

Scheduling and Due-Date Quotation in a MTO Supply Chain

Philip Kaminsky

Onur Kaya

*Dept. of Industrial Engineering and Operations Research
University of California, Berkeley*

November 2006, Revised August 2007

Abstract

We consider a manufacturer, served by a single supplier, who has to quote due dates to arriving customers in a make-to-order production environment. The manufacturer is penalized for long lead times, and for missing due dates. In order to meet due dates, the manufacturer has to obtain components from a supplier. We model this manufacturer and supplier as a two-machine flow shop, consider several variations of this problem, and design effective due-date quotation and scheduling algorithms for centralized and decentralized versions of the model. We perform extensive computational testing to assess the effectiveness of our algorithms, and to compare the centralized and decentralized models in order to quantify the value of centralized control in a make-to-order supply chain. Since complete information exchange and centralized control is not always practical or cost-effective, we explore the value of partial information exchange for this system.

1 Introduction

We consider a problem faced by a manufacturer who produces customized products based on specific customer orders (a make-to-order, or MTO, system). When a customer arrives at the system with an order, the manufacturer typically has to quote a maximum completion time (in the literature, commonly called a due date) to the customer. In addition, the manufacturer has to decide on an appropriate processing sequence or schedule for the orders. Clearly, to effectively quote due dates and make sequencing decisions, the manufacturer needs to have a comprehensive understanding of his internal systems and processes, and of

the characteristics of arriving orders. Most prior research into these combined due date quotation/scheduling problems focuses on a single manufacturer, and on quoting due-dates based on conditions of that manufacturer's systems. Increasingly, however, portions of manufacturing processes are outsourced, and if this is the case, a manufacturer may not be able to effectively quote due dates and make sequencing decisions based exclusively on the knowledge of his or her own manufacturing systems.

Indeed, many manufacturers work with one or more suppliers to obtain subcomponents or parts necessary to produce customer orders, and the relationship between a manufacturer and his suppliers directly impacts scheduling and the due-date quotation(DDQ) decisions. On one hand, some suppliers are simply unreliable – a manufacturer may not have much of an idea when subcomponents will be delivered. This, of course, makes scheduling and due date quotation decisions very difficult. On the other hand, a supplier might be very quick and very reliable, or slow but reliable, and these characteristics will also impact the manufacturer's decisions. Unfortunately, the incentives of suppliers to provide a particular quality of service may not align with those of the manufacturer.

Issues related to these are frequently addressed in the literature with the use of supply chain coordination schemes. As many authors have observed, supply chain management and coordination has gained importance in recent years as businesses feel the pressure of increased competition and as managers have begun to understand that a lack of coordination can lead to decreased profits and service levels. There is a large and growing amount of literature on this subject, but the vast majority of this research focuses on make-to-stock (MTS) 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(MTO) systems. This is particularly true as more and more supply chains move to a mass customization-based approach to satisfy customers and to decrease inventory costs. Fast and on-time delivery is a critical performance measure for any firm that operates as a make-to-order system, and the manufacturer-supplier relationship dramatically impacts the delivery of products and system performance.

In this paper, we consider stylized models of such a supply chain, with a single manufacturer and a single supplier, in order to begin to quantify the impact of manufacturer-supplier relations on effective scheduling and due date quotation. Modeling this system as a two-machine flow shop, we analyze a make-to-order system in this simple supply chain setting and develop effective algorithms for scheduling and due-date quotation in both centralized and decentralized versions of this model. Building on this analysis, we explore the value of centralized control in this supply chain, and develop schemes for managing the supply chain in the absence of centralized control and with only partial information exchange.

As mentioned above, researchers have introduced a variety of models in an attempt to understand effective

due date quotation. Kaminsky and Hochbaum [16] and Cheng and Gupta [5] survey due date quotation models in detail. The majority of earlier papers on due-date quotation have been simulation based. For instance, Eilon and Chowdhury [9], Weeks [30], Miyazaki [23], Baker and Bertrand [1], and Bertrand [2] consider various due date assignment and sequencing policies, and in general demonstrate that policies that 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, for example, Brucker[3], Kahlbacher[13], Panwalkar et al.[24], Hall and Posner [12], Seidmann et al.[27], and Chand and Chhajed [4]); however, these results don't extend in an obvious way to more complex models. Some researchers approach lead time quotation models within a queuing theoretic framework. For example, Wein [31] 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. 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 [8] consider such a system using a queuing model, and provide effective heuristics under various problem characteristics. Duenyas [7] extends their results to multiple customer classes, with different net revenues and lead time preferences.

Kaminsky and Lee [21] introduce a new due date quotation model that 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 with several variations, characterize the asymptotic performance of this algorithm, and then analyze asymptotic probabilistic bounds on its performance.

In this paper, we extend previous work in due date quotation by exploring due date quotation in supply chains. In particular, we focus on a two member 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 customized 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

quoted lead time and lateness.

Our objective is to develop effective online 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. We focus on online models, in which information about a particular order's characteristics (arrival time and processing time) is not available until the order arrives at the manufacturer, as this lack of knowledge of future arrivals is a key characteristic of real world systems. In particular, we focus on three online 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, and very little information is exchanged. Indeed, in our discussions with the managers of several small manufacturing firms, this is typical of their relationships with many suppliers. Building on our first two models, we develop a decentralized model with additional information exchange, in which both the manufacturer and the 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. This model allows to explore the value of partial information exchange, and to determine if and when the cost and difficulty of implementing a centralized system are worth it.

Some of our models can also be interpreted as traditional manufacturing flowshops, particularly in the centralized case, where a single agent manages each of the stages. As pointed out in Kaminsky and Simchi-Levi [15] and Pinedo [25], most of the flow shop related research has focused on minimizing the makespan since the other objectives such as minimizing total completion time are very difficult to analyze. Indeed, most versions of these models are NP hard. In addition, it is in the context of supply chains that decentralized models make the most sense, so previous research in this area has not typically considered this notion. Thus, we focus on developing effective heuristics for these models, using the tools of probabilistic analysis, as well as computational testing, in order to 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, since researchers have often found that asymptotically optimal heuristics also give good results for smaller problem instances.

To put our work into perspective, we briefly discuss other research related to probabilistic analysis of scheduling problems. To the best of our knowledge, little 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 [6], Loulou [22] and Frenk and Rinnooy Kan [10], who analyze the parallel machine scheduling problem when the objective is to minimize the makespan, and Spaccamela et al.[28] and Webster [29], who analyze the parallel machine *weighted completion time* model. Ramudhin et al.[26] analyze the 2-machine flow shop makespan model. Kaminsky and Simchi-Levi [14], [15] and Xia, Shanthikumar, and Glynn [32] analyze the flow shop average completion time problem. Finally, Kaminsky [17] 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 the dates are in this case given rather than determined by the model.

In the next section, we introduce our models and sketch our main results.

2 Model and Main Results

We model two parties, a supplier and a manufacturer, working to satisfy customer orders. Customer orders, or jobs, i , $i = 1, 2, \dots, n$, arrive at the manufacturer at time r_i , and the manufacturer quotes a due date for each order, d_i , when it arrives. To begin processing each order, the manufacturer requires a component specifically manufactured to order by the supplier. The component requires processing time p_i^s at the supplier, and the order requires processing time p_i^m . (To clarify the remaining exposition, we will refer to an **order** or a **job** with processing time p_i^s at the supplier and p_i^m at the manufacturer.) Recall that we are focusing on **online** versions of this model, in which information about a specific order's characteristics (arrival time and processing time) is not available until the order arrives at the manufacturer (that is, until its release time r_i). As observed above, this model can also be interpreted as a traditional two machine flowshop, where the supplier and the manufacturer are machines in series. We consider several versions of this model, which we briefly describe here and discuss in more detail in subsequent paragraphs. In the centralized version of the model, the entire system is operated by a single entity, who is aware of processing times both at the supplier and at the manufacturer. In the simple decentralized model, the manufacturer and the supplier are assumed to work independently, and each is unaware of the job's processing time at the other stage. Finally, in the decentralized model with additional information exchange, the supplier quotes a due date to the manufacturer as a mechanism for limited information exchange.

