (1982), Common Due Date Assignment to Minimize Total Penalty for the One Machine Scheduling Problem. *Operations Research* **30**, pp. 391-399.

[10] Hall, N.G. and M.E. Posner (1991), Earliness-Tardiness Scheduling Problems, I: Weighted Deviation of Completion Times about a Common Due Date. *Operations Research* **39**, pp. 836-846.

[11] Seidmann, A., S.S. Panwalker, and M.L. Smith (1981), Optimal Assignment of Due Dates For a Single Processor Scheduling Problem. *International Journal of Production Research* **19**, pp. 393-399.

[12] Chand, S. and D. Chhajed (1992), A Single Machine Model for Determination of Optimal Due Date and Sequence. *Operations Research* **40**, pp. 596-602.

[13] Keskinocak, P., R. Ravi, and S. Tayur (1997), Algorithms for Reliable Lead Time Quotation. *GSIA Working Paper*, Carnegie Mellon University, Pittsburgh, PA.

[14] Wein, L. M. (1991) Due-date Setting and Priority Sequencing in a Multiclass M/G/1 Queue. *Management Science* **37**, pp. 834-850.

[15] Duenyas, I. (1995), Single Facility Due Date Setting with Multiple Customer Classes. *Management Science* **41**, pp. 608-619.

[16] Duenyas, I and W.J. Hopp (1995), Quoting Customer Lead Times. *Management Science* **41**, pp. 43-57.

[17] Spearman, M., and R.Q. Zhang (1999), Optimal Lead Time Policies. *Management Science* **45**, pp. 290-295.

[18] Kaminsky, P. and D. Simchi-Levi (2001), Probabilistic Analysis of an Online Algorithm for the Single Machine Completion Time Problem with Release Dates. *Operations Research Letters* **29** pp. 141-148.

[19] Gazmuri (1985), Probabilistic Analysis of a Machine Scheduling Problem. *Mathematics of Operations Research* **10** pp. 328-339.

[20] Xia, C., G. Shanthikumar, and P. Glynn (2000), On The Asymptotic Optimality of The SPT Rule for The Flow Shop Average Completion Time Problem. *Operations Research* **48**, pp. 615-622.

[21] Kaminsky, P. and D. Simchi-Levi (1998), Probabilistic Analysis and Practical Algorithms for the Flow Shop Weighted Completion Time Problem. *Operations Research*, **46**, pp. 872-882.

[22] Pinedo, M. (1995), *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Inc. Englewood Cliffs, New Jersey.

[23] Coffman, E.G., G. N. Frederickson and G. S. Lueker (1982), Probabilistic Analysis of the LPT Processor Scheduling Heuristic. *Deterministic and Stochastic Scheduling*, D. Reidel Publishing Company, M. A. H. Dempster et al. (eds.), pp. 319-331.

[24] Loulou, R. (1984), Tight Bounds and Probabilistic Analysis of Two Heuristics for Parallel Processor Scheduling. *Mathematics of Operations Research* **9**, pp. 142-150.

[25] Frenk, J.B.G. and A. H. G. Rinnooy Kan (1987), The Asymptotic Optimality of the LPT rule. *Mathematics of Operations Research* **12**, pp. 241-254.

[26] Spaccamela, A.M., W.S. Rhee, L. Stougie and S. van de Geer (1992), Probabilistic Analysis of the Minimum Weighted Flowtime Scheduling Problem. *Operations Research Letters* **11**, pp. 67-71.

[27] Webster, S. (1993), Bounds and asymptotic results for the uniform parallel processor weighted flow time problem. *Operations Research Letters*, **41**, pp. 186-193.

[28] Ramudhin, A, J.J. Bartholdi, J. Calvin, J.H. Vande Vate, and G. Weiss (1996) A Probabilistic Analysis of 2-Machine Flowshops. *Operations Research* **44**, pp. 899-908.

[29] Lee, Z.H. and P. Kaminsky (2003) On-line Algorithms for Due Date Quotation with Lateness Penalties. *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference* , pp. 3056-3069.

[30] Kaminsky, P and D. Hochbaum (2004) Due Date Quotation Models and Algorithms. *To be published as a chapter in the forthcoming book Handbook on Scheduling Algorithms, Methods and Models, Joseph Y. Leung (ed.), Chapman Hall/CRC.*

[31] Simchi-Levi, D., P. Kaminsky and E. Simchi-Levi(2004)*Managing The Supply Chain: The Definitive Guide for the Business Professional*, McGraw-Hill Trade: New York.

[32] Kaminsky, P. and Z-H Lee (2004) Analysis of On-line Algorithms for Due Date Quotation. *Submitted for publication.*

[33] Kaminsky, P. (2003) The Effectiveness of the Longest Delivery Time Rule for the Flow Shop Delivery Time Problem. *Naval Research Logistics* **50**(3) pp. 257-272