NSF GRANT #0092854
NSF PROGRAM NAME: MES/OR

MTO-MTS Production Systems in Supply Chains

Philip M. Kaminsky
University of California, Berkeley

Onur Kaya
University of California, Berkeley

Abstract: Increasing cost pressures have led supply chain managers to focus on running increasingly lean and efficient supply chains, with minimal inventory. Indeed, more and more firms are relying on pull or make-to-order (MTO) supply chains to minimize cost and waste. At the same time, increasing competitive pressures are leading to an increased emphasis on customer service. An important element of customer service, of course, is having make-to-stock (MTS) items in stock, and delivering make-to-order (MTO) products quickly and by the promised due date.

In this project, we consider a variety of models designed to provide insight into the operation of combined MTS-MTO supply chains. We primarily focus on a simple stylized model of a two facility supply chain featuring a manufacturer served by a single supplier. Initially, we model a pure MTO supply, in which customers arrive at the manufacturer and place orders. The manufacturer needs to quote a due date to the customer when the order is placed, and the manufacturer needs to receive a component from the supplier before completing the manufacturing process. We design effective algorithms for production sequencing and due date quotation in both centralized and decentralized versions of this supply chain, characterize the theoretical properties of these algorithms, and compare the performance of the centralized and decentralized versions of the supply chain under various conditions.

We also consider combined MTS-MTO versions of this supply chain. In these models, the manufacturer and supplier have to decide which items to produce to order, and which items to produce to stock. Inventory levels must be set for the MTS items, and due dates need to be quoted for the MTO items. In addition, sequencing decisions must be made. We consider several versions of these models, design effective algorithms to find inventory levels, to quote due dates, and to make sequencing decisions in both centralized and decentralized settings, and use these algorithms to assess the value of joint MTS-MTO systems, as well as the value of centralization under various conditions.

Finally, we consider more complex supply networks, and employ the results from the simple two facility supply chains described above to design effective heuristics for the MTS-MTO decision, inventory levels, sequencing, and due date quotation in these more complex supply chains. We employ these heuristics to answer a variety of questions about the value of centralization, and of using a combined MTS-MTO approach in complex supply chains.

1. Introduction: Supply chain management can be viewed as the combination of the approach and information technology that integrates suppliers, manufacturers, and distributors of products or services into one cohesive process in order to satisfy customer requirements. Traditionally, orders have been the only mechanism for order exchange between firms; however, information technology now allows firms to share more extensive information such as demand, inventory data, etc. quickly and inexpensively. With the help of this information sharing, firms can now coordinate their processes more easily and increase overall system efficiency.

In this project, we analyze the inventory decisions, scheduling, and lead time quotation in a variety of supply chains structures, develop approaches to minimize the total cost in these supply chains, and compare centralized and decentralized versions of the supply chain to begin to quantify the value of centralization and information exchange.

Minimizing inventory holding costs and quoting reliable and short lead times to customers are clearly conflicting objectives in supply chains with stochastic demand and processing times. Ideally, companies would like to initiate production every time a customer order arrives in order to avoid inventory holding costs; however, this strategy is likely to lead to long waiting times for order delivery. Every time a customer arrives, the firm must quote a due date for the order. Long quoted lead times lead to customer dissatisfaction, lost sales and decreased profits. On the other hand, quoting short lead times increases the risk of missing the delivery dates, which also has negative implications for the firm. Also, for a company that produces multiple products with different char-

acteristics, the decision of when to produce each order also effects the completion times and thus the lead times of many other products. In this project, we analyze these trade-offs, and develop approaches to effectively address them.

Optimal supply chain performance requires the execution of a precise set of actions, however, those actions are not always in the best interest of the members in the supply chain, i.e. the supply chain members are primarily concerned with optimizing their own objectives, and that self-serving focus often results in poor performance. However, optimal performance can be achieved if the firms in the supply chain coordinate by contracting on a set of transfer payments such that each firm's objective becomes aligned with the supply chain's objective. See Cachon [6] for an extensive survey of supply chain coordination with contracts and Chen [10] for an extensive survey on information sharing and supply chain coordination. Cachon and Lariviere [7], Li [25], Jeuland and Shugan [17], Moorthy [27], Ingene and Parry [15] and Netessine and Rudi [28] are a few of the researchers that analyze the benefits of information sharing in supply chains and how to operate such systems for different objectives. Also, Bourland et al. [5], Chen [9], Gavirneni et al. [13], Lee et al. [23] and Aviv and Federgruen [2] show that sharing demand and inventory data improves the supply chain performance for several different objectives.

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 (see Simchi-Levi, Kaminsky, and Simchi-Levi [30]). 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. MTO systems have many unique issues and need to be operated very differently from MTS systems. In an MTO system, firms need to find an effective approach to scheduling their customers' orders, and they also need to quote short and reliable due-dates to their customers.

However, not all firms employ a pure MTO or MTS system. Although some firms make all of their products to order while some others make them to stock, there are also a number of firms that maintain a middle ground, where some items are made to stock and others are made to order. The decision on using either an MTO strategy or an MTS strategy at a facility heavily depends on the characteristics of the system. In supply chains using a combined system, holding inventory at some of the stages of the chain and using an MTO strategy at other facilities might decrease the costs dramatically without increasing the lead times. Because of this, companies are starting to employ a hybrid approach, a "push-pull" strategy (i.e. a combined MTO-MTS system), holding inventory at some of the facilities in their supply chain and producing to order in others. For example, a company with a diverse product line and customer base can be best served with an appropriate combination of MTO-MTS systems. Also, a supplier with one primary customer and several smaller customers might be able to operate more profitably with a combined MTO-MTS system. The importance of inventory management as outlined above has also increased with the growing prevalence of e-commerce. In today's world, e-commerce end customers expect high levels of service and speedy and on-time deliveries.

In the first part of this project, we consider a single 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 consider several variations of this problem, and design effective due-date quotation and scheduling algorithms for centralized and decentralized versions of the model.

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. 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 [19] and Cheng and Gupta [31] 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 [11], Weeks [32], Miyazaki [26], Baker and Bertrand [3], and Bertrand [4] 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.

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.

In the second part of the project, we consider a combined MTO-MTS supply chain composed of a manufacturer, served by a single supplier working in a stochastic multi-item environment. The manufacturer and the supplier have to decide which items to produce to stock and which ones to order. The manufacturer also has to quote due dates to arriving customers for make-to-order products. The manufacturer is penalized for long lead times, missing the quoted lead times and for high inventory levels. We consider several variations of this problem, and design effective heuristics to find the optimal inventory levels for each item and also design effective scheduling and lead time quotation algorithms for centralized and decentralized versions of the model.

We analyze the conditions under which an MTO or MTS strategy would be optimal to use for both the manufacturer and the supplier and find the optimal inventory levels. We also design effective scheduling and lead time quotation algorithms based on the algorithms in the first part of this thesis. We consider several variations of this problem. In particular, we focus on three online models. In the first model, the centralized model, both facilities are controlled by the same agent, who decides on the inventory levels at both facilities, schedules the jobs and quotes a due date to the arriving customer in order to achieve the end objective. In the second model, we develop a decentralized model with full information, in which the manufacturer and the supplier work independently from each other and make their own decisions but the manufacturer has complete information about both his own processes and the supplier. This model allows to explore the value of information exchange, and to determine if and when the cost and difficulty of implementing a centralized system are worth it. In the last model, the simple decentralized model, the manufacturer has no information about the supplier and makes certain assumptions about in order to decide on his inventory values and quote a due date to arriving customers. 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. We compare the centralized and decentralized models through extensive computational analysis to assess the value of centralization and information exchange.

In literature, MTO/MTS models are generally studied for single stage systems. Several researchers, such as Williams [33], Federgruen and Katalan [12] and Carr and Duenyas [8] assume that the decision of which items to produce-to-order and which ones to stock is made in advance and they try to find the best way to operate that system. Others, like Li [24], Arreola-Risa and DeCroix [1] and Rajagopalan[29], the make-to-stock/make-to-order distinction is a decision variable determined within the the model. In contrast to these models, we consider the scheduling and inventory decisions together and we analyze supply chain systems instead of a single facility model. We also integrate lead time quotation into these models in our research.

Finally, we consider more complex supply chain networks composed of several facilities with different relationships to each other. In this model, we consider a supply chain composed of several facilities and managed by a single decision agent who has complete information about these facilities. In addition, there are external suppliers to this supply chain outside the control of the manager of the supply chain, and this manager has only limited information about these external suppliers. Under these conditions, using the results from the two-facility supply chain, we design effective heuristics to be used by the manager to find optimal inventory levels, to sequence the orders at each facility, and to quote reliable and short lead times to customers.

In contrast to the models in literature, we integrate due-date quotation issues into combined MTO-MTS systems, consider different schedules and focus on supply chain models of this system. We consider scheduling, inventory and lead time quotation decisions together in our study. We develop models that provide guidance for deciding when to use MTS and when to use MTO approaches for single facility and for supply chain models, and for how to effectively operate the system to minimize system wide-costs in each case. We also quantify the value of centralization and information in this system by building decentralized and centralized models and obtaining good solutions to these models. To the best of our knowledge, this is the first study that analytically explores inventory decisions and lead time quotation together in the context of a supply chain, and that explores the impact of the supplier-manufacturer relationship on this system.

We perform extensive computational testing for each of the cases above to assess the effectiveness of our algorithms, and to compare the centralized and decentralized models in order to quantify the value of centralized control and information exchange in these supply chains. Since complete information exchange and centralized control is not always practical or cost-effective,

we also explore the value of partial information exchange for these systems. In this project, we are attempting to find answers to questions such as:

1. Which items should be produced MTO and which ones MTS at each step of the supply chain and what are the optimal levels of inventory for MTS items?
2. Which item should be produced next when a facility becomes available for production?
3. What is the optimal due-date that should be quoted to each customer at the time of arrival?
4. What is the benefit of a centralized supply chain as opposed to decentralized systems?
5. How much of the gains associated with centralization can be achieved through information exchange between supply chain members?

2. Scheduling and Due-Date Quotation in an MTO Supply Chain: We begin our analysis by considering a pure MTO supply chain and we focus on the scheduling and due-date quotation issues in this system. In this section, we consider a single 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. We consider several variations of this problem, and design effective due-date quotation and scheduling algorithms for centralized and decentralized versions of this model. We also complete an extensive computational experiment to evaluate the effectiveness of our algorithms and to assess the value of centralization and information exchange in this system.

2.1. Model and Main Results: We model two parties, a supplier and a manufacturer, working to satisfy customer orders. Customer orders, or jobs, i, $i = 1, 2, .., n$ 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 **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 times) is not available until the order arrives at the manufacturer (that is, until its release time r_i). 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 study, 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, for 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

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

where $T_i = (C_i - d_i)^+$ is the tardiness 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.

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 this model and these 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 identical independent 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 at 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 2.3.1, 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 resulting from applying this heuristic to an n job instance, and Z_n^* to be the optimal objective value for this instance.

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:*

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

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 that he is aware of the mean interarrival rate of jobs to his facility. In Section 2.2, 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 manufacturers knowledge about the supplier's system. Since the manufacturer is unaware of p_i^s values, he can't inquire 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 2.3.2, we propose an asymptotically optimal heuristic called $SPTA - SLC_{SD}$. Define $Z^{SPTA-SLC_{SD}}$ to be the objective function resulting from applying this heuristic to an n job instance, and Z_n^* to be the optimal objective value for this instance given the information available to the manufacturer.

Theorem 2. *Consider a series of randomly generated problem instances of the 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,*

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

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 2.3.3, we explain this heuristic in detail. We define $Z_n^{SPTA-SLC_{DIE}}$ to be the objective function resulting from applying this heuristic to an n job instance, and Z_n^* to be the optimal objective value for this instance given the information available to the manufacturer.

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,*

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

In Section 2.4, 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 centralized control.

In the next section, we introduce a preliminary model, and analyze this model. In section 2.3, we present our models, algorithms, and results in detail, and in section 2.4, we present the computational analysis of our heuristics and a comparison of centralized and decentralized

supply chain due date quotation models using our heuristics.

2.2. Preliminary: The Single Facility Model:

2.2.1. The Model: Although our ultimate goal is to analyze multi-facility systems, we begin with a preliminary analysis of a single facility 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, nonpreemptively, on a single machine. Each job has an associated type $l, l = 1, 2, ...k$, and each type has an associated finite processing time p_l, $p_l < \infty$. 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 study 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 $P_l, l = 1..k$ 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 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 for this model (and throughout this thesis) 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, [20]), we have found that a two-phase approach is asymptotically optimal. 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. This is the intuition behind the heuristic presented below.

2.2.2. 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 [18] 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 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. 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 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:

$$d_i = r_i + R_i + \sum_{j=1}^{pos[i]} (p_{list[j]}) + slack_i \qquad (2.1)$$

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 thesis, 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.

The remainder of this section 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_l, 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, [14]), 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:

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

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 [14]. Gross and Harris [14] 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:

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

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 $slack_i = (n-i)\psi_l\mu_l$

Also, if $\frac{\psi_l\mu_l}{\lambda} \geq 1$, 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 $slack_i = (n-i)\psi_l\mu_l$.

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 according to the shortest processing time available (SPTA) rule.

Due-Date Quotation:

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

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

2.2.3. 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 job instance, and let Z_n^* be the optimal objective function value for that instance.

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,*

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

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

2.3. 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 2.2, and develop effective algorithms for scheduling and due-date quotation for both the centralized and decentralized versions of these systems. These algorithms allow us to compare the value of centralization and information exchange in supply chains under a variety of different conditions.

2.3.1. The Centralized Model:

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.

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 [34] and Kaminsky and Simchi-Levi [21] 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,*

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

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:

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 $max\{t_i^{ms} + t_i^{mm} + slack_i^m - (d_i^s - r_i), 0\}$, 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 $slack_i^m$, $min\{\frac{(d_i^s - r_i - p_i^s)}{\lambda}, (n-i)\}$ 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.

Recall Theorem 1 in Section 2.1 which states that $SPTA_p - SLC$ is asymptotically optimal.

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,

ALGORITHM 2: SPTA$_p$-SLC

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

Due-Date Quotation:

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

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

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

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

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

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

we focus on the bottleneck facility and use an SPTA based schedule focusing on the processing times at the bottleneck facility. Then, again utilizing our two-phase approach, we adjust our due-date quotation algorithm for that schedule, accordingly. Please refer to Kaya [22] for more detail on the scheduling and due-date quotation algorithms for unbalanced cases.

2.3.2. 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 manufacturer are independent firms, with relatively limited information about each other. Implementing centralized control in these supply chains is typically 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.

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 suppliers 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 the customer 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.

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 the customer, 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:

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 Theorem 2 in Section 2.1 states that $SPTA - SLC_{SD}$ is asymptotically optimal under the conditions described above.

