Batch Ordering and Batch Replenishment Policies for MTS-MTO Manufacturing Systems

by

Eman Almehdawe

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Eman Almehdawe

Abstract

Hybrid Make-To-Stock (MTS)-Make-To-Order (MTO) manufacturing is a well known policy that captures the benefits of both MTS and MTO policies. This manufacturing policy is adopted by many manufacturing firms because it allows for production based on customer specifications while keeping short response times. We study a hybrid MTS-MTO manufacturing system which consists of two processing stages and an intermediate buffer between these two stages. We propose two separate scenarios for ordering and replenishment of components from the first stage which will give more realistic guidance for practitioners. The first scenario is batching customer orders before being released to the first stage. The second scenario is batch replenishment of common components from the first stage. Most existing MTS-MTO models focus on one-for-one ordering and replenishment strategies. We enhance these models by introducing a batch ordering policy to account for economies of scale in ordering when there is an ordering cost associated with each order placed for common components. We use queueing theory to model the system behavior and use the matrix-geometric method to evaluate system performance under the new ordering policy. Afterwards, we develop an optimization model with the objective to minimize the system overall costs. The purpose of our optimization model is to find the optimal intermediate buffer size and the optimal order quantity for the system. In the second scenario, we introduce the batch replenishment policy from stage 1. This policy is suitable when stage 1 and stage 2 are physically distant and there is a shipping cost incurred when components are transferred from stage 1 to stage 2. The decision variables in this model are the intermediate buffer size and the shipping quantity.

We show that the base stock policy is sub-optimal when there is an ordering cost

incurred for ordering components. The savings from adopting the batch ordering policy are high and the response time for most customer orders is not affected. When there are shipping costs and shipping time between the two stages, we show that the right selection of the system decision variables can have a large impact on the total cost incurred by the system.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Elizabeth Jewkes, for her guidance, encouragement, constructive critiques, enthusiasm and patience throughout this research. I would like to thank her for her financial support. She has been a great mentor and guide to me. This thesis would not have been possible without her.

I would also like to thank my readers, Professor Miguel Anjos and Professor Rangaraja Sundarraj, for their valuable comments, insights and feedback.

Above all, I wish to thank my husband Saleh for his enormous support, encouragement and love throughout this work. This thesis would not have been possible without his encouragement, patience and guidance. Also, my sincere thanks to my wonderful daughter Noor for her patience when I was busy in my research.

Lastly and most importantly, I am indebeted to my parents, my siters Reema and Maryam, my brothers Ayman, Ashraf, Mohammad, and Atiah for their love and support throughout this thesis. For them I dedicate this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to globalization, competition is increasing amongst companies where flexibility, quality, cost, and response time play a major role. The major challenge in today's industry is how to increase product variety and at the same time decrease cost and lead time. Manufacturing systems are usually categorized into Make-To-Stock (MTS) systems and Make-To-Order (MTO) systems. In a MTS system, the facility produces according to a forecast of customer demand, and completed jobs enter a finished goods inventory, which in turn serves customer demand. In a MTO system, the facility produces according to customer requests and no finished goods inventory is kept (Wein, 1992). The main advantage of MTS over the MTO system is that it allows for immediate satisfaction of customer demand. The main drawback for MTS system is the high inventory costs incurred for holding finished goods inventory, especially when there is a high variety of products offered to customers. Thus, MTS systems are usually suitable for high volume and low variety products. Whereas MTO systems are suitable for low volume and high variety products. The advantage of the MTO policy is that there is no need to carry inventory of finished

products, and hence, no inventory cost is incurred. The main disadvantage of this policy is that the response times may become quite long if the load is high (Adan and Wal, 1998).

Today's competition is urging companies to provide a high variety of products and to keep the response time as low as possible. One of the proposed solutions for winning in this competitive environment is to adopt a hybrid MTS-MTO policy, which helps in reducing inventory holding cost and decreases response time for orders by balancing the advantages of the MTS and the MTO policies. The ability to quickly assemble and deliver custom products is a winning competitive strategy; customers get what they want and the manufacturer avoids the costs of shortages and overages (Serwer, 2002). A well-known and successful example of a company that adopted this strategy is Dell Computer Corporation. The customer gets the exact machine he/she wants, at a lower cost and more quickly than competitors (Serwer, 2002).

1.1 Hybrid MTS-MTO Manufacturing Systems

Besides the above mentioned extreme policies, Youssef, Delft, and Dallery (2004) suggest a combined policy that can be used in the following manner: The upstream manufacturing system is controlled according to a MTS policy, and the downstream part of the manufactuing system is controlled by a MTO policy, which is called the hybrid policy, as shown in Figure 1.1. This kind of a hybrid policy combines the advantages of both the MTO and MTS policies. It reduces the order fulfillment delay relative to MTO. It also lowers inventory cost since inventory is held only for components which is lower due to order pooling (Gupta and Benjafaar, 2004).

The inventory cost is also less under this policy because demand information is better forecasted for components when it is closer to customers due to risk pooling. This policy is suitable for manufacturing systems that provide a wide variety of products and still the response time is a great advantage. The hybrid MTS-MTO policy is widely used in the electronics industry and other similar markets where many product configurations can be produced from common components (Gupta and Weerawat, 2006). Donk (2001) also presents a hybrid policy which is used in the food processing industries which must deliver a wide variety of products and keep costs as low as possible.

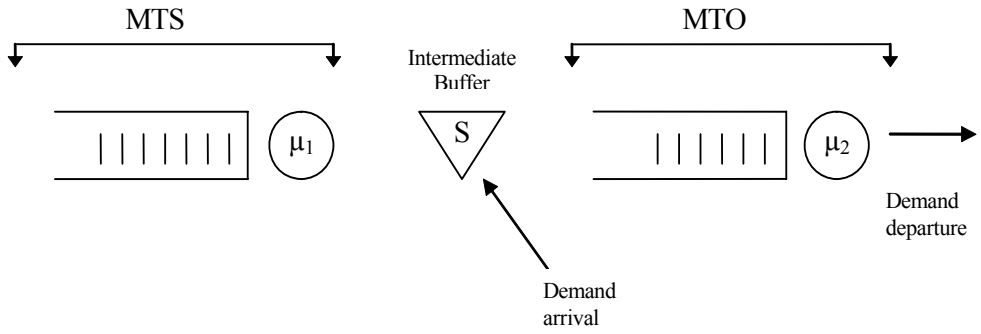

Figure 1.1: A Typical Hybrid Manufacturing System

Most research in hybrid MTS-MTO systems assume a base stock policy for the control of inventory in the intermediate buffer. The base stock policy works as follows: whenever there is a demand arrival for the end product, the inventory level decreases by one, and an instantaneous replenishment order is released to the upstream stage to make up for the used unit. Demand is assumed to occur one at a time. This policy is widely adopted in the literature because of the ease of modeling, although such a policy is not necessarily optimal when there is an

ordering cost associated with each replenishment order released. Veinott (1965) shows that a batch ordering policy is optimal when there is an ordering cost in the system. It is costly to pay an ordering cost each time an order is placed for one component. On the other hand, the base stock policy is costly when there is a shipping cost associated with each replenishment shipment to the inventory buffer. In real manufacturing systems, a base stock policy may not be optimal when there are ordering or shipping costs incurred in the replenishment process.

Most of the research in hybrid systems also assume that the replenished orders arrive instantaneously to the intermediate buffer, and that there is no shipping cost between stage 1 and the intermediate buffer. These assumptions could lead to misleading decisions if the two stages are physically distinct. The consideration of the shipping time and shipping cost may lead to different operating policies.

1.2 Contributions of this Work

The primary contribution of this thesis is the introduction of a batch ordering policy to the typical hybrid MTS-MTO system. Most of the literature adopts a base stock policy for convenience, but as a result, the analysis with this policy cannot incorporate the impact of ordering and shipping costs. This policy is not always optimal when there is a fixed ordering cost associated with each order. The base stock policy is easy to implement but is not realistic and may lead to wrong conclusions when there is a cost associated with each order placed in the system. Our batch ordering policy is a generalization of the base stock policy and represents a more realistic modeling of a common manufacturing problem faced by decision makers. Adopting the batch ordering policy may benefit the manufacturer by saving costs. It may affect the response time in the system but in our work we

show that batching orders will affect only a small percentage of orders when the there is no upper limit for the customer delay. We also show that the base stock policy produces sub-optimal solutions in some settings.

Another contribution of this thesis is the introduction of a transportation time and/or a shipping cost to the typical hybrid MTS-MTO system. We introduce this policy when there is a shipping cost incurred for the replenished components. Our proposed policy is a generalization of the instantaneous replenishment assumption in most hybrid systems models developed in the literature. This model is more realistic and can help decision makers select their optimal decision variables when there is a shipping cost and a shipping time associated with the replenishment process of orders.

1.3 Outline of the Report

The remainder of this report is organized as follows: In Chapter 2, we review the literature on hybrid MTS-MTO systems, which follows two main streams of research: Multi-stage Production/Inventory systems, and Delayed Product Differentiation. In each research stream we review the main models developed and discuss the limitations of these models. In Chapter 3, we introduce the batch ordering policy. Then we evaluate the system performance under the new proposed policy using the matrix-geometric method developed by Neuts in 1981, and finally we build an optimization model to find the optimal buffer size and the optimal batch size and compare the results with the base stock policy. In Chapter 4, we present a model for batching replenishment orders. We solve for the system performance measures, and we develop an optimization model to find the optimal buffer size and the optimal shipping lot size, then we compare the results for different shipping speeds.

Finally, conclusions and future research directions are discussed in Chapter 5.

Chapter 2

Literature Review

This thesis focuses on the design of a MTS-MTO manufacturing system when ordering and replenishment costs are considered. We identify the optimal buffer size and the optimal batch size for ordering or replenishment, when there is an ordering cost incurred in the case of batch ordering, or when there is a shipping cost incurred in the case of a batch replenishment policy.

Hybrid MTS-MTO systems have been studied in the literature from several different perspectives. We focus on two streams of research that are helpful in understanding the MTS-MTO systems, and provide a base for our proposed research. MTS-MTO manufacturing systems are related to the research stream in multi-stage Production/Inventory systems (PI). These models generally consist of multiple production stages separated by inventory buffers. Whereas the general case for a multi-stage PI system is to have a buffer for finished goods inventory at the end stage which allows for the immediate satisfaction of customer demand, MTS-MTO systems have no finished goods inventory. Most of the works in multi-stage PI systems adopt the base stock policy for the replenishment of items through

the system. They assume no ordering cost is incurred when an order is placed.

Another stream of research that is closely related to the simple hybrid MTS-MTO system is in the Delayed product Differentiation (DD) area. DD is a hybrid system, in which a common product platform is built to stock and then differentiated by assigning a customer specified features after the demand is realized (Gupta and Benjafaar, 2004). Most of the analytical models developed for DD systems focus on the optimal intermediate buffer size between stages as well as on the optimal point of differentiation for products. They also look at the costs of redesigning product processes. The later two research questions are not addressed in our work.

2.1 Multi-stage Production/Inventory systems

The closest work to our model was developed by Lee and Zipkin in 1992. They study a multi-stage production system in tandem, as depicted in Figure 2.1. They assume that final customer demand follows a Poisson process and each unit production time is exponentially distributed. The system is controlled via a base stock policy. The customer demand at the end stage triggers a demand at its predecessor stage and a unit of material to stage 1, where units move from an output buffer to the next one only in response to a demand arrival. If the finished goods buffer, or any other buffer in the system is empty, the order is backlogged. They develop an approximation scheme for the system expected number of backorders, and the expected number of semi-finished goods inventory at each buffer. Their approximation scheme has been used by many authors since. However, in their work they assume no setup costs or setup times, and, consequently, no batching of units into lots. They also assume no shipping cost from one stage to the next. Their model is not a hybrid

system since they assume the existence of finished goods inventory buffer at the downstream stage. If we set the capacity of the downstream buffer equal to zero and the number of stages equal to 2, then this will be similar to the hybrid system we are studying.

In the next chapter we use Lee and Zipkin's approximation results for the performance measures to validate our model when $S_J = 0$ for a two stage tandem system.

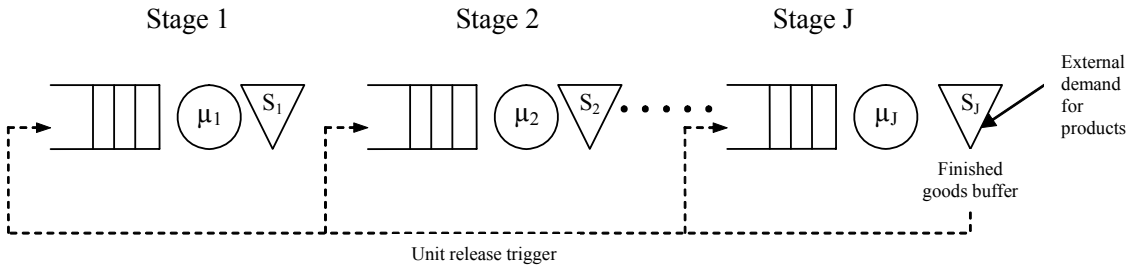

Figure 2.1: A schematic representation of a multi-stage P/I system

Gupta and Selvaraja (2006) extended Lee and Zipkin's model by providing a near-exact solution for the system performance measures for a capacitated serial supply system. They show that Lee and Zipkin's approximation for the system performance measures overestimates the congestion in a series system with multiple stages. They use the matrix-geometric method to find the optimal base-stock policy that minimizes the inventory and backordering costs. They also do not consider shipping costs nor ordering costs in their model.

Gupta and Weerawat (2006) studied the special case of no inventory buffer after the last process for a two stage inventory system. They investigate the coordination effects between a manufacturer and a supplier. They developed an optimization

model to compare the revenue results when the two parties coordinate their decisions, versus acting separately. When there is coordination, the manufacturer offers the supplier a share of the revenues. They assume that the revenue is a function of the delivery delay.

Liu, Liu, and Yao (2004) present a decomposition approach to find the optimal inventory bufffer sizes for a multi-stage inventory system while maintaining a certain service level for customer orders. The decomposition is performed by treating the queue length at each stage as an independent sum of regular orders and backorders. The system performance measures are the fill rate, and the expected Work In Process (WIP) at each stage. Afterwards, they use these measures in an optimization model to minimize the overall inventory at each stage, while maintaing an overall prespecified service level.

While the previous mentioned works assume constant replenishment lead time, Levi and Zhao (2005) considered a stochastic replenishment lead time for a multi-stage supply chain network, which has a tree shape. Boute, Lambrecht, and Houdt (2007) assumed load dependent lead times for their production inventory system, i.e., as the load increases, the replenishment lead time increases.

Most of the studies in the literaure focus on a base-stock policy for controlling serial supply systems. Bonvik, Couch, and Gershwin (1997) compared the serial supply system control policies: Minimum base-stock policy, base-stock policy, Constant Work In Process (CONWIP), kanban, and a hybrid policy that includes a base stock policy and a kanban policy. They show that the CONWIP and the hybrid policies give significantly better response to changes in the demand rate. Veatch and Wein (1994) perform a comparison between these policies, using dynamic programming, for a two station tandem production/inventory system. They show that

base stock policies are never optimal for such a system because they accumulate large quantities of WIP which may remain for long periods of time. They showed that a base stock policy is close to optimal at some parameter settings and that the optimal policy to be a hybrid policy. Duenyas and Pantana-anake (1998) extended Veatch and Wein's work using a Markov Decision Process that relaxes the exponential processing time in the Veatch and Wein's model to a general stationary distribution.

All of the above work assumes no setup time or setup cost incurred in the system when an order is placed. This assumption was relaxed in Li and Liu's (2006) work, where they used an (s, S) policy in a two-stage production system, for which they assume that station 1 produces semi-finished product to stock, and station 2 produces finished product to order from WIP. These two stations are in tandem, and there is a significant setup time at the upstream station. The production times at both stations are exponentially distributed random variables. The capacity of station 1 is high, so that when there are sufficient supplies for station 2, it can be switched off. There is a setup time at station 1 each time it restarts the operation, so that batch production at station 1 is necessary for efficient capacity utilization. A batch production control rule is used at the upstream station, with the objective of minimizing the WIP level while maintaining a required busy probability at the downstream station. To characterize the system performance under this rule, Li and Liu construct a discrete time Markov model for the status of station 1. This Markov model is used to find the analytical performance measures to be included in the optimization model. The objective of the optimization model is to find the batch production rule for station 1 that minimizes the WIP between the two stations while keeping high utilization of station 2. Li and Liu's model is similar to our model in considering set up costs for station 1 which made it reasonable to

adopt the batch ordering policy. Their decision variables (s, S) are similar to our decision variables (B, S) but the optimization problem is different since they do not consider customer delays. Instead, they focus on the utilization of station 2. Moreover, they assume the existence of finished goods inventory after station 2.

The base stock policy has been extensively used in the modeling of multi-stage production/inventory systems because of the ease of implementation with this policy. This policy is optimal when there is no ordering or setup cost, and both holding and shortage costs are proportional to the volume of on-hand inventory or shortage (Boute, Lambrecht and Houdt, 2007). It also provides a benchmark on how much inventory is needed to provide a certain service level. Veinott (1965) considered a batch ordering policy for a single location inventory system with constant lead time. He shows that when the system incurs unit ordering costs, holding costs, and penalty costs, then a batch ordering policy is optimal. This optimality does not hold when a fixed charge for placing an order is considered. There is some work on batch ordering policies where the flow of material is in batches, but this stream of research assumes that units are processed in batches, not in single units as our research. Chen (2000) extended Veinott's model to a multi-echelon inventory system in which material flow is in batches. Moinsade and Lee (1986) studied the batch size problem for a repair system where a depot fills the orders from multiple location service centers in batches. Axsater (1993) studied the same problem and provided exact evaluation for such a policy. He and Jewkes (1997) explored the relationship between the batch size and flow time in a single server queue, where Poisson demand arrivals are batched before being processed. They derive the Laplace-Stieltjes Transform for three types of flow times considered in the system. Thereafter, they derive the optimal batch size that minimizes the expected flow time or its variance. The batch ordering policy they adopt is similar to our work but they apply it for a

single stage system.