In this paper, we propose algorithms for these models. Our goal in developing these algorithms is to provide a simple and asymptotically optimal online scheduling and due date quotation heuristic for either the manufacturer and the supplier individually in the decentralized system, or for both in the centralized system, that works well even for congested systems, so that we can compare the performance of these systems. Since firms are faced with a tradeoff between quoting short due dates, and meeting these due dates, we consider an objective function that captures both of these concerns. In particular, we are attempting to minimize the

total cost function

$$Z_n = \sum_{i=1}^n (c^d d_i + c^T T_i)$$

where $T_i = (C_i - d_i)^+ = \max\{C_i - d_i, 0\}$ is the tardiness of job i , C_i is the actual completion time of job i and c^d and c^T are the unit due date and tardiness costs for the model. Clearly, $c^T > c^d$, or otherwise it will be optimal for all due dates to be set to 0. We note that to the best of our knowledge, this objective has not been analyzed in either the machine scheduling or supply chain literatures.

As mentioned above, we use the tools of probabilistic analysis, as well as computational testing, to characterize the performance of these heuristics, and to compare these models. In particular, we focus on asymptotic probabilistic analysis of these models and heuristics. In this type of analysis, we consider a sequence of randomly generated instances of this model, with processing times drawn from independent identical distributions bounded above by some constant, and with arrival times determined by generating inter-arrival times drawn from independent identical distributions bounded above by some constant. Processing times are assumed to be independent of inter-arrival times. The processing time of a job at the supplier and the manufacturer may be generated from different distributions.

Recall that any algorithm for this problem has to both set due dates for arriving jobs, and determine job sequences at the supplier and the manufacturer. In Section 4.1, for the centralized model described above, we detail a series of heuristics, most notably one called (for reasons that will subsequently become clear) $SPTA_p - SLC$, that are used to determine due dates as jobs arrive, and to sequence jobs both at the supplier and at the manufacturer. We define $Z_n^{SPTA_p - SLC}$ to be the objective function value resulting from applying this heuristic to an n job instance of this due date quotation and sequencing problem, and Z_n^* to be the optimal objective function value for this instance of the due date quotation and sequencing problem. We prove the following theorem in Section A.3 of the online supplement to our paper.

Theorem 1. *Consider a series of randomly generated problem instances of the centralized model of size n . Let interarrival times be i.i.d. random variables bounded above by some constant; the processing times at each facility be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. Also, if the processing times at the supplier and the manufacturer are generated from independent and exchangeable distributions, then for an n job instance, using $SPTA_p - SLC$ to quote due dates and sequence jobs satisfies almost surely:*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{SPTA_p - SLC} - Z_n^*}{Z_n^*} = 0$$

In the simple decentralized model, recall that the manufacturer and the supplier work independently, and they both attempt to minimize their own costs. When the customer arrives at the manufacturer, the

manufacturer needs to quote a due date, even though he is not aware of either the processing time of the job at the supplier, or of the supplier's schedule. We assume that the manufacturer is aware of the number of jobs at the supplier (since this is equal to the number of jobs he sent there, minus the number that have returned), that he is aware of the average processing time at the supplier, and the mean interarrival rate of jobs to his facility. In Section 3, in a preliminary exploration, we consider a single facility due date quotation model that is analogous to our two-stage model, except that jobs only need to be processed at a single stage (in other words, our model, but with no supplier necessary). We develop an asymptotically optimal algorithm for this single facility model. For the purpose of understanding the performance of the decentralized system, we assume that the supplier uses this asymptotically optimal single facility model. Then, we explore the problem from the perspective of the manufacturer, and determine an effective due date quotation and sequencing policy for the manufacturer, given our assumptions about the limits of the manufacturer's knowledge about the supplier's system. Since the manufacturer is unaware of p_i^s values, he can't acquire any information about the supplier's schedule and location of a job at the supplier queue and assumes that each arriving job is scheduled in the middle of the existing jobs in the supplier queue. Under this assumption, in Section 4.2, we propose an asymptotically optimal heuristic called $SPTA - SLC_{SD}$. Define $Z_n^{SPTA-SLC_{SD}}$ to be the objective function value resulting from applying this heuristic to an n job instance, and Z_n^* to be the optimal objective function value for this instance given the information available to the manufacturer. We prove the following theorem in Section A.4 of the online supplement.

Theorem 2. *Consider a series of randomly generated problem instances of the simple decentralized model of size n . Let interarrival times be i.i.d. random variables bounded above by some constant; the processing times at the manufacturer be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. If the manufacturer uses $SPTA - SLC_{SD}$, then under manufacturer's assumptions about the supplier's schedule, almost surely,*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{SPTA-SLC_{SD}} - Z_n^*}{Z_n^*} = 0$$

When we explore our decentralized model with information exchange, we assume that when orders arrive at the manufacturer, that the manufacturer has the same information as in the simple decentralized model described above, and that in addition, the supplier uses our asymptotically optimal single facility algorithm for sequencing and to quote a due date to the manufacturer. The manufacturer in turn uses this due date in his due date quotation and sequencing heuristic, $SPTA - SLC_{DIE}$. In Section 4.3, we explain this heuristic in detail. We define $Z_n^{SPTA-SLC_{DIE}}$ to be the objective function value resulting from applying this heuristic to an n job instance, and Z_n^* to be the optimal objective function value for this instance given the information available to the manufacturer. We prove the following theorem in Section A.5 of the online supplement.

Theorem 3. *Consider a series of randomly generated problem instances of size n . Let interarrival times be i.i.d. random variables bounded above by some constant; the processing times at each facility be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. If the manufacturer uses $SPTA-SLC_{DIE}$, and the supplier uses a locally asymptotically optimal algorithm, then almost surely,*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{SPTA-SLC_{DIE}} - Z_n^*}{Z_n^*} = 0$$

In Section 5, we present a computational analysis of these algorithms for a variety of different problem instances, and compare the centralized and decentralized versions of the model. We see that the proposed algorithms are effective even for small numbers of jobs, and that the objective function values approach the optimal values quite quickly as the number of jobs increases. Also, we characterize conditions under which the centralized model performs considerably better than the decentralized model, and calculate this “value of centralization” under various conditions. Of course, as we mentioned previously, in many cases implementing centralized control is impractical or prohibitively expensive, so we also explore the value of simple information exchange in lieu of completely decentralized control.

In the next section of this paper, we introduce a preliminary model, and analyze this model. In Section 4, we present our models, algorithms, and results in detail, and in Section 5, we present the computational analysis of our heuristics and a comparison of centralized and decentralized supply chain due date quotation models using our heuristics. The proofs of the theorems are presented in the online supplement to this paper.

3 Preliminary: The Single Facility Model

3.1 The Model

Although our ultimate goal is to analyze multi-facility systems, we begin with a preliminary analysis of a single facility (or single machine) system. We focus on developing asymptotically optimal scheduling and due date quotation heuristics for this system, and consider cases when the system is congested (the arrival rate is greater than the processing rate), and cases when it is not.