2.3.3. The Decentralized Model with Additional Information Exchange: As we will see in our computational analysis in Section 2.4, 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.

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 with 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.

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 2.2 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

ALGORITHM 5: SPTA-SLC$_{SD}$

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

Due-Date Quotation:

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

$d_i^s = r_i + \frac{q_i^s \mu^s}{2} + slack_i^s$

$$slack_i^s = \begin{cases} min\{(n-i), \frac{(\mu^s \frac{q_i^s}{2})}{\lambda - \frac{\mu^s}{2}}\} \frac{\mu^s}{2} & if \quad \lambda - \frac{\mu^s}{2} > 0 \\ (n-i)\frac{\mu^s}{2} & otherwise \end{cases}$$

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

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

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

$\lambda^m = max\{\lambda, \mu^s\}$

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

ALGORITHM 6: SPTA-SLC$_{DIE}$

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

Due-Date Quotation:

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

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

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

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

$\lambda^m = max\{\lambda, \mu^s\}$

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

is based on the slack algorithm SL for the single facility case. We summarize this approach below:

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 2.1 which states that $SPTA-SLC_{DIE}$ is asymptotically optimal.

2.4. 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.

2.4.1. 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 can 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 exponential distributions, hold the arrival rate constant, and vary the mean processing time, and relative due date and tardiness costs. The ratios of $Z_n^{SPTA-SL}$ to the lower bound for different combinations of n, c^T

and μ values with $c^d = 1$, $\lambda = 1$ and exponential inter-arrival and processing time distributions are presented in Table 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(process time)/E(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 use exponentially distributed inter-arrival and processing times. The ratios of the objective function $Z_n^{SPTA_p-SLC}$ and the total tardiness, T, to the lower bound for different combinations of n, μ^s and μ^m with $c^d = 1$, $c^T = 2$, $\lambda = 1$ and exponential inter-arrival and processing time distributions 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 a 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.

2.4.2. 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 2.3. Table 3 shows the ratios of the objective values of centralized and decentralized models obtained by simulating the system for $n = 3000$ jobs, where each entry shows the average of five runs with different random number streams. We used different combinations of μ^s and μ^m with $c^d = 1$, $c^T = 2$, $\lambda = 1$ and exponential inter-arrival and processing time distributions in this simulation. 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 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.

3. An Analysis of a Combined Make–to-Order/Make-to-Stock System: After analyzing the pure MTO model, we consider a combined MTO-MTS supply chain composed of a manufacturer, served by a single supplier working in a stochastic multi-item

Table 1: Ratios of $Z_n^{SPTA-SL}$ to the lower bound

# 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
μ=0.5					μ=1.5				
10	1.00962	1.01815	1.02882	1.09278	10	1.03373	1.06192	1.09715	1.30856
100	1.00258	1.00285	1.00317	1.00514	100	1.05480	1.05898	1.06421	1.09557
1000	1.00033	1.00036	1.00039	1.00062	1000	1.01558	1.01798	1.02098	1.03898
5000	1.00007	1.00008	1.00009	1.00014	5000	1.00753	1.00794	1.00846	1.01159
10000	1.00003	1.00003	1.00003	1.00006	10000	1.00541	1.00560	1.00583	1.00720
μ=1					μ=2				
10	1.10481	1.11671	1.13157	1.22079	10	1.04609	1.07794	1.11775	1.35663
100	1.01426	1.01605	1.01828	1.03165	100	1.07541	1.08011	1.08600	1.12129
1000	1.00677	1.00777	1.00903	1.01657	1000	1.01140	1.01636	1.02256	1.05976
5000	1.00270	1.00291	1.00319	1.00482	5000	1.00901	1.00925	1.00953	1.01128
10000	1.00177	1.00191	1.00219	1.00317	10000	1.00457	1.00491	1.00534	1.0079

Table 2: Ratios of the objective function $Z_n^{SPTA_p-SLC}$ and the total tardiness, T, to the lower bound

# 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
μ^s=1	μ^m=1			μ^s=2	μ^m=1			μ^s=5	μ^m=1	
10	1.0202	0.0118		10	1.0344	0.0102		10	1.0462	0.0121
100	1.0217	0.0059		100	1.0618	0.0096		100	1.0495	0.0122
1000	1.0087	0.0022		1000	1.0239	0.0052		1000	1.0127	0.0086
5000	1.0037	0.0010		5000	1.0102	0.0017		5000	1.0067	0.0023
μ^s=1	μ^m=2			μ^s=2	μ^m=2			μ^s=5	μ^m=2	
10	1.0344	0.0137		10	1.0263	0.0294		10	1.0471	0.0118
100	1.0374	0.0040		100	1.0635	0.0160		100	1.0507	0.0134
1000	1.0133	0.0025		1000	1.0241	0.0057		1000	1.0137	0.0075
5000	1.0071	0.0012		5000	1.0108	0.0039		5000	1.0080	0.0024
μ^s=1	μ^m=5			μ^s=2	μ^m=5			μ^s=5	μ^m=5	
10	1.0552	0.0144		10	1.0553	0.0259		10	1.0467	0.0349
100	1.0494	0.0053		100	1.0722	0.0159		100	1.0196	0.0338
1000	1.0191	0.0021		1000	1.0330	0.0046		1000	1.0117	0.0108
5000	1.0104	0.0016		5000	1.0143	0.0046		5000	1.0046	0.0089

Table 3: Ratios of the objective values of centralized and decentralized models

μ^s	μ^m	SD/cen	DIE/cen	SD/DIE		μ^s	μ^m	SD/cen	DIE/cen	SD/DIE
0.5	0.5	1.00211627	1.00215099	0.9999653		2	0.5	2.07968368	1.50892076	1.37825903
0.5	1	1.0025820	1.00308953	0.99949407		2	1	2.08234862	1.5133057	1.37602638
0.5	2	1.01378038	1.01417906	0.99960689		2	2	1.94756315	1.44041736	1.35208253
0.5	5	1.00604253	1.00615847	0.99988477		2	5	1.31118576	1.05354521	1.24454626
1	0.5	1.02696404	1.02616592	1.00077776		5	0.5	2.4082784	1.86390096	1.29206350
1	1	1.02055926	1.01543491	1.00504646		5	1	2.40706065	1.86304885	1.29200082
1	2	1.02373349	1.01594891	1.00766237		5	2	2.35828183	1.82580914	1.29163655
1	5	1.00838256	1.00560849	1.00275859		5	5	2.08264443	1.63575747	1.27319878

environment. In this case, both facilities are allowed to carry some level of inventory for each type of product instead of operating a pure MTO system. So, in addition to designing effective scheduling and lead time quotation algorithms, we want to find the optimal inventory levels that should be carried at each facility for this combined system. In this system, the manufacturer and the supplier have to decide which items to produce to stock and which ones to order. The manufacturer also has to quote due dates to arriving customers for make-to-order products. The manufacturer is penalized for long lead times, missing the quoted lead times and for high inventory levels. In the following sections, we consider several variations of this problem, and design effective heuristics to find the optimal inventory levels for each item and also design effective scheduling and lead time quotation algorithms for centralized and decentralized versions of this model. We also make extensive computational experiments to evaluate the effectiveness of our algorithms, to analyze the benefits of the combined MTO-MTS systems versus pure MTO or MTS systems and to compare the centralized and decentralized supply chains.

3.1. Properties of the Model: We model two parties, a supplier and a manufacturer in a flow shop setting. Customer orders, or jobs, j, arrive at the manufacturer at time r_j, and the manufacturer quotes a due date for each order, d_j, when it arrives. We assume that there are k types of jobs with mean processing times μ_i for each type i=1,2...k. Jobs arrive to the system with rate λ and each arriving job has a probability δ_i of being type i. We assume exponentially distributed, stationary and independent inter-arrival times, so, each job type i has an arrival rate of $\lambda_i = \lambda\delta_i$. To begin processing each order, the manufacturer requires a component specifically manufactured to order by the supplier. The component type i requires mean processing time μ_i^s at the supplier, and the order requires mean processing time μ_i^m at the manufacturer.

We assume that a base-stock policy is used for inventory control of MTS items and starting with R_i units of inventory, whenever a demand occurs, a production order is sent to replenish inventory. $R_i = 0$ means a make-to-order production system is employed for job type i. If demand is higher than the inventory level, then the extra demand is backlogged and satisfied later when it is produced. It is also assumed that the system is not congested and the interarrival times follow an exponential distribution.

We aim to minimize the total expected inventory plus lead time plus tardiness costs in this system. Thus, the objective function to minimize is

$$Z = \sum_{i=1}^{k} \{h_i E[I_i] + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\} \quad (3.1)$$

where h_i is the unit inventory holding cost, c_i^d is the unit lead time cost and c_i^T is the unit tardiness cost for type i. $E[I_i]$ denotes the mean amount of inventory, $E[d_i]$ denotes the lead time mean and $E[W_i - d_i]$ denotes the mean tardiness.

We consider three 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 the inventory levels and processing times both at the supplier and the manufacturer. So, this single entity decides on the inventory levels for each class as well as the production schedule and the lead times that should be quoted to each customer. In the decentralized, full information model, the manufacturer and the supplier are assumed to work independently, but the manufacturer has full information about both his processes and the supplier. They both try to minimize their own costs in a sequential game theoretic approach. The supplier acts first to determine his optimal inventory levels and then the manufacturer acts to minimize his own costs using the optimal inventory levels for the supplier. However, in the simple decentralized model, the manufacturer and the supplier still works independently from each other but now the manufacturer has no information about the supplier except the average delivery times of orders from the supplier.

The objective of our problem is to determine the optimal inventory levels, a sequence of jobs and a set of due dates such that the total cost $Z = \sum_{i=1}^{k} \{h_i E[I_i] + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$ is minimized. Clearly, to optimize this expression, we need to coordinate due date quotation, sequencing and inventory management and an optimal solution to this model would require simultaneous consideration of these three issues. However, the approach we have elected to follow for this model (and throughout this thesis) is slightly different. Observe that in an optimal off-line solution to this model, lead times would equal waiting times of jobs in the system. Thus, the off-line problem becomes equivalent to minimizing $\sum_{i=1}^{k} \{h_i E[I_i] + c_i^d E[W_i]\}$. Of course, in an online schedule, it is impossible to both minimize this function 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. In this approach, we first determine a scheduling approach designed to effectively minimize the sum of waiting times, and then based on that schedule, we find the optimal inventory levels to minimize $\sum_{i=1}^{k} \{h_i E[I_i] + c_i^d E[W_i]\}$ and design a due date quotation approach that presents due dates that are generally close to the completion times suggested by our scheduling approach. This is the intuition behind the heuristic presented below.

In section 2, we present effective scheduling and lead

time quotation algorithms for pure MTO versions of this system. We benefit from the properties of those algorithms and present modified versions of them for this system. However, note that, our results for the optimal inventory levels can be generalized to other schedules and lead time quotation algorithms as long as the schedule is independent of the workload or inventory levels in the system. The schedule only effects the stationary distributions of the number of jobs in the system. We consider SPTA and FCFS scheduling algorithms for our calculations but the results can be generalized to different schedules by recalculating the stationary distributions only. We consider different schedules in section 3.4 and compare them by computational analysis.

In the next section, we introduce a preliminary single facility model and analyze this model. In section 3.3, we present our supply chain models, algorithms, and results in detail, and in section 3.4, we present the computational analysis and a comparison of centralized and decentralized models.

3.2. Single Facility Model:

3.2.1. The Model: Although our ultimate goal is to analyze multi-facility systems, we begin with a preliminary analysis of a single facility system. We focus on finding the optimal inventory levels for this system and the conditions under which an MTO strategy or an MTS strategy would be optimal for this facility. We also focus on designing effective scheduling and due date quotation heuristics for this system.

In this section, we consider a single facility that combines make-to-stock and make-to-order policies to minimize its inventory and lead time related costs. Our objective in this system is to find an effective operating structure to minimize the inventory and lead time related costs. In solving this problem, we need to determine the optimal values for base-stock levels R_i for each job type as well as finding an effective scheduling and lead time quotation algorithm for these jobs to minimize the total costs. The model operates exactly as explained in the previous section but with just a single manufacturer without any supplier. We can think of this single facility model as a special case of the supply chain model where the processing times at the supplier are all 0, so the components are available to the manufacturer as soon as an order arrives. Also, the supplier doesn't need to hold any inventory and there is no cost related to the supplier.

Define two queues for the manufacturer in this model, the production queue and the order queue. Whenever a job arrives at the system, if that item exists in inventory, the demand is immediately satisfied from the inventory. Since that order is immediately satisfied, we don't place that job in the order queue. However, that job is still placed on the production queue in order to replenish the inventory. When an order arrives and there isn't any inventory of that item left, that job is placed at both the order queue and the production queue and a lead time needs to be quoted for that item. Thus, the order queue includes just the unsatisfied orders at any time while the production queue includes the items that are going to be produced both to satisfy orders and to replenish inventory. The order queue is just a subset of the production queue.

The production queue operates exactly the same way as a pure make-to-order system because even though we have inventory at hand for an item, we place that job in the production queue to replenish inventory. Thus, having inventory for an item does not impact the production process, but does decrease the due date costs since we satisfy those orders immediately and don't put them in the order queue.

Since the production queue operates exactly the same way as a pure MTO system, we employ the following algorithm, named SPTA-LTQ, for scheduling and lead time quotation, which is very similar to the algorithm SPTA-SL, designed for a pure MTO model single facility case in Section 2.2.

3.2.2. Analysis and Results: As mentioned in Section 3.1, we elect to employ a heuristic that first attempts to find a schedule to minimize the total completion times, and then finds the optimal inventory levels to minimize a function of the inventories and waiting times of the jobs in the system based on this schedule and then sets lead times that approximate the waiting times of jobs in the system with that schedule using the state of the system at the time of the arrival of the order, in an effort to minimize our objective function. 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. For a pure MTO system, although the problem of minimizing completion times is NP-Hard, Kaminsky and Simchi-Levi [18] found that the SPTA rule is *asymptotically optimal* for this problem, that is SPTA rule is optimal for minimizing the sum of completion times as the number of jobs goes to ∞. Also, note that this approach to sequencing does not take quoted due date or inventory levels into account, and is thus easily implemented.

For the lead time quotation algorithm LTQ presented above, we present the following lemma.