There is some work in the literature that compares the performance of a pure MTO versus a pure MTS policies in which a base stock policy was also adopted. They do not combine the MTS and MTO policies as in our research. One of the first models for combined MTS and MTO systems is due to Williams (1984). Williams assumes that the MTS items are produced in batches of fixed size, requests for which are triggered by a (Q, r) policy. Orders for products are of random size with geometric interarrival times. There are priority items and regular items in his model and priority is given to the batch with the largest waiting time. First, he derives an approximation for the items' lead time and optimal reorder level (r) for a given batch size (Q) in which he assumes a Poisson arrival for each product class. Then, he approximates the results to a nonlinear cost function of batch size (Q). Rajagopalan (2002) develops a nonlinear integer programming model for a company that incurs a setup time, has a limited capacity, and experiences congestion. The model selects which items are to be processed via a MTS and which items are to be processed via a MTO policy, based on demand, setup time, processing time, and unit holding costs. Federgruen and Katalan (1999) look at the performance measures for both policies in terms of inventory level and waiting time distribution. Arreola-Risa and Decroix (1998) derive optimality conditions for MTO versus MTS based on demand and capacity utilization.

Youssef, Delft, and Dallery (2004) compare a First In First Out (FIFO) and a priority rule for a hybrid MTS-MTO system under stochastic assumptions. They consider two products; one has high volume demand and the other has low volume demand. The inventory is managed by a base stock policy. They found that under the priority rule, the total cost is much lower for achieving the same service level constraint.

2.2 Delayed product Differentiation

The other stream of research that is directly related to the hybrid MTS-MTO system is in Delayed product Differentiation (DD). This concept was first introduced by Alderson in 1950. DD models consist of a number of common stages for all product platforms, and a customization stage/stages for platform-specific demands, as shown in Figure 2.2. The common stages are analogous to the MTS stage in the hybrid MTS-MTO system, while the customization stages are similar to the MTO stage. Hence, a DD system can be used to model a hybrid MTS-MTO system when there is no inventory held for the end product. Conceptually, DD tries to exploit the commonalities between products, and delay the point of diffferentiation to be as close as possible to the point demand requirements are realized. The work done recently by Gupta and Benjaafar (2004) is related to the hybrid MTS-MTO systems. They model a two-stage production system in which both the characteristics of MTS-MTO and delayed differentiation are considered. They assume no setup costs and unsatisfied demand is fully backorderd. The lead time is load dependent and demand for each product occurs according to a Poisson process. They also assume that inventory for semi-finished and finished products is controlled via a base-stock policy. They use Lee and Zipkin's 1992 approximation scheme to develop performance measures for the system and optimal intermediate buffer size, to compare optimal costs under pure MTS, pure MTO, and DD systems. They observe that if the first stage utilization is very high, then a DD system is preferred to a MTS system. They also examine the point of differentiation for a serial production/inventory system. Again, this model is suitable for industries that do not incur ordering costs or shipping costs during the replenishment of items through the system. Otherwise, a base stock assumption for the control of intermdiate in-

ventory in the system could lead to incorrect results. The performance measures used in this model are the expected inventory and backorders in the system. In addition to these performance measures, we use the expected delay in the system.

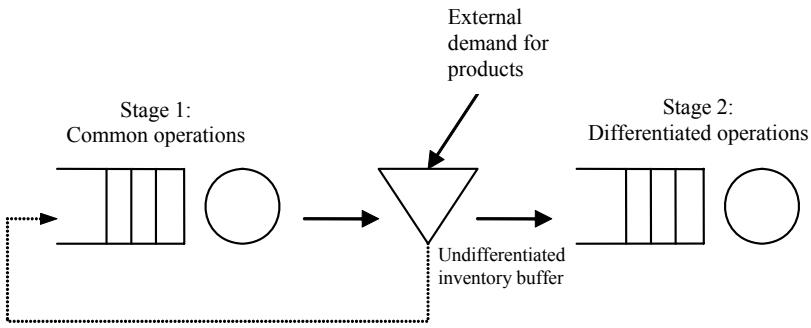

Figure 2.2: A schematic representation of a DD system

Lee and Tang (1997) develop a simple strategic planning model that captures the costs and benefits associated with the redesign of products and processes, to determine the point at which products are differentiated. The paper focuses on three approaches for delayed product differentiation: standardization, modular design, and process restructuring. Standardization refers to using common components or processes. Modular design refers to decomposing the complete product into sub modules that can be easily assembled together. Process restructuring refers to resequencing process steps in making a product. They develop a discrete time model where the demand for the end product is normally distributed. They consider a manufacturer that produces two end products. These two products share k common operations, and then are customized in N-k operations. For the control of inventory, they use order-up-to policy to keep the model simple. They also apply their model to special cases in which the lead time and the inventory costs will not be affected when the point of differentiation is delayed. The costs they consider

are the design cost, processing cost, and the inventory cost at intermediate stages. They assume constant lead time at each stage. They show that delaying the point of differentiation lowers the inventory and improves the service level in the system due to risk pooling.

Swaminathan and Tayur (1999) consider a manufacturer whose product line consists of several products each defined by a subset of components. The goal is to simultaneously design an efficient assembly sequence for the product, and determine the type and target inventory levels of semi-finished inventory that enable delayed differentiation. The objective function measures the cost that needs to be incurred while designing the components, in order to enable such an assembly sequence.

Another model that assumes constant lead time was developed by Aviv and Federgruen (2001). They model a multi-item inventory system which consists of two stages; the first stage is used to produce common intermediate items, and the second stage is the differentiation stage for those items. Each stage has its own lead time and stage 1 has a limited capacity. They examine the benefits of differentiation when demands are seasonally fluctuating, and possibly correlated.

Our work focuses on a two stage production/inventory system where the first stage is MTS and the second stage is MTO. There is an intermediate buffer for common components between the two stages. We propose a batch ordering policy in which orders are accumulated until a prespecified limit is reached at which time an order for common components is issued to the first stage. We also propose a batch replenishment policy in which orders are accumulated in a shipping buffer before being shipped to the intermediate buffer at stage 2. This thesis extends existing MTS-MTO models by considering the impact of ordering and replenishment costs on operating decisions. Most of the queueing based work we presented earlier adopted the base stock policy for the control of inventory in the system, mainly

because of the ease of its analytical implementation. However, the base stock policy has proved to be sub-optimal when there is an ordering cost incurred whenever an order is placed. Other models which adopted a batch ordering policy did not consider the queuing and thus, the congestion in the system. Our work is unique in the sense that it combines the queuing analysis along with the batch ordering and replenishment policies. We also model the randomness in the replenishment lead time and consider the cost of shipping which was not addressed by the previous models. Some of the previous work developed an approximation scheme for the system performance, in our work we utilize a Markov chain model to derive the exact performance measures in the system.

In Chapter 3 we use a batch ordering policy to account for economies of scale in setup and ordering costs. We use matrix-geometric methods to calculate the system performance metrics. These measures are analyzed and used in an optimization model to find the optimal batch size and the optimal intermediate buffer size for different system settings. In Chapter 4, we present a batch replenishment policy model which is suitable for systems that incur shipping costs and time between the two stages.

Chapter 3

The MTS-MTO Model with Batch Ordering

In this chapter, we consider a two stage MTS-MTO system in which stage 1 produces components or items to stock, and stage 2 is a customization stage for these components based on customer requirements. The decision variable in these models is the intermediate buffer size. Most hybrid MTS-MTO systems introduced in the literature assume a base stock policy with an order release each time a demand arrival occurs for the end product. This policy could be non-optimal when there is an ordering cost associated with each order placed.

We consider the batching of orders to explore the possibility of optimizing the total costs. We use the matrix-geometric method developed by Neuts in 1981 to evaluate the system performace. Then we compare our model performance results with Lee and Zipkin's (1992) approximation for the special case when the replenishment policy is a base stock policy or the batch size equals 1. Afterwards, we find the optimal combination of the buffer size and the batch size that minimizes

the system overall costs. This model is suitable for companies that incur ordering cost each time an order is placed. It is also suitable for companies that incur setup cost before production in stage 1.

3.1 Model Description

We consider a two stage production/inventory system where the manufacturing process consists of two major stages: stage 1 is the manufacturing stage of common components from raw material. Stage 2 is the customization step/s for the common components based on the demand requirements, Figure 3.1. Customization is triggered for common components whenever there is a demand arrival. i.e., upon the arrival of a demand for the end product, a common component is released from the intermediate buffer and then queued for the customization process at stage 2. We assume that each demand requires just one unit of the final product and the customization processing time for these common components is not a function of the product type, which is quite realistic for a lot of electronic industry companies. When an order arrives and finds the intermediate buffer empty of common components, then the order is backordered. Backorders are filled whenever the common components arrive to the intermediate buffer. The manufacturing stage has a finite capacity; however, we assume that it is large enough that it can handle all the demand requirements.

Ordering the common components is managed by a batch ordering policy; orders are accumulated until a batch size of B orders is reached, then an order is released to the stage 1 queue, to replenish the buffer of common components. The accumulation of orders is carried out because of the ordering or the setup costs incurred at stage 1.

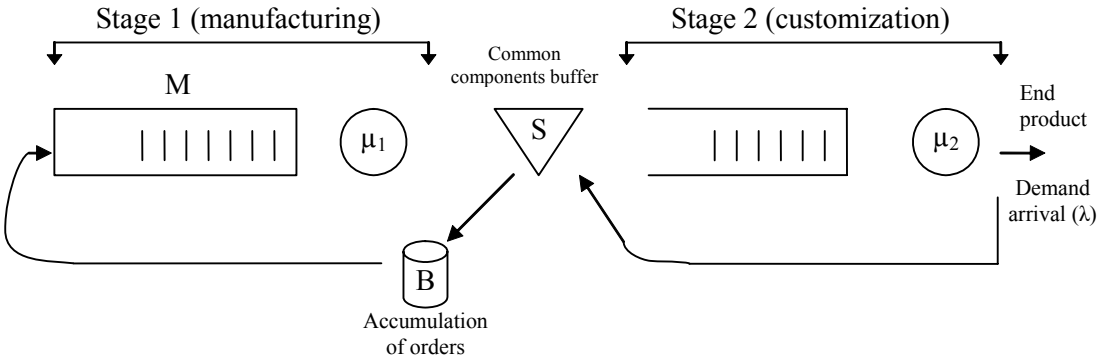

Figure 3.1: The MTS-MTO system with batch ordering policy

Each stage is capacitated and has one server that processes the orders sequentially on a First Come First Serve (FCFS) priority rule.

The system reaction upon the arrival of a demand for the end product can be summarized in the following steps:

- If the common components buffer is not empty and a demand for the end product arrives, then one common component is released from the buffer and is sent to join the queue of stage 2, which serves the demands on a FCFS basis. This demand arrival will also trigger the orders accumulated to increase by one and the common components buffer contents to decrease by one.

- If the buffer is empty, then the order is backlogged. The demand arrival triggers the orders accumulated to increase by one.

- When the orders accumulated reach the pre-specified limit B, an order with a quantity of B will be released and sent to the queue of stage 1, where the orders are processed one at a time also on a FCFS basis.

- When a common component finishes processing at stage 1, it is transferred immediately to the common components buffer.

The model assumptions can be summarized in the following points:

- The common components buffer size is S.
- We assume Poisson arrival of customer demand with rate λ.
- We assume that customer orders consist of 1 item and all orders carry the same priority.
- We assume an exponential processing time at stage 1 and stage 2, with rates μ_1 and μ_2 respectively.
- We consider backordering of unsatisfied demand when the buffer is empty with a limit M (supplier capacity).
- An order for common components is released in batches of size B, and the maximum batch size allowed is S. This is because the intermediate buffer capacity should not be exceeded upon the arrival of a replenishment order.
- We assume that customers are served on a FCFS priority rule.
- The stage 2 queue has unlimited capacity.

The research question this model addresses is to find the optimal buffer size, S, and the optimal ordering batch size, B that minimize the total costs incurred by the whole system. A variety of objective functions can be used to find the optimal buffer size and the optimal batch size. In our work, the objective function is composed of the ordering cost and the inventory holding costs for common components and a delay penalty for orders.

3.2 The Markov Chain

The model described above can be represented by a continuous-time Markov chain with a generator matrix (Q). To describe the system precisely at time t we need to define three state variables as follows:

1. $N_2(t)$: The queue occupancy before stage 2 at time t, including the one in process. We define the level of the Markov chain as the subset of all states that have the same $N_2(t)$. $\{N_2(t) : N_2(t) = 0, 1, ...\}$

2. $N_1(t)$: The queue occupancy before stage 1 at time t, including the one in process. This state variable also is used to define the number of common components in the intermediate buffer. i.e., if $N_1(t) = i$, then the intermediate buffer contains $S - i$ components if $S - i \geq 0$, or there are $S - i$ backorders if $S - i < 0$. We define the first sub-level of the Markov chain as the subset of all states that have the same $N_1(t)$. $\{N_1(t) : N_1(t) = 0, 1, ..., M\}$.

3. $J(t)$: The number of orders for common components accumulated to form a batch at time t. We define thesecond sub-level of the Markov chain as the subset of all states that have the same $J(t)$. $\{J(t) : J(t) = 0, 1, ..., B - 1\}$

The system state changes upon an arrival of a customer order or when there is a service completion at either stage. These possible transitions from the current state ($N_2(t)$, $N_1(t)$, $J(t)$) into other states are summarized in Table 3.1. The notation (t) has been suppressed for notation simplicity.

The infinitesimal generator matrix (Q) of the system continuous Markov chain has a unique block tridiagonal structure as follows:

Event	Condition	System next state	Rate	Explanation
Customer Arrival	$J < B-1$ $N_1 < S$	$(N_2+1, N_1, J+1)$	λ	No batch is formed and there is inventory in the buffer
	$J < B-1$ $N_1 \geq S$	$(N_2, N_1, J+1)$	λ	No batch is formed and there is no inventory in the buffer
	$J = B-1$ $N_1 < S$	$(N_2+1, N_1+B, 0)$	λ	A batch is formed and there is inventory in the buffer
	$J = B-1$ $N_1 \geq S$	$(N_2, N_1+B, 0)$	λ	A batch is formed and there is no inventory in the buffer
Service completion	$N_1 \leq S$	(N_2, N_1-1, J)	μ_1	Service completion at stage 1
	$N_1 > S$	(N_2+1, N_1-1, J)	μ_1	Service completion at stage 1
	$N_2 > 0$	(N_2-1, N_1, J)	μ_2	Service completion at stage 2

Table 3.1: Model 1 state transitions

$$Q = \begin{pmatrix} B_0 & A_0 & 0 & 0 & 0 \\ A_2 & A_1 & A_0 & 0 & \ddots \\ 0 & A_2 & A_1 & A_0 & \ddots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix}$$

where A_0 represents the rate matrix at which the system moves up one level, A_2 is the rate matrix at which the system moves back one level, A_1 is the rate matrix at which the system returns to the same level, and B_0 is the rate matrix at which the system returns to the boundary level (level 0). As we can see, the process can move only to an adjacent level upon an arrival or a departure from stage 2 queue. We denote the level of the Markov chain by the subset of all states that have the same number of units at stage 2. The first sub-level in the generator matrix Q is the subset of all states that have the same number of units at stage 1, while the second sub-level is the subset of all states that have the same number of accumulated orders in the orders buffer.

We denote the steady state probability distribution of the number of units in the system by π, this matrix is the unique solution for the set of equations:

$$\begin{aligned} \pi Q &= \mathbf{0} && (3.1) \\ \pi e &= \mathbf{1} \\ \pi &\geq \mathbf{0} && (3.2) \end{aligned}$$

where e is a column vector of ones of appropriate size. The first sets of equations are the balance equations for the Markov Chain, and the second sets are the normalization conditions which are used to find a unique solution for the system of equations. The steady state probability distribution matrix π can be calculated in different ways. Our approach is based on Neuts observation in 1981 for the repetitive structure of the generator matrix Q. We describe Neuts' matrix-geometric method in the next section.

3.3 The Matrix-geometric Approach

In this section we describe a computational procedure, based on the Matrix-geometric approach that was developed by Neuts in 1981. It can be used to find exact performance measures of any system with defined set of states, and its infinitesimal generator matrix has a tridiagonal structure. We treat the queue at stage 1 as a finite queue, but one of sufficiently large size that the desired performance measures are quite accurate. The stage 2 queue will be treated as an infinite queue; and the queue of accumulated orders will be treated as a finite capacity queue. From this method we calculate the steady state probability distribution for the expected

number of units at each stage and in the orders buffer. The key idea is that the queue occupancy in stage 2 can be modeled as a Quasi-Birth-Death (QBD) process. A Markov chain is called a QBD if one step transitions from a state are restricted to states in the same level or in the two adjacent levels (Latouche and Ramaswami, 1999). This property allows us to develop a matrix-geometric solution for its steady state probability distribution.

The essential problem in determining the steady state probability distribution of a Markov process is solving a set of linear, flow balance, equations, where there is an equation associated with each state of the system. For systems with a large or possibly infinite number of states, exact solutions can only be obtained if one can exploit structural properties of these balance equations. Neuts developed a body of results that allows one to exploit repetitive structure. If the states of the Markov process can be grouped into vectors which possess a certain repetitive structure, then a recursive procedure can be used to determine the stationary state probabilities of any vector in terms of the probabilities for the previous vector (Nelson, 1991).

The generator matrix (Q) has two portions; the boundary portion and the repeating portion. The general form for the balance equations in the repeating portion is as follows:

$$\pi_{j-1}A_0 + \pi_j A_1 + \pi_{j+1}A_2 = \mathbf{0}, \quad j \geq 2 \tag{3.3}$$

The steady state probability distribution for the boundary states (π_0) is obtained by the relation:

$$\pi_0(B_0 + RA_2) = \mathbf{0} \tag{3.4}$$

subject to the normalization condition $\pi_0(I - R) = \mathbf{1}$, where R is the minimal non-negative solution to the matrix quadratic equation:

$$A_0 + RA_1 + R^2A_2 = \mathbf{0} \tag{3.5}$$

R can be calculated by different ways; we use Latouche and Ramaswami's (1999) method which calculates R from the following recursive relation:

$$R_i = -(A_0 + R_{i-1}^2 A_2)A_1^{-1} \tag{3.6}$$

until $|R_i - R_{i-1}| \leq \varepsilon$, where ε is a very small number and represents the accuracy of the matrix R, and $R_0 = 0$. Then recursively we find the steady state probability distribution of the repeating portion from the following relationship:

$$\pi_{i+1} = \pi_i R, \quad i \geq 0 \tag{3.7}$$

where i represents the level in the generator matrix.