In this model, we need to process a set of jobs, non-preemptively, on a single machine. Each job has an associated type $l = 1, 2, \dots, k$, and each type has an associated finite processing time $p_l < \infty$. At the time that job i arrives at the system, r_i , the operator of the system quotes a due date 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 will use the tools of asymptotic probabilistic analysis to characterize the performance of the heuristics we propose in this paper 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 a job of 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 λ .

The objective of our problem is to determine a sequence of jobs and a set of due dates such that the total cost $Z_n = \sum_{i=1}^n (c^d d_i + c^T [C_i - d_i]^+)$ is minimized, where C_i denotes the completion time of the order i . Clearly, to optimize this expression, we need to coordinate due date quotation and sequencing, and an optimal solution to this model would require simultaneous sequencing and due date quotation. However, the approach we have elected to follow to this model (and throughout the paper) is slightly different. Observe that in an optimal offline solution to this model, due dates would equal completion times since $c^T > c^d$, or otherwise it is optimal for all due dates to be set to 0. Thus, the problem becomes equivalent to minimizing the sum of completion times of the tasks. Of course, in an online schedule, it is impossible to both minimize the sum of completion times of jobs, and set due dates equal to completion times, since due dates are assigned without knowledge of future arrivals, some of which may have to complete before jobs that have already arrived in order to minimize the sum of completion times. However, for related problems (Kaminsky and Lee [18]), we have found that surprisingly, a two-phase approach is asymptotically optimal as the number of jobs get large. In this type of approach, we first determine a scheduling approach designed to effectively minimize the sum of completion times, and then we design a due date quotation approach that presents due dates that are generally close to the completion times suggested by our scheduling approach. While intuition might suggest that this approach would get arbitrarily bad as the number of jobs increases, we have found the opposite to be true. This is the intuition behind the heuristic presented below.

3.2 The Heuristic

As mentioned above, we have employed a heuristic that first attempts to minimize the total completion times, and then sets due dates that approximate these completion times in an effort to minimize the objective. The heuristic we propose sequences 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. As we observed in the introduction, although the problem of minimizing completion times is NP-hard, Kaminsky and Simchi-Levi [14] found that the SPTA rule is

asymptotically optimal for this problem. Also, 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 by SPTA, when a job completes processing, the first job in 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. However, if the system is not empty at time r_i , job i is inserted into the appropriate place in the waiting list. Let R_i be the remaining time of the job in process at the arrival of job 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:

$$d_i = r_i + R_i + \sum_{j=1}^{pos[i]} (p_{list[j]}) + slack_i \quad (3.1)$$

where $slack_i$ is some additional time added to the due date in order to account for future arrivals 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.

Throughout this paper, we name our heuristics in two parts, where the first part (before the hyphen) refers to the sequencing rule, and the second part refers to the due date quotation approach. Following this convention, we call this approach **SPTA-SL**, where the SPTA refers to the sequencing rule, and the SL refers to the due date quotation rule based on calculated completion at arrival plus the insertion of slack. In the next section, we analytically demonstrate the effectiveness of the **SPTA-SL** approach. Surprisingly, this decomposed sequential approach is asymptotically optimal for the overall due date quotation and sequencing problem.

The remainder of this subsection focuses on determining an appropriate value for $slack_i$. To do this, we need to estimate the total processing times of jobs that arrive before we process job i and have processing times less than the processing time of job i . We complete this calculation for one job at a time.

Define M_i 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 job i of type l , at the time of its arrival, such that

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

Let ψ_l be the probability that an arriving job has processing time less than p_l . Also, let $\mu_l = E[p|p < p_l]$

be the expected processing time of a job given that it is less than p_i , and let λ be the mean interarrival time. Then, the slack for job i can be calculated using an analogous approach to busy period analysis in queueing theory (see, e.g., Gross and Harris, [11]), where only those jobs that are shorter than job i are considered “new arrivals” for the analysis, since other jobs will be processed after job i and thus won’t impact job i ’s completion time. Note that the sequence of jobs to be processed before job i doesn’t impact job i ’s completion time, so for our analysis we can assume any sequence that is convenient (even if it is not the sequence that we will ultimately use, as long as we consider only those jobs that will be processed before job i in the sequence we actually use). In particular, we assume for the purpose of our analysis that first we process all the jobs that are already there when job i arrives, which takes the amount of time M_i . During this time, suppose that K jobs with processing time shorter than i arrived. Then, at the end of M_i , we have K jobs on hand that will impact the completion time of job i , and we pick one of them arbitrarily. Now, we imagine that the job just arrived when we selected it and that there are no other jobs in the system, and calculate that job’s busy period – the time until a queue featuring that job, and other arrivals shorter than it, will remain busy. We don’t consider any of the other K jobs until the busy period of this first job is completed. Then, we move to considering the second of the K jobs when the server becomes idle (when it finishes the busy period of the first job) and calculate its busy period, and so on, until we have considered all K jobs.

Thus, we can write the delayed busy period of job i with M_i as:

$$B_i(M_i) = M_i + \text{slack}_i = M_i + \sum_{j=1}^{\tilde{A}(M_i)} B_j$$

where $\tilde{A}(M_i)$ is the actual number of arrivals with processing time less than p_i during M_i (the arrivals after i that will be processed before job i) and B_j is the “busy period” of each of these jobs as defined in Gross and Harris [11]. Gross and Harris [11] show that for an M/G/1 queue, if $\frac{\mu_i}{\lambda/\psi_i} < 1$, then

$$E[B_i(M_i)] = \frac{M_i}{1 - \frac{\mu_i}{\lambda/\psi_i}} = \frac{M_i \lambda}{\lambda - \mu_i \psi_i}.$$

This suggests that we can approximate the slack value we are looking for by using this relationship:

$$\text{slack}_i = E[B_i(M_i)] - M_i = \frac{M_i \mu_i \psi_i}{\lambda - \mu_i \psi_i}$$

However, since we consider a problem instance of size n , it may be that all of the jobs have arrived before job i is processed, in which case the slack value will be equal to $\text{slack}_i = (n - i)\psi_i \mu_i$

Also, if $\frac{\psi_i \mu_i}{\lambda} \geq 1$, then the expected delayed waiting time is longer than the expected time for all the remaining jobs to arrive, and thus the slack value is again equal to $\text{slack}_i = (n - i)\psi_i \mu_i$.

We summarize the scheduling and due date quotation rule for job i of type l in Algorithm 1:

ALGORITHM 1: SPTA-SL

Scheduling: Sequence and process each job according to the shortest processing time available (SPTA) rule.

Due-Date Quotation:

$$d_i = r_i + M_i + p_i + slack_i$$

$$slack_i = \begin{cases} \min\left\{\frac{M_i \psi_l \mu_l}{\lambda - \psi_l \mu_l}, (n - i) \psi_l \mu_l\right\} & \text{if } \frac{\psi_l \mu_l}{\lambda} < 1 \\ (n - i) \psi_l \mu_l & \text{otherwise} \end{cases}$$

3.3 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 to an n job instance, and let Z_n^* be the optimal objective function value for that instance. We prove the following theorem in Section A.1 of the online supplement.

Theorem 4. *Consider a series of randomly generated problem instances of size n meeting the requirements described above. Let interarrival times be i.i.d. random variables bounded above by some constant; the processing times be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. Then, almost surely,*

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

In other words, **SPTA-SL**, which sequentially considers sequencing first and then due date quotation, is asymptotically optimal for the overall sequencing and due date quotation problem.