Lemma 1. *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 be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. Also, let $Z_n^{SPTA} = \sum_{i=1}^{n} c^d C_i$ denote the total weighted delivery times of orders where $C_i = r_i + W_i$ is the delivery time of order i and $Z_n^{SPTA-LTQ} = \sum_{i=1}^{n} \{c^d d'_i + c^T (C_i - d'_i)^+\}$*

ALGORITHM 7: SPTA-LTQ

Scheduling: Sequence the jobs in the production queue according to shortest processing time available (SPTA) rule.

lead time Quotation:

$$d_i = \begin{cases} 0 & if \;\; I_i > 0 \;\; at \;\; r_i \\ E[p_i] + E[M_j] + \frac{E[M_j]\lambda\psi_i\tau_i}{1-\lambda\psi_i\tau_i} & otherwise \end{cases}$$

where j is the job in the production queue that will be used to satisfy order i. M_j is the workload in front of job i at the time of arrival, ψ_i is the probability that an arriving job has processing time less than p_i and $\tau_i = E[p|p < p_i]$ is the expected processing time of a job given that it is less than p_i

denote the total due-date plus tardiness costs with the algorithm SPTA-LTQ where $d'_i = r_i + d_i$ is the quoted due-date. Then, the lead time quotation algorithm LTQ is asymptotically optimal to minimize this objective function assuming SPTA scheduling rule is used, that is almost surely,

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

We also state the following lemma for the lead time quotation algorithm LTQ presented above:

Lemma 2. *For the model explained above, for every class i, the expected value of the quoted lead time for class i is equal to the expected waiting time of a class i job, that is $E[d_i] = E[W_i]$ where W_i denotes the actual waiting time of job i in the order queue.*

We use the SPTA-LTQ algorithm for scheduling and lead time quotation as above and based on this schedule, we find the optimal inventory levels for each class in the subsequent parts of this section. However, note that different schedules can also be considered and the results of our subsequent analysis can be easily modified by updating the stationary distributions of the number of jobs in the system. We consider the above algorithm for our calculations since it is an effective one to quote reliable and short due-dates. Also, for the supply chain models, in Section 3.3, we present effective algorithms for scheduling and lead time quotation and use them in our analysis to find the optimal inventory levels, but again, our results regarding the optimal inventory levels can be generalized to different scheduling and lead time quotation algorithms easily.

To find the optimal inventory levels, we aim to minimize the following objective function:

$$\sum_{i=1}^{k} \{h_i E[I_i] + c_i^d E[W_i]\} \tag{3.2}$$

where $E[I_i]$ is the expected inventory level for job type i and $E[W_i]$ is the expected waiting time for job type i in the order queue. There is an obvious tradeoff between the inventory costs and the waiting time costs in this objective. We can decrease the waiting times of the class i jobs by holding additional inventory of that type but that will increase the inventory costs. Observe that holding additional inventory for an item effects the order queue only and does not effect the production queue. Thus, the waiting times and the lead time quotation procedure for the other classes aren't effected by this.

Lemma 3. *Objective function 3.2 is equivalent to the function $\sum_{i=1}^{k}\{h_i E[I_i] + c_i E[N_i]\}$ where $c_i = c_i^d/\lambda_i$ and $E[N_i]$ is the expected number of job type i in the order queue.*

Arreola-Risa and DeCroix [1] considers the same objective function $\sum_{i=1}^{k}\{h_i E[I_i] + c_i E[N_i]$ to minimize by considering a FCFS schedule. We begin by restating some of their results, expand them to include SPTA schedules and compare the two types of schedules in this section.

Lemma 4.

$$\begin{aligned} & \sum_{i=1}^{k}\{h_i E[I_i] + c_i E[N_i]\} \\ = \;& \sum_{i=1}^{k} h_i \sum_{x=0}^{R_i}(R_i - x) f_i(x) + c_i \sum_{x=R_i}^{\infty}(x - R_i) f_i(x) \end{aligned} \tag{3.3}$$

where $f_i(x)$ denotes the probability of having x jobs of type i in the production queue.

Lemma 5. *Dividing the problem into k subproblems according to their types, and the solving for each type individually gives the optimal solution for the whole problem.*

So, our problem decreases to minimizing $h_i \sum_{x=0}^{R_i}(R_i - x) f_i(x) + c_i \sum_{x=R_i}^{\infty}(x - R_i) f_i(x)$ for each i.

Theorem 6. *The optimal level of inventory R_i is the minimum value $x \geq 0$ that satisfies*

$$F_i(x) \geq \frac{c_i}{c_i + h_i}$$

Corollary 1. *Produce item i MTO if and only if $F_i(0) \geq \frac{c_i}{c_i+h_i}$*

Corollary 2. *An item's production moves towards MTO if its unit lead time cost, processing time or arrival rate decreases or unit holding cost increases.*

Note that the results above regarding optimal inventory levels hold for a variety of queueing disciplines, not restricted to a single server queue or an SPTA schedule, and they are also independent of the arrival or manufacturing process. However, these characteristics effect $F_i(x)$, the stationary distribution of the number in the system. To assess the effectiveness of our scheduling our algorithm SPTA and to explore how the schedule used in the system effects $F_i(x)$ and the objective function, we analyze two different schedules SPTA and FCFS and present the following two corollaries for these schedules. We also compare the objective function using these schedules through computational analysis in section 3.4. Also, one can extend these results for other kinds of schedules or queues with different characteristics by only considering the changes in $F_i(x)$ for that queue.

Corollary 3. *If FCFS scheduling rule is used in the production queue, instead of SPTA, assuming an M/G/1 queue, to minimize 3.3, it is optimal to produce product i MTO if and only if:*

$$\sum_{j=1}^{k} \delta_j E[e^{-\lambda_i \mu_j}] \leq \frac{(1-\delta_i) r_i}{r_i - (1-\rho)\delta_i}$$

where $r_i = \frac{c_i}{c_i+h_i}$, $\delta_i = \frac{\lambda_i}{\lambda}$ and $\rho = \sum_{i=1}^{k} \lambda_i \mu_i$

Corollary 4. *If SPTA scheduling rule is used in the production queue, assuming it is an M/G/1 queue, it will be optimal to produce product i MTO if and only if:*

$$\frac{(1-\rho)(\xi_i) + \lambda_b(1-\gamma_b(\xi_i)}{\lambda_i \gamma_i(\xi_i)} \geq r_i$$

where $\xi_i = \lambda_i + \lambda_a - \lambda_a \nu_a(\lambda_i)$, $r_i = \frac{c_i}{c_i+h_i}$, $\rho = \sum_{i=1}^{k} \lambda_i \mu_i$, $\lambda_a = \sum_{j=1}^{i-1} \lambda_j$ is the total arrival rate of jobs shorter than class i, $\lambda_b = \sum_{j=i+1}^{k} \lambda_j$ is the total arrival rate of jobs longer than class i, $\gamma_i(z) = E[e^{-z p_i}]$ is the Laplace transform associated with the processing time of class i and $\nu_a(z)$ is the solution of the equation $\nu_a(z) = \gamma_a(z + \lambda_a - \lambda_a \nu_a(z))$.

3.3. Supply Chain Models: In this section, we analyze the inventory decisions, scheduling and due-date quotation issues for two-stage supply chains using the results from our analysis in section 3.2. We develop effective heuristics to find the optimal inventory levels at both facilities and design effective algorithms for scheduling and due-date quotation for both the centralized and decentralized versions of these systems. These algorithms allow us to compare the value of centralization and information exchange in supply chains under a variety of different conditions.

When a manufacturer is working with a supplier, the components may not be immediately available to the manufacturer at the arrival time of an order. The manufacturer has to wait for some time for the components to arrive from his supplier before he can start working on that order. Thus, the supplier-manufacturer relationship effects the optimal levels of inventories that should be held as well as the scheduling and lead time quotation decisions.

We model this system as a two facility flow shop with a manufacturer and a supplier where both parties can choose to stock some of the items and use a make-to-order strategy for the others in a multi-item, stochastic environment. We assume that the supplier and the manufacturer employs a one-to-one replenishment strategy and a base-stock policy for inventory control of their items. The manufacturer starts with an inventory of R_i^m units of finished goods and the supplier starts with an inventory of R_i^s units of semi-finished goods that the manufacturer needs to complete his production.

We again define the production and order queues similar to the way we did in the single facility model, but in this case, for both the supplier and the manufacturer. When an order arrives, if that item is in the manufacturer's inventory, the order is immediately satisfied and a lead time of 0 is quoted. However, a production order of that class is sent to both the supplier and the manufacturer to replenish the inventory of the manufacturer. That order is not placed in the manufacturer's order queue until the semi-finished goods are delivered to the manufacturer by the supplier. If the supplier also has inventory of that class, he sends it directly to the manufacturer and that order appears in the manufacturer's production queue immediately. The supplier still places that order in his production queue to replenish his own semi-finished goods inventory. If that item is neither in the manufacturer's nor the supplier's inventory, then a lead time is quoted to the customer and a production order is sent to both facilities to satisfy this order. This order appears immediately in the supplier's production queue and after it is delivered from the supplier, it appears in the manufacturer's production queue. The manufacturer has to wait for some time for the semi-finished goods to be delivered to him by the supplier to put that order in his production queue. However, if the supplier has this item in its inventory, then that order immediately appears in the manufacturer's production queue as well as the supplier's production queue. A shorter lead time is quoted in this case since the customer only needs to wait for the production at the manufacturer and waiting time at the supplier is 0.

Let x_i^s and x_i^m denote the amount of jobs of type i in the supplier's and manufacturer's production queue, re-

spectively. Then, let N_i^s denote the amount of jobs of class i waiting in supplier's order queue that should be delivered to the manufacturer, I_i^s denote the amount of semi-finished goods inventory of class i, N_i^m denote the total number of customers of class i waiting in the system for their orders to be delivered and I_i^m denote the amount of finished goods inventory at the manufacturer of class i. Then,

$$\begin{aligned} N_i^s &= max\{x_i^s - R_i^s, 0\} \\ I_i^s &= max\{R_i^s - x_i^s, 0\} \\ N_i^m &= max\{x_i^m + max(x_i^s - R_i^s, 0) - R_i^m, 0\} \\ I_i^m &= max\{R_i^m - x_i^m - max(x_i^s - R_i^s, 0), 0\} \end{aligned} \tag{3.4}$$

3.3.1. The Centralized Supply Chain Model:

The Model In some systems, the manufacturer has a close relationship and perhaps even complete control over his supplier. For example,the manufacturer and the supplier may belong to the same firm. In those cases, a central agent that has complete information about both parties and makes all the decisions about both firms will obviously be much more effective in minimizing the total costs in the system than the individual parties would be.

In this section, we assume that the manufacturer and the supplier work as a single entity and they are both controlled by the same agent that has all the information about both sides. The decisions about the scheduling at both facilities and lead time quotation for the customer as well as the inventory levels for each party are made by this agent.

In the centralized model, our objective function to minimize is:

$$\sum_{i=1}^{k} h_i^s E[I_i^s] + h_i^m E[I_i^m] + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+ \tag{3.5}$$

where h_i^s is the unit holding cost of semi-finished goods at the supplier and h_i^m is the unit holding cost of finished goods at the manufacturer.

Analysis and Results In this system, we use an approach that is similar to the one we employed for the single facility case. We first find the optimal inventory levels for a schedule that is independent of the workload or inventory levels in the system (e.g. using a FCFS schedule in both facilities is such a schedule), to minimize the objective function

$$Z(R^s, R^m) = \sum_{i=1}^{k} \{h_i^s E[I_i^s] + h_i^m E[I_i^m] + c_i^d E[W_i]\} \tag{3.6}$$

Then, we present an effective scheduling algorithm consistent with this model to minimize the total waiting times of the jobs in the system and a lead time quotation algorithm that matches these waiting times.

Using the definitions 3.4 and the equations $c_i^d = \lambda c_i$ and $E[W_i] = E[N_i^m]/\lambda_i$ due to Little's law, and writing them explicitly, we get the objective function 3.7 to minimize in terms of the inventory amounts and the number of jobs in the production queues of the supplier and the manufacturer both of which operate as pure MTO systems.

$$\begin{aligned} & Z(R^s, R^m) = \sum_{i=1}^{k} Z(R_i^s, R_i^m) \\ = & \sum_{i=1}^{k} \{h_i^s E[I_i^s] + h_i^m E[I_i^m] + c_i E[N_i^m]\} \\ = & \sum_{i=1}^{K} \{h_i^s \sum_{y_i^s=0}^{R_i^s} (R_i^s - y_i^s) P(x_i^s = y_i^s) \\ + & h_i^m [\sum_{y_i^s=0}^{R_i^s-1} \sum_{y_i^m=0}^{R_i^m} (R_i^m - y_i^m) P(x_i^s = y_i^s, x_i^m = y_i^m) \\ + & \sum_{y_i^s=R_i^s}^{R_i^s+R_i^m} \sum_{y_i^m=0}^{R_i^s+R_i^m-y_i^s} (R_i^s + R_i^m - y_i^s - y_i^m) \\ & * P(x_i^s = y_i^s, x_i^m = y_i^m)] \\ + & c_i [\sum_{y_i^s=0}^{R_i^s-1} \sum_{y_i^m=R_i^m}^{\infty} (y_i^m - R_i^m) P(x_i^s = y_i^s, x_i^m = y_i^m) \\ + & \sum_{y_i^s=R_i^s}^{\infty} \sum_{y_i^m=R_i^s+R_i^m-y_i^s}^{\infty} (y_i^s + y_i^m - R_i^s - R_i^m) \\ & * P(x_i^s = y_i^s, x_i^m = y_i^m)]\} \end{aligned} \tag{3.7}$$

where R^s and R^m are the array of inventory levels at the supplier and the manufacturer and x_i^s and x_i^m are the number of class i jobs at the supplier's and manufacturer's production queue, respectively.

Observe that due to our assumption about the schedule used in the facilities, both the supplier and the manufacturer's production queue operates independent of R^m. In addition, the supplier's production queue is independent of R^s. However, the manufacturer's production queue depends on R^s, thus $f_m(x)$, the stationary distribution of number of jobs at the manufacturer, is a function of R^s.

Theorem 7. *For fixed inventory levels R_i^s for each class i at the supplier, the optimal levels of inventory for the manufacturer are the minimum R_i^m values that satisfy:*

$$\begin{aligned} & P(x_i^s > R_i^s, x_i^s + x_i^m \leq R_i^s + R_i^m) \\ + & P(x_i^s \leq R_i^s, x_i^m \leq R_i^m) \geq \frac{c_i}{c_i + h_i^m} \end{aligned} \tag{3.8}$$

Corollary 5. *If the manufacturer is working with a pure MTO supplier, then the manufacturer's optimal inventory levels for each class are the minimum R_i^m values that satisfy:*

$$P(x_i^s + x_i^m \leq R_i^m) \geq \frac{c_i}{c_i + h_i^m} \tag{3.9}$$

When we look at the supplier inventory levels, observe that $f_m(x)$, the probabilities of the number of jobs at the manufacturer's production queue, depends on the inventory levels R^s at the supplier which makes the problem very hard to solve analytically.

However, we can find the optimal solutions under some special conditions. We state the following theorem for one of these conditions.