For the hybrid system described above, we can notice that the arrival process to the first stage follows the Erlang distribution with parameters (B, λ), and the service time is exponentially distributed with rate μ_1, so the queue to this stage is a PH/M/1 queue. Due to the buffer existence between the MTS stage and the MTO stage, the arrival process for the second stage is not exponential anymore, which is the reason that such a system is difficult to solve for its exact performance measures and therefore, approximation schemes were developed in the literature for its performance measures. This system can be modeled as a Markov chain which has the generator matrix Q described earlier. The building blocks of the generator matrix which are: B_0, A_0, A_1, and A_2, square matrices of order $(M + 1) * B$. The

A_0 matrix which represents the rate at which the number of units in stage 2, $N_2(t)$, increases by 1 has the following general form:

$$A_0 = \begin{array}{c} \\ 0 \\ \vdots \\ B \\ \vdots \\ S-1 \\ S \\ S+1 \\ \vdots \\ M-B+1 \\ \vdots \\ M \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & \cdots & B & \cdots & S-1 & S & S+1 & \cdots & M-B+1 & \cdots & M \end{array} \\ \begin{pmatrix} A_{00} & & A_{01} & & & & & & & & \\ & \ddots & & \ddots & & & & & & & \\ & & \ddots & & \ddots & & & & & & \\ & & & \ddots & & \ddots & & & & & \\ & & & & A_{00} & & \ddots & & & & \\ & & & & & 0 & & \ddots & & & \\ & & & & & \mu_1 I & \ddots & & A_{01} & & \\ & & & & & & \ddots & \ddots & & & \\ & & & & & & & \ddots & \ddots & & \\ & & & & & & & & \ddots & \ddots & \\ & & & & & & & & & \ddots & \ddots \end{pmatrix} \end{array}$$

where each level in A_0 represents the possible states for the number of units at stage 1, $N_1(t)$, and each entry in A_0 is a matrix of size B. A_{01} is the rate at which $N_1(t)$ increases by B units and $N_2(t)$ increases by 1 upon the arrival of a customer order when the buffer is not empty and the number of orders in the orders buffer is $B-1$. A_{01} is defined as:

$$A_{01} = \begin{pmatrix} 0 & \mathbf{0} \\ \lambda & 0 \end{pmatrix},$$

where $\mathbf{0}$ is a square matrix of zeros of size $B-1$.

A_{00} : is a square matrix of size B and it represents the rate at which $N_1(t)$ returns to the same state while $N_2(t)$ increases by 1 upon the arrival of an order and the orders buffer has less than $B-1$ units. It has the following general form:

$$A_{00} = \begin{pmatrix} 0 & \lambda I_0 \\ 0 & 0 \end{pmatrix},$$

where I_0 is an identity matrix of size $B-1$.

The rate at which $N_1(t)$ decreases by 1 while $N_2(t)$ increases by 1 is $\mu_1 I$. This represents the case of service completion at stage 1 and there is a backorder placed. The unit released from the stage 1 will directly enter the queue for stage 2 to be processed.

A_2 matrix represents the rate at which the number of units at stage 2, $N_2(t)$, decreases by 1 upon a service completion at stage 2. It has the following general form:

$$A_2 = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ M \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & \cdots & M \end{array} \\ \begin{pmatrix} \mu_2 I & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \mu_2 I \end{pmatrix} \end{array}$$

where I is an identity matrix of size B. $\mu_2 I$ represents the rate at which $N_1(t)$ returns to the same state while $N_2(t)$ decreases by 1 whenever there is a service completion at stage 2.

A_1 represents the rate at which $N_2(t)$ returns to the same level in the repeating portion. Since the infinitesimal generator matrix Q columns must sum to 0, it follows that:

$$A_1 = -(A_0 + A_2), \quad A_0 = -B_0$$

which results in the following formula for A_1 :

$$A_1 = B_0 - \mu_2 * I_2$$

where:

I_2 : is an identity matrix of size $(M+1) * B$.

In other words, the rate at which $N_2(t)$ returns to the same level is $1-$ the rate at which the system either moves one level up or one level down.

The boundary matrix B_0 is the rate at which $N_2(t)$ returns to level 0. This matrix has the following general form:

$$
B_o = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ S-1 \\ S \\ \vdots \\ S+B \\ \vdots \\ M-1 \\ M \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} 0 & 1 & \cdots & S-1 & S & \cdots & S+B & \cdots & M-1 & M & \end{array} \\
\begin{pmatrix}
-\lambda I & & & & & & & & & & \\
\mu_1 I & B_{00} & & & & & & & & & \\
 & \ddots & \ddots & & & & & & & & \\
 & & \mu_1 I & B_{00} & & & & & & & \\
 & & & & B_{00}+B_{01} & B_{03} & & & & & \\
 & & & & & \ddots & \ddots & & & & \\
 & & & & & & \ddots & \ddots & & & \\
 & & & & & & & \ddots & \ddots & & \\
 & & & & & & & & B_{00}+B_{01}+B_{02} & \ddots & \\
 & & & & & & & & \ddots & \ddots &
\end{pmatrix}
\end{array}
$$

where:

I: is an identity matrix of size B.

The diagonal entries of B_o (B_{00}, B_{01}, and B_{02}) are square matrices of size B and are calculated such that the sum of the rows for the generator matrix Q equals 0.

$$
B_{00} = \begin{pmatrix} -(\lambda+\mu_1) & 0 \\ 0 & -(\lambda+\mu_1) \end{pmatrix},
$$

$$
B_{01} = \begin{pmatrix} 0 & \lambda I_0 \\ 0 & 0 \end{pmatrix},
$$

where I_0 is an identity matrix of size $B-1$,

$$B_{02} = \begin{pmatrix} 0 & 0 \\ 0 & \lambda I_0 \end{pmatrix},$$

B_{03} is a square matrix of size B. It represents the rate at which $N_1(t)$ increases by B steps while $N_2(t)$ returns to the same level upon the arrival of a customer demand when there are no backorders and the orders buffer contains $B-1$ orders. It has the following general form:

$$B_{03} = \begin{pmatrix} 0 & \mathbf{0} \\ \lambda & 0 \end{pmatrix},$$

where $\mathbf{0}$ is a square matrix of zeros of size $B-1$.

The rate at which $N_1(t)$ decreases by 1 and $N_2(t)$ returns to the same level is $\mu_1 I$. This represents the case when there is a service completion at stage 1 and no backorders in the system.

3.4 Implementation

Using the matrix-geometric method described earlier, we use Matlab 7.0 to solve for the stationary probability distribution (π) of the system, see Appendix A for Matlab codes. For computational purposes we set the maximum queue length of stage 2 large enough so that the impact of truncating the state space is minimal and the probability that the queue length is beyond this limit is close to zero. After some testing for this limit, we set it to 100. Also we set the maximum capacity for stage 1 (M) large enough such that the probability an order is lost is very small and consequently, the computed steady state probability distribution is accurate. After some testing with M, we set it to 50.

The utilization of stage i (ρ_i) is defined by $\rho_i = \dfrac{\lambda}{\mu_i}$ where $i = 1, 2$.

For most of the remainder of this work our focus is on balanced capacity MTS-MTO systems, i.e., we set $\mu_1 = \mu_2$, and for different utilization rates we vary λ only.

The steady state probability distribution we computed has the following general form:

$$\pi = \begin{pmatrix} \pi_{0,0} & \pi_{0,1} & \cdots & \pi_{0,B(M+1)} \\ \pi_{1,0} & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & \vdots \\ \pi_{100,0} & \cdots & \cdots & \pi_{100,B(M+1)} \end{pmatrix}$$

The columns represent the steady state probability distribution for the levels in the generator matrix, which is stage 2 steady state probability distribution in this case. The rows have 2 levels; stage 1 steady state probability distribution, and orders accumulation steady state probability distribution. This matrix can be reduced to π^*, which is the matrix of the steady state probability distribution for the number of units in the system, regardless of the batching quantity, as follows:

$$\pi^* = \begin{pmatrix} \sum_{i=0}^{B-1} \pi_{0,i} & \cdots & \sum_{i=BM}^{B(M+1)} \pi_{0,i} \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{B-1} \pi_{100,i} & \cdots & \sum_{i=BM}^{B(M+1)} \pi_{100,i} \end{pmatrix}$$

To find the distribution of the steady state number of units at stage 1 ($\pi 1$), we sum over the rows of the π matrix which yields a $(1,\ B(M+1))$ vector ($\pi 1^*$). Then we sum over the batch size, as follows :

$$\pi 1^* = \left[\sum_{i=0}^{100} \pi_{i,0}, \quad \cdots, \quad \sum_{i=0}^{100} \pi_{i,B(M+1)} \right]$$

$$\pi 1 = \left[\sum_{i=0}^{B-1} \pi 1_i^* \quad \cdots \quad \sum_{i=BM+1}^{B(M+1)} \pi 1_i^* \right] = \left[\pi 1_0, \quad \cdots, \quad \pi 1_M \right]$$

The steady state distribution of the steady state number of items at stage 2 ($\pi 2$) is calculated by summing over the columns of the previous matrix which yields:

$$\pi 2 = \sum_{j=0}^{B(M+1)} \pi_{i,j} = \begin{bmatrix} \pi 2_0 \\ \vdots \\ \pi 2_{100} \end{bmatrix}$$

We have computed all the necessary steady state probability distributions that are necessary to calculate the system steady state performance measures under different settings. Still we need to make sure that the system is operating under normal conditions and is stable. Therefore, we derive the stability conditions which are important to check before any runs are conducted.

Stability Conditions:

In order to have a stable system, the Markov chain should be positive recurrent (Neuts, 1981). This condition is represented by the following relationship:

$$\pi A_2 \cdot \mathbf{1} > \pi A_0 \cdot \mathbf{1}$$

In other words, the rate of moving down one level in the Markov chain must exceed the rate of moving up one level, in order to have a stable system. This condition can be explicitly defined for our model as:

$$\begin{bmatrix} \pi_{0,0} & \cdots & \pi_{0,B(M+1)} \\ \pi_{1,0} & \cdots & \cdots \\ \vdots & \cdots & \cdots \\ \pi_{100,0} & \cdots & \cdots \end{bmatrix} \cdot \begin{bmatrix} \mu_2 \\ \vdots \\ \vdots \\ \mu_2 \end{bmatrix} > \begin{bmatrix} \pi_{0,0} & \cdots & \pi_{0,B(M+1)} \\ \pi_{1,0} & \cdots & \cdots \\ \vdots & \cdots & \cdots \\ \pi_{100,0} & \cdots & \cdots \end{bmatrix} \cdot \begin{bmatrix} \lambda \\ \vdots \\ \mu_1 \\ \vdots \end{bmatrix}$$

Which reduces to:

$$\begin{bmatrix} \mu_2 \cdot \sum_{i=0}^{B(M+1)} \pi_{0,i} \\ \vdots \\ \vdots \\ \mu_2 \cdot \sum_{i=0}^{B(M+1)} \pi_{100,i} \end{bmatrix} > \begin{bmatrix} \lambda \cdot \sum_{i=0}^{BS} \pi_{0,i} + \mu_1 \cdot \sum_{i=BS+1}^{B(M+1)} \pi_{0,i} \\ \vdots \\ \vdots \\ \lambda \cdot \sum_{i=0}^{BS} \pi_{100,i} + \mu_1 \cdot \sum_{i=BS+1}^{B(M+1)} \pi_{100,i} \end{bmatrix}$$

Now let :

$$\alpha_1(j) = \sum_{i=0}^{B(M+1)} \pi_{j,i}, \qquad \alpha_2(j) = \sum_{i=0}^{BS} \pi_{j,i}, \qquad \alpha_3(j) = \sum_{i=BS+1}^{B(M+1)} \pi_{j,i},$$

then the stability conditions reduces to the following series of conditions:

$$\frac{\lambda\alpha_2(j) + \mu_1\alpha_3(j)}{\mu_2\alpha_1(j)} < 1, \quad j = 1, .., B(M+1)$$

These stability conditions show that for each level in the system, the total probability that the system moves up one level (upon an arrival or a service completion) should be less than the total probability the system moves down one level. In other words, the total drift up should be less than the total drift down for the system to be stable.

3.5 Basic Performance Measures

The measures we use to evaluate the system performance under various parameter settings for (B, S, and ρ_i) are: the expected number of units at stage 1, $E(N_1)$, the expected number of units at stage 2, $E(N_2)$, the expected number of backorders, $E(O)$, order fulfillment delay, $E(D)$, and the expected number of semi finished inventory, $E(I)$. These performance measures are obtained from the steady state probability distribution of the number of units in the system (π). The expected number of units at stage 1, $E(N_1)$, is calculated from the steady state stationary distribution of the number of units at stage 1 from the following relationship:

$$E(N_1) = \sum_{i=0}^{M} \pi 1_i . i$$

The expected number of units at stage 2, $E(N_2)$, is calculated from the stationary distribution of the number of units at stage 2 by the relation:

$$E(N_2) = \sum_{j=1}^{100} \pi 2_j . j$$

The expected number of backorders, $E(O)$, is calculated from the stationary distribution of the number of units at stage 1 when the queue length is greater than the buffer size as follows:

$$E(O) = E(N_1 \mid N_1 > S) = \sum_{i=S+1}^{M} \pi 1_i . (i - S)$$

The expected number of semi finished inventory, $E(I)$, is the sum of the common components buffer contents, $E(I1)$, and the units waiting for processing in front of stage 2, $E(N_2)$. It is calculated as follows:

$$E(I1) = \sum_{i=0}^{M} \pi 1_i . \max(0, (S - i))$$
$$E(I) = E(N_2) + E(I1)$$

The delay of an order is defined as the time from an arrival of customer order until it is fulfilled. The expected order fulfillment delay is computed as a weighted average of the expected delay in the system for customers who find the intermediate buffer empty, $E(D1)$, and the customers who find the intermediate buffer non-empty, $E(D2)$, as follows:

$$E(D) = p1 * E(D1) + p2 * E(D2)$$
$$p1 = p(N_1 > S) = \sum_{i=S+1}^{M} \pi 1_i$$
$$p2 = p(N_1 \leq S) = 1 - p1$$

where $p1$ is the probability that an order upon arrival is backordered, and $p2$ is the probability that an order upon arrival is fulfilled from the intermediate buffer. By Little's Law (Ross, 2006), the expected delay is the ratio of the expected number of units in a queue over the arrival rate, for our model this reduces to:

$$E(D1) = \frac{E(N_2) + E(O)}{\lambda}$$
$$E(D2) = \frac{E(N_2)}{\lambda}$$

First, to validate our model results we compare the special case of our model, when the batch size equal 1, with Lee and Zipkin's (1992) approximation results for the performance measures. In their paper they study a multi-stage production system in tandem. They assume demand follows a Poisson process and unit production times are exponentially distributed. The system is controlled by a base stock

policy. The customer demand at the end stage triggers a demand at its predecessor stage and a unit of material to stage 1. They solve for a system of two stages and find the expected number of backorders and the expected number of semi finished goods inventory. We compare their approximation results with our exact results from the matrix-geometric method when the batch size equals 1, and for different values of S, μ_1, and μ_2. The results are summarized in Table 3.2. The % Deviation for the expected inventory and the expected backorders is calculated as follows:

$$\%Deviation = \frac{Approximation - Exact}{Approximation} * 100\%$$

As Gupta and Selvaraja (2006) showed in their work, the approximation developed by Lee and Zipkin provides an upper bound on the actual performance measures, and hence overestimates the congestion in the system. This is because in their approximation scheme, Lee and Zipkin assume that each stage of the system operates as an M/M/1 queue. We also notice that as the intermediate buffer size (S) increases, the percent error between the exact method and Lee and Zipkin's approximation decreases. This means that increasing the buffer size up to a certain limit will make the assumption of having two separate M/M/1 queues valid. Since the deviations provided by the Matrix-geometric method are small, this validates our method. We continue to compare the system performance under the batch ordering policy we introduced earlier in this chapter.

3.6 Basic Performance Measure Analysis

To study the system behavior under the batch ordering policy, we vary separately the batch size, the intermediate buffer size, and the system utilization.