4 Supply Chain Models, Heuristics, and Analysis

In this section, we analyze the scheduling and due-date quotation decisions for two-stage supply chains using the results from our analysis in Section 3, and develop effective algorithms for scheduling and due-date quotation for both the centralized and decentralized versions of this system. These algorithms allow us to compare the value of centralization and information exchange in supply chains under a variety of different conditions.

4.1 The Centralized Model

4.1.1 The Model

For the centralized case, the system can be modeled as, in effect, a two-facility flow shop. We assume that the manufacturer and the supplier work as a single entity and that they are both controlled by the same agent that has complete information about both stages. The decisions about the scheduling at both facilities and due date setting for the customer are made by this agent.

4.1.2 The Heuristic and Main Results

Recall that in the single facility case, we utilized a known asymptotic optimality result for the completion time problem as a basis for our sequencing rule, and then designed a due date quotation rule so that due dates were close to the completion times. For this model, we employ the same two phase approach, but we first need to determine an asymptotically optimal scheduling rule for the related completion time problem, and then design an asymptotically optimal due date quotation heuristic for the sequence.

Xia, Shantikumar and Glynn [32] and Kaminsky and Simchi-Levi [15] independently proved that for a flow shop model with m machines, if the processing times of a job on each of the machines are independent and exchangeable, (i.e. p_i^j and p_i^k are independent and exchangeable for all pairs of machines (j, k) for every job i), processing the jobs according to the shortest total processing time $p_i = \sum_{j=1}^m p_i^j$ at the first facility (supplier) and processing the jobs on a FCFS basis at the others (manufacturer) is asymptotically optimal if all the release times are 0. We extend this result in Theorem 5, focusing on a 2-facility flow shop model, to include the case where all the release times are not necessarily 0, and jobs are scheduled by shortest available total processing time at the supplier. We denote this heuristic $SPTA_p$ (because we schedule the jobs based on total processing time). Let Z_n^* be the minimum possible value for the total completion time objective and $Z_n^{SPTA_p}$ be the total completion time of the jobs with the heuristic explained above for an n job instance. Then, we have the following theorem for this scheduling rule.

Theorem 5. *Consider a series of randomly generated problem instances of size n . If the processing times of jobs at the supplier and the manufacturer are generated from independent and exchangeable distributions, and if the jobs are scheduled using the instance, scheduling the jobs according to $SPTA_p$ is asymptotically optimal for the objective of minimizing the total completion time $Z_n = \sum_{i=1}^n C_i$. In other words, almost surely,*

$$\lim_{n \rightarrow \infty} \frac{Z_n^{SPTA_p} - Z_n^*}{Z_n^*} = 0$$

Observe that although this heuristic generates a permutation schedule, it is asymptotically optimal over all possible schedules, not just permutation schedules.

We base the first phase of our Algorithm 2 on Theorem 5, and then generate due dates similarly to the ones for our single facility approach. We call our due-date quotation rule *SLC* since it is based on the slack algorithm *SL* for the single facility case. The scheduling and due-date quotation algorithm called *SPTA_p – SLC* is stated as below:

ALGORITHM 2: SPTA_p-SLC

Scheduling: Process the jobs according to *SPTA_p* at the supplier and FCFS at the manufacturer.

Due-Date Quotation:

$$d_i^m = d_i^s + p_i^m + \max\{t_i^{ms} + t_i^{mm} + \text{slack}_i^m - (d_i^s - r_i), 0\}$$

$$d_i^s = r_i + p_i^s + M_i^s + \text{slack}_i^s$$

$$\text{slack}_i^s = \begin{cases} (n-i)pr\{p < p_i\}E\{p^s|p < p_i\} & \text{if } \lambda - pr\{p < p_i\}E\{p^s|p < p_i\} \leq 0 \\ \min\{(n-i)pr\{p < p_i\}E\{p^s|p < p_i\}, \frac{M_i^s pr\{p < p_i\}E\{p^s|p < p_i\}}{\lambda - pr\{p < p_i\}E\{p^s|p < p_i\}}\} & \text{otherwise} \end{cases}$$

$$t_i^{ms} = \sum_{i \in A} p_i^m \quad \text{where A=set of jobs in supplier queue scheduled before job } i \text{ at time } r_i.$$

$$t_i^{mm} = \sum_{i \in B} p_i^m \quad \text{where B=set of jobs in manufacturer queue at time } r_i$$

$$\text{slack}_i^m = \min\left\{\frac{(d_i^s - r_i - p_i^s)}{\lambda}, (n-i)\right\}pr\{p < p_i\}E\{p^m|p < p_i\}$$

The due date set of equations listed above is similar to those for the single facility case, adjusted for the centralized model. Essentially, we approximate the amount of workload, both at the supplier and at the manufacturer, that will be processed before job *i* if the *SPTA_p* scheduling rule is employed.

The due date, d_i^m , is equal to the sum of d_i^s , the approximated finish time of job *i* at the supplier, p_i^m , the processing time at the manufacturer, and $\max\{t_i^{ms} + t_i^{mm} + \text{slack}_i^m - (d_i^s - r_i), 0\}$, the approximate waiting time of job *i* at the manufacturer queue. $t_i^{ms} + t_i^{mm}$ denotes the sum of the processing times of the jobs that are already in the system and scheduled before job *i* at time r_i and slack_i^m approximates the workload at manufacturer of future arrivals that will be scheduled before job *i* while it waits in the supplier queue. In the calculation of slack_i^m , $\min\left\{\frac{(d_i^s - r_i - p_i^s)}{\lambda}, (n-i)\right\}$ denotes the approximate number of jobs that will arrive after r_i , and multiplying this by $pr\{p < p_i\}E\{p^m|p < p_i\}$ approximates the length of the subset of these jobs that will be scheduled before job *i* at supplier. These jobs will arrive at the manufacturer before job *i* and since we use FCFS at the manufacturer, they will also be processed before job *i* there. When setting the due date, we subtract $d_i^s - r_i$ since this is the approximate amount of work that will be processed at manufacturer while job *i* is still at the supplier.

Observe that once again we take a two-phase approach to a problem that intuitively has to be solved in

one-stage, and recall that in spite of this, Theorem 1 in Section 2 states that $SPTA_p - SLC$ is asymptotically optimal. We prove this result in Section A.3 of the online supplement.

Unbalanced processing times If the processing times at the supplier and the manufacturer are not exchangeable as assumed in the previous case, we adjust our scheduling and due-date quotation algorithm $SPTA_p - SLC$ to reflect the properties of the unbalanced system. For scheduling, we approach the system to balance the workloads at both facilities so that the total completion time is minimized. Since the processing times are unbalanced, we focus on the bottleneck facility and use an SPTA based schedule focusing on the processing times at the bottleneck facility. Again utilizing our two-phase approach, we adjust our due-date quotation algorithm for that schedule accordingly. Refer to Kaya [19] for more detail on the scheduling and due-date quotation algorithms for unbalanced cases.

4.2 The Simple Decentralized Model

While some supply chains are relatively easy to control in a centralized fashion, most often this is not the case. Even if the stages in a supply chain are owned by a single firm, information systems, control systems, and local performance incentives need to be designed and implemented in order to facilitate centralized control. In many cases, of course, the supplier and the manufacturer are independent firms, with relatively limited information about each other. Implementing centralized control in these supply chains is frequently even more difficult and costly, since the firms need to coordinate their processes, agree on a contract, implement an information technology system for their processes, etc. Thus, for either centrally owned or independent firms, centralization might not be worth the effort if the gains from centralization are not big enough.