Theorem 8. *For the centralized model, if $h_i^s \geq h_i^m$ for a product type i, then an MTO strategy for type i at the supplier, that is holding no inventory of type i at the supplier, is optimal.*

For other cases, finding the optimal solution is very hard since the manufacturer's production queue depends on the inventory levels R^s at the supplier which makes the problem very hard to trace analytically. To find an approximation on the optimal R^s values, we assume that the change in the stationary distributions of the number of jobs at the manufacturer is negligible w.r.t. a change in the amount of the inventory levels at the supplier and try to find the optimal R^s values using this approximation. In that case, we can divide the problem into K subproblems and analyze each class separately. Still, we can't state a result similar to Theorem 7 for the inventory values at the supplier, since the objective function 3.7 doesn't possess the convexity structure in R_i^s for fixed R_i^m. (i.e. $Z(R_i^s + 1, R_i^m) - Z(R_i^s, R_i^m)$ is not nondecreasing in R_i^s for every R_i^m.)

So, to determine the optimal levels of R^s, we employ a one-dimensional search on R_i^s. For each R_i^s, we calculate the optimal values of R_i^m, calculate the total cost using the objective function 3.7 and pick the pair with minimum cost for each class i. However, we can decrease the search space using some properties of the objective function.

Lemma 6. *For this model, the function $Z(R_i^s, R_i^m)$ as given in 3.7 is supermodular for every class i.*

Theorem 9. *Let $\bar{R}_i^s \geq 0$ be the minimum value that satisfies*

$$P(x_i^s \leq \bar{R}_i^s) \geq \frac{c_i}{c_i + h_i^s}$$

Then, the optimal level of inventory $R_i^{s*} \leq \bar{R}_i^s$

So, there is no need to search for R_i^{s*} beyond $\bar{R}_i^s$.

Corollary 6. *It is optimal for the supplier to use an MTO strategy to produce product type i if $F_i^s(0) \geq \frac{c_i}{c_i + h_i^s}$*

In general, for the fixed inventory value R_i^m at the manufacturer, if for every R_i^s, $\frac{\Delta^2 Z(R_i^s, R_i^m)}{\Delta^2 R_i^s} \geq 0$, then the following theorem holds. An example of this case occurs, when $h_i^s \geq h_i^m$.

Theorem 10. *For fixed inventory levels R_i^m at the manufacturer, if $\frac{\Delta^2 Z(R_i^s, R_i^m)}{\Delta^2 R_i^s} \geq 0$, the optimal levels of inventory at the supplier are the minimum value R_i^s that satisfies:*

$$\begin{aligned} & (h_i^m + c_i) P(x_i^s > R_i^s, x_i^s + x_i^m \leq R_i^s + R_i^m) \\ + \quad & (h_i^s + c_i) P(x_i^s \leq R_i^s) \geq c_i \end{aligned} \tag{3.10}$$

We present effective scheduling and lead time quotation algorithms in Section 2 for a pure MTO supply chain system. Using the same ideas as in those algorithms, we design modified versions of them for our system with inventories. For the centralized supply chain model, assuming independent and exchangeable processing times, the algorithm $SPTA_p - LTQ_C$ is outlined below.

Note that this algorithm is consistent with our assumptions for this model since the schedule is independent of the workload or inventory in the system and the lead times quoted with this algorithm satisfies the relation $E[d_i] = E[W_i]$.

For the lead time quotation algorithm LTQ_C presented above, we present the following lemma.

Lemma 7. *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. Also, let $Z_n^{SPTA_p} = \sum_{i=1}^n c^d C_i$ denote the total weighted delivery times of orders with the $SPTA_p$ schedule where $C_i = r_i + W_i$ is the delivery time of order i and $Z_n^{SPTA_p - LTQ_C} = \sum_{i=1}^n \{c^d d_i' + c^T (C_i - d_i')^+\}$ denote the total due-date plus tardiness costs with the algorithm $SPTA_p - LTQ_C$ where $d_i' = r_i + d_i$ is the quoted due-date. Then, the lead time quotation algorithm LTQ_C is asymptotically optimal to minimize this objective function for this centralized system assuming that $SPTA_p$ scheduling rule is used to sequence jobs, that is almost surely,*

$$\lim_{n \to \infty} \frac{Z_n^{SPTA_p - LTQ_C} - Z_n^{SPTA_p}}{Z_n^{SPTA_p}} = 0$$

The schedule used in the system only effects the stationary distributions of the number of jobs in the system, (i.e.f(x)). For the schedule we presented above, the supplier is using an SPTA rule w.r.t. total processing time

ALGORITHM 8: SPTA$_p$-LTQ$_C$
Scheduling: Process the jobs according to $SPTA_p$ (SPTA based on total processing time $p_i = p_i^s + p_i^m$) at the supplier and FCFS at the manufacturer.
lead time Quotation:

$$d_i^m = \begin{cases} 0 & if \;\; I_i^m > 0 \;\; at \;\; r_i \\ E[p_i^m] + t_i^{mm} & if \;\; I_i^m = 0, \;\; I_i^s > 0 \;\; at \;\; r_i \\ d_i^s + E[p_i^m] + max\{t_i^{ms} + t_i^{mm} + slack_i^m - d_i^s, 0\} & otherwise \end{cases}$$

where
$d_i^s = E[p_i^s] + E[M_i^s] + \frac{E[M_i^s]\lambda pr\{p<p_i\}E\{p^s|p<p_i\}}{1-\lambda pr\{p<p_i\}E\{p^s|p<p_i\}}$
$t_i^{ms} = \sum_{j\in A} E[p_j^m]$ where A=set of jobs in supplier queue scheduled before job i and will be sent to manufacturer.
$t_i^{mm} = \sum_{j\in B} E[p_j^m]$ where B=set of jobs in manufacturer queue at time r_i
$slack_i^m = (d_i^s - p_i^s)\lambda pr\{p < p_i\}E\{p^m|p < p_i\} + \sum_{j\in L} min\{(d_i^s - p_i^s)\lambda_j, I_j^s\}E[p_j^m]$ where L is the set of jobs that will be scheduled after i at the supplier

of the jobs and the manufacturer is scheduling his jobs according to FCFS. Because the production queue at the supplier is independent of R^s and R^m and it operates just like the single facility described in Section 3.2, the supplier can find the stationary distribution of the number in his system using the probability generating function for a single facility queue.

However, when we look at the manufacturer side, the inventory levels at the supplier effects the processes at the manufacturer, since the whole supply chain is described as an *inventory queue*. This makes the problem very difficult analytically. We use the common decomposition approach to approximate the stationary distributions of the number of jobs in the manufacturer side. Note that, when the interarrival times are exponentially distributed in a queueing model like the well known Jackson network model presented first in Jackson[16], the departure process is poisson distributed. Since the inter-arrival time to our system is exponentially distributed, we approximate the departure process from the supplier with a poisson distribution and thus we assume that the arrivals to the manufacturer are poisson. So, we treat the processes at the manufacturer as a single facility with multiple classes with poisson arrivals for FCFS schedule.

3.3.2. The Decentralized Supply Chain, Full Information 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 manufacturer are independent firms, with relatively limited information about each other. Implementing centralized control in these supply chains is typically 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.

Although centralization in supply chains is generally a difficult and costly thing to do, at the same time, with a decentralized system, companies might lose a lot of their profits. In some cases, instead of completely centralizing the system, the companies might just choose to share all their information with each other to increase their profits. Thus, we are motivated by the fact that information exchange in some supply chains might increase the profits high enough so that complete centralization will be unnecessary. In this case, the manufacturer has all the information about the whole system but has no control over the supplier's decisions. For this system, we find the optimal inventory levels for the manufacturer and the supplier as well as an effective scheduling and due-date quotation algorithm. We also analyze the differences between this decentralized model and the centralized model through computational analysis in the next section.

The Model In this decentralized supply chain model, we assume that the two parties work independently from each other and aim to minimize their own costs. However, the manufacturer has full information about the processes at the supplier as well as his own processes. Since the supplier works independently from the manufacturer and tries to minimize his own costs, the results from the single facility case applies for the supplier to determine the optimal inventory levels at his facility and to sequence the orders to minimize his own completion times. Then, for that sequence and inventories at the supplier, we find the optimal inventory levels and design an effective scheduling and lead time quotation algorithm for the manufacturer.

Analysis and Results Since the supplier is independent from the manufacturer, he operates exactly the same way as the single facility explained in the previous section. Thus, the supplier's objective is to minimize $\sum_{i=1}^{k}\{h_i^s E[I_i^s] + c_i^d E[d_i^s] + c_i^T E[W_i^s - d_i^s]^+\}$. Since the supplier works independently from the manufacturer, he uses the scheduling and lead time quotation algorithm as explained in the single facility case and the results in section 3.2 holds for the supplier. Thus, the optimal inventory levels for class i jobs at the supplier, R_i^s, are the minimum $x \geq 0$ that satisfy

$$F_i^s(x) \geq \frac{c_i}{c_i + h_i^s}$$

Using the relations given in section 3.2, the optimal inventory levels for the supplier can be obtained.

Corollary 7. *The optimal inventory levels for the supplier in the decentralized, full information case will be greater than or equal to the optimal level in the centralized case if the same schedules are used for both cases.*

Assuming that the supplier uses the optimal inventory levels for his facility, the manufacturer tries to minimize his own costs for those fixed R^s values. So, we try to minimize the objective function:

$$\sum_{i=1}^{k}\{h_i^m E[I_i^m] + c_i E[N_i^m]\}$$

Using the definitions 3.4 and writing them explicitly, for fixed R^s, we will get the objective function for the manufacturer to minimize as

$$\begin{aligned}
Z(R^s, R^m) &= \sum_{i=1}^{k}\{h_i^m E[I_i^m] + c_i E[N_i^m]\} \\
&= \sum_{i=1}^{K}\{h_i^m \sum_{y_i^s=0}^{R_i^s-1}\sum_{y_i^m=0}^{R_i^m}(R_i^m - y_i^m) \\
&\quad * P(x_i^s = y_i^s, x_i^m = y_i^m) \\
&+ \sum_{y_i^s=R_i^s}^{R_i^s+R_i^m}\sum_{y_i^m=0}^{R_i^s+R_i^m-y_i^s}(R_i^s + R_i^m - y_i^s - y_i^m) \\
&\quad * P(x_i^s = y_i^s, x_i^m = y_i^m) \\
&+ c_i[\sum_{y_i^s=0}^{R_i^s-1}\sum_{y_i^m=R_i^m}^{\infty}(y_i^m - R_i^m)P(x_i^s = y_i^s, x_i^m = y_i^m) \\
&+ \sum_{y_i^s=R_i^s}^{\infty}\sum_{y_i^m=R_i^s+R_i^m-y_i^s}^{\infty}(y_i^s + y_i^m - R_i^s - R_i^m) \\
&\quad * P(x_i^s = y_i^s, x_i^m = y_i^m)]\} \qquad (3.11)
\end{aligned}$$

Observe that both the supplier and the manufacturer's production queue operates independent of R^m for fixed R^s.

Then, for fixed inventory levels R_i^s for each class i at the supplier, the optimal levels of inventory R_i^m for the manufacturer can be found by Theorem 7. Also, the optimal inventory levels for a manufacturer working with a pure MTO supplier satisfy corollary 5.

Corollary 8. *The optimal inventory level for the manufacturer in the decentralized, full information case will be less than or equal to the optimal level in the centralized case if the same schedules are used both cases.*

Corollary 9. *Using an MTO strategy for class i jobs at the manufacturer is optimal iff*

$$P(x_i^s \leq R_i^s, x_i^m \leq 0) \geq \frac{c_i}{c_i + h_i^m} \qquad (3.12)$$

where R_i^s is the optimal inventory level for class i at the supplier.

Since the manufacturer is working independently from the supplier, he also uses an SPTA schedule according to his own processing times to sequence his jobs and employs a lead time quotation algorithm similar to the centralized model since he has full information about the supplier. So, we use the following scheduling and lead time quotation algorithm $SPTA{-}LTQ_{DFI}$ for the manufacturer which satisfies $E[d_i] = E[W_i]$.

For the lead time quotation algorithm LTQ_{DFI} presented above, we present the following lemma.

Lemma 8. *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. Also, let $Z_n^{SPTA} = \sum_{i=1}^{n} c^d C_i$ denote the total weighted delivery times of orders with the $SPTA$ schedule where $C_i = r_i + W_i$ is the delivery time of order i and $Z_n^{SPTA-LTQ_{DFI}} = \sum_{i=1}^{n}\{c^d d_i' + c^T(C_i - d_i')^+\}$ denote the total due-date plus tardiness costs with the algorithm $SPTA-LTQ_{DFI}$ where $d_i' = r_i + d_i$ is the quoted due-date. Then, the lead time quotation algorithm LTQ_{DFI} is asymptotically optimal to minimize the objective function for this decentralized system with full information assuming that SPTA schedule is used for sequencing jobs, that is almost surely,*

$$\lim_{n\to\infty} \frac{Z_n^{SPTA-LTQ_{DFI}} - Z_n^{SPTA}}{Z_n^{SPTA}} = 0$$

In this case, the supplier is working as a single facility and using SPTA rule w.r.t. his own processing times. So, the stationary distributions of the number of jobs at the supplier can be found using the pgf for a single facility queue.

The manufacturer is also scheduling his jobs according to SPTA. We again use the decomposition approach

ALGORITHM 9: SPTA-LTQ$_{DFI}$
Scheduling: Schedule the jobs using SPTA rule according to p^m.
lead time Quotation:

$$d_i^m = \begin{cases} 0 & if\ \ I_i^m > 0\ \ at\ \ r_i \\ E[p_i^m] + t_i^{mm} + slack_i^m & if\ \ I_i^m = 0,\ \ I_i^s > 0\ \ at\ \ r_i\ \ where\ \ slack_i^m = \frac{t_i^{mm}\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}{1-\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}} \\ d_i^s + E[p_i^m] + M_i^m + slack_i^m & otherwise \end{cases}$$

where
$d_i^s = E[p_i^s] + E[M_i^s] + slack_i^s$
$slack_i^s = \frac{E[M_i^s]\lambda pr\{p^s<p_i^s\}E\{p^s|p^s<p_i^s\}}{1-\lambda pr\{p^s<p_i^s\}E\{p^s|p^s<p_i^s\}}$
$M_i^m = \max\{t_i^{ms} + t_i^{mm} + sl_i^m - d_i^s, 0\}$
$t_i^{ms} = \sum_{j\in A} E[p_j^m|j \in A]$ where A=set of jobs of class k at supplier queue that will be sent to manufacturer s.t. $E[p_k^m] < E[p_i^m]$ and $E[p_k^s] < E[p_i^s]$.
$t_i^{mm} = \sum_{j\in B} E[p_j^m|j \in B]$ where B=set of jobs at manufacturer queue at time r_i with $E[p^m] < E[p_i^m]$.
$sl_i^m = (slack_i^s + E[M_i^s])\lambda pr\{p^s < p_i^s, p^m < p_i^m\}E\{p^m|p^m < p_i^m\} + \sum_{j\in L} min\{(slack_i^s + E[M_i^s])\lambda_j, I_j^s\}E[p_j^m|j \in L]$ where L is the set of jobs of class k s.t. $E[p_k^m] < E[p_i^m]$ and $E[p_k^s] > E[p_i^s]$.
$slack_i^m = \frac{M_i^m\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}{1-\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}$

to approximate the stationary distribution of the number of jobs at the manufacturer's site, similar to the centralized model and assume that the arrivals to the manufacturer are exponentially distributed. Thus, the stationary distribution of the number of jobs at the manufacturer can also be found by using the pgf for a single facility queue. However, since the manufacturer and supplier have different processing times and thus different schedules, their stationary distributions will also be different although both are found through the same pgf.