μ_1	μ_2	S	**Approximation**		**Exact**		**% Deviation**	
			expected inventory	expected backorders	expected inventory	expected backorders	expected inventory	expected backorders
1.25	1.25	1	4.200	7.200	4.121	7.120	-1.885	-1.108
		3	5.048	6.048	4.865	5.865	-3.616	-3.028
		5	6.311	5.311	6.114	5.114	-3.122	-3.712
		7	7.839	4.839	7.670	4.669	-2.159	-3.510
		9	9.537	4.537	9.405	4.405	-1.383	-2.920
1.25	1.5	1	2.200	5.200	2.146	5.146	-2.443	-1.045
		3	3.048	4.048	2.944	3.943	-3.418	-2.588
		5	4.311	3.311	4.210	3.210	-2.340	-3.064
		7	5.839	2.839	5.758	2.758	-1.379	-2.856
		9	7.537	2.537	7.478	2.477	-0.786	-2.358
1.25	2.0	1	1.200	4.200	1.169	4.169	-2.560	-0.745
		3	2.048	3.048	2.001	3.000	-2.316	-1.575
		5	3.311	2.311	3.271	2.270	-1.210	-1.759
		7	4.839	1.839	4.810	1.809	-0.599	-1.607
		9	6.537	1.537	6.517	1.517	-0.303	-1.328
1.5	1.25	1	4.333	5.333	4.229	5.229	-2.409	-1.957
		3	5.593	4.593	5.440	4.440	-2.736	-3.332
		5	7.263	4.263	7.158	4.158	-1.445	-2.462
		7	9.117	4.117	9.058	4.058	-0.650	-1.439
		9	11.052	4.052	11.021	4.021	-0.276	-0.754
1.5	1.5	1	2.333	3.333	2.258	3.258	-3.206	-2.244
		3	3.593	2.593	3.493	2.493	-2.782	-3.854
		5	5.263	2.263	5.197	2.197	-1.252	-2.912
		7	7.117	2.117	7.080	2.080	-0.516	-1.735
		9	9.052	2.052	9.033	2.033	-0.208	-0.919
1.5	2.0	1	1.333	2.333	1.288	2.288	-3.348	-1.913
		3	2.593	1.593	2.542	1.542	-1.950	-3.174
		5	4.263	1.263	4.233	1.233	-0.703	-2.372
		7	6.117	1.117	6.101	1.101	-0.257	-1.407
		9	8.052	1.052	8.044	1.044	-0.095	-0.730
2.0	1.25	1	4.500	4.500	4.400	4.400	-2.228	-2.228
		3	6.125	4.125	6.059	4.059	-1.083	-1.607
		5	8.031	4.031	8.009	4.009	-0.275	-0.547
		7	10.008	4.008	10.001	4.001	-0.066	-0.164
		9	12.002	4.002	12.000	4.000	-0.015	-0.044
2.0	1.5	1	2.500	2.500	2.422	2.422	-3.132	-3.132
		3	4.125	2.125	4.073	2.073	-1.255	-2.436
		5	6.031	2.031	6.013	2.013	-0.292	-0.868
		7	8.008	2.008	8.003	2.003	-0.068	-0.272
		9	10.002	2.002	10.000	2.000	-0.015	-0.075
2.0	2.0	1	1.500	1.500	1.449	1.449	-3.433	-3.433
		3	3.125	1.125	3.093	1.093	-1.027	-2.853
		5	5.031	1.031	5.020	1.020	-0.214	-1.043
		7	7.008	1.008	7.005	1.005	-0.049	-0.342
		9	9.002	1.002	9.001	1.001	-0.011	-0.095

Table 3.2: Model results for B=1 compared to Lee and Zipkin's Approximate results

B	2	3	4	5	6	7	8	9	10
$E(N_1)$	2.741	3.136	3.540	3.938	4.338	4.725	5.096	5.452	5.810
$E(N_2)$	2.620	2.663	2.695	2.722	2.740	2.778	2.833	2.900	2.565
$E(D)$	1.748	1.780	1.806	1.829	1.850	1.886	1.935	1.995	1.787
$E(O)$	0.274	0.386	0.496	0.598	0.703	0.803	0.893	0.976	1.066
$E(I1)$	7.533	7.249	6.956	6.660	6.365	6.078	5.797	5.524	5.256
$E(I)$	10.153	9.913	9.651	9.381	9.106	8.855	8.630	8.424	7.821
$p1$	0.01	0.02	0.03	0.04	0.05	0.06	0.08	0.09	0.11

Table 3.3: Performance evaluation with various batch sizes

1. The effect of Batch size: We run the model for different batch sizes when $S = 10$, $\mu_1 = \mu_2 = 2$, and $\lambda = 1.5$. The results are summarized in Table 3.3.

As we can see from Figure 3.2, the stage 1 queue is affected by the batching policy while stage 2 does not see that effect. This is because units are processed one at a time at stage 1 and then released, either to the buffer, or directly to stage 2 (if the order was backordered). Due to the larger queue length at stage 1, the expected delay in the system will increase as the batch size increases. This is because orders are not released directly, they will wait until the batch size limit is reached, and then orders are released in bulk. Hence, there will be usually either a queue at stage 1 or no units at all. The expected number of backorders increases as the batch size increases because the probability an order arrives and finds the buffer empty ($p1$) increases as the batch size increases. The expected intermediate inventory decreased from 10.2 to 7.8 units in the previous table (24% decrease in intermediate inventory when batch size increased from 2 to 10 in the previous example). The reason for this decrease is that, when orders are accepted and batched, components will spend less time in the intermediate buffer, while they will wait more before the orders are processed.

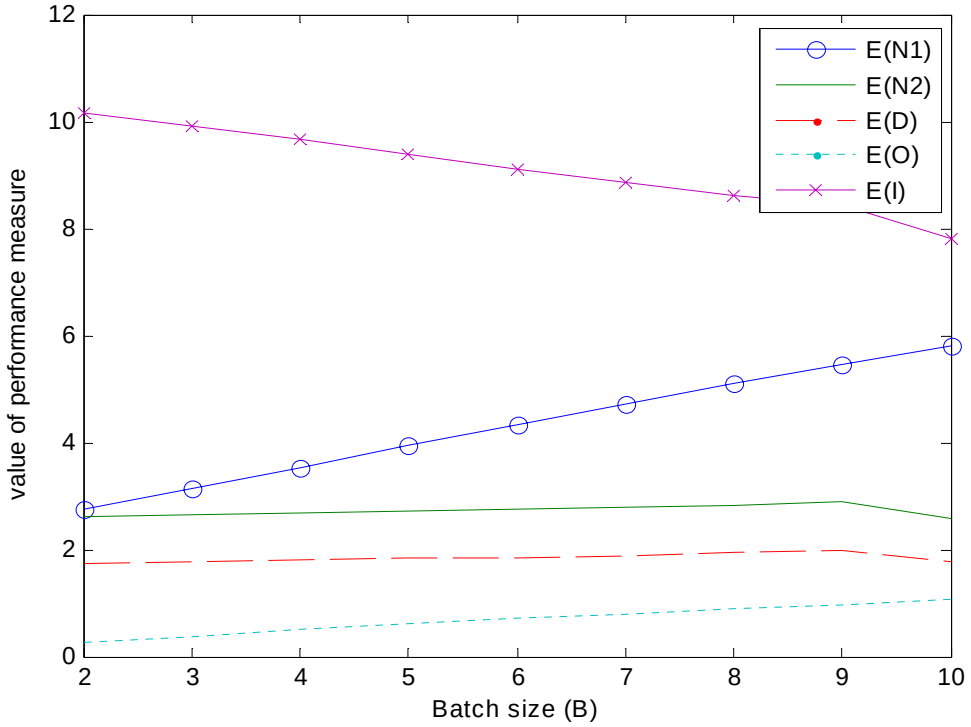

Figure 3.2: Performance evaluation with various batch sizes

2. The effect of Intermediate buffer size: We set $B = 2$, $\mu_1 = \mu_2 = 2$, and $\lambda = 1$. We vary the intermediate buffer size, the results are summarized in Table 3.4.

As we can see from Figure 3.3, the expected queue length in front of stage 1, and the expected number of backorders decreases sharply as the buffer size increases. This is because the increase in the buffer size means that the system can handle more orders without the need of backlogging. The intermediate inventory increases substantially as the buffer size increases because we are holding more inventory in the buffer. Some of these units will spend a longer time in the buffer before being processed. On the other hand, the expected queue length in front of stage 2 is not

S	2	3	4	5	6	7	8	9	10	11	12
$E(N_1)$	5.817	2.593	1.965	1.458	1.290	1.200	1.165	1.149	1.143	1.140	1.140
$E(N_2)$	0.723	0.900	0.911	0.956	0.972	0.985	0.991	0.995	0.997	0.999	0.999
$E(D)$	1.419	1.008	0.942	0.963	0.974	0.985	0.991	0.995	0.997	0.999	0.999
$E(O)$	5.061	1.714	0.987	0.423	0.217	0.101	0.050	0.024	0.012	0.006	0.003
$E(I1)$	1.244	2.121	3.022	3.966	4.927	5.902	6.886	7.875	8.869	9.865	10.863
$E(I)$	1.967	3.021	3.933	4.922	5.899	6.887	7.877	8.871	9.866	10.864	11.862

Table 3.4: Performance evaluation with various buffer sizes

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$E(N_1)$	0.076	0.162	0.276	0.436	0.653	0.932	1.273	1.670	2.113	2.593
$E(N_2)$	0.053	0.113	0.181	0.259	0.346	0.442	0.547	0.659	0.777	0.900
$E(D)$	0.528	0.563	0.604	0.649	0.697	0.750	0.808	0.870	0.937	1.008
$E(O)$	0.001	0.009	0.041	0.113	0.239	0.424	0.670	0.972	1.323	1.714
$E(I1)$	2.925	2.847	2.765	2.678	2.586	2.492	2.397	2.302	2.210	2.121
$E(I)$	2.977	2.959	2.946	2.936	2.932	2.935	2.944	2.962	2.987	3.021

Table 3.5: Performance evaluation with various arrival rates

affected by the increase in the buffer size, due to the fact that units are queued in front of stage 2 whenever there is a demand and the system is not starving. Also, the expected delay in the system is not affected by the increase in the buffer size when there are no backorders.

3. Effect of changing the system utilization: we vary the arrival rate (λ) for when $\mu_1 = \mu_2 = 2$, we fix $S = 3$, and $B = 2$. The results are summarized in Table 3.4.

As we can see from Figure 3.4, the queue length in front of stage 1, and the queue length in front of stage 2 increases as the arrival rate increases. The expected time each order needs to spend in the system increases as the arrival rate increases

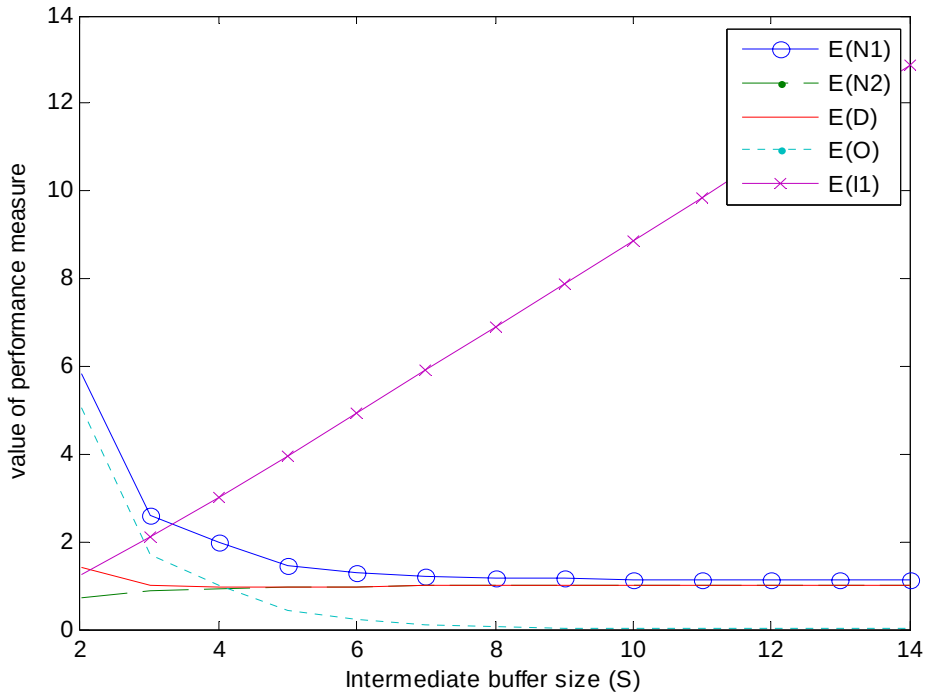

Figure 3.3: Performance evaluation with various buffer sizes

because there will be more congestion in the system. The expected common components inventory in the buffer decreases as the arrival rate increases, which is quite expected because units in the buffer spend less time in the system. While the total common components inventory is not affected by the changes in the arrival rate, the expected number of backorders increases as the arrival rate increases which is quite intuitive.

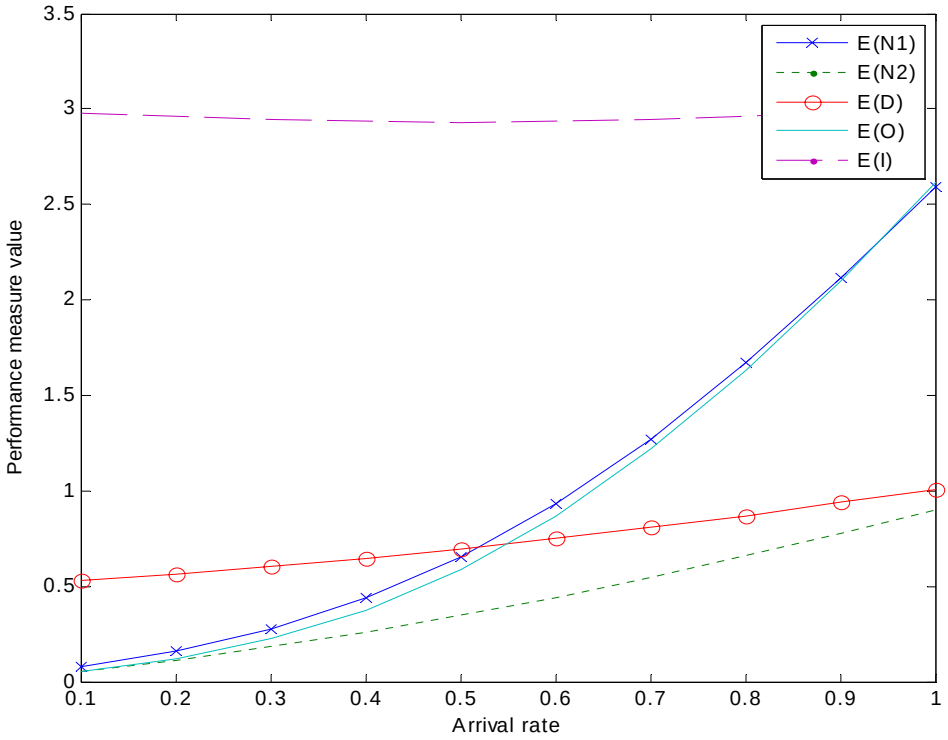

Figure 3.4: Performance evaluation with various arrival rates

3.7 Optimization Model

The problem we are trying to solve using our optimization model is to find the optimal batch size (B) and the optimal buffer size (S) which leads to a minimum expected operating total cost. The costs incurred are the holding costs of common components, an ordering cost for components, and the expected penalty cost of customer order fulfillment delay. The optimization problem we are looking at is:

$$\min_{B,S} \left\{ TC(B,S) = C_h * E(I) + C_O * \frac{\lambda}{B} + C_D * E(D) \right\}$$
$$S.t. \qquad 0 < B \leq S$$

B, S integers

where:

C_h: cost of holding inventory for common components between the two stages ($/unit/unit time).

C_O: cost of ordering common components ($/order).

C_D: penalty cost for customer order fulfillment delay ($/unit/unit time).

3.8 Computational Results

To illustrate the methodology we solve an example to find the optimal (B, S) by calculating the total cost for different combinations of (B, S). We use Matlab 7.0 for find the optimal combinations. We vary the system utilization by changing the arrival rate and fixing the processing rates at each stage. We set $\mu_1 = \mu_2 = 2$. The arrival rates we considered are 1.0, 1.5 and 1.8 corresponding to 50%, 75% and 90% utilization rates respectively. We also set the capacity of stage 1 to be large enough so that there are no lost customers. We vary S and B, while keeping the validity of the condition $S \geq B$. We then use the derived performance measures from the previous section along with some cost parameters to find the optimal combination of (B, S) that minimizes the total cost for the system. The cost parameters considered correspond to low, medium, and high penalties. The ordering cost parameters considered are: 1.0, 5.0, and 10.0. The holding inventory cost parameters considered are: 0.1, 0.25, and 2.0, and the cost parameters considered for the penalty of order delay are: 0.5, 2.0, 5.0 and 10.0. For each set of cost parameters we calculate the

optimal (B, S) as shown in Figure 3.5. The optimal solutions for different utilization rates are summarized in Table 3.6.