Typically, if a supply chain is decentralized, the supplier and the manufacturer have only limited information about each other. The manufacturer is unaware of the processes at his supplier and needs to make his own decisions without any information from the supplier. For example, the manufacturer may only be aware of the average time it takes for the supplier to process and deliver an order. For this type of decentralized supply chain, we develop an effective approach for scheduling orders and quoting due dates to customers with limited information about the supplier.

4.2.1 The Model

In this section, we consider a setting in which the manufacturer and the supplier work independently and each tries to minimize his or her own costs. When the customer arrives at the manufacturer and places an order, the manufacturer needs to immediately quote a due date, although the manufacturer has limited

supplier-side information. In particular, the manufacturer has to quote due dates to the customers without knowledge of the supplier's schedule or knowledge of the location of any incomplete orders in the supplier's queue, and thus without knowledge or control of when the materials for that order will arrive from the supplier.

We assume that the manufacturer only knows the average processing time of jobs at the supplier, as well as the average interarrival time of orders to the system and the processing time of jobs at his own facility. The manufacturer doesn't know the processing time of jobs at the supplier or the schedule of the supplier. Thus, the manufacturer has to quote due dates to customers using only the knowledge of his own shop, mean processing times at the supplier, and knowledge of the number of jobs at the supplier, since this is equal to the number of orders that have arrived at the manufacturer minus the number of orders that the supplier completed and sent to the manufacturer.

4.2.2 The Heuristic and Main Results

In this decentralized case, we focus on the manufacturer's problem since he is the one who quotes the due dates to customers, and we try to find an effective scheduling rule/due date quotation heuristic to minimize the manufacturer's total cost given limited supplier information.

For this model, we employ the same two-phase approach that we used before, by first determining an asymptotically optimal scheduling rule to minimize the total completion times, and then designing a due date quotation heuristic to match the completion times with that schedule. In this case, since the manufacturer is working independently from the supplier and has no information about the processing times or the scheduling rule used in the supplier side, to minimize the total completion times, it will be asymptotically optimal for him to use the SPTA scheduling rule according to his own processing times p^m .

Based on this schedule, to find an effective due date quotation heuristic for the manufacturer, we use the same due date setting ideas as before. However, in this case, since the manufacturer is unaware of the processes at the supplier, we use estimates of the conditions at the supplier site – these estimates replace the information that we used in the centralized case.

The manufacturer has no information about the supplier except the number of jobs at the supplier side, q_i^s , at time r_i . Although using an SPTA schedule according to the processing times at the supplier, p^s , is asymptotically optimal to minimize the total completion times at the supplier, the manufacturer is unaware of p_i^s values, so that he can't infer any information about the supplier's schedule or the location of a job at the supplier queue. Thus, to quote a due-date, we assume that an arriving job is located in the middle of the existing jobs in the supplier queue, and that the future arrivals will be scheduled in front of this job with

probability 1/2. This is reasonable given that the manufacturer doesn't know anything about the schedule used by the supplier, about the processing time of that job, or about the other jobs at the supplier queue. We develop an asymptotically optimal scheduling and due date quotation algorithm for the manufacturer given that the manufacturer is using this assumption about the supplier's status.

We call the due-date quotation heuristic SLC_{SD} for this simple decentralized case since it is based on the slack algorithm SL for the single facility case. The manufacturer's scheduling and due date setting heuristic, $SPTA - SLC_{SD}$ for this case follows:

ALGORITHM 3: SPTA- SLC_{SD}

Scheduling: Process the jobs according to shortest processing time, p_i^m , at the manufacturer.

Due-Date Quotation:

$$\begin{aligned}
 d_i^m &= d_i^s + p_i^m + \omega_i^m + slack_i^m \\
 d_i^s &= r_i + \frac{q_i^s \mu^s}{2} + slack_i^s \\
 slack_i^s &= \begin{cases} \min\{(n-i), \frac{(\mu^s \frac{q_i^s}{2})}{\lambda - \frac{\mu^s}{2}}\} \frac{\mu^s}{2} & \text{if } \lambda - \frac{\mu^s}{2} > 0 \\ (n-i) \frac{\mu^s}{2} & \text{otherwise} \end{cases} \\
 \omega_i^m &= \max\{t_i^{mm} + \frac{(d_i^s - r_i)\Theta_i}{\mu^s} - (d_i^s - r_i), 0\} \\
 t_i^{mm} &= \sum_{j \in B} p_j^m \quad \text{where } B = \text{set of jobs at manufacturer queue with } p^m < p_i^m \text{ at time } r_i \\
 \Theta_i &= pr\{p^m < p_i^m\} E\{p^m | p^m < p_i^m\} \\
 \lambda^m &= \max\{\lambda, \mu^s\} \\
 slack_i^m &= \begin{cases} \{(n-i + q_i^s) - \frac{(d_i^s - r_i)}{\mu^s}\} \Theta_i & \text{if } \lambda^m - \Theta_i \leq 0 \\ \min\{\frac{\omega_i^m}{\lambda^m - \Theta_i}, (n-i + q_i^s - \frac{(d_i^s - r_i)}{\mu^s})\} \Theta_i & \text{otherwise} \end{cases}
 \end{aligned}$$

In the above equations, d_i^s is the approximate completion time of job i in the supplier side assuming that each arriving job is scheduled in the middle of the supplier queue. ω_i^m denotes the approximate queue length in front of job i when it arrives to the manufacturer from the supplier and $slack_i^m$ denotes the approximated length of the jobs that will arrive to the manufacturer after job i but will be processed there before job i . The value $\frac{(d_i^s - r_i)}{\mu^s}$ approximates the number of jobs that will be finished before job i at the supplier and will bring an extra workload to the manufacturer and $q_i^s + (n-i) - \frac{(d_i^s - r_i)}{\mu^s}$ approximates the maximum number of jobs that can arrive to the manufacturer from the supplier after job i .

We denote the objective value with this due date setting heuristic for the simple decentralized case $Z_n^{SPTA-SLC_{SD}}$ for an n job instance, and recall that once again in spite of the two-phase nature of the algorithm, Theorem 2 in Section 2 states that $SPTA - SLC_{SD}$ is asymptotically optimal under the conditions described above. We prove this result in section A.4 of the online supplement.

4.3 The Decentralized Model with Additional Information Exchange

As we demonstrate in our computational analysis in Section 5, there are significant gains that result from centralizing the control of this system. On the other hand, as we discussed above, there are frequently significant expenses and complexities inherent in moving to a centralized supply chain, if it is possible at all. Thus, firms may be motivated to consider limited or partial information exchange to achieve some of the benefits of centralization. Indeed, it may be that limited information exchange achieves many of the benefits of centralized control, rendering complete centralization unnecessary. In this section, we start to explore this question, by considering the case in which the supplier shares some of the information about his processes through a mechanism, so that presumably, the manufacturer can quote better due-dates to his customers. For this system, we find an effective scheduling and due-date quotation algorithm and analyze the gains by simple information exchange.

4.3.1 The Model

In particular, we assume that the supplier shares limited information with the manufacturer by quoting intermediate due dates. In other words, when a customer order arrives at the manufacturer, the manufacturer immediately places an order to the supplier, and the supplier quotes a due date for the suppliers subsystem to the manufacturer. The manufacturer can use this supplier due date, along with knowledge of his own shop and the time that the order will take him, to quote a due date to the customer. However, the model is in all other ways the same as the simple decentralized model. The manufacturer still doesn't know anything about the schedule the supplier uses or the processing times of the jobs at the supplier. In this case, however, the manufacturer can use the due date given by the supplier to better estimate the completion times of orders at the supplier, and can thus quote more accurate due dates to the customer.