3.3.3. The Simple Decentralized Model: In many real-world systems, the supplier and the manufacturer are independent firms and they have very little information or no information at all 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. Most of the time, the manufacturer is only aware of the average time it takes for the supplier to process an order and send it to him. In such a decentralized supply chain system, we find the optimal inventory levels for the manufacturer and an effective way to schedule the orders and to quote due-dates to the customers without any information from the supplier.

The Model In this model, we again assume that the manufacturer and the supplier work independently from each other, but distinct from the full information model, the manufacturer has no information about the supplier, and thus he can't deduce the production schedule or inventory levels at the supplier. The supplier still behaves the same way as before and the results obtained in the previous section for the supplier still hold.

However, in this case, the manufacturer only knows the average time it takes for the supplier to deliver a job type i, denoted by $E[d_i^s]$. Thus, for each job, the manufacturer acts as if each job of type i is going to be delivered to him from the supplier after $E[d_i^s]$ time units. Based on this assumption, we determine the optimal inventory levels for the manufacturer and design an effective scheduling and lead time quotation algorithm.

Analysis and Results Since the supplier acts as a single facility as in the previous section, the optimal inventory levels for class i jobs, R_i^s are the minimum values that satisfy

$$F_i^s(R_i^s) \geq \frac{c_i}{c_i + h_i^s}$$

However, in this case, since the manufacturer only knows the average time it takes for an order of type i to be delivered by the supplier, when an order arrives, he assumes that order will be delivered by the supplier to him after $E[d_i^s]$ time units. Observe that if we model the supplier as a $M/D/\infty$ queue with deterministic processing times $E[d_i^s]$ for type i and without any inventories, each job of type i will take exactly $E[d_i^s]$ time units to be delivered from the supplier to the manufacturer as assumed by the manufacturer in our system for this case and we can use the same analysis and the results in previous sections for this model. Thus, the objective function for the

manufacturer to minimize is

$$Z(R^m) = \sum_{i=1}^{k}\{h_i^m E[I_i^m] + c_i E[N_i^m]\}$$

$$= \sum_{i=1}^{K}\{h_i^m[\sum_{y_i^s+y_i^m=0}^{R_i^m}(R_i^m - y_i^s - y_i^m)P(x_i^s + x_i^m = y_i^s + y_i^m)]$$

$$+c_i[\sum_{y_i^m+y_i^s=R_i^m}^{\infty}(y_i^s + y_i^m - R_i^m)P(x_i^s + x_i^m = y_i^s + y_i^m)]\}$$

For this case, since we model the supplier as a $M/D/\infty$ queue with no inventories, using Corollary 5, the optimal levels of inventory are the minimum R_i^m values that satisfy:

$$P(x_i^s + x_i^m \leq R_i^m) \geq \frac{c_i}{c_i + h_i^m}$$

We find the stationary distributions of the jobs at the supplier using the $M/D/\infty$ queue model with processing times $E[d_i^s]$. Since we assume that each order is delivered to the manufacturer at time $r_i + E[d_i^s]$ by the supplier, the processes at the supplier doesn't effect the arrival process to the manufacturer and the distribution of the interarrival time of jobs to the manufacturer follows the same exponential distribution of the interarrival time of jobs to the system. Thus, the arrival process of the jobs to the manufacturer is also poisson. Both the supplier and manufacturer are working as a single facility in this case, and thus using the relations for a single facility queue, we can find the stationary distributions of the number of jobs and the inventory levels for the manufacturer and the supplier.

For this case, since the manufacturer focuses only on his individual objective, he uses an SPTA schedule according to his own processing times. We assume that the manufacturer has no information about the supplier and is unaware of the schedule or inventory levels there, but that from his previous experience, he can deduce an average delivery time for each class of products. So, he uses this piece of information to quote due dates to his customers. Thus, we use the following scheduling and lead time quotation algorithm $SPTA - LTQ_{SD}$ for this simple decentralized model:

For the lead time quotation algorithm LTQ_{SD} presented above, we present the following lemma.

Lemma 9. *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 the manufacturer be also i.i.d random variables and bounded and the processing times and interarrival times be independent of each other. Also, let $Z_n^{SPTA} = \sum_{i=1}^{n} c^d C_i$ denote the total weighted delivery times of orders with the $SPTA$ schedule where $C_i = r_i + W_i$ is the delivery time of order i and $Z_n^{SPTA-LTQ_{SD}} = \sum_{i=1}^{n}\{c^d d_i' + c^T(C_i - d_i')^+\}$ denote the total due-date plus tardiness costs with the algorithm $SPTA - LTQ_{SD}$ where $d_i' = r_i + d_i$ is the quoted due-date. Then, the lead time quotation algorithm LTQ_{SD} is asymptotically optimal to minimize this objective function for this simple decentralized system assuming that SPTA schedule is used for sequencing jobs, that is almost surely,*

$$\lim_{n\to\infty} \frac{Z_n^{SPTA-LTQ_{SD}} - Z_n^{SPTA}}{Z_n^{SPTA}} = 0$$

3.4. Computational Analysis: Using the inventory values determined by the results in the previous sections and the scheduling and lead time quotation algorithms, we designed several computational experiments to assess the effectiveness of our algorithms, how the combined MTS/MTO systems differ from pure MTO or MTS systems and the differences between centralized and decentralized supply chains. For the centralized system, we assume that both parties cooperate and use the heuristics explained in section 3.3.1 for finding the inventory levels, scheduling and lead time quotation. For the decentralized systems, we assume that each facility acts independently from each other and use the heuristics that will minimize their own costs as explained in sections 3.3.2 and 3.3.3 for finding the inventory levels, scheduling and lead time quotation. We implement our heuristics using C++ and present the results in the following sections.

3.4.1. Effectiveness of a Combined System and Efficiency of Heuristics for a Single Facility: First, we consider a single facility system using $n = 1000$ jobs for each simulation. Then, for each of these n job instances, the ratios of the total costs $Z = \sum_{i=1}^{k}\{h_i E[I_i] + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$, on average, are displayed in Table 4 where $E[I_i]$ denotes the average inventory, $E[d_i]$ denotes the average quoted lead time and $E[W_i - d_i]^+$ denotes the average tardiness in a simulation run for product type i.

We consider different scenarios regarding different number of job types k and different multipliers h, c^d and c^T for cost functions to evaluate the effect of these parameters on our system. For each of these cases, we designed 10 scenarios with different arrival and processing rates, each having exponential distributions. Each value in Table 4 denotes the average of these 10 scenarios.

We present the comparison of combined MTO-MTS system with pure systems and with different schedules for a single facility using $c^d = 2$ and different combinations of h, c^T and k in Table 4. The first and second columns contain the comparison of the combined MTS/MTO system using SPTA-LTQ algorithm with pure MTS and MTO systems, respectively. Observe that the combined system, on average, provides a 20% decrease in costs as opposed to the pure MTS systems and a 15% decrease as opposed to the pure MTO systems. Also, in the third column, we compare the costs with the SPTA-LTQ algorithm for this combined system with a relevant

ALGORITHM 10: SPTA-LTQ$_{SD}$
Scheduling: Process the jobs according to shortest processing time at the manufacturer.
lead time Quotation:

$$d_i = \begin{cases} 0 & if\ \ I_i^m > 0\ \ at\ \ r_i \\ E[p_i^m] + t_i^{mm} + slack_i^m & if\ \ I_i^m = 0,\ \ I_i^s > 0\ \ at\ \ r_i\ \ where\ \ slack_i^m = \frac{t_i^{mm}\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}{1-\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}} \\ E[d_i^s] + E[p_i^m] + M_i^m + slack_i^m & otherwise \end{cases}$$

where

$E[d_i^s]$ is the average delivery time of class i products from the supplier.
$M_i^m = max\{t_i^{mm} + \frac{E[d_i^s]pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}{E[p^s]} - E[d_i^s], 0\}$
$slack_i^m = \frac{M_i^m\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}{1-\lambda pr\{p^m<p_i^m\}E\{p^m|p^m<p_i^m\}}\}$

lead time quotation algorithm for a FCFS schedule. We see that SPTA-LTQ algorithm decreases the costs about 15% on average as opposed to a FCFS-LTQ algorithm.

We also see the effect of the parameters on the system performance in Table 4. Unsurprisingly, as the unit inventory holding cost increases while all other parameters remain constant, the combined system moves toward a pure MTO system and gives much better results than pure MTS systems since holding inventory becomes much more costly as h increases. The system is effected the same way as the unit due-date cost, c^d, decreases because in that case lead times become less important and an MTO system becomes much more attractive as c^d decreases. We also see that SPTA-LTQ algorithm gives much better results than FCFS-LTQ algorithm, as h increases (alternatively c^d decreases) because the schedule has an important effect on completion times, thus duedates, in MTO systems, while for MTS systems, since the orders are satisfied from the inventory, minimizing the completion times of the jobs is not so important. On the other hand, as the unit tardiness cost, c^T, increases while all other parameters remain constant, MTO systems give worse results and pure MTS systems become much more attractive. Also, we see that the performance difference between SPTA-LTQ algorithm and FCFS-LTQ algorithm becomes less and less as c^T increases and FCFS-LTQ algorithm start to give better results as c^T becomes very high. This happens because, with FCFS-LTQ algorithm, the future arrivals don't effect the completion times of the jobs that have already arrived, thus we can quote the lead times exactly as the completion times of the jobs and there will be no tardiness. However, with SPTA-LTQ algorithm, there will be some tardiness cost and as c^T increases, the cost of tardiness overcomes the gains in total completion times with SPTA-LTQ algorithm and the total costs with FCFS-LTQ becomes lower than the total costs with SPTA-LTQ algorithm.

3.4.2. Effect of Inventory Decisions on Supply Chains: We next explore the effect of inventory decisions on our system without considering the lead time quotation. So, for a fixed schedule, we compare the objective functions $Z' = \sum_{i=1}^{k}\{h_i^s E[I_i^s] + h_i^m E[I_i^m] + c_i^d E[W_i]\}$ to explore only the effect of inventory decisions in this system unaffected by lead time quotation. Recall that we made some assumptions about the interaction between the supplier and the manufacturer regarding the stationary distributions of the number of jobs at the manufacturer and we also assumed that inventory values of different types at the supplier don't effect the stationary distributions of other types at the manufacturer. We explore how these assumptions effect the optimal inventory levels and the objective function with this computational study. We also compare the centralized and decentralized versions of the combined MTS/MTO supply chain with this objective function. Table 5 shows the ratios of objective functions Z' for different cases using $c^d = 2$ and different combinations of h^s, h^m and k.

To explore the effectiveness of our algorithms, we compare the objective function using our inventory levels for the centralized model with the minimum objective function for that case. In our simulations, we find the minimum objective functions by trying all the possible combinations of inventories of each type for the supplier and the manufacturer and selecting the one that gives the best objective value. We use these minimum objective values as lower bounds in our simulations. However, this process takes a very long time, especially when the number of types are big. Although we can find the optimal inventories by trying all possible solutions for the cases we analyzed in this experiment since they are relatively of small size, note that this method becomes almost impossible to apply as the problem size gets bigger and bigger. For example consider the case when there are 10 different jobs and the inventory of each job type at the supplier and the manufacturer can take a value from 0 to 5. In that case we need to evaluate $(5 * 5)^{10}$ different possible solutions for the trial and error method and if each of them takes a nanosecond(10^{-9} seconds), the whole process takes more than a day and becomes almost impossible to apply for even bigger problems. However,

Table 4: Comparison of combined MTO-MTS system with pure systems and with different schedules for a single facility

$\frac{h=0.5}{c^T=2.5}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$	$\frac{h=1}{c^T=2.1}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$
k=3	0.963	0.526	0.942	k=3	0.829	0.880	0.894
k=5	0.972	0.613	0.973	k=5	0.798	0.844	0.850
k=10	0.947	0.734	0.926	k=10	0.778	0.831	0.724
$\frac{h=1}{c^T=2.5}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$	$\frac{h=1}{c^T=3}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$
k=3	0.832	0.906	0.913	k=3	0.847	0.873	0.922
k=5	0.814	0.827	0.897	k=5	0.825	0.829	0.908
k=10	0.795	0.872	0.762	k=10	0.808	0.820	0.795
$\frac{h=2}{c^T=2.5}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$	$\frac{h=1}{c^T=5}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$
k=3	0.724	0.936	0.846	k=3	0.880	0.845	0.931
k=5	0.719	0.934	0.823	k=5	0.896	0.796	0.911
k=10	0.683	0.967	0.695	k=10	0.913	0.753	0.924
$\frac{h=5}{c^T=2.5}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$	$\frac{h=1}{c^T=10}$	$\frac{Z_{MTO-MTS}}{Z_{MTS}}$	$\frac{Z_{MTO-MTS}}{Z_{MTO}}$	$\frac{Z_{SPTA-LTQ}}{Z_{FCFS-LTQ}}$
k=3	0.573	1	0.837	k=3	0.925	0.743	1.032
k=5	0.551	1	0.776	k=5	0.946	0.719	1.131
k=10	0.472	0.987	0.672	k=10	0.953	0.651	1.096
average	0.754	0.858	0.838		0.866	0.799	0.926

with our heuristic, there is no interaction between the different types and we consider them separately. Also, we only do a one-dimensional search over the supplier's inventory levels up to an upper bound. Thus, for the same case, we only need to evaluate $5 * 10$ possible solutions at the worst case which can be done immediately and can be easily applied to problems of much bigger size.

We construct our lower bound by using the inventory values found by the trial and error method that gives the minimum objective value and compare it with the objective value obtained by using the inventory values found by our heuristic. The first column in Table 5 compares these two objective functions for the centralized supply chain model. We see that the inventory values found by our heuristic are very close to the optimal inventory values and there is only a 5% difference, on average, between the minimum costs and the costs obtained by using our inventory values. The difference is mainly due to our assumption that having inventory of type i at the supplier don't effect the stationary distributions of the number of jobs at the manufacturer which isn't the case in reality.

When we compare the centralized and decentralized models, we see that the costs with the centralized model is, on average, 10% less than the decentralized model with full information. The cost savings due to inventory decisions increase to more than 15% when we compare the centralized model with the simple decentralized case. We also see that, without centralization, if the manufacturer has full information about the supplier, he can decrease the costs by about 7% by only adjusting his own inventory levels without changing anything else. We see that if the manufacturer had control over the supplier, he could cut his costs significantly. However, even if the manufacturer didn't have control over the supplier but had full information about the whole system, he still can cut his costs and increase his profits.

We can also see the effects of the parameters h^s, h^m, c^d and k on the system performance. We see that as the inventory holding cost at the supplier, h^s, increases while everything else remains the same, our heuristic gives closer results to the lower bound. A intuitive explanation for this is; as h^s increases, the optimal inventory levels at the supplier and their effect on the manufacturer decreases as we assumed in our approximation. Besides, if the supplier uses a pure MTO strategy, then our heuristic finds the optimal solution since our approximations become exact for a pure MTO supplier. The same effect occurs as h^m decreases because in that case it would be better to carry inventories at the manufacturer instead of the supplier and the inventory levels at the supplier and their effect on the manufacturer decreases as above. Similarly, as c^d decreases, the inventory levels at the supplier decreases, too and our heuristic gives closer results to the optimal solution.

When we compare the supply chain models, we see that as h^s decreases, decentralized models give closer results to the centralized one. This is because as h^s decreases, the optimal inventory levels at the supplier for the centralized model become close to the upper bound we presented in Theorem 9 which is also the optimal inventory level for the supplier in the decentralized models assuming that the same schedule is used in both cases. Thus, as h^s decreases, the inventory levels at the supplier, thus the inventory levels at the manufacturer for the centralized and decentralized models become close to each other and the objective values with the decentral-

ized models become closer to the centralized model objective value. We can see the same effect as h^m increases, because in the centralized model as h^m increases, the inventory levels at the manufacturer decreases and the inventory levels at the supplier increases becoming close to the upper bound. Thus, as explained above, the decentralized models give closer results to the centralized one.

3.4.3. Effectiveness of a Combined System and Efficiency of Heuristics for Supply Chains: We then complete a similar study which includes lead time quotation, and compare the objective functions $Z = \sum_{i=1}^{k}\{h_i^s E[I_i^s]+h_i^m E[I_i^m]+c_i^d E[d_i]+c_i^T E[W_i-d_i]^+\}$. We use the multipliers $h^s = 0.5$, $h^m = 1$, $c^d = 2$ and $c^T = 2.5$ for this case. Through this analysis, we compare the costs for the whole system for the centralized and decentralized models. In the first column in Table 6, we compare the total costs for the centralized model using the algorithm $SPTA_p - LTQ_C$ with the algorithm that schedules the jobs according to FCFS in both systems and quotes lead times similar to our heuristic. We conclude that the schedule used to produce the jobs has an important effect on the total costs and we see that, on average, our heuristic performs about 20% better than the commonly used schedule FCFS in the industry. Observe from Table 4 that this was also the case for the single facility case.

Also, when we compare the second columns of both tables, we see that having full information about the supplier helps the manufacturer a lot to quote better lead times and there is very little difference between costs due to lead time quotation for centralized and decentralized with full information cases. The difference between costs is mainly because of inventory decisions in this comparison.

However, when we compare the simple decentralized model with the centralized model, we see that the cost difference increases significantly. Since the manufacturer has very little information about the supplier, he can no longer quote reliable due-dates and the costs increase dramatically and the costs with the simple decentralized model are about 40% worse than the centralized model. Observe that, although there weren't much difference between the decentralized model with full information and the simple decentralized model in Table 5, in Table 6, this difference increases significantly to 30% mainly due to lead time quotation. We conclude that having full information about the supplier is critical to the manufacturer for lead time quotation. In addition, the inventory costs can be decreased dramatically if the manufacturer has complete control over the supplier in addition to the full information case.

4. Complex Supply Chain Networks: In this section, we consider more complex supply chain networks composed of several facilities with different classes of relationships. Utilizing from the results from the previous sections, we design effective heuristics for determining inventory levels, sequencing the orders at each facility and quoting reliable and short lead times to customers for these complex networks. We also complete extensive computational experiments to evaluate the effectiveness of our algorithms and to analyze the benefits of the combined MTO-MTS systems versus pure MTO or MTS systems.

4.1. Model: We model a multi-facility supply chain that is composed of a main manufacturer and its internal and external suppliers. The manufacturer receives orders from the customers and delivers it immediately if it has that product in its inventory or quotes a due date to the customer if it doesn't. We assume that there are a total of N facilities in the supply chain and there are K types of different products this firm offers to its customers, each with a different demand rate. The manufacturer wants to quote short and reliable due-dates and doesn't want to keep too much inventory either. Thus, we want to minimize the objective function $\sum_{i=1}^{K}\{(\sum_{j=1}^{N} h_{ij}E[I_{ij}]) + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$ composed of the inventory costs, lead time costs and tardiness costs.

If the processing times are short and the total time required to produce a product is less than the desired lead time, then the company doesn't need to keep any inventory and uses an MTO approach. However, if the total processing time is more than the desired lead time, due to the stochastic nature of demand and lead times, the company has to start processing before the orders actually arrive and keep some work-in-process or finished goods inventory to achieve the desired lead time. Instead of keeping only finished goods inventory at the manufacturer, we can choose to store inventory at other facilities and use an MTO approach at others. We aim to find the optimal locations to store inventory in this chain to minimize the system-wide inventory costs and design effective algorithms to quote short and reliable lead times.

We consider a supply chain like in Figure 1 where the same firm owns some of the facilities at different locations drawn with rectangles and also has independent outside suppliers drawn with circles. The company has total control over its own facilities; however, it has no control over outside suppliers. We can also think of the internal suppliers as different machines in a facility where a single decision agent makes all of the decisions and the external suppliers as the suppliers of this company which this decision agent can not control. The decision agent can choose to stock some WIP inventory in this system instead of choosing to stock only the finished goods inventory.

Since the demand is stochastic, the firm needs to carry some inventory to respond to customer orders in a short time. We assume that an initial inventory amount (i.e. safety stocks) of x_{ij} for raw materials and y_i for finished

Table 5: Effect of inventory decisions on the centralized and decentralized systems

$\frac{h^s=0.5}{h^m=2}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$	$\frac{h^s=0.5}{h^m=0.6}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$
k=3	0.916	0.963	0.896	0.930	k=3	0.964	0.834	0.794	0.952
k=5	0.933	0.925	0.849	0.917	k=5	0.959	0.877	0.808	0.921
k=10	0.929	0.923	0.832	0.901	k=10	0.967	0.892	0.836	0.937
$\frac{h^s=1}{h^m=2}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$	$\frac{h^s=0.5}{h^m=1}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$
k=3	0.921	0.935	0.868	0.928	k=3	0.943	0.945	0.890	0.942
k=5	0.949	0.914	0.855	0.935	k=5	0.909	0.856	0.788	0.920
k=10	0.952	0.922	0.862	0.934	k=10	0.918	0.875	0.808	0.923
$\frac{h^s=1.9}{h^m=2}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$	$\frac{h^s=0.5}{h^m=5}$	$\frac{Z'_{LB}}{Z'_{Cen}}$	$\frac{Z'_{Cen}}{Z'_{DFI}}$	$\frac{Z'_{Cen}}{Z'_{SD}}$	$\frac{Z'_{DFI}}{Z'_{SD}}$
k=3	0.969	0.856	0.802	0.937	k=3	0.922	0.961	0.881	0.917
k=5	1	0.839	0.784	0.934	k=5	0.911	0.975	0.874	0.896
k=10	1	0.803	0.771	0.960	k=10	0.936	0.953	0.841	0.882
average	0.952	0.898	0.835	0.931		0.939	0.908	0.836	0.921

Table 6: Comparison of centralized and decentralized supply chains for combined MTO-MTS system

	$Z_{SPTA_p-LTQ_C}/Z_{FCFS-LTQ}$	Z_{Cen}/Z_{DFI}	Z_{Cen}/Z_{SD}	Z_{DFI}/Z_{SD}
k=3	0.859	0.954	0.679	0.712
k=5	0.775	0.857	0.575	0.671
k=10	0.682	0.893	0.584	0.654
Average	0.772	0.902	0.613	0.680

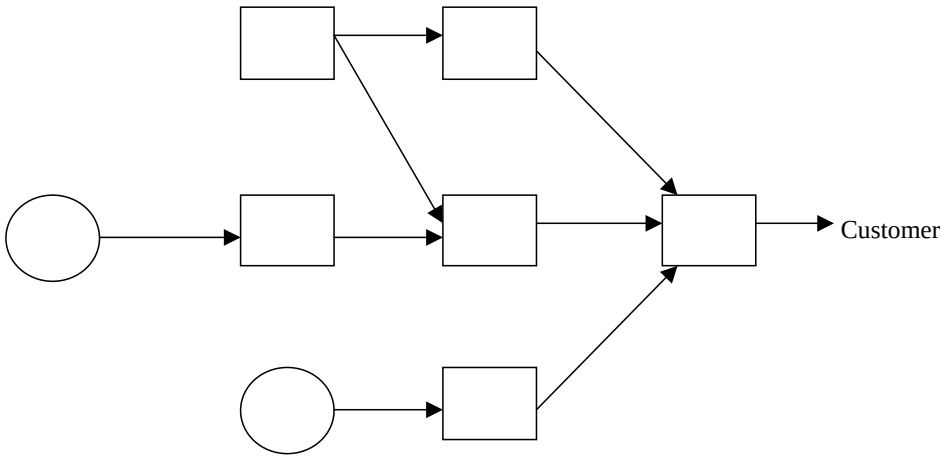

Figure 1: Supply Chain Structure

goods are held at facility i. A customer order triggers a production order in the system and the orders have to wait a lead time depending on the amount of initial inventories. We desire short lead times without carrying too much inventory, so we aim to minimize a cost function:

$$Z = \sum_{i=1}^{K}\{(\sum_{j=1}^{N} h_{ij}E[I_{ij}]) + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$$

that is composed of the lead times and the safety stocks in the whole system.

Each order k requires a processing time of p_{ik} at facility i and a transshipment time t_{ijk} to move from facility i to j. Also for each individual order l, w_{il} denotes the waiting time of order l at facility i, i.e. w_{il} is the time starting with the arrival of the required parts for order l to facility i and ending with the processing of that order at facility i. Note that if there is no queue at facility i when order l receives to facility i, $w_{il} = p_{ik}$. Otherwise, $w_{il} = p_{ik} + \{$waiting time in queue of facility $i\}$. Also, we assume that each facility quotes a due-date f_{il} for each order to its downstream member and eventually the manufacturer to its customers.

For this system, we need to decide on a scheduling rule and a due-date quotation algorithm as well as how much inventory to store at each of the facilities. The objective of our problem is to determine the optimal inventory levels, a sequence of jobs and a set of due dates such that the total cost $\sum_{i=1}^{K}\{\sum_{j=1}^{N} h_{ij}E[I_{ij}] + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$ is minimized. Clearly, to optimize this expression, we need to coordinate due date quotation, sequencing and inventory management and an optimal solution to this model would require simultaneous consideration of these three issues. However, the approach we have elected to follow for this model (and throughout this project) is slightly different. Observe that in an optimal offline solution to this model, lead times would equal actual waiting times of jobs in the system. Thus, the problem becomes equivalent to minimizing $\sum_{i=1}^{K}\{(\sum_{j=1}^{N} h_{ij}E[I_{ij}]) + c_i^d E[W_i]\}$. Of course, in an online schedule, it is impossible to both minimize this function 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. In this approach, we first determine a scheduling approach designed to effectively minimize the sum of completion times, and then based on that schedule, we find the optimal inventory levels to minimize $\sum_{i=1}^{K}\{(\sum_{j=1}^{N} h_{ij}E[I_{ij}]) + c_i^d E[W_i]\}$ and then with these inventories, we design a due date quotation approach that presents due dates that are generally close to the completion times suggested by our scheduling approach.

4.2. Analysis and Results: For this model, we use the same ideas for scheduling and lead time quotation as we use in previous sections and adjust them for this more complex situation. However, when we first attempted to use the same analysis that we did in section 3 to find the optimal inventory levels at each of the facilities, we see that the model becomes very complex and difficult to analyze. Thus, we develop another approach to approximate the optimal inventory levels that should be stored at each facility for each product type for this system. We explain our approach to find the inventory levels as well as the algorithms we use for scheduling and lead time quotation in detail in the following sections.

4.2.1. Single Product Type Model: In this section, we assume that the firm produces only one type of product and that they use FCFS scheduling rule at all facilities since there is no difference between orders.

To find the optimal inventory levels at each of the facilities, assuming $x \leq p$, we use the relation $l = p - x$ at each facility where p is the time units required to process an order, l is the lead time and x is the length of time that the safety stock that we carry is worth for a specified service level. Every time a customer order arrives, we start to produce one item to replenish inventory or to satisfy future orders. Then, the demand for the first x time units is satisfied from the inventory and since the first order finishes processing at time p, after the inventory is depleted at time x, the first order that arrives, can only be satisfied at time p causing a lead time $l = p - x$. Note that this is a conservative approach and overestimates the amount of inventory that we need to carry. In reality, the same lead time might be achieved by carrying much less inventory but holding this amount of inventory ensures that the specified service level is achieved for this lead time and every order is satisfied with a lead time less than l for sure at this service level.

We define the parameters and variables and write the LP formulation of the model below:

Parameters:

$i =$ subscript used to describe the facilities in the supply chain starting with 1 denoting the manufacturer.

$v_i =$unit value of parts produced at facility i

$p_i =$ processing time of parts at facility i

$t_{ij} =$ transshipment time to move parts from facility i to j

$h_{ij}^1 =$ unit raw material inventory cost at facility i for the parts produced at facility j

$h_i^2 =$ unit finished part inventory cost at facility i.

$c^d =$ cost of a unit increase in response time to orders.