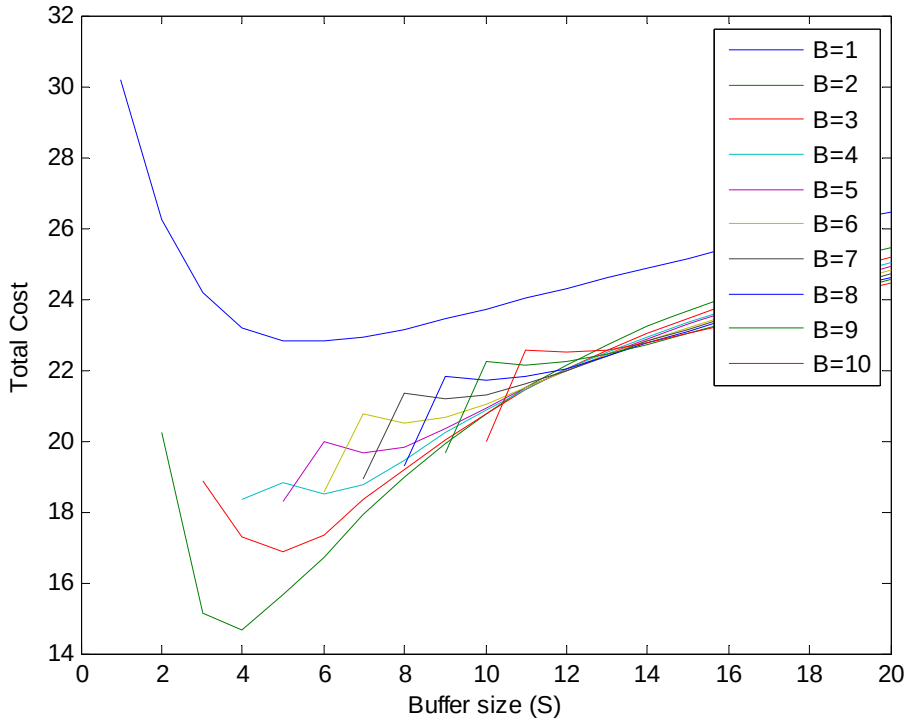

Figure 3.5: Total cost for (0.1, 5.0, 10.0) cost parameters and 75% utilization

From results shown in Table 3.6 we can make the following observations:

- For items that have high inventory cost with respect to ordering cost and when the utilization of the system is low, it is optimal to use the simple base stock policy, otherwise batching orders is optimal.
- For items that have high inventory cost but medium ordering cost, it is optimal to order in batches of 2-3, unless customers are too sensitive to wait

Cost parameter			**50% utilization**		**75% utilization**		**90% utilization**	
C_h	C_O	C_D	(S, B)	Total cost	(S, B)	Total cost	(S, B)	Total cost
0.1	1.0	0.5	(4, 4)	1.069	(5, 5)	1.588	(3, 2)	1.991
		2.0	(5, 5)	2.446	(4, 2)	3.735	(4, 2)	4.222
		5.0	(5, 5)	5.173	(4, 2)	7.587	(4, 2)	8.554
		10.0	(5, 5)	9.718	(4, 2)	14.007	(4, 2)	15.774
	5.0	0.5	(8, 8)	1.738	(10, 10)	2.426	(8, 8)	3.299
		2.0	(8, 8)	3.106	(8, 8)	5.080	(5, 3)	7.240
		5.0	(7, 7)	5.840	(7, 7)	10.269	(4, 2)	12.154
		10.0	(7, 7)	10.385	(4, 2)	17.007	(4, 2)	19.374
	10.0	0.5	(11, 11)	2.256	(15, 15)	3.030	(12, 12)	4.200
		2.0	(11, 11)	3.641	(13, 13)	5.7938	(7, 7)	8.813
		5.0	(10, 10)	6.410	(10, 10)	11.217	(5, 3)	15.802
		10.0	(9, 9)	11.005	(7, 7)	19.956	(4, 2)	23.874
0.25	1.0	0.5	(3, 3)	1.544	(3, 3)	2.103	(3, 3)	2.453
		2.0	(4, 4)	3.001	(3, 2)	4.277	(3, 2)	4.757
		5.0	(4, 4)	5.809	(4, 2)	8.213	(4, 2)	9.204
		10.0	(5, 5)	10.360	(4, 2)	14.633	(4, 2)	16.424
	5.0	0.5	(5, 5)	2.524	(6, 6)	3.386	(6, 6)	4.186
		2.0	(5, 5)	3.888	(6, 6)	5.939	(4, 4)	7.910
		5.0	(5, 5)	6.615	(6, 6)	11.045	(4, 2)	12.804
		10.0	(6, 6)	11.154	(4, 2)	17.633	(4, 2)	20.024
	10.0	0.5	(7, 7)	3.335	(9, 9)	4.342	(9, 9)	5.465
		2.0	(7, 7)	4.698	(9, 9)	6.993	(6, 6)	9.668
		5.0	(7, 7)	7.425	(8, 8)	12.215	(5, 3)	16.601
		10.0	(7, 7)	11.970	(6, 6)	20.805	(4, 2)	24.524
2.0	1.0	0.5	(1, 1)	4.434	(2, 2)	5.795	(2, 2)	6.117
		2.0	(1, 1)	6.045	(2, 2)	8.640	(2, 2)	9.262
		5.0	(1, 1)	9.267	(3, 2)	14.050	(3, 2)	15.201
		10.0	(1, 1)	14.637	(3, 2)	20.830	(3, 2)	22.726
	5.0	0.5	(2, 2)	7.144	(2, 2)	8.795	(2, 2)	9.717
		2.0	(3, 3)	9.255	(2, 2)	11.640	(2, 2)	12.862
		5.0	(3, 3)	12.399	(3, 2)	17.050	(3, 2)	18.801
		10.0	(3, 3)	17.638	(3, 2)	23.830	(3, 2)	26.326
	10.0	0.5	(3, 3)	9.349	(3, 3)	11.645	(3, 3)	13.042
		2.0	(3, 3)	10.921	(3, 3)	14.292	(3, 3)	16.376
		5.0	(3, 3)	14.065	(3, 3)	19.587	(3, 3)	23.045
		10.0	(4, 4)	18.888	(3, 2)	27.580	(3, 2)	30.826

Table 3.6: Optimal policies for different utilization settings

where orders should be issued one at a time.

- When ordering cost is high with respect to holding inventory cost, then irrespective of the customer sensitivity to wait, it is optimal to order in large batches.

- The batching policy outperforms the simple base stock policy unless the inventory holding cost is higher than the ordering cost.

- When the system utilization is low, it is optimal to have the batch size equal to the buffer size irrespective of the holding cost or ordering cost, and as the utilization is increasing or the cost of delay is very high, then the buffer size should be larger than the batch size.

- For the same cost combinations and when the results for the optimal S and B are the same for all utilizations of the system, then the cost is increasing as the utilization of the system is increasing. This is because the delay is increasing and consequently it costs more.

To compare the optimal system results under our proposed policy with the base stock policy that was used extensively in the literature, we run our model when $B = 1$, and find the optimal buffer size (S) for the 75% utilization rate. The results are summarized in Table 3.7.

As we can see from Table 3.7, 17% of the cases solved had the buffer size equal to that in the base stock policy and the batching policy, otherwise the results are different. We can notice also that the total cost for the base stock policy is always higher than the total cost incurred in the batching policy, and the savings from

Cost parameter			**Batching policy**		**Base stock policy**	
C_h	C_O	C_D	(S, B)	Total cost	(S)	Total cost
0.1	1.0	0.5	(5, 5)	1.588	3	2.995
		2.0	(4, 2)	3.735	5	6.046
		5.0	(4, 2)	7.587	6	11.987
		10.0	(4, 2)	14.007	6	21.832
	5.0	0.5	(10, 10)	2.426	3	8.995
		2.0	(8, 8)	5.080	5	12.046
		5.0	(7, 7)	10.269	6	17.987
		10.0	(4, 2)	17.007	6	27.832
	10.0	0.5	(15, 15)	3.030	3	16.495
		2.0	(13, 13)	5.7938	5	19.546
		5.0	(10, 10)	11.217	6	25.487
		10.0	(7, 7)	19.956	6	35.332
0.25	1.0	0.5	(3, 3)	2.103	2	3.583
		2.0	(3, 2)	4.277	4	6.802
		5.0	(4, 2)	8.213	5	12.864
		10.0	(4, 2)	14.633	6	22.794
	5.0	0.5	(6, 6)	3.386	2	9.583
		2.0	(6, 6)	5.939	4	12.802
		5.0	(6, 6)	11.045	4	18.864
		10.0	(4, 2)	17.633	6	28.794
	10.0	0.5	(9, 9)	4.342	2	17.083
		2.0	(9, 9)	6.993	4	20.302
		5.0	(8, 8)	12.215	5	26.364
		10.0	(6, 6)	20.805	6	36.294
2.0	1.0	0.5	(2, 2)	5.795	1	9.235
		2.0	(2, 2)	8.640	2	13.386
		5.0	(3, 2)	14.050	2	20.547
		10.0	(3, 2)	20.830	3	31.400
	5.0	0.5	(2, 2)	8.795	1	15.235
		2.0	(2, 2)	11.640	2	19.386
		5.0	(3, 2)	17.050	2	26.547
		10.0	(3, 2)	23.830	3	37.400
	10.0	0.5	(3, 3)	11.645	1	22.735
		2.0	(3,3)	14.292	2	26.886
		5.0	(3, 3)	19.587	2	34.047
		10.0	(3, 2)	27.580	3	44.900

Table 3.7: Comparison between the batching policy and a base stock policy

using this policy range from 32% to 82% which supports the conclusion that the batching policy outperforms the base stock policy.

Some Managerial Insights

- The batch ordering policy increases the congestion in front of the MTS stage, but it does not much affect the queue length in front of the MTO stage.
- The batch ordering policy does not increase the response time in the system for the customers whose orders are filled directly from the intermediate buffer. On the other hand, only customers whose orders are backlogged will have longer response time due to the batching policy.
- Initially there is an unexpected advantage for the batching of orders which is the decrease in the inventory holding cost with this policy. But since batching delays the orders from being transformed into WIP or in other words, batching orders delays the physical arrival of replenishment orders this advantage becomes quite expected.
- The base stock policy outperforms the batch ordering policy only when the system utilization is low and the inventory holding cost is very high with respect to the ordering cost. As the system utilization increases, the base stock policy advantage decreases.
- When the inventory costs are high and the system utilization is high, the use of the batching policy is recommended because it will decrease the inventory costs.

In this chapter, we have introduced the first variation of the pure MTS-MTO system by batching orders before a replenishment order is released when there is

a fixed ordering cost associated with each order. In the next chapter we introduce another variation for the pure MTS-MTO system by batching the replenished orders when shipping cost and time are considered.

Chapter 4

The MTS-MTO Model with Batch Replenishment

In the previous chapter, we introduced a batch ordering policy. This policy is suitable for systems that incur an ordering cost whenever an order is placed for common components. In this chapter, we present an extension to the original MTS-MTO model introduced earlier. This model is suitable for situations when the two stages are physically distinct and there is a shipping cost and time to send common components between the two stages. In this model, stage 1 represents a manufacturer who produces common components and accumulates these components, then he ships them to an assembly center whose job is to customize these components based on customer demand. To take advantage of economies of scale in replenishment, we assume that replenishment occurs in batches of size C. The system is modeled as a continuous time Markov chain and then solved using the matrix-geometric method. We then find the optimal combination of the shipping quantity (C) and the intermediate buffer size (S) that minimizes the overall costs of the system. The costs

considered are the shipping cost of common components between the two stages, holding costs for inventory, and a penalty cost for customer fulfillment delay.

4.1 Model Description

In this chapter we develop a model that is suitable for a manufacturing system which consists of two stages: stage 1 is the manufacturing stage of common components from raw material (MTS). Stage 2 is the customization step/s for the common components based on the demand requirements (MTO), as shown in Figure 4.1. This model is suitable for companies which have assembly centers in multiple locations where they do the manufacturing of products in a central location (MTS stage) and then ship these components to assembly or distribution centers (MTO stage) who perform a customization process for the common components based on customer requirements. It is not realistic to assume that the components are replenished after being processed at stage 1 one by one, especially if the inventory for components is located far from the manufacturer. Instead, after being processed at stage 1, the common components are accumulated and then shipped together to the distribution center where each is customized based on customer requirements. Customization is triggered for components whenever there is a demand arrival. i.e., upon the arrival of a demand for the end product, a common component is released from the intermediate buffer (common components buffer) and then sent immediately for the customization process at stage 2. We assume that each demand requires just one unit of the final product. If the intermediate buffer is empty of common components upon the arrival of a customer order, then the customer order is backordered. After the production process from raw material at stage 1, the manufacturer accumulates

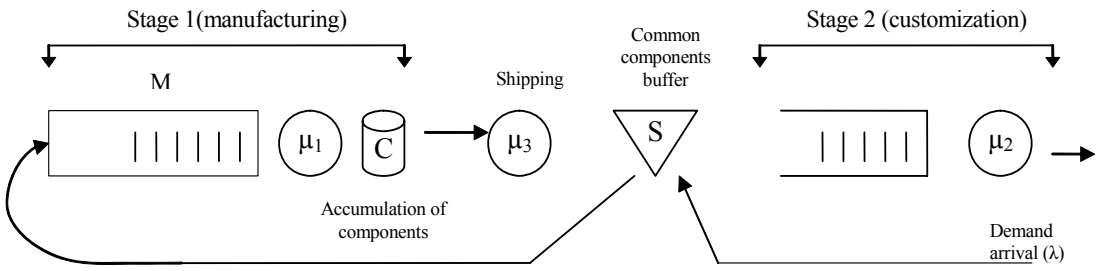

Figure 4.1: Hybrid MTS-MTO system with batch replenishment policy

the components and ships them in bulk to the intermediate buffer in order to take advantage of economies of scale in replenishment.

The system reaction upon the arrival of a demand for the end product can be summarized in the following steps:

- If the intermediate buffer is not empty and a demand arrives for the end product, then one common component is released from the intermediate buffer and is sent immediately for processing at stage 2. Demands are served on a FCFS basis in this stage. This demand arrival will also trigger an order for common components of size one to be released to stage 1.

- If the buffer is empty, then the order is backordered, and it will also trigger an order for common components of size one to be released to stage 1.

- After finishing the processing at stage 1, common components are accumulated until a pre-specified limit (C) of common components is reached. Then these components are shipped to the intermediate buffer with an exponential time. This will increase the intermediate buffer contents by C units.

The model assumptions can be summarized in the following points:

We assume:

- The common components buffer size is S.
- Poisson arrival of customer demand with rate λ.
- Orders consist of one product and all orders carry the same priority.
- Exponential processing times at stage 1 and stage 2 with rates μ_1 and μ_2 respectively.
- Exponential shipping time from stage 1 to the intermediate buffer with rate μ_3.
- The maximum shipping quantity (C) allowed is S. This is because the intermediate buffer capacity should not be exceeded upon the arrival of a replenishment shipment.
- Customers are served on a FCFS basis.
- Stage 2 queue has unlimited capacity.

The research question this model addresses is to find the optimal buffer size (S) and the optimal shipping quantity (C) that minimizes the total costs incurred by the whole system. The costs considered in this model are the inventory holding costs for common components, the shipping cost between the two stages, and a penalty cost for customer order fulfillment delay.

4.2 The Markov Chain

The model described above can be represented by a continuous-time Markov chain with a generator matrix (Q). This Markov chain can be described by three state variables:

1. $N_2(t)$: The number of units queued at stage 2 including the one in process. $\{N_2(t) : N_2(t) = 0, 1, ...\}$

2. $N_1(t)$: The number of units queued at stage 1 including the one in process. $\{N_1(t) : N_1(t) = 0, 1, ..., M\}$

3. $K(t)$: The number of accumulated common components waiting for shipping in the shipping buffer. $\{K(t) : K(t) = 0, 1, ..., C\}$

The system state may change only when there is an arrival of an order, arrival of a shipment, or service completion at either stage. The possible changes of the system state variables $(N_2(t), N_1(t), K(t))$ to other states are summarized in Table 4.1. The notation (t) has been suppressed for notation simplicity.

The number of items in the system forms a Markov chain. The infinitesimal generator matrix (Q) of the continuous Markov chain has a unique block diagonal structure as follows:

$$Q = \begin{pmatrix} B_0 & A_0 & 0 & 0 & 0 \\ A_2 & A_1 & A_0 & 0 & \ddots \\ 0 & A_2 & A_1 & A_0 & \ddots \\ 0 & 0 & \ddots & \ddots & \ddots \end{pmatrix}$$

Event	Condition	System next state	Rate	Explanation
Customer arrival	$N_1 < S$	$(N_2 + 1, N_1 + 1, K)$	λ	Demand arrival and there is inventory in the buffer
	$N_1 \geq S$	$(N_2, N_1 + 1, K)$	λ	Demand arrival and there is no inventory in the buffer
Service completion	$N_1 > 0$	$(N_2, N_1 - 1, K + 1)$	μ_1	Service completion at stage 1
	$N_2 > 0$	$(N_2 - 1, N_1, K)$	μ_2	Service completion at stage 2
Shipping of components	$K = C$ $N_1 < S$	$(N_2, N_1, 0)$	μ_3	Shipping and there are no backorders
	$K = C$ $N_1 \geq S$	$(N_2 + 1, N_1, 0)$	μ_3	Shipping and there are backorders

Table 4.1: Model 2 state transitions

where B_0, A_0, A_1, and A_2 are square matrices of order $(M + 1) * (C + 1)$. A_0 matrix represents the rate matrix at which the system moves up one level, A_2 is the rate matrix at which the system moves down one level, A_1 is the rate matrix at which the system returns to the same level, and B_0 is the rate matrix at which the system returns to the boundary level (level 0). We denote the level of the Markov chain by the subset of all states that have the same number of units at stage 2. As we can see, the process can move only to an adjacent level upon an arrival or a departure from stage 2 queue. The first sub-level in the generator matrix Q is the subset of all states that have the same number of units at stage 1, while the second sub-level is the subset of all states that have the same number of accumulated common components in the shipping buffer. A_0 matrix has the following general form:

$$A_0 = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ S-1 \\ S \\ \vdots \\ M \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & \cdots & S-1 & S & \cdots & M \end{array} \\ \begin{pmatrix} 0 & \lambda I & & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ & & & & \lambda I & & \\ & & & & A_{01} & & \\ & & & & & \ddots & \\ & & & & & & A_{01} \end{pmatrix} \end{array}$$

where each level in A_0 represents the possible states for the number of units at stage 1, $N_1(t)$, and each entry in A_0 is a matrix of size $C+1$. A_{01} is the rate at which $N_1(t)$ returns to the same state while $N_2(t)$ increases by 1 when there is a shipping incurred between stages and there is a backorder placed. Each entry in A_{01} represents the rate at which the shipping buffer state changes. A_{01} is defined as:

$$A_{01} = \mu_3 . \begin{pmatrix} 0 & \mathbf{0} \\ 1 & 0 \end{pmatrix},$$

where $\mathbf{0}$ is a square matrix of zeros of size C.

λI represents the rate at which $N_1(t)$ increases by 1 and $N_2(t)$ increases by one level upon an arrival for a customer order when there are common components available in the intermediate buffer, where I is an identity matrix of size $C+1$.

A_2 is the rate matrix at which $N_2(t)$ decreases one level. Each entry in A_2 is a square matrix of size $C+1$ and represents the rate at which $N_1(t)$ returns to the same state while $N_2(t)$ decreases by one level. This represents the case when there is a service completion at stage 2. It has the following general form:

$$A_2 = \begin{pmatrix} \mu_2 I & & \\ & \ddots & \\ & & \mu_2 I \end{pmatrix}$$

A_1 represents the rate at which $N_2(t)$ returns to the same level in the repeating portion. Since the columns of the infinitesimal generator matrix Q must sum to 0, it follows that:

$$A_1 = -(A_0 + A_2), \quad A_0 = -B_0$$

which results in the following formula for A_1 :

$$A_1 = B_0 - \mu_2 * I_2$$

where:

I_2 : is an identity matrix of size $(M+1) * (C+1)$.

In other words, the rate at which $N_2(t)$ returns to the same level is $1-$ the rate at which the system either moves one level up or one level down.