4.3.2 The Heuristic and Main Results

Once again, we employ a two-phase approach to due date quotation and scheduling, and since the manufacturer is independent from the supplier in this case, as in the simple decentralized case, it once again makes sense for the manufacturer to schedule using the SPTA rule.

Similarly, since supplier works independently from the manufacturer, he acts as a single facility and tries to minimize his own costs. Thus, we assume that the supplier uses the asymptotically optimal **SPTA-SL** heuristic for scheduling and due date quotation described in Section 3 for the single facility case. Thus, the

supplier quotes due dates to the manufacturer according to the following rule:

$$d_i^s = r_i + p_i^s + M_i^s + slack_i^s$$

However, the manufacturer is unaware of the schedule used by the supplier, and instead uses the due dates quoted by the supplier to estimate the completion times of the orders at the supplier and sets his due dates accordingly.

We call the due-date quotation heuristic SLC_{DIE} for this decentralized case with information exchange since it is based on the slack algorithm SL for the single facility case. We summarize this approach below:

ALGORITHM 4: SPTA- SLC_{DIE}

Scheduling: Process the jobs according to shortest processing time, p^m , at the manufacturer.

Due-Date Quotation:

$$d_i^m = d_i^s + p_i^m + \omega_i^m + slack_i^m$$

$$\omega_i^m = \max\{t_i^{mmm} + \frac{(d_i^s - r_i)\Theta_i}{\mu^s} - (d_i^s - r_i), 0\}$$

$$t_i^{mmm} = \sum_{j \in B} p_j^m \quad \text{where } B = \text{set of jobs at manufacturer queue with } p^m < p_i^m \text{ at time } r_i$$

$$\Theta_i = pr\{p^m < p_i^m\}E\{p^m | p^m < p_i^m\}$$

$$\lambda^m = \max\{\lambda, \mu^s\} \quad slack_i^m = \begin{cases} \{(n - i + q_i^s) - \frac{(d_i^s - r_i)}{\mu^s}\}\Theta_i & \text{if } \lambda^m - \Theta_i \leq 0 \\ \min\{\frac{\omega_i^m}{\lambda^m - \Theta_i}, (n - i + q_i^s - \frac{(d_i^s - r_i)}{\mu^s})\}\Theta_i & \text{otherwise} \end{cases}$$

We denote the objective value for an n job instance with this due date setting heuristic for this decentralized case with information exchange $Z_n^{SPTA-SLC_{DIE}}$. Recall Theorem 3 in Section 2 which states that $SPTA - SLC_{DIE}$ is asymptotically optimal. We prove this result in Section A.5 of the online supplement.

5 Computational Analysis

Using these scheduling and due date quotation heuristics, we designed computational experiments to explore how these asymptotically optimal heuristics work even for smaller instances, and to better understand the value of centralization.

5.1 Efficiency of the Heuristics

To assess the performance of the heuristics, we compared objective values for various problem instance sizes to lower bounds. Observe that in an optimal offline solution to this model, due dates equal completion times, and the problem becomes equivalent to the problem of minimizing the sum of completion times of the orders.

Also note that for a single facility online problem, a preemptive SPTA schedule minimizes the total completion times of the jobs for the preemptive version of the problem, and is thus a lower bound on the non-preemptive completion time problem. Therefore, if we use that schedule and set the due dates equal to the job completion times, and ignore the lateness component of the objective, we have a lower bound on the objective value of our problem.

For the supply chain models, we use the same lower bound, focusing only on the processing times at one of the facilities, assuming that the job's waiting time at the queue of the other facility is zero. We use a preemptive SPTA schedule at our chosen facility, and then set the due-date of a job equal to the completion time of that job at the chosen facility plus the processing time of the job at the other facility, once again ignoring the lateness component of our objective. For the case with exchangeable processing time distributions, we can select either facility to focus on, and for the non-exchangeable cases, we focus on the bottleneck facility. In all cases, we have a lower bound on the completion time at one facility, and have added only the processing time at the other facility, ignoring capacity constraints at that facility, so this is clearly a lower bound on our objective.

For the single facility case, we simulate the system using different number of jobs arriving to the facility, and we use algorithm *SPTA – SL* to sequence and quote due-dates. We generate our problem sequences using two different distributions, exponential and normal distributions. We use the same arrival rate $\lambda = 1$ for every instance and vary the processing times, and relative due date and tardiness costs. The results are presented in Table 1 where the first number in every box is the result from using exponential distribution with processing mean μ and arrival rate $\lambda = 1$ and the second number is the result from using a normal distribution with processing mean μ , standard deviation $\mu/2$ and inter-arrival mean $\lambda = 1$ with standard deviation 0.5. We truncate each generated numbers to 0.1 if it is less than 0.1. Observe that as the number of jobs, n , increases, $Z_n^{SPTA-SL}$ rapidly approaches the lower bound. The rate of convergence differs for different μ/λ (i.e. $E(\text{process time})/E(\text{interarrival time})$) ratios and for different cost values but in all cases it converges quite quickly to the lower bound as n gets larger. The heuristic performs well even for small number of jobs. Note that each entry in Table 1 reflects the average of five runs with different random number streams.

For the centralized supply chain model, using the due date quotation and scheduling algorithm *SPTA_p – SLC* and its modifications for the unbalanced cases, we simulate the system with different numbers of jobs for different combinations of mean processing times μ^s and μ^m to explore the effectiveness of these heuristics, even for small problem instances. For these experiments, we generate the inter-arrival and processing times from exponential and normal distributions and for each instance in the table, the first numbers are the results from using exponential distribution with processing means μ^s and μ^m and arrival rate $\lambda = 1$ and

Table 1: Ratios of $Z_n^{SPTA-SL}$ to the lower bound for different combinations of n , c^T and μ values with $c^d = 1$

# of jobs	$c^T=1.1$	$c^T=1.5$	$c^T=2$	$c^T=5$	# of jobs	$c^T=1.1$	$c^T=1.5$	$c^T=2$	$c^T=5$
$\mu=0.5$					$\mu=1.5$				
10	1.00962 1.00643	1.01815 1.01912	1.02882 1.06729	1.09278 1.13226	10	1.03373 1.05023	1.06192 1.09419	1.09715 1.11902	1.30856 1.16420
100	1.00258 1.00303	1.00285 1.00364	1.00317 1.00452	1.00514 1.00635	100	1.05480 1.07436	1.05898 1.080329	1.06421 1.08714	1.09557 1.09342
1000	1.00033 1.00021	1.00036 1.00034	1.00039 1.00043	1.00062 1.00078	1000	1.01558 1.012243	1.01798 1.014929	1.02098 1.019925	1.03898 1.026732
5000	1.00007 1.00005	1.00008 1.00006	1.00009 1.00008	1.00014 1.00011	5000	1.00753 1.00836	1.00794 1.00965	1.00846 1.010944	1.01159 1.012721
10000	1.00003 1.00002	1.00003 1.00003	1.00003 1.00003	1.00006 1.00004	10000	1.00541 1.00543	1.00560 1.00613	1.00583 1.00626	1.00720 1.00697
$\mu=1$					$\mu=2$				
10	1.10481 1.06632	1.11671 1.10029	1.13157 1.14213	1.22079 1.18790	10	1.04609 1.05722	1.07794 1.06834	1.11775 1.09521	1.35663 1.21325
100	1.01426 1.01821	1.01605 1.02617	1.01828 1.02892	1.03165 1.03384	100	1.07541 1.03923	1.08011 1.05029	1.08600 1.08324	1.12129 1.13002
1000	1.00677 1.00344	1.00777 1.00731	1.00903 1.00982	1.01657 1.01532	1000	1.01140 1.00824	1.01636 1.01029	1.02256 1.01775	1.05976 1.03566
5000	1.00270 1.00199	1.00291 1.00306	1.00319 1.00355	1.00482 1.00423	5000	1.00901 1.00661	1.00925 1.00743	1.00953 1.00994	1.01128 1.01075
10000	1.00177 1.00092	1.00191 1.00107	1.00219 1.00173	1.00317 1.00278	10000	1.00457 1.00578	1.00491 1.00598	1.00534 1.00633	1.00790 1.00739