$S =$ set of facilities that belong to the firm

$E =$ set of external suppliers

$P_i =$ set of facilities that are immediate predecessors of facility i

f_i for $\forall i \in E =$ committed response time of orders from external supplier $i \in E$

Variables:

x_{ij} = units of time that the safety stock of raw materials received from facility j and stored at facility i to achieve the desired service level.

y_i = units of time that the finished parts stored at facility i that achieves the desired service level.

w_i = waiting time of orders at facility i, i.e. w_i is the average time starting with the arrival of the required parts for the production of an order at facility i and ending with the processing of that order at facility i. Note that if there is no queue at facility i when an order receives to facility i, $w_i = p_i$. Otherwise, $w_i = p_i + \{$waiting time in queue of facility $i\}$.

f_i for $\forall i \in S$ = total lead time of orders up to facility $i \in S$, i.e lead time between the arrival of a customer to the system and the completion of the parts required for that order at facility i.

$$
\begin{aligned}
Min \quad & \sum_{i=1}^{N}[\sum_{j=1}^{N}\{h_{ij}^1 x_{ij}\} + h_i^2 y_i] + c^d f_1 \\
s.t. \quad & max\{max_{i\in P_j}\{max\{f_i + t_{ij} - x_{ij}, 0\}\} \\
& \qquad +w_i - y_i, 0\} \le f_j \qquad for\ \ \forall j \in S \\
& All\ \ variables \ge 0
\end{aligned}
\tag{4.1}
$$

In this system, if we assume that there is no congestion or queues in the system (e.g. there are assembly lines or infinite servers at all the facilities and each order is put on the assembly line and starts processing as soon as the required materials for that order receives to that facility), there will be no queues and we use only the processing times as the waiting times at the facilities. However, if we assume that there is a single server (or $M < \infty$ servers) at a facility, then there will be congestion and queues in the system. In that case, we approximate the waiting time of orders at facility i by using the expected waiting time of a job in that queue with arrival rate D_i and mean processing time p_i. We use $E[W_i]$ instead of p_i and use these values in the LP formulation.

Also, if we assume that the manufacturer has a committed response time to its customers, which is generally the case in competitive markets, we fix the committed response time and try to minimize the total inventory costs with the above LP model with the fixed lead time as a parameter instead of a variable.

For due-date quotation, observe that if there is inventory at a facility, the components are immediately satisfied from that facility and the order doesn't have to wait any time for the processing at or before that facility. If the waiting time of an order at a facility is constant and is not effected by the other orders (e.g. an assembly line system or an infinite server queue system), then the due-dates are quoted by the following algorithm.

However, if there are queues in the system, then the lead time of an order is effected by the other orders and the state of the system at time r_k, the arrival time of order k to the system. For this model, since there is only one product and a FCFS scheduling rule is used, we don't need to consider future arrivals and we can quote the due dates by only considering the current state of the system at the time of arrival. An order has to wait only for the processing of orders that are already at the queue of a facility. Let w_i^k denote the waiting time of an order k at facility i.(Note that $w_i^k = p_i$ for the assembly line system) and U_i^k denote the set of orders that arrived to the system before r_k but not yet finished processing at facility i. Then, the due-date for an order k is quoted according to the following algorithm.

4.2.2. Multiple Product Types Model: In this section, we assume that the firm produces multiple products with different characteristics, (i.e different processing times, arrival rates, and supply chain architecture). In this case, in addition to deciding on inventory levels and due-date quotation, we also design an effective scheduling algorithm to process the jobs at each facility since products have different characteristics and the sequence of jobs will effect the completion times and response times to customers.

For the multiple product model, again if we assume that there is no congestion or queues in the system (e.g. there are assembly lines at all the facilities and each order is put on the assembly line and starts processing as soon as the required materials for that order receives to that facility), then the model becomes exactly the same as the single product case because the different orders and product types don't effect each other and don't cause any congestion in the system. In this case, we use the same LP model 4.1 to decide on the inventory amounts and the due-date quotation algorithm 11 to quote due-dates. There is no need for a scheduling algorithm since each arriving job is immediately placed under process without waiting in this system.

However, if we assume that there is a single server (or $M < \infty$ servers at a facility), then there will be congestion and queues in the system. In that case, we approximate the waiting time of orders at facility i by using a similar approach as in the single product model. For the schedule used at facility i, we find the expected waiting time of an order type l in that queue with arrival rate D_i^l and mean processing time p_i^l. In this case, the expected waiting time for type l, $E[W_i^l]$, depends on the schedule used in that facility and other products. We use $w_i^l = E[W_i^l]$ as the waiting time of jobs of type l at facility i and use these values in the LP formulation 4.1.

For this case, we also use the same idea for due-date quotation as in Algorithm 12. However, in this case, the scheduling algorithm doesn't have to be FCFS and the facilities might choose other sequences to minimize total completion times. Although the problem of minimizing completion times at a single facility is NP-Hard, Kamin-

ALGORITHM 11:

Step 1: At the time of arrival of an order k to the system, denoted by r_k, form a new subgraph, G', of the supply chain considering the inventories at time r_k. Starting from the manufacturer, do a depth-first search on the graph and add a facility to the subgraph if there is no inventory at hand at that facility. If there is inventory at a facility do not add that to the subgraph and do not go further from that node.

Step 2: Let the length between two facilities i and j be $l_{ij} = p_i + t_{ij}$.

Step 3: Add a node 0 at the end of the subgraph and connect it with all the facilities that don't have a predecessor and let $l_{0j} = 0$. Using these l_{ij} values, find the longest path from node 0 to the manufacturer.

Step 4: Set the length of the longest path as the lead time for that order.

ALGORITHM 12:

Step 1: Set $d_i^k = 0$ if there is inventory of type l at facility i and $d_i = f_i$ if $i \in E$.

Step 2: Form the subgraph G' as in Algorithm 11 and put a facility i into set F if facility $i \in G'$ and $i \in S$. Set F denotes the set of facilities that belong to the firm and haven't been quoted a due-date yet.

Step 3: If facility $i \in F$ and $j \notin F$ for $\forall j \in P_i$, then set due-date for order k at facility i as below and delete facility i from set F.

$d_i^k = max_{j \in P_i}\{d_j^k + t_{ji}^k\} + w_i^k$

$w_i^k = max\{sum_{j \in U_i^k} p_i^j - max_{j \in P_i}\{d_j^k + t_{ji}^k\}, 0\} + p_i^k$

Step 4: Stop if set F is empty and set d_1^k as the lead time for order k. Return to step 3 otherwise.

sky and Simchi-Levi [18] shows that the SPTA rule is *asymptotically optimal* for this problem. Under the SPTA heuristic, each time a job completes processing, the shortest available job which has yet not been processed is selected for processing. Also, note that this approach to sequencing does not take quoted due date into account, and is thus easily implemented.

We consider FCFS and SPTA scheduling algorithms to use at the facilities. If FCFS schedule were used at all facilities, then the due-date quotation algorithm would be exactly the same as Algorithm 12. However, if SPTA schedule is used at a facility, then we consider future arrivals, too, in addition to the current state of the system because future arrivals might be scheduled before previous orders and increase their delivery times.

Xia, Shantikumar and Glynn [34] and Kaminsky and Simchi-Levi [21] independently proved that for a flow shop model with n machines, if the processing times of a job on each of the machines are independent and exchangeable, processing the jobs according to the shortest total processing time $p_i = \sum_{j=1}^{n} p_i^j$ at the first facility and processing the jobs on a FCFS basis at the others is asymptotically optimal if all the release times are 0.

We design a scheduling algorithm for our system based on this result. For each product type, we find the longest path in the production network of that product and find the total processing time of each product type by summing the processing types on this path. Then, we schedule the jobs according to shortest total processing times at the facilities that are at the end of the supply chain network and use a FCFS schedule at the other facilities. We explain our scheduling algorithm with due-date quotation below in algorithm 13.

In this system let Pr_l denote the arrival probabilities for each product type $l = 1..K$ with sum equal to 1. Let set M_l denote the set of product types that are going to be scheduled before product type l and $\psi_l = \sum_{k \in M_l} Pr_k$ is the probability that an arriving job is going to be scheduled before job type l. $\mu_i^l = \sum_{k \in M_l}\{Pr_k E[p_i^k]\}$ is the expected processing time of such a job at facility i and λ is the mean inter-arrival time of orders. Also, N_i^k denotes the number of orders that arrived after order k but scheduled before it at facility i and w_i^k denotes the waiting time of order k at facility i. Note that $w_i^k = p_i$ for the assembly line system . Then, the scheduling and due-date quotation algorithm for an order k is as below:

4.3. Computational Analysis: We performed several computational experiments in order to evaluate the performance of the algorithms explained above for different cases, and we also compared the objective of solutions based on our heuristics with the objective functions achieved using traditional methods, and with a lower bound. We design a complex supply chain network using different processing times, transshipment times between facilities, unit holding costs and unit waiting costs and implement our heuristics in C++. Whenever needed, we solve the LP model 4.1 using ILOG AMPL/CPLEX 7.0.

Consider a supply chain network for a single product type as shown in figure 2. The meanings of the numbers in figure 2 are explained in figure 3. In this network, S_i denotes the facilities that belong to the same firm and E_i denotes the external suppliers.

ALGORITHM 13:
Scheduling:

Step 1: Find the total time required to process each product type l(i.e. the longest path from the suppliers at the end of the chain to the manufacturer in the supply chain for product type l.) and denote it by T_l.

Step 2: Define set L to be set of facilities that have no internal supplier. Schedule the jobs according to shortest T_l at a facility i if $i \in L$ and use FCFS at the other facilities.

lead time Quotation:

Step 3: Set $N_i^k = 0$ for all i, $d_i^k = 0$ if there is finished goods inventory of type l at facility i and $d_i = f_i$ if $i \in E$.

Step 4: Form the subgraph G' as in Algorithm 11 and put a facility i into set F if facility $i \in G'$ and $i \in S$. Set F denotes the set of facilities that belong to the firm and haven't been quoted a due-date yet.

Step 5: If facility $i \in F$ and $j \notin F \; \forall j \in_i$, then set due-date for order k at facility i as below and delete facility i from set F. Let U_i^k denote the set of orders that are already in the system at time r_k and is scheduled before job k but not yet finished processing at facility i

If facility $i \in L$, then

$$d_i^k = max_{j \in P_i}\{d_j^k + t_{ji}^k\} + p_i^k + w_i^k + slack_i^k$$

$$w_i^k = max\{sum_{j \in U_i^k} p_i^j - max_{j \in P_i}\{d_j^k + t_{ji}^k\}, 0\}$$

$$slack_i^k = \begin{cases} min\{\frac{w_i^k \psi^l \mu_i^l}{\lambda - \psi_l \mu_i^l}, (n-i)\psi_l \mu_i^l\} & if \;\; \lambda - \psi_l \mu_i^l > 0 \\ (n-i)\psi_l \mu_i^l & otherwise \end{cases}$$

$$N_i^k = slack_i^k / \mu_i^l$$

If facility $i \notin L$

$$d_i^k = max_{j \in P_i}\{d_j^k + t_{ji}^k\} + p_i^k + w_i^k$$

$$w_i^k = max\{sum_{j \in U_i^k} p_i^j + max_{j \in P_i}\{N_j\}\mu_i^l - max_{j \in P_i}\{d_j^k + t_{ji}^k\}, 0\}$$

$$N_i = max_{j \in P_i} N_j$$

Step 6: Stop if set F is empty and set d_1^k as the lead time to the customer. Return to step 5 otherwise.

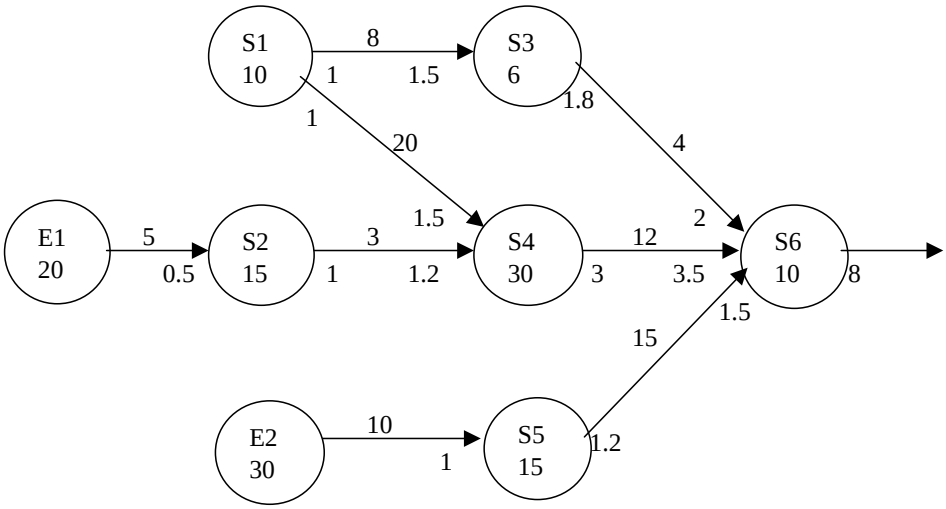

Figure 2: Supply Chain Network Example

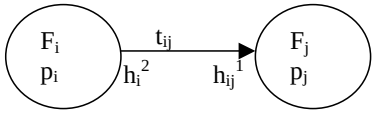

Figure 3: Explanation of the values

Table 7: Comparison of combined strategy with pure strategies for an assembly line system

	h=8,c^d=10	h=8,c^d=5	h=4, c^d=5
$Z_{MTO-MTS}/Z_{MTS}$	0.526	0.451	0.814
$Z_{MTO-MTS}/Z_{MTO}$	0.424	0.710	0.655

Table 8: Comparison of combined strategy with pure strategies for a single server model

	h=8,c^d=10	h=8,c^d=5	h=4, c^d=5
$Z_{MTO-MTS}/Z_{MTS}$	0.581	0.515	0.842
$Z_{MTO-MTS}/Z_{MTO}$	0.492	0.769	0.677

4.3.1. Effect of Inventory Positioning without Congestion Effects in the System: We start by examining the case where the waiting time of a job at facility i is deterministic and equal to p_i as in an assembly line model or an infinite server model with deterministic processing times. Since, the demand is stochastic, the firm needs to keep some safety stock to achieve the desired service level. In Table 7, we compare the optimal objective values of the formulation 4.1 allowing to hold inventory at every facility as opposed to holding no inventory at all or holding only finished good inventories. The ratios of the objective function values of formulation 4.1 with the combined strategy over the costs with pure MTS and MTO strategies for different combinations of inventory holding cost at the last facility, h, and unit lead time cost, c^d are shown in Table 7. Traditionally, firms either use an MTO strategy without keeping any inventory or an MTS strategy keeping only finished goods inventory. However, as we show in Table 7, using a combined strategy is much more beneficial for minimizing the cost function $\sum_{j=1}^{N} h_j E[I_j] + c^d E[W]$ where $E[I_j]$ denotes the average inventory at facility j and $E[W]$ denotes the average waiting time of customers in the whole system. For example, for the case where the holding costs are as shown in figure 2 except that the holding cost for finished goods at the last facility is 4 ($h = 4$) and $c^d = 5$, with a pure MTS strategy, we need to keep a finished goods inventory of 95 units with a cost of 380. With a pure MTO strategy, the lead time will be 95 and the cost is 475. However, with a combined strategy with $y_{S_1} = 10$, $y_{S_2} = 15$, $y_{S_4} = 12$, $y_{S_6} = 40$, $x_{E_1,S_2} = 25$, $x_{E_2,S_5} = 40$ $x_{S_1,S_4} = 20$ $x_{S_2,S_4} = 3$, the total cost will be 307.1. The cost with the combined strategy is significantly lower than the costs with pure strategies.

Consider another example in which the industry lead time is 30 and the firm tries to achieve this lead time by keeping some inventory. If the firm only keeps finished goods inventory, then $y_{S_6} = 65$ and the total inventory cost is 260. However, by keeping inventory at other facilities, with the same lead time, the total inventory cost can be decreased to 187.1 with $y_{S_1} = 10$, $y_{S_2} = 15$, $y_{S_6} = 22$, $x_{E_1,S_2} = 25$, $x_{E_2,S_5} = 28$ $x_{S_1,S_4} = 20$ $x_{S_2,S_4} = 3$. In addition, the firm will be able to cut the lead time by half to 15 with a cost of 247.1 which is even less than the cost with the initial strategy.

4.3.2. Effect of Inventory Positioning with Congestion Effects for Single Product Type Model: If we consider a single server model for each facility with stochastic processing times, we find the mean waiting time of a job at each facility and use these values as the waiting times. In that case, the ratios of the objective values in formulation 4.1 with the combined strategy over the pure strategies are shown in Table 8.

Note that these costs are found through the objective functions in our LP formulation which are calculated using the maximum lead time for the specified service level for all jobs and using the corresponding safety stock levels. However, in reality each job has a different lead time depending on the congestion in the system and the inventory levels fluctuate due to the stochastic nature of demand and production rates. We simulate this stochastic system assuming that there is a single server at each facility with mean processing times and other values as given in figure 2 and mean inter-arrival time 50. We calculate the costs considering the fluctuations in the inventory values throughout the simulation and using the waiting time of each job in the system as the lead time which is different for each job. We assume exponential inter-arrival and processing times at each facility independent from each other. Using our heuristics, we make 10 runs for each of the simulations with different random number streams and using $n = 5000$ jobs and present the comparison of objective values $\sum_{j=1}^{N} h_j E[I_j] + c^d E[d] + c^T E[W-d]^+$ on average in the following tables.

For the single product type model, we start our simulations with the initial inventory levels found by our LP formulation and every time a customer order arrives to the system, a new production order is given at that time. We compare the objective functions $\sum_{j=1}^{N} h_j E[I_j] + c^d E[d] + c^T E[W-d]^+$ with this combined model to pure MTO and MTS models where both systems are operated as the combined system but with the initial inventories all 0 for the MTO model and there is only finished goods inventory for the MTS model. The ratios of the costs are in 9 for different h and c^d combinations with $c^T = 12$.

In this simulation analysis, to assess the effectiveness of our lead time quotation algorithm, we also compare the lead times quoted for this single type system to the actual waiting times of the jobs in the system using $c^d = 5$ and $c^T = 7$. Let $Z^n_{LT} = \sum_{i=1}^{n} \{c^d d_i + c^T (W_i - d_i)^+\}$ denote the total lead time plus tardiness costs, $Z^n_{DD} = Z_{LT} + \sum_{i=1}^{n} r_i$ denote the total due-dates plus tardiness costs, $Z^n_W = \sum_{i=1}^{n} \{c^d W_i\}$ denote the total waiting times of the jobs in the system and $Z^n_C = Z_W + \sum_{i=1}^{n} r_i$ denote the total completion times of the jobs. We present ratios for these values for different number of jobs, n, in Table

Table 9: Simulation analysis of combined strategy compared to pure strategies

	h=8,c^d=10	h=8,c^d=5	h=4, c^d=5
$Z_{MTO-MTS}/Z_{MTS}$	0.833	0.810	0.892
$Z_{MTO-MTS}/Z_{MTO}$	0.785	0.871	0.827

Table 10: Comparison of lead times and due-dates to actual waiting times and completion times

	n=10	n=100	n=1000	n=5000
Z_W^n/Z_{LT}^n	0.891	0.950	0.964	0.962
Z_C^n/Z_{DD}^n	0.925	0.967	0.992	0.996

10. As we see in Table 10, the lead times quoted with our algorithm are very close to the actual waiting times and Z_{DD}^n approach to Z_C^n as n gets bigger since the effect of the release times increase with n.

4.3.3. Effectiveness of the Algorithms for Multiple Product Types: We then considered multiple product types and designed a simulation study to assess how our algorithms work for the multiple product type case. In addition to the single product type we considered above, now assume that there are 4 more product types with arrival probabilities and mean processing times as shown in Table 11, everything else remaining the same for all product types. We first consider a 3-product type model using the first two product types in Table 11 and then a 5-product type model considering all the product types.

In this case, we use an SPTA schedule and a LTQ algorithm as explained in algorithm 13. To explore the effectiveness of the SPTA schedule, we compared the total waiting times of jobs with the SPTA schedule to that of a lower bound. If we consider only the bottleneck facility (the facility with slowest processing rate) and use a SPTA schedule with preemption in that facility and assume that the waiting time of any job in a queue of any other facility is zero, then the total weighted waiting time of jobs at this system will be a lower bound for our model. Let Z_{LB} denote the lower bound for the total weighted waiting times of jobs in the system and $Z_{SPTA} = \sum_{i=1}^{n}\{c^d W_i\}$ denote the total weighted waiting times with the SPTA-based schedule. The comparison of the total waiting times with our heuristic to that of the lower bound are presented in Table 12 for different number of jobs. Also, to explore the effectiveness of the LTQ algorithm, we present the ratios of the total quoted lead times plus tardiness costs, $Z_{SPTA-LTQ} = \sum_{i=1}^{n}\{c^d d_i + c^T(W_i - d_i)^+\}$ for this case over Z_{SPTA} and Z_{LB} in Table 12 using the same weights for different product types, $c^d = 5$ and $c^T = 7$. Note that the optimal off-line LTQ algorithm quotes lead times that are exactly equal to the waiting times of the jobs in the system which is equal to Z_{SPTA}, thus Z_{SPTA} is a lower bound for the LTQ algorithm with the SPTA-based schedule and Z_{LB} is a lower bound among all schedules. As seen in this table, the difference between Z_{SPTA} and the lower bound are less than 20% and the lead time quotation algorithm gives results that are less than 7% worse than Z_{SPTA}. Thus, we conclude that our scheduling and lead time quotation algorithms are effective in minimizing the objective function $\sum_{i=1}^{n}\{c^d d_i + c^T(W_i - d_i)^+\}$

In Table 13, we present the ratios of the total costs $\sum_{i=1}^{K}\{(\sum_{j=1}^{N} h_{ij}E[I_{ij}]) + c_i^d E[d_i] + c_i^T E[W_i - d_i]^+\}$ by using the inventory values obtained from our LP formulation and using the scheduling and lead time quotation algorithm as explained in algorithm 13 to that of the total costs with pure strategies and with a FCFS schedule, for $n = 5000$ jobs using $c^T = 12$ and different weights for h and c^d. As we see in Table 13, the costs with the combined strategy is about 20% less on average than the pure strategies. Also, the SPTA based schedule used for this model decreases the total costs by about 10% compared to a FCFS schedule. Also, we see that as c^d decreases, the combined system moves toward an MTO system while as h decreases, an MTS system gives better results.

5. Conclusion: In this project, we considered a supply chain in a stochastic, multi-item environment and designed effective algorithms for the optimal inventory levels, scheduling of the jobs and lead time quotation to customers. We also analyzed the value of centralization and information exchange by considering centralized and decentralized versions of this supply chain.

In the first part of the project, 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.

In the second part of the project, we consider stylized

Table 11: Arrival probabilities and mean processing times of product types

	P(l)	p_{E_1}	p_{E_2}	p_{S_1}	p_{S_2}	p_{S_3}	p_{S_4}	p_{S_5}	p_{S_6}
l=2	0.3	15	10	25	5	45	15	20	10
l=3	0.15	5	15	10	10	15	20	25	15
l=4	0.25	10	20	30	15	10	5	10	20
l=5	0.1	20	5	5	10	5	10	30	15

Table 12: Comparison of SPTA schedule and the LTQ with the lower bound for K=3 and K=5 product types

K=3	n=10	n=100	n=1000	n=5000	K=5	n=10	n=100	n=1000	n=5000
Z_{LB}/Z_{SPTA}	0.962	0.813	0.847	0.833		0.859	0.790	0.822	0.816
Z_{SPTA}/Z_{LT}	0.874	0.933	0.952	0.947		0.941	0.922	0.925	0.931
Z_{LB}/Z_{LT}	0.848	0.753	0.802	0.786		0.807	0.735	0.766	0.751

models of a combined MTO-MTS supply chain, with a single manufacturer and a single supplier, in order to find the optimal inventory values that should be carried at each facility and to assess the impact of manufacturer-supplier relations on inventory decisions and effective lead time quotation. In our models, we consider several variations of inventory, scheduling and lead time quotation problems in order to minimize a function of the total inventory, lead times and tardiness. We derive the optimality conditions for both the centralized and decentralized versions, under which an MTO or MTS system should be used for each product at each facility and we present algorithms to find the optimal inventory levels. We also present effective lead time quotation and scheduling algorithms for centralized and decentralized versions of this model that are computationally found to be effective. We also investigate the value of coordination schemes involving information sharing between supply chain members for this system. Through computational analysis, we see that costs can be cut dramatically by using a combined system instead of pure MTO or MTS systems and information exchange between the supplier and the manufacturer is critical for effective lead time quotation. Also, we see that if centralization is not possible, an information exchange in the decentralized model can also improve the level of performance substantially.

In the final part of the project, we consider stylized models of more complex MTO-MTS supply chains with multiple facilities in order to find the optimal inventory values that should be carried at each facility and to design effective scheduling and lead time quotation algorithms. Through computational analysis, we see that combined MTO-MTS systems give much better performance than pure MTO or MTS systems and an SPTA based algorithm for scheduling the jobs performs much better than the generally used FCFS approach. We also assess that information exchange is critical for short and reliable lead time quotation.

Of course, these are stylized models, and real world systems have many more complex characteristics that are not captured by these models. Nevertheless, this is to the best of our knowledge the first study that analytically explores inventory decisions, scheduling and lead time quotation together in the context of an MTO/MTS supply chain, and that explores the impact of the supplier-manufacturer relationship on this system.

In the future, we 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.

6. References:

[1] Arreola-Risa, A. and DeCroix, G.A. (1998), Make-to-Order versus Make-to-Stock in a Production-Inventory System with General Production Times. *IIE Transactions* **30**, no.8, pp.705-713.

[2] Aviv, Y. and A. Federgruen (1998) The Operational Benefits of Information Sharing and Vendor Managed Inventory (VMI) Programs *Working paper*,Washington University and Columbia University.

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

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

[5] Bourland, K.E., S. Powell and D. Pyke (1996) Exploiting Timely Demand Information to Reduce Inventories *European Journal of Oper. Research* **92**, pp. 239-253.

[6] Cachon, G.P. (2002) Supply Chain Coordination with Contracts *Handbooks in Operations*

Table 13: Comparison of combined strategy with pure strategies for multiple product type models

K=3	h=8,c^d=10	h=8,c^d=5	h=4, c^d=5	K=5	h=8,c^d=10	h=8,c^d=5	h=4, c^d=5
$Z_{MTO-MTS}/Z_{MTS}$	0.732	0.695	0.887		0.762	0.717	0.869
$Z_{MTO-MTS}/Z_{MTO}$	0.666	0.851	0.789		0.724	0.871	0.812
Z_{SPTA}/Z_{FCFS}	0.943	0.918	0.930		0.885	0.874	0.891

Research and Management Science: Supply Chain Management, North-Holland.

[7] Cachon, G.P. and M.A. Lariviere (1997) Contracting to Assure Supply: How to Share Demand Forecasts in a Supply Chain *Management Science* **47**(5), pp. 629-646.

[8] Carr, S. and Duenyas, I.(2000), Optimal Admission Control and Sequencing in a Make-to Stock, Make-to-Order Production System. *Operations Research* **48**, no.5, pp.709-720.

[9] Chen, F. (1998) Echelon Reorder Points, Installation Reorder Points, and the Value of Centralized Demand Information *Management Science* **44**, pp. S221-S234.

[10] Chen, F. (2002) Information Sharing and Supply Chain Coordination *Handbooks in Operations Research and Management Science.*

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

[12] Federgruen, A. and Katalan, Z. (1999), Impact of Adding a Make-to-Order Item to a Make-to-Stock Production System. *Management Science* **45**, no.7, pp.980-994.

[13] Gavirneni, S., R. Kapuscinski and S. Tayur (1999) Value of Information in Capacitated Supply Chains *Management Science* **45**, pp. 16-24.

[14] Gross, D. and Harris C. (1985) *Fundamentals of Queueing Theory* John Wiley and Sons: New York.

[15] Ingene, C. and M. Parry (1995) Coordination and Manufacturer Profit Maximization: The Multiple Retailer Channel *Journal of Retailing* **71**, pp. 129-151.

[16] Jackson, J.R. (1963) Job Shop-like Queuing Systems. *Management Science* **10** pp.131-142.

[17] Jeuland, A.P. and S.M. Shugan (1983) Managing Channel Profits *Marketing Science* **2**, pp. 239-272.

[18] 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.

[19] 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.*

[20] Kaminsky, P. and Z-H. Lee (2005) Asymptotically Optimal Algorithms for Reliable Due Date Scheduling. *Submitted for publication.*

[21] Kaminsky, P. and D. Simchi-Levi (2001) The Asymptotic Optimality of the Shortest Processing Time Rule for the Flow Shop Completion Time Problem. *Operations Research* **49** pp. 293-304.

[22] Kaya, O. (2006). Scheduling, Due-Date Quotation and Inventory Issues in MTO-MTS Supply Chain Systems. PhD Thesis, University of California, Berkeley, USA.

[23] Lee, H.L.,K.C. So and C.S. Tang (2000) The Value of Information Sharing in a Two-Level Supply Chain *Management Science* **46**(5), pp. 626-643.

[24] Li, L. (1992) The Role of Inventory in Delivery Time Competition *Management Science* **38**, no.2, pp.182-197.

[25] Li, L. (2001) Information Sharing in a Supply Chain with Horizontal Competition *Management Science* **48**(9), pp. 1196-1212.

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

[27] Moorthy, K. (1987) Managing Channel Profits: Comment *Marketing Science* **6**, pp. 375-379.

[28] Netessine, S., N. Rudi (2004)Supply Chain Structures on the Internet and the Role of Marketing-Operations Interaction *Handbook of*

Quantitative Supply Chain Analysis: Modeling in the E-Business Era, Kluwer Academic Publishers, pp.607-642

[29] Rajagopalan, S. (2002) Make-to-Order or Make-to-Stock: Model and Application. *Management Science* **48**, no.2, pp.241-256.

[30] 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.

[31] 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.

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

[33] Williams, T.M. (1984), Special Products and Uncertainty in Production/Inventory Systems. *European Journal of Operations Research* **15**, pp.46-54.

[34] 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.