The boundary matrix B_0 is the rate at which $N_2(t)$ returns to level 0. This matrix has the following general form:

$$B_o = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ S \\ S+1 \\ \vdots \\ M-1 \\ M \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & & 1 & & \cdots & & S & & S+1 & \cdots\ M-1 & M \end{array} \\ \begin{pmatrix} -\lambda I + B_{03} + B_{02} & & & & & & & & & & \\ B_{00} & -\lambda I + B_{03} + B_{01} + B_{02} & & & & & & & & & \\ & \ddots & \ddots & & & & & & & & \\ & & \ddots & -\lambda I + B_{03} + B_{01} & \lambda I & & & & & & \\ & & & \ddots & \ddots & \ddots & & & & & \\ & & & & \ddots & \ddots & \ddots & & & & \\ & & & & & \ddots & \ddots & \lambda I & & & \\ & & & & & & B_{00} & B_{03} + B_{01} & & & \end{pmatrix} \end{array}$$

B_{00} represents the rate at which $N_1(t)$ decreases by one state upon a service completion on stage 1 while $N_2(t)$ returns to the boundary level. It has the following form:

$$B_{00} = \mu_1. \begin{pmatrix} 0 & I_0 \\ 0 & 0 \end{pmatrix},$$

where I_0 is an identity matrix of size C.

B_{01}, B_{02}, and B_{03} are square matrices of size $C+1$. These matrices are calculated utilizing the fact that the sum of the rows of the generator matrix is equal to 0. These matrices have the following general forms:

$$B_{01} = -\mu_1. \begin{pmatrix} I_0 & 0 \\ 0 & 0 \end{pmatrix},$$

$$B_{02} = \mu_3. \begin{pmatrix} 0 & \mathbf{0} \\ 1 & 0 \end{pmatrix},$$

$$B_{03} = -\mu_3. \begin{pmatrix} \mathbf{0} & 0 \\ 0 & 1 \end{pmatrix},$$

where $\mathbf{0}$ is a square matrix of zeros of size C.

λI represents the rate at which $N_1(t)$ increases by 1 and $N_2(t)$ returns to the boundary level when there is a customer demand arrival and the common components buffer is empty, where I is an identity matrix of size $C + 1$.

4.3 Implementation

We notice that the queue occupancy at stage 2 can be modeled as a Quasi-Birth-Death process which allows us to develop a matrix geometric solution for its steady

state probability distribution. Using the matrix-geometric method described earlier, we solve for the steady state probability distribution (π) of the system. We use Matlab 7.0 to perform these computations. See Appendix B for Matlab codes. For computational purposes we set the maximum stage 2 queue length large enough so that the impact of truncating the state space is minimal and the probability that it is beyond this limit is close to zero. After some testing for this limit we set it to 100. We also set the maximum capacity for stage 1 (M) large enough such that there are no lost orders and consequently, the computed stationary probability distribution is accurate. After some testing with M we set it to 50.

The utilization of stage i (ρ_i) is defined by $\rho_i = \dfrac{\lambda}{\mu_i}$.

In the remainder of our work we focus on balanced capacity MTS-MTO systems, i.e, we set $\mu_1 = \mu_2$, and for different utilization rates we vary λ only.

The steady state probability distribution we computed has the following general form:

$$\pi = \begin{pmatrix} \pi_{0,0,0} & \cdots & \pi_{0,0,C} & \cdots & \pi_{0,M,C} \\ \pi_{1,0,0} & \cdots & \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \pi_{100,0,0} & \cdots & \vdots & \cdots & \pi_{100,M,C} \end{pmatrix}$$

where $\pi_{i,j,k}$ represents the probability of having i units at stage 2, j units at stage 1, and k units in the shipping buffer. We can find the steady state probability distribution for the number of units in stage 2 ($\pi 2$) from π. $\pi 2$ is a $(100, 1)$ vector and is calculated by the following relationship:

$$\pi 2 = \sum_{j=0,k=0}^{j=M,k=C} \pi_{i,j,k}$$

and the steady state probability distribution for the number of units at stage 1 ($\pi 1$) is a $(1, M+1)$ vector and is calculated as follows :

$$\pi 1 = \sum_{i=0,k=0}^{i=100,k=C} \pi_{i,j,k}$$

while the steady state probability distribution for the number of units in the shipping buffer ($\pi 3$) is a $(1, C+1)$ vector and is calculated from the following relationship:

$$\pi 3 = \sum_{i=0,j=0}^{i=100,j=M} \pi_{i,j,k}$$

We have computed all the necessary steady state probability distributions that are necessary to calculate the system steady state performance measures under different settings. Still we need to make sure that the system is operating under normal conditions and is stable. Therefore, we derive the stability conditions which are important to check before any runs are conducted.

Stability Condition:

In order to have a stable system, the Markov chain should be positive recurrent (Neuts,1981). This conditions is represented by the following relationship:

$$\pi A_2 \cdot \mathbf{1} > \pi A_0 \cdot \mathbf{1}$$

Since it is difficult to get an explicit form for the stability conditions due to the generator matrix form, we check the stability of the system using the previous relationship in each run.

4.4 Basic Performance Measures

The measures we use to evaluate the system performance are: the expected number of units at stage 1 $E(N_1)$, the expected number of units at stage 2 $E(N_2)$, the expected number of units in the shipping buffer $E(C)$, the expected number of backorders $E(B)$, order fulfillment delay $E(D)$, the expected number of units in the intermediate buffer $E(I1)$, and the expected number of semi finished inventory in the system $E(I)$. These performance measures are calculated from the stationary probability distribution of the number of units in the system (π).

The expected number of units at stage 1, $E(N_1)$, is calculated from the stationary probability for the number of units at stage 1 using the following equation:

$$E(N_1) = \sum_{i=0}^{M} \pi 1_i . i$$

where the subscript i represents the i^{th} element in $\pi 1$ vector.

The expected number of units at stage 2, $E(N_2)$, is calculated from the stationary probability distribution of the number of units at stage 2 using the following equation:

$$E(N_2) = \sum_{j=0}^{100} \pi 2_j . j$$

where the subscript j represents the j^{th} element in $\pi 2$ vector.

The expected number of common components in the shipping buffer, $E(C)$, is calculated from the stationary probability distribution of the number of units in the shipping buffer from the following relationship:

$$E(C) = \sum_{k=0}^{C} \pi 3_k . k$$

where the subscript k represents the k^{th} element in $\pi 3$ vector.

The expected number of backorders, $E(B)$, in the system is calculated from the stationary probability distribution of the number of units in the system, the number of backorders when $N_1 = i$ and $C = k$ is $\max(0, i + k - S)$. It is calculated from the following relationship:

$$E(B) = E(N_1 \mid N_1 + C > S) = \sum_{i=0,k=0}^{i=M,k=C} \pi_{i,k} . \max(0, i + k - S)$$

The expected number of common components in the intermediate buffer, $E(I1)$, is calculated from the following relationship, where $r_{i,k}$ is the number of common components in the intermediate buffer when $N_1 = i$ and $C = k$ and equals $\max(0, S - i - k)$:

$$E(I1) = \sum_{i=0,k=0}^{i=M,k=C} \pi_{i,k} . r_{i,k} .$$

The expected number of common components in the system, $E(I)$, is the sum of the common components in the intermediate buffer, $E(I1)$, the shipping buffer, $E(C)$, shipped quantity in transit, $\dfrac{C}{\mu_3}$, and common components at stage 2, $E(N_2)$. It is calculated from the following relationship:

$$E(I) = E(N_2) + E(I1) + \frac{C}{\mu_3} + E(C)$$

The order fulfillment delay, $E(D)$, is the expected time from the arrival of a customer order until it is fulfilled. It is calculated as the weighted average of the expected delay when an order arrives and finds the buffer empty, $E(D1)$, with the expected delay when an order arrives and finds the intermediate buffer non-empty, $E(D2)$. It is calculated in the following way:

$$E(D) = p1 * E(D1) + p2 * E(D2)$$

$$p1 = p(N_1 \mid N_1 + C > S) = \sum_{i=0,k=0}^{i=M,k=C} \pi_{i,k}. \min(1, \max(0, i + k - S))$$

$$p2 = p(N_1 \mid N_1 + C \leq S) = 1 - p1$$

where $p1$ is the probability that an order upon arrival finds the intermediate buffer empty, and $p2$ is the probability that an order upon arrival is fulfilled from the intermediate buffer. By Little's Law [35], the expected delay is the ratio of the expected number of units in a queue over the arrival rate, for our model this reduces to:

$$E(D1) = \frac{E(N_2) + E(B)}{\lambda}$$

$$E(D2) = \frac{E(N_2)}{\lambda}$$

4.5 Basic Performance Measure Analysis

To study the system behavior under the batch replenishment policy we introduced earlier, we vary separately the arrival rate (λ), the shipping quantity size (C), and the intermediate buffer size (S).

1. Performance evaluation with various arrival rates: we fix the processing rates at stage 1 and stage 2 to 2. We set the shipping rate to 1, $S = 3$, $C = 2$ and vary the arrival rate from 0.1 to 1.0. The results are summarized in Table 4.2 for the various system performance measures.

As we can see from Figure 4.2, the queue length in front of stage 1, the expected number of backorders, and the expected delay in the system increase rapidly as the

λ	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
$E(N_1)$	0.061	0.147	0.270	0.446	0.705	1.111	1.810	3.238	7.471	24.767
$E(N_2)$	0.053	0.111	0.177	0.249	0.324	0.395	0.449	0.467	0.423	0.313
$E(D)$	0.527	0.558	0.593	0.633	0.687	0.784	1.027	1.799	5.125	22.008
$E(B)$	0.001	0.008	0.029	0.080	0.190	0.417	0.906	2.090	6.043	23.076
$E(I1)$	2.346	2.181	1.999	1.798	1.574	1.324	1.045	0.734	0.387	0.070
$E(C)$	0.594	0.680	0.760	0.837	0.910	0.982	1.051	1.119	1.185	1.239
$E(I)$	4.993	4.972	4.936	4.884	4.809	4.701	4.546	4.320	3.995	3.622

Table 4.2: Performance evaluation with various arrival rates

arrival rate increases from 0.1 to 1.0. On the other hand, the expected queue length in front of stage 2 increases only slightly. The total expected intermediate inventory in the system decreased because with increased arrival rate, common components spend less time in the intermediate buffer. The system behavior under the increased arrival rate can be explained as follows: When the arrival rate increases up to a certain limit (0.8 in this case), the system cannot handle the orders arriving after this limit because the intermediate buffer is starving and hence, the expected queue length in front of stage 1 and the expected delay increases rapidly.

2. Performance evaluation with various shipping lot sizes: we vary the shipping lot size from 1 to 10 to observe the system behavior under different shipping quantities. We set $\lambda = 1$, $\mu_1 = 2$, $\mu_2 = 2$, $\mu_3 = 1$, and $S = 10$. The results are summarized in Table 4.3.

As we can see from Figure 4.3, the queue length in front of stage 1, the expected number of backorders, and the expected delay in the system decrease drastically as the shipping lot size increases from 1 to 3, then this decrease is slight afterwards.

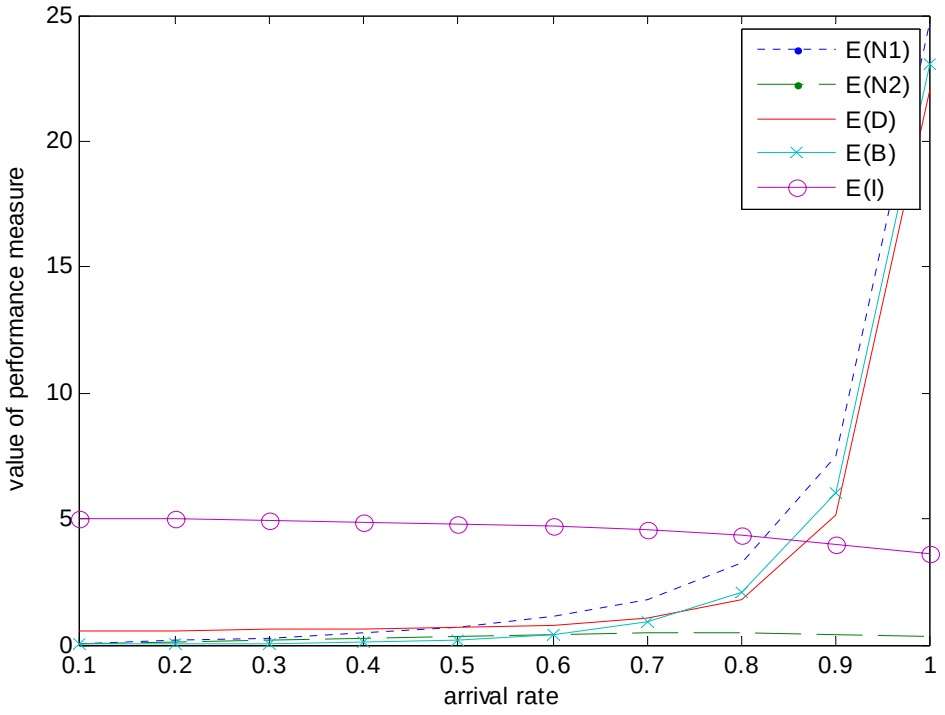

Figure 4.2: Performance evaluation with various arrival rates

This is because when the shipping quantity C is less than 3, the system is congested and the shipping process delays the orders from being filled because the shipping rate is slow with respect to processing rates (shipping is the bottleneck process). When we increase C beyond a certain limit (3 in this case), then the shortages are not incurred as much. This explanation holds for the behavior of the total expected intermediate inventory in the system, because when C is less than 3 the expected number of units in the intermediate buffer $E(I1)$ is close to 0 and then increases as the shipping lot size increases. The expected queue length in front of stage 2 increases slightly as we increase the shipping lot size.

C	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$E(N_1)$	48.391	24.767	4.391	2.684	2.124	1.847	1.681	1.571	1.493	1.434
$E(N_2)$	0.427	0.410	0.867	0.943	0.965	0.975	0.980	0.984	0.986	0.987
$E(D)$	40.151	14.986	1.277	1.125	1.124	1.154	1.215	1.319	1.492	1.771
$E(B)$	39.057	16.905	0.887	0.285	0.181	0.159	0.169	0.204	0.266	0.366
$E(I1)$	0.000	0.899	4.800	5.417	5.371	5.119	4.784	4.418	4.048	3.696
$E(C)$	0.667	1.239	1.697	2.184	2.686	3.194	3.703	4.214	4.725	5.236
$E(I)$	2.094	4.548	10.363	12.544	14.022	15.287	16.468	17.616	18.759	19.919

Table 4.3: Performance evaluation with various shipping lot sizes

S	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
$E(N_1)$	2.124	2.124	2.124	2.124	2.124	2.124	2.124	2.124	2.124	2.124	2.124
$E(N_2)$	0.965	0.976	0.983	0.988	0.992	0.994	0.996	0.997	0.998	0.999	0.999
$E(D)$	1.124	1.084	1.057	1.039	1.027	1.018	1.013	1.009	1.006	1.004	1.003
$E(B)$	0.181	0.126	0.087	0.060	0.042	0.029	0.020	0.014	0.010	0.007	0.005
$E(I1)$	2.686	2.686	2.686	2.686	2.686	2.686	2.686	2.686	2.686	2.686	2.686
$E(C)$	5.371	6.315	7.277	8.250	9.232	10.219	11.210	12.204	13.200	14.197	15.194
$E(I)$	14.022	14.977	15.946	16.924	17.909	18.899	19.892	20.887	21.884	22.881	23.880

Table 4.4: Performance evaluation with various intermediate buffer sizes

3. Performance evaluation with various intermediate buffer sizes: We vary S from 10 to 20 to observe the system behavior under various buffer sizes. We set $C = 10$, $\mu_1 = \mu_2 = 2$, $\lambda = 1$, and $\mu_3 = 0.2$. The results are summarized in Table 4.4.

As we can see from Figure 4.4, the expected delay and consequently, the expected number of backorders decreases as the intermediate buffer size increases, which is quite intuitive. The expected queue length in front of stage 1 does not change because changing the buffer size won't affect the arrival process. The expected queue length in front of stage 2 also does not change much as we increase the

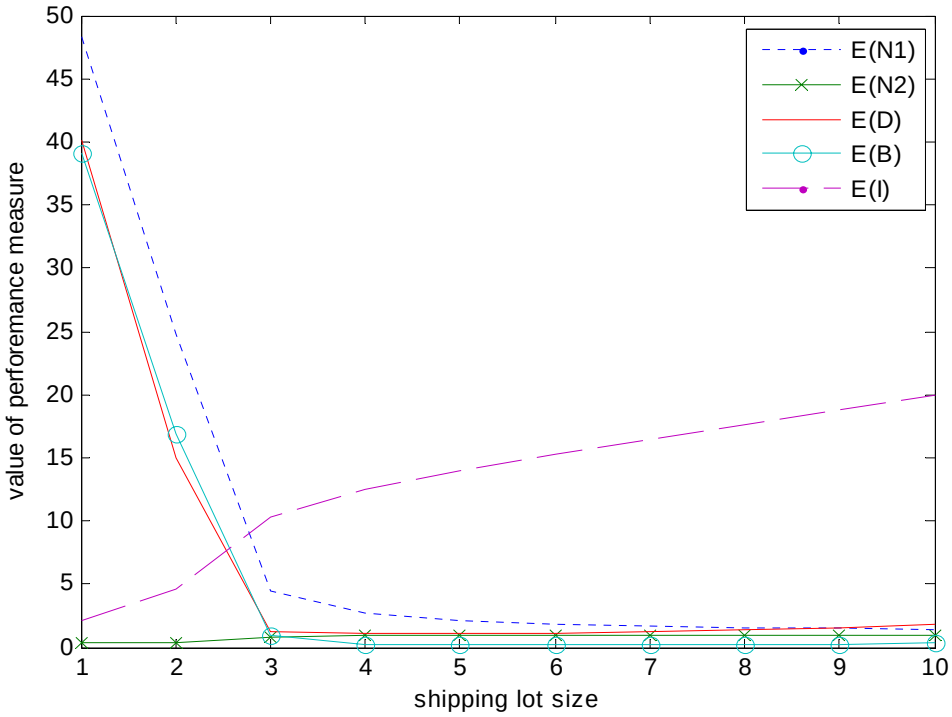

Figure 4.3: Performance evaluation with various shipping lot sizes

buffer size. The total intermediate inventory in the system increases because the expected units in the shipping buffer $E(C)$ increases while the expected inventory in the intermediate buffer $E(I1)$ is not affected.

4.6 Optimization Model

The problem we are trying to solve using our model is to find the optimal shipping lot size (C), and the optimal intermediate buffer size (S), which will lead to the minimum total cost. The costs incurred are holding inventory cost of semi finished

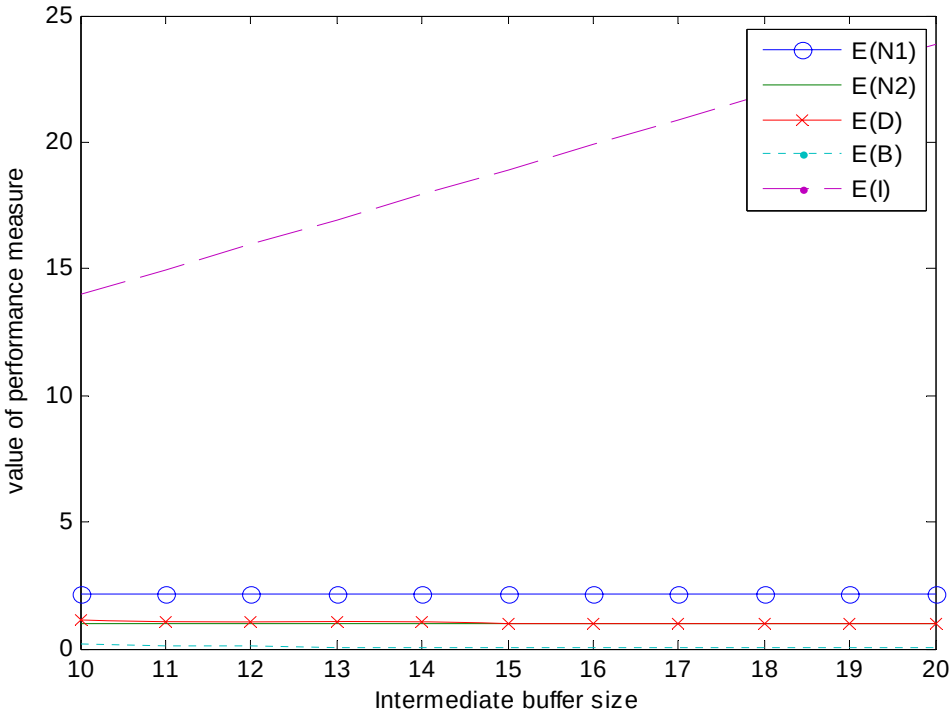

Figure 4.4: Performance evaluation with various buffer sizes

units in the system, shipping cost of semi finished units, and a penalty cost for customer order fulfillment delay. The optimization problem we are looking at is:

$$\min_{C,S} \left\{ TC(C,S) = C_h * E(I) + C_S * \frac{\lambda}{C} + C_D * E(D) \right\}$$
$$S.t. \quad 0 < C \leq S$$
$$C,\ S \quad \text{integers}$$

where:

C_h: the cost of holding inventory for semi finished units between the two stages per unit per unit time.

C_s: the cost of shipping for the semi finished units per shipment.

C_D: a penalty cost for customer order fulfillment delay per order per unit time.

The cost of shipping is calculated by multiplying the number of shipments $(\frac{\lambda}{C})$ by a fixed cost of shipping for each shipment.

4.7 Computational Results

We solve an example to find the optimal combination of (C, S) for the previous optimization problem by calculating the total cost for different combinations of (C, S). We vary the system utilization rate, where applicable, to analyze how the optimal solution changes under various utilization rates. We set the capacity of stage 1 to be large enough so that there are no lost customers. For the shipment rate (μ_3) we consider two options; the first one is fast shipment $(\mu_3 = 1)$, and the second one is slow shipment $(\mu_3 = 0.2)$. We set $\mu_1 = \mu_2 = 2$. We vary S and C keeping the condition $S \geq C$ in these variations. We set the cost parameters to represent low, medium, and high charges for each type. We assume the following values for the cost parameters:

$C_h = 0.1$, 0.25, and 2.0

$C_S = 1.0$, 5.0, and 10.0

$C_D = 0.5$, 2.0, and 5.0.

1. Slow delivery case: The results for the different combinations of cost parameters for the slow delivery case are summarized Table 4.5. Where $\rho_1 = \rho_2 = 50\%$. From these results we can draw the following observations:

- The system is not stable at utilization rates higher than 50%. This is because when the shipping takes a long time, then the intermediate buffer will be empty most of the time and waiting for the shipments to come. At higher utilization rates than 50%, the system is starving and as a result unstable. For this system with the previous defined parameters to be stable at higher utilization rates, then stage 1 should be very fast to be able to replenish orders, the rate should be greater or equal 8.

- As the penalty cost for customer delay increases, the optimal intermediate buffer size increases because holding more inventory in the intermediate buffer will decrease the probability an order is backlogged and hence, has to wait more time to be replenished. The optimal shipping quantity increases slightly as the penalty cost for customer delay increases for the same reason, but since the shipping takes a long time the major increase will be on the intermediate buffer size. On the other hand, as the holding inventory cost increases, the optimal intermediate buffer size decreases which is quite obvious.

- When the penalty cost for customer delay is very low, then most of the time, it is optimal to set the shipping lot size equal the intermediate buffer size. This can be explained as follows: having high penalty for customer delay will push C to be as large as possible, and the maximum allowed C by definition is S. Hence, $S = C$ in this case unless the cost of shipping is very high.

2. Fast delivery case: The results for the different combinations of cost parameters for the fast delivery case are summarized Table 4.6. Where $\rho_1 = \rho_2 =$ 50%, 75%. From these results we can draw the following observations:

Cost parameters			**50% utilization**	
C_h	C_s	C_D	(S, C)	Total cost
0.1	1.0	0.1	(8, 6)	1.4586
		2.0	(18, 7)	4.4561
		5.0	(21 ,8)	7.8402
	10.0	0.1	(9, 9)	2.7245
		2.0	(18, 10)	5.5131
		5.0	(21, 10)	8.8302
	20.0	0.1	(12, 12)	3.6959
		2.0	(21, 13)	6.4085
		5.0	(24, 13)	9.6845
0.25	1.0	0.1	(6, 6)	4.2502
		2.0	(14, 8)	8.2440
		5.0	(18, 8)	12.1120
	10.0	0.1	(6, 6)	4.2502
		2.0	(14, 8)	8.2440
		5.0	(18, 8)	12.1120
	20.0	0.1	(8, 8)	5.7073
		2.0	(15, 9)	9.4672
		5.0	(18, 9)	13.2650
2.0	1.0	0.1	(20, 1)	7.1784
		2.0	(8, 6)	26.0047
		5.0	(11, 6)	34.5287
	10.0	0.1	(4, 4)	14.9802
		2.0	(8, 6)	27.5047
		5.0	(11, 6)	36.0287
	20.0	0.1	(4, 4)	17.4802
		2.0	(8, 6)	29.1713
		5.0	(11, 6)	37.6953

Table 4.5: Optimal policies for slow delivery

- When the shipping time is fast compared to the processing times at both stages, then the optimal intermediate buffer size and the optimal shipping lot size are always less than the sizes when the shipping is slow. This is because in fast shipping the intermediate buffer is filled faster and as a result, there will be less shortages for the same cost parameters compared to slow delivery case.

- As the holding inventory cost is increasing, the intermediate buffer size and the shipping lot size gets closer together. This is more noticed when the shipping time is small compared to processing times. This is because as we increase the holding inventory cost, the optimal buffer size is decreasing. But since we have the constraint $S \geq C$, then the smallest buffer size allowed is C, this will drive $S = C$ when holding inventory cost is high.

- As the utilization increases for the system, the (S, C) combination gets larger to handle the increase in demand arrivals.

Some Managerial Insights

- If the demand arrival is high, the batch replenishment policy could be costly because orders are waiting to be shipped for long times.

- The settings of the decision variables (S, C) affects the behavior of the system. The shipping lot size C should be set large enough so that there is no starving in the system, and as the shipping time increases, this quantity should be increased so that the system can handle the orders without the increase in response time.

Cost parameters			**50% utilization**		**75% utilization**	
C_h	C_s	C_D	(S, C)	Total cost	(S, C)	Total cost
0.1	1.0	0.1	(3, 3)	0.9008	(4, 4)	1.1857
		2.0	(3, 2)	2.8830	(8, 4)	4.9468
		5.0	(4, 2)	5.8173	(9, 4)	10.4173
	10.0	0.1	(8, 8)	2.5892	(11, 11)	2.8479
		2.0	(10, 8)	4.8062	(16, 11)	7.3356
		5.0	(12, 8)	7.9481	(10, 5)	13.5105
	20.0	0.1	(12, 12)	3.6045	(11, 11)	3.7570
		2.0	(14, 12)	5.8283	(16, 11)	8.2447
		5.0	(16, 12)	8.9672	(19, 11)	14.5537
0.25	1.0	0.1	(2, 1)	1.3704	(3, 3)	1.8087
		2.0	(2, 2)	3.4575	(8, 3)	6.1385
		5.0	(2, 2)	6.4575	(9, 3)	11.7393
	10.0	0.1	(5, 5)	4.0198	(6, 6)	4.9226
		2.0	(6, 5)	6.3645	(7, 5)	9.2490
		5.0	(7, 5)	9.6555	(9, 5)	15.0955
	20.0	0.1	(7, 7)	5.6334	(11, 11)	6.2527
		2.0	(8, 7)	8.0356	(13, 11)	11.1894
		5.0	(9, 7)	11.3556	(16, 11)	17.8844
2.0	1.0	0.1	(1, 1)	5.2126	(2, 2)	6.5463
		2.0	(2, 2)	8.8200	(3, 3)	14.9040
		5.0	(2, 2)	12.2130	(5, 3)	22.7360
	10.0	0.1	(2, 2)	11.1711	(3, 3)	13.1739
		2.0	(2, 2)	13.3200	(3, 3)	19.4040
		5.0	(2, 2)	16.7130	(5, 3)	27.2360
	20.0	0.1	(3, 3)	15.8393	(3, 3)	18.1739
		2.0	(3, 3)	18.0167	(4, 4)	23.7140
		5.0	(3, 3)	21.4547	(5, 4)	31.2110

Table 4.6: Optimal policies for fast delivery

- The shipping time is a critical issue in the design of MTS-MTO systems decision variables.

In this chapter, we have introduced the second variation of the pure MTS-MTO system by batching replenishment orders for common components after processing at stage 1. This model is suitable when the two stages are physically distinct. In the next chapter we conclude our work and set some future directions for our research.

Chapter 5

Conclusions and Future Research

Manufacturing systems, in general, can be categorized into MTS or MTO systems. The main advantage of a MTS system is that it allows for the immediate satisfaction of customer orders due to the existence of finished goods inventory in the system. The disadvantage of a MTS is that it incurs high costs to the company to keep the finished goods inventory. The main advantage of a MTO system is that there are no inventory costs incurred since there is no finished goods inventory kept in the system. The disadvantage of a MTO is the high response time associated with each order. The combined MTS-MTO manufacturing policy combines the advantages of a MTS and a MTO. It allows for production based on customer requirements with less inventory costs and less response times. Most of the models developed in the literature for these systems assume a base stock policy for the control of inventory in the intermediate buffer. Little of the literature has explored order consolidation or batching which may be beneficial if the manufacturer incurs setup/ordering costs or shipping costs. This thesis explored two possible ways in which MTS-MTO systems could be adapted to take advantage of economies of scale in either ordering

or replenishing common components. The primary contribution of this work is to show the potential benefit of such batching and to demonstrate that there can be substantial savings to the manufacturer, but little cost to the consumer.

The first scenario we considered was the batching of orders, we developed a model for a MTS-MTO manufacturing system when there is an ordering cost incurred whenever an order is placed for common components. This model is a generalization for the base stock policy. We modeled the behavior of the system as a continuous Markov chain. Then we implemented the matrix-geometric method to compute the exact performance measures for the new proposed system. Then we developed an optimization model with the objective of minimizing the system overall costs. This model was used to find the optimal buffer size and the optimal batch size under the new batch ordering policy. We showed that the base stock policy is not always optimal under the new settings and compared the savings when a batch ordering policy is adopted.

The second scenario we considered was the batching of replenishment orders for common components. We developed a model for the MTS-MTO manufacturing systems that is more realistic when there is a shipping cost and/or a shipping time incurred during the replenishment process of orders and the two stages are physically distinct. We modeled the behavior of the system as a continuous Markov chain. Then we implemented the matrix-geometric method to compute the exact performance measures for the new proposed system. We developed an optimization model with the objective of minimizing the system overall costs. This model was used to find the optimal buffer size and the optimal shipping lot size under the new batch replenishment policy.

Future research should investigate more scenarios for the previous proposed models. In particular, we may consider the following deviations:

- The focus of this work was mainly on the system overall costs where the two stages are assumed to work cooperatively and the decision making process was centralized. Future work includes exploiting the difference when each of the two players is trying to minimize his own overall costs or maximize his own profit. In this case, we will deal with two objective functions that are contradicting. This is because the common components supplier, who experiences setup or shipping costs, will try to increase the batch size as much as possible. While the manufacturer who performs the customization will wish to decrease the batch size as low as possible to save on inventory costs. Multiple scenarios for such a multi-objective optimization problem may result, which in turn will affect the final decision variables and consequently, the total costs paid by each party. We also may investigate the difference between the saving margins for each party when the batching policy is adopted.

- In future research we may consider state dependent arrivals where the arrival rate decreases as the congestion increases in the system. In this case, there will be lost customers and the rate of losing customers will increase as the congestion increases in the system. The optimization problem will be modified to account for the lost customers. This may be added to the objective function as a penalty cost whenever there is a lost customer. We may investigate the effect of this variation on the final decision variables and the total cost incurred by the system.

- Since introducing the new batching policy has decreased the system overall costs but increased the response time (delay), in future research we may consider adding a service level constraint for the customer delay. The service level constraint will limit each order delay from exceeding a certain limit

instead of limiting the total average delay in the system as we assumed. This variation may affect the optimal decision variables we obtained earlier.

- In our work we considered one class of priority for orders, in future research we may consider different classes of customers. We may consider high priority customers who are willing to pay more for having their orders delivered in a guaranteed time frame and low priority customers who are not sensitive to the order fulfillment delay. The high priority customers orders will be processed in the MTO stage first and then the low priority customers. In order to model this case, we need to include the unit price and the extra charge for high priority customers in our optimization model. This is a realistic problem that a lot of computer assembly companies adopt to serve different types of customers. We may consider a profit maximization problem to find the optimal decision variable in the system.

All this work will lead to a better understanding of such complex MTS-MTO systems.

Bibliography

[1] Adan, I. and J. Wal, (1998). Combining make to order and make to stock. *OR Spectrum*. **20(2)** 73-81.

[2] Arreola-Risa, A. and G. Decroix, (1998). Make-to-order versus make-to-stock in a production/inventory system with general production times. *IIE Transactions* **30** 705-713.

[3] Aviv, W. and A. Federgruen, (2001). Capacitated Multi-Item Inventory Systems with Random and Seasonally Fluctuating Demands: Implications for Postponement Strategies. *Management Science*. **47(4)** 512-531

[4] Axstar, S., (1993). Exact and Approximate Evaluation of Batch-Ordering Policies for Two-Level Inventory Systems.*Operations Research*. **41(4)** 777-785.

[5] Benjafaar, S. and D. Gupta, (1999). Workload allocation in multi-product, multi-facility production systems with setup times. *IIE Transactions.* **31** 339-352.

[6] Bonvik A., C. Couch and S. Gershwin, (1997). A comparison of production-line control mechanisms. *International Journal of Production Research*. **35(3)** 789-804.

[7] Boute, R., M. Lambrecht and B. Houdt, (2007). Performance Evaluation of a Production/Inventory System with Periodic Review and Endogenous Lead Times. *Naval Research Logistics*. **54** 462-473.

[8] Breuer, L. and D. Baum, (2005). An Introduction to Queuing Theory and Matrix-Analytic Methods. Springer.

[9] Buzacott, J., (1989). Queueing models of Kanban and MRP controlled production systems. *Engineering Costs and Production Economist*. **17** 3-20.

[10] Buzacott, J. and J. Shanthikumer, (1993). Stochastic Models of Manufacturing Systems. Prentice Hall Inc.

[11] Carr, S. and I. Duenyas, (2000). Optimal Admission Control and Sequencing in a Make-To-Stock/Make-To-Order Production System. *Operations Research*. **48(5)** 709-720.

[12] Chakravarthy, S. and A. Alfa, (1997). Matrix-Analytic Methods in Stochastic Models. Marcel Dekker Inc. USA. Volume 183.

[13] Chen, F., (2000). Optimls Policies for Multi-echelon Inventory Problems with Batch Ordering.*Operations Research* **48(3)** 376–389.

[14] Donk, D. (2001). Make to stock or make to order: The decoupling point in the food processing industries. *International Journal of Production Economics.* **69(3)** 297-306.

[15] Dobson, G. and C. Yano, (2002). Product offering, pricing, and make-to-stock/make-to-order decisions with shared capacity. *Production and Operations Management*. **11(2)** 293-312.

[16] Duenyas, I. and P. Pantana-anake, (1998). Base-stock control for single-product tandem make-to-stock systems. *IIE Transactions.* **30** 31-39.

[17] Federgruen, A. and Z. Katalan, (1999). The Impact of Adding a Make-to-Order Item to a Make-to-Stock Production System. *Management Science.* **45(7)** 980-994.

[18] Grassmann, W. (2000). Computational Probability.

[19] Gupta, D. and S. Benjafaar, (2004). Make-to-order, Make-to-sock, or delay product differentiation? A common framework for modeling and analysis. *IIE Transactions.* **36** 529-546.

[20] Gupta, D. and N. Selvaraju, (2006). Performance Evaluation and Stock Allocation in Capacitated Serial Supply Systems. *Manufacturing & Service Operations Management* **8(2)** 169–191.

[21] Gupta, D. and W. Weerawat, (2006). Supplier–manufacturer coordination in capacitated two-stage supply chains. *European Journal of Operational Research.* **175** 67-89.

[22] He, M. and E. Jewkes, (1997). Flow time distributions in queues with customer batching and setup times. *INOFR.* **35(1)** 76-91.

[23] Hoekstra, S. and J. Romme, (1992). Integral Logistic Structures: Developing Customer-oriented Goods Flow, *McGraw-Hill*, London,

[24] Latouche, G. and V. Ramaswami, (1999). Introduction to Matrix Analytic Methods in Stochastic Modeling. *ASA-SIAM*, Alexandria, Virginia.

[25] Lee, H. and C. Tang, (1997). Modeling the costs and benefits of Delayed Product Differentiation. *Management Science.* **43** 40-53.

[26] Lee, Y. and P. Zipkin, (1992). Tandem queues with planned inventories. *Operations Research.* **40(5)** 936-946.

[27] Levi, D. and Y. Zhao, (2005). Safety Stock Positioning in Supply Chains with Stochastic Lead Times. *Manufacturing & Service Operations Management.* **7(4)** 295–318.

[28] Li, H. and L. Liu, (2006). Production control in a two-stage system. *European Journal of Operational Research.* **174(2)** 887-904.

[29] Liu, L., X. Liu and D. Yao, (2004). Analysis and Optimization of a Multistage Inventory-Queue System. *Management Science.* **50** 365–380.

[30] Moinzade, K. and H. Lee, (1986). Batch Size and Stocking Levels in Multi-Echelon Repairable Systems. *Management Science.* **32(12)** 1567-1581.

[31] Nelson, R., (1991). Matrix Geometric Solutions in Markov Models; A Mathematical Tutorial. *IBM Research Division. T.J. Watson Research Center.*

[32] Neuts, M.F. (1981). Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach, Johns Hopkins University Press, Baltimore.

[33] Papadopoulos H.T. and C. Heavey, (1996). Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research.* **92** 1-27.

[34] Rajagopalan, S., (2002). Make to Order or Make to Stock: Model and Application. *Management Science.* **48(2)** 241–256.

[35] Ross, S., (2006). Introduction to Probability Models. ELSEVIER, $8^{th}ed.$

[36] Serwer, A., (2002). Dell does domination. *Fortune Magazine.* **145(2)** 70–75.

[37] Swaminathan, J. and S. Tayur, (1999). Managing design of assembly sequences for product lines that delay product differentiation. *IIE Transactions.* **31** 1015-1025.

[38] Veatch M. and L.Wein, (1994). Optimal Control of a Two-Station Tandem Production/Inventory System. *Operations Research.* **42(2)** 337-350.

[39] Veinott, A., (1965). The Optimal Inventory Policy for Batch Ordering. *Operations Research.* **13(3)** 424-432.

[40] Wein, L., (1992). Dynamic Scheduling of a Multiclass Make to Stock Queue. *Operations Research.* **40(4)** 724-735.

[41] Williams, T.M., (1984). Special products and uncertainty in production/inventory systems. *European Journal of Operational Research.* **15** 46-54.

[42] Youssef, K., C. Delft and Y. Dellery, (2004). Efficient Scheduling Rules in a Combined Make-to-Stock and Make-to-Order Manufacturing System. *Annals of Operations Research.* **126** 103–134.

Appendix A

Matlab Code for Batch Ordering Policy

The Matrix-Geometric method was implemented using Matlab 7.0. The code for calculating the steady state probability distribution and the performance measures is as follows:

```
%%%%%example on the simple case of batch ordering
%%%%%Input Parameters
clc
clear
B=2; % batch size
lmdas=.1:.1:1.0; % demand arrival rate
mu1=2; % service rate at stage 1
mu2=2; % service rate at stage 2
S=3; % buffer size
```

```
M=50; % stage 1 capacity
%%%% Definition of Matrices
for w=1:10;
    lmda=lmdas(w);
I=eye(B);
%%%%B0 Matrix:
Bo=zeros((M+1)*B,(M+1)*B);
Bo(1:B,1:B)=-lmda*I;

count=1;
for i=1:S
  Bo(B+count:B-1+count+B,count:B+count-1)=mu1*I;
  count=count+B;
end

count=1;
for i=1:M
  Bo(B+count:2*B-1+count,B+count:2*B+count-1)=-(lmda+mu1)*I;
  count=count+B;
end

I0=eye(B-1);
C0=zeros(B,1);
R0=zeros(1,B-1);
B00=horzcat(C0,vertcat(lmda*I0,R0));
B01=horzcat(C0,vertcat(R0,lmda*I0));
```

```
Z0=zeros(B-1,B-1);
B02=horzcat(vertcat(R0',lmda),vertcat(Z0,R0));
count=1;
for i=1:M-S+1
    Bo(S*B+count:S*B+count+B-1,S*B+count:S*B+count+B-1)=
Bo(S*B+count:S*B+count+B-1,S*B+count:S*B+count+B-1)+B00;
   count=count+B;
end
count=1;
for i=1:2
    Bo((M-1)*B+count:(M-1)*B+count+B-1,(M-1)*B+count:(M-1)*B+count+B-1)
=Bo((M-1)*B+count:(M-1)*B+count+B-1,(M-1)*B+count:(M-1)*B+count+B-1)+B01;
   count=count+B;
end
count=1;
for i=1:1;
    Bo(S*B+count:S*B+count+B-1,M*B+count:M*B+count+B-1)=B02;
end
Bo;

%%%% Ao Matrix:
Ao=zeros((M+1)*B,(M+1)*B);
Aoo=zeros(B,B);
for i=1:B-1
    Aoo(i,i+1)=lmda;
end
```

```
Aoo;
count=1;
for i=1:S
    Ao(count:count+B-1,count:count+B-1)=Aoo;
   count=count+B;
end

count=1;
for i=1:M-S
    Ao((S+1)*B+count:(S+1)*B+count+B-1,S*B+count:S*B+count+B-1)=mu1*I;
    count=count+B;
end
A01=horzcat(vertcat(R0',lmda),vertcat(Z0,R0));
count=1;
for i=1:S
    Ao(count:count+B-1,B*B+count:B*B+count+B-1)=A01;
    count=count+B;
end;
Ao;
%%%% A1 Matrix:
A1=zeros((M+1)*B,(M+1)*B);
IR=eye((M+1)*B);
A1=Bo - mu2*IR;
A1;

%%%% A2 Matrix:
```

```
I1=eye((M+1)*B,(M+1)*B);
A2=I1*mu2;
A2;

%%%% finding R matrix:
eps=0.000000001;
R0=zeros((M+1)*B,(M+1)*B);
eps1=1.0;
while eps1>eps
    R1=-(Ao+(R0*R0*A2))*inv(A1);
    eps1=max(max(abs(R0-R1)));
    R0=R1;
end;
R=R0;

%%%%% calculatiog pi0 from normalization and boundary:

I2=eye((M+1)*B);
v=zeros(1,(M+1)*B);
one=ones((M+1)*B,1);
pi0=zeros(1,(M+1)*B);
new=Bo+(R*A2);
nor=inv(I2-R)*one;
new(:,1)=nor;
v(1)=1;
pi0=v*inv(new);
```

```
%%%%% calculating pi by recursive relation with pi0:
pi=zeros (100,(M+1)*B);
pi(1,:)=pi0;
for i=2:100;
    pi1=pi0*R;
   for j=1:(M+1)*B;
        pi(i,j)=pi1(j);
    end;
   pi0=pi1;
end;
pi;
s=sum (sum(pi));  % sum of S.S probabilty matrix should equal 1

%%%% Stability Condition:
a1=zeros(B*(M+1),1);
a2=zeros(B*(M+1),1);
a3=zeros(B*(M+1),1);
for i=1:100;
a1(i)=sum(pi(i,:));
a2(i)=sum(pi(i,1:(B*S)+1));
a3(i)=sum(pi(i,(B*S)+2 : B*(M+1)));
RHS(i)=(lmda*a2(i)+ mu1*a3(i))/(mu2*a1(i));
% the RHS should be less than 1 to have a stable system
end;
```

```
%%%%% Perforemance Measures:
%%%% Steady State probabilities for each state
pii2=zeros(100,1);
pii1=zeros(1,B*(M+1));
pii2=sum(pi');
pii2=pii2';             %stage 2 steady state probabilties
pii1=sum(pi);
pa=zeros(1,M+1);        %stage 1 steady state probabilties

count = 1;
for i=1:M+1;
    for j=1:B;
        pa(1,i)=pii1(1,j+count-1)+pa(1,i);
    end;
    count=count+B;
end;

EN2(w)=0;
for i=1:100;            % expexted number of units at stage 2
EN2(w)=pii2(i,1)*(i-1)+EN2(w);
end;
EN2(w);

EN1(w)=0;
for i=1:M+1;            % expextwd number of units at stage 1
EN1(w)=pa(1,i)*(i-1)+EN1(w);
```

```
end;
EN1(w);

%%% To calculate the expected number of units in system:
 p=zeros(100,M+1);       % probability for the number of units
 for l=1:100;
 count=1;
     for i=1:M+1;
         for j=1:B;
         p(l,i)=pi(l,j+count-1)+p(l,i);
         end;
         count = count+B;
     end;
 end;

%%%%% expected number of Backorders
EB2(w)=0;
for i=S:M+1;
    EB2(w)=pa(1,i)*max(0,(i-S-1))+EB2(w);
end;
EB2(w);
EB(w)=EB2(w)+EN2(w);
%%%%%%%%%expected delay for the customers
P2=0;
for i=1:S+1;
    P2=pa(1,i)+P2;
```

```
end;
P1=1-P2;
ED1(w)=EB(w)/lmda;
ED2(w)=EN2(w)/lmda;
ED(w)=P1*ED1(w)+P2*ED2(w)
%%%%% expected number of semi finished inventory
EI1(w)=0;
for i=1:M+1;
    EI1(w)=pa(1,i)*max(0,(S-i+1))+EI1(w);
end;
EI1(w);
EI(w)=EI1(w)+EN2(w);

EN2;      % expextwd number of units at stage 1
EN1;      % expextwd number of units at stage 2
ED;       % expected delay in the system
EB2;       % expextwd number of Backorders
plot(lmdas,EN1,lmdas,EN2,lmdas,ED,lmdas,EB,lmdas,EI)
xlabel('Arrival rate');
ylabel('Perforemance measure value');
%title('Plot of several perforemance measures','FontSize',12);
legend('E(N1)','E(N2) ','E(D)','E(B)','E(I)');
per=[EN1;EN2;ED;EB2;EI1;EI]
```

Appendix B

Matlab Code for Batch Replenishment Policy

The Matrix-Geometric method was implemented using Matlab 7.0. The code for calculating the steady state probability distribution and the performance measures is as follows:

```
%%%%%example on the simple case of batch replenishment policy
%%%%%Input Parameters
clc
clear
C=10;        % batch size
lmda=1;
mu1=2;      % service rate at stage 1
mu2=2;      % service rate at stage 2
mu3=.2;      % shipping to DC
SS=10:1:10;        % buffer size
```

```
M=50;       % stage 1 capacity
%%%% Definition of Matrices
for w=1:1;
    S=SS(w);
I=eye(C+1);
I1=eye(C);
Z=zeros(C,C);
%%%%B0 Matrix:
Bo=zeros((M+1)*(C+1),(M+1)*(C+1));
B03=zeros((C+1),(C+1));
B03((C+1),(C+1))=-mu3;
B02=zeros((C+1),(C+1));
B02((C+1),1)=mu3;
B00=zeros((C+1),(C+1));
C0=zeros(C+1,1);
R0=zeros(1,C);
B00=horzcat(C0,vertcat(mu1*I1,R0));
B01=zeros((C+1),(C+1));
B01=horzcat(vertcat(-mu1*I1,R0),C0);
Bo(1:C+1,1:C+1)=-lmda*I+B03+B02;
Bo(M*(C+1)+1:M*(C+1)+C+1,M*(C+1)+1:M*(C+1)+C+1)=B01+B03;
count=1;
for i=1:M-S+1;
    Bo(C+count+1:2*(C+1)+count-1,C+count+1:2*(C+1)+count-1)=
-lmda*I+B01+B02+B03;
   count=count+C+1;
```

```
end
count=1;
for i=1:M-S;
    Bo(S*(C+1)+count:S*(C+1)+count+C,S*(C+1)+count:S*(C+1)+count+C)=
-lmda*I+B01+B03;
   count=count+C+1;
end
count=1;
for i=1:M
  Bo(C+1+count:C+count+C+1,count:C+count)=B00;
  count=count+C+1;
end
count=1;
for i=1:M-S
    Bo(S*(C+1)+count:S*(C+1)+count+C,(S+1)*(C+1)+count:
(S+1)*(C+1)+count+C)=lmda*I;
    count=count+C+1;
end
Bo;

%%%% Ao Matrix:
Ao=zeros((M+1)*(C+1),(M+1)*(C+1));
A01=zeros((C+1),(C+1));
A01((C+1),1)=mu3;
count=1;
for i=1:S;
```

```
    Ao(count:C+count,C+count+1:2*(C+1)+count-1)=lmda*I;
   count=count+C+1;
end
count=1;
for i=1:M-S+1;
    Ao(S*(C+1)+count:S*(C+1)+count+C,S*(C+1)+count:S*(C+1)+count+C)=A01;
   count=count+C+1;
end
Ao;

%%%% A1 Matrix:
A1=zeros((M+1)*(C+1),(M+1)*(C+1));
IR=eye((M+1)*(C+1));
A1=Bo - mu2*IR;
A1;

%%%% A2 Matrix:
I1=eye((M+1)*(C+1),(M+1)*(C+1));
A2=I1*mu2;
A2;

%%%% finding R matrix:
eps=0.000000001;
R0=zeros((M+1)*(C+1),(M+1)*(C+1));
eps1=1.0;
while eps1>eps
```

```
    R1=-(Ao+(R0*R0*A2))*inv(A1);
    eps1=max(max(abs(R0-R1)));
    R0=R1;
end;
R=R0;
R;
% %%%% calculatiog pi0 from normalization and boundary:

I2=eye((M+1)*(C+1));
v=zeros(1,(M+1)*(C+1));
one=ones((M+1)*(C+1),1);
pi0=zeros(1,(M+1)*(C+1));
new=Bo+(R*A2);
nor=inv(I2-R)*one;
new(:,1)=nor;
v(1)=1;
pi0=v*inv(new)

%%%% calculating pi by recursive relation with pi0:
pi=zeros (100,(M+1)*(C+1));
pi(1,:)=pi0;
for i=2:100;
    pi1=pi0*R;
   for j=1:(M+1)*(C+1);
        pi(i,j)=pi1(j);
    end;
```

```
   pi0=pi1;
end;
pi;
s=sum (sum(pi))  % sum of S.S probabilty matrix should equal 1

%Stability condition
onee=ones((C+1)*(M+1),1);
aa=A2*onee;
ab=pi*aa;
ac=Ao*onee;
ad=pi*ac;
for i=1:100;
    RHS(i)=ad(i)/ab(i);  % should be less than one
end;
%%%%% Perforemance Measures:
%%%% Steady State probabilities for each state
pii2=zeros(100,1);
pii1=zeros(1,(C+1)*(M+1));
pii2=sum(pi');
pii2=pii2';             %stage 2 steady state probabilties
pii1=sum(pi);
pa=zeros(1,M+1);        %stage 1 steady state probabilties
pc=zeros(1,C+1);        %shipping buffer steady state probabilities

count = 1;
for i=1:M+1;
```

```
    for j=1:(C+1);
        pa(1,i)=pii1(1,j+count-1)+pa(1,i);
    end;
    count=count+(C+1);
end;

EN2(w)=0;
for i=1:100;            % expexted number of units at stage 2
EN2(w)=pii2(i,1)*(i-1)+EN2(w);
end;
EN2(w);

EN1(w)=0;
for i=1:M+1;            % expextwd number of units at stage 1
EN1(w)=pa(1,i)*(i-1)+EN1(w);
end;
EN1(w);

for i=1:C+1;
count = 1;
    for j=1:(M+1);
        pc(1,i)=pii1(1,count+i-1)+pc(1,i);
        count=count+(C+1);
    end;
end;
```

```
EC(w)=0;
for i=1:C+1;            % expextwd number of units at shipping buffer
EC(w)=pc(1,i)*(i-1)+EC(w);
end;
EC(w);
%%% To calculate the expected number of units in system:
 p=zeros(100,M+1);
 for l=1:100;
 count=1;
     for i=1:M+1;
         for j=1:(C+1);
         p(l,i)=pi(l,j+count-1)+p(l,i);
         end;
         count = count+(C+1);
     end;
 end;

E2 = zeros(100,1);
for j=1:100;
    for i=1:(M+1);
    E2(j,1)= p(j,i)*(i-1)+E2(j,1);
    end;
end;
E2;
E(w)=0;
for i=1:100;
```

```
E(w)=E2(i,1)*(i-1)+E(w);
end;
E(w);           %%%total expected customers in the system

% D(w)=E(w)/lmda;        % expected delay in the system
%%%%% expected number of Backorders
count=1;
EB2(w)=0;
for i=1:(M+1);
    for j=1:C+1
    EB2(w)=pii1(1,count)*max(0,(i+j-S-2))+EB2(w);
    count=count+1;
    end;
end;
EB2(w);
EB(w)=EB2(w);
%%%%%%%%expected delay for the customers
count=1;
P1(w)=0;
for i=1:(M+1);
    for j=1:C+1
    P1(w)=pii1(1,count)*min(1,max(0,(i+j-S-2)))+P1(w);
    count=count+1;
    end;
end;
P2(w)=1-P1(w);
```

```
ED1(w)=(EB2(w)+EN2(w)+EC(w))/lmda
ED2(w)=EN2(w)/lmda
ED(w)=P1(w)*ED1(w)+P2(w)*ED2(w)
%%%%% expected number of semi finished inventory
EI1(w)=0;       % intermediate inventory at the buffer
count=1;
for i=1:M+1;
    for j=1:C+1;
    EI1(w)=pii1(1,count)*max(0,(S-i-j+2))+EI1(w);
    count=count+1;
    end
end;
EI2(w)=0;       % intermediate inventory at the shipping buffer
for i=1:C+1;
    EI2(w)=pc(1,i)*(i-1)+EI2(w);
end;
EI(w)=EI1(w)+EN2(w)+ C/mu3 + EC(w);
end;
plot(SS,EN1,SS,EN2,SS,ED,SS,EB,SS,EI)
xlabel('Intermediate buffer size