the second numbers are the results from using a normal distribution with processing means μ^s and μ^m and standard deviations $\mu^s/2$ and $\mu^m/2$ and inter-arrival mean $\lambda = 1$ with standard deviation 0.5. We truncate each generated numbers to 0.1 if it is less than 0.1. The ratios of the objective function $Z_n^{SPTA_p-SLC}$ and the total tardiness, T , to the lower bound are in Table 2 where each entry represents the average of five runs with different random number streams. As the number of jobs increases, $Z_n^{SPTA_p-SLC}/LB$ ratio approaches to 1 and, even for smaller instances, that ratio is still very close to one. Also, the T/LB ratio converges to 0 as the number of jobs increases, and that ratio is also very close to zero even for small number of jobs. Thus, the asymptotically optimal due-date quotation algorithm and its variants seem to work well, even for small numbers of jobs.

5.2 Comparison of Centralized and Decentralized Models

The ultimate goal of this work is to compare centralized and decentralized make-to-order supply chains, and to explore the value of information exchange in this system. To that end, we prepared a computational study to investigate the differences between the centralized and decentralized versions of this supply chain. For the centralized model we use the algorithm $SPTA_p - SLC$ for scheduling and lead-time quotation and for the decentralized cases, we assume that the supplier uses a SPTA schedule according to his own processing times p^s and the manufacturer uses the scheduling and due-date quotation algorithms described in Section 4. Table 3 shows the ratios of the objective values obtained by simulating the system for $n = 3000$ jobs, where each entry shows the average of five runs with different random number streams. For these experiments, we again generate the inter-arrival and processing times from exponential and normal distributions and for each instance in the table, the first numbers are the results from using exponential distribution with processing means μ^s and μ^m and arrival rate $\lambda = 1$ and the second numbers are the results from using a normal distribution with processing means μ^s and μ^m and standard deviations $\mu^s/2$ and $\mu^m/2$ and inter-arrival mean $\lambda = 1$ with standard deviation 0.5, truncated as before. In this table cen denotes the objective value for the centralized case, SD denotes the objective value for the simple decentralized model and DIE denotes the objective value with the decentralized model with information exchange.

Our experiments demonstrate that when the mean processing time at the supplier is smaller than the mean interarrival time, i.e. when there is no congestion at the supplier side, the centralized and decentralized models lead to very similar performance. However, as the congestion at the supplier begins to increase, the value of information and centralization also increases and the centralized model starts to lead to much better results than the decentralized ones. Similarly, if the mean processing time at the supplier is held constant, as the mean processing time at the manufacturer increases, the value of centralization decreases. As the supplier becomes more congested than the manufacturer, the supplier is the bottleneck facility that

Table 2: Ratios of $Z_n^{SPTA_p-SLC}/LB$ and T/LB for different combinations of n , μ^s and μ^m with $c^d = 1$, $c^T = 2$, $\lambda = 1$

# jobs	$\frac{Z_n^{SPTA_p-SLC}}{LB}$	T/LB	# jobs	$\frac{Z_n^{SPTA_p-SLC}}{LB}$	T/LB	# jobs	$\frac{Z_n^{SPTA_p-SLC}}{LB}$	T/LB
	$\mu^s=1, \mu^m=1$			$\mu^s=2, \mu^m=1$			$\mu^s=5, \mu^m=1$	
10	1.0202	0.0118	10	1.0344	0.0102	10	1.0462	0.0121
	1.0754	0.0201		1.0803	0.0345		1.0771	0.0321
100	1.0217	0.0059	100	1.0618	0.0096	100	1.0495	0.0122
	1.0356	0.0092		1.0379	0.0111		1.0425	0.0146
1000	1.0087	0.0022	1000	1.0239	0.0052	1000	1.0127	0.0086
	1.0113	0.0015		1.0149	0.0045		1.0227	0.0067
5000	1.0037	0.0010	5000	1.0102	0.0017	5000	1.0067	0.0023
	1.0032	0.0009		1.0053	0.0012		1.0061	0.0018
	$\mu^s=1, \mu^m=2$			$\mu^s=2, \mu^m=2$			$\mu^s=5, \mu^m=2$	
10	1.0343	0.0137	10	1.0263	0.0244	10	1.0471	0.0118
	1.0596	0.0178		1.0644	0.0192		1.0562	0.0203
100	1.0374	0.0040	100	1.0635	0.0160	100	1.0507	0.0134
	1.0277	0.0058		1.0339	0.0131		1.0409	0.0174
1000	1.0133	0.0025	1000	1.0241	0.0057	1000	1.0137	0.0075
	1.0156	0.0031		1.0149	0.0042		1.0117	0.0062
5000	1.0071	0.0012	5000	1.0108	0.0039	5000	1.0080	0.0024
	1.0094	0.0014		1.0132	0.0035		1.0137	0.0041
	$\mu^s=1, \mu^m=5$			$\mu^s=2, \mu^m=5$			$\mu^s=5, \mu^m=5$	
10	1.0552	0.0144	10	1.0553	0.0259	10	1.0467	0.0349
	1.0543	0.0155		1.0612	0.0242		1.0633	0.0264
100	1.0494	0.0053	100	1.0722	0.0159	100	1.0196	0.0138
	1.0335	0.0104		1.0376	0.0123		1.0759	0.0294
1000	1.0191	0.0021	1000	1.0330	0.0046	1000	1.0117	0.0108
	1.0142	0.0019		1.0237	0.0045		1.0358	0.0163
5000	1.0104	0.0016	5000	1.0143	0.0046	5000	1.0046	0.0039
	1.0092	0.0015		1.0117	0.0042		1.0162	0.0098

Table 3: Ratios of the objective values of centralized and decentralized models for different combinations of μ^s and μ^m with $c^d = 1$, $c^T = 2$, $\lambda = 1$

μ^s	μ^m	SD/cen	DIE/cen	SD/DIE	μ^s	μ^m	SD/cen	DIE/cen	SD/DIE
0.5	0.5	1.0021	1.0022	0.9999	2	0.5	2.0796	1.5089	1.3782
		1.0036	1.0028	1.0079			1.8727	1.4143	1.3241
0.5	1	1.0025	1.0031	0.9994	2	1	2.0823	1.5133	1.3760
		1.0037	1.0033	1.0039			1.8634	1.4080	1.3234
0.5	2	1.0137	1.0142	0.9996	2	2	1.9475	1.4404	1.3521
		1.0086	1.0054	1.0032			1.7962	1.3559	1.3247
0.5	5	1.0060	1.0062	0.9998	2	5	1.3111	1.0535	1.2445
		1.0163	1.0124	1.0038			1.2455	1.0634	1.1712
1	0.5	1.0269	1.0261	1.0007	5	0.5	2.4082	1.8639	1.2921
		1.0532	1.0471	1.0058			2.1473	1.5782	1.3606
1	1	1.0205	1.0154	1.0050	5	1	2.4070	1.8630	1.2920
		1.0337	1.0102	1.0233			2.1406	1.5633	1.3692
1	2	1.0237	1.0159	1.0076	5	2	2.3582	1.8258	1.2916
		1.0224	1.0096	1.0126			2.0053	1.5118	1.3264
1	5	1.0083	1.0056	1.0027	5	5	2.0826	1.6357	1.2731
		1.0105	1.0083	1.0021			1.7554	1.3481	1.3021

ultimately determines system performance, but the manufacturer has limited knowledge of the bottleneck facility and thus gives inaccurate due dates to the customers. However, if the manufacturer is the bottleneck, the manufacturer already has the information about his processing times which have the majority of impact on system performance. The value of information about processing times at the supplier is limited because they don't have tremendous impact on system performance. In other words, in this case, the value of information is lower.

These experiments suggest that if there is little or no congestion at the supplier, or if the manufacturer is significantly more congested than the supplier, centralizing control of the system is likely not worth the effort (at least, for the objective we are considering). However, if the congestion at the supplier increases, the value of centralization increases and total costs can be dramatically decreased by centralizing the system.

Although centralization can significantly decrease costs in some cases, if centralization is not possible or very hard to implement, simple information exchange might also help to decrease costs. As seen in Table 3, the losses due to decentralization can be cut in half by simple information exchange. However, even with information exchange, the costs of decentralized models are much higher, about 80% in some cases, than centralized ones. So, if the congestion at the supplier is high, centralization is worth the effort it takes to design and implement information systems, design and implement supply contracts, etc. However, if centralization is not possible, simple information exchange can also improve the level of performance dramatically.

6 Conclusion

We consider stylized models of a supply chain, with a single manufacturer and a single supplier, in order to quantify the impact of manufacturer-supplier relations on effective due date quotation. In our models, we consider several variations of scheduling and due-date quotation problems in MTO supply chains in order to minimize a function of the total quoted due-dates plus tardiness. We present due-date quotation and scheduling algorithms for centralized and decentralized versions of this model that are asymptotically optimal, and that are computationally found to be effective for relatively small problem instances. We also investigate the value of coordination schemes involving information sharing between supply chain members for this system. Through computational analysis, we see that if the processing rate at the supplier is larger than the arrival rate, i.e. when there is no congestion at the supplier side, the centralized and decentralized models perform similarly. However, as the congestion at the supplier starts to increase, the centralized model performs significantly better, and the value of information and centralization increases dramatically. Also, if centralization is not possible, a simple information exchange in the decentralized model can also improve

the level of performance dramatically, although not as significantly as centralized control.

Of course, these are stylized models, and real world situations have many more complex characteristics that are not captured by these models. Nevertheless, this is to our knowledge the first paper that analytically explores due date quotation in the context of a supply chain, and that explores the impact of the supplier-manufacturer relationship on due-date quotation.

In the future, we hope to evaluate more complex supply chain systems and see how the supply chain architecture can impact scheduling and due-date quotation decisions. Also, a pure MTO approach is not appropriate for many supply chains, especially if the processing times are long. We are currently investigating combined MTO-MTS supply chains, in which a facility can choose to stock some product types and produce others to order. We are in the process of developing approaches for making the MTS/MTO decision, as well as developing effective scheduling and due-date quotation algorithms for these systems. We also intend to expand this research to consider different functions of lead-time in the objective function. In some systems, the manufacturer doesn't have to accept all orders and has the option to reject certain orders. Pricing and capacity decisions can also be incorporated into these model. In all of these models and variants, the manufacturer needs to develop strategies for system design, and for scheduling and and due-date quotation.

Acknowledgement: We would like to thank the associate editor and anonymous referees for their helpful comments and suggestions. This material is based upon work supported by the National Science Foundation under Grants DMI-0092854 and DMI-0020439.

References

- [1] K.R. Baker, K.R. and J.W.M. Bertrand (1981), A Comparison of Due-Date Selection Rules. *AIIE Transactions* **13**, pp. 123-131.
- [2] Bertrand, J.W.M. (1983), The Effect of Workload Dependent Due-Dates on Job Shop Performance. *Management Science* **29**, pp. 799-816.
- [3] Brucker, P. (1998), *Scheduling Algorithms*. Springer, New York.
- [4] Chand, S. and D. Chhajed (1992), A Single Machine Model for Determination of Optimal Due Date and Sequence. *Operations Research* **40**, pp. 596-602.
- [5] 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.

- [6] 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.
- [7] Duenyas, I. (1995), Single Facility Due Date Setting with Multiple Customer Classes. *Management Science* **41**, pp. 608-619.
- [8] Duenyas, I and W.J. Hopp (1995), Quoting Customer Lead Times. *Management Science* **41**, pp. 43-57.
- [9] Eilon, S. and I.G. Chowdhury (1976), Due Dates in Job Shop Scheduling. *International Journal of Production Research* **14**, pp. 223-237.
- [10] 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.
- [11] Gross, D. and Harris C. (1985), *Fundamentals of Queueing Theory* John Wiley and Sons: New York.
- [12] 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.
- [13] Kahlbacher, H. (1992), Termin-und Ablaufplanung - ein analytischer Zugang, *Ph.D. thesis*, University of Kaiserslautern, cited in [3]
- [14] Kaminsky, P. and D. Simchi-Levi (2001), Probabilistic Analysis of an On-line Algorithm for the Single Machine Completion Time Problem with Release Dates. *Operations Research Letters* **29** pp. 141-148.
- [15] 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.
- [16] 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.
- [17] 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
- [18] Kaminsky, P. and Z-H. Lee (2005), Asymptotically Optimal Algorithms for Reliable Due Date Quotation. *Submitted for publication*.

- [19] Kaya, O. (2006), MTO-MTS Production Systems in Supply Chains. PhD Thesis, University of California, Berkeley, USA.
- [20] Keskinocak, P., R. Ravi, and S. Tayur (1997), Algorithms for Reliable Lead Time Quotation. *GSIA Working Paper*, Carnegie Mellon University, Pittsburgh, PA.
- [21] Lee, Z.H. and P. Kaminsky (2003), Online Algorithms for Due Date Quotation with Lateness Penalties. *Proceedings of the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference*, pp. 3056-3069.
- [22] Loulou, R. (1984), Tight Bounds and Probabilistic Analysis of Two Heuristics for Parallel Processor Scheduling. *Mathematics of Operations Research* **9**, pp. 142-150.
- [23] Miyazaki, S. (1981), Combined Scheduling System for Reducing Job Tardiness in a Job Shop. *International Journal of Production Research* **19**, pp. 201-211.
- [24] 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.
- [25] Pinedo, M. (1995), *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Inc. Englewood Cliffs, New Jersey.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] Webster, S. (1993), Bounds and asymptotic results for the uniform parallel processor weighted flow time problem. *Operations Research Letters*, **41**, pp. 186-193.
- [30] Weeks, J.K. (1979), A Simulation Study of Predictable Due-Dates. *Management Science* **25**, pp. 363-373.
- [31] Wein, L. M. (1991), Due-date Setting and Priority Sequencing in a Multiclass M/G/1 Queue. *Management Science* **37**, pp. 834-850.
- [32] 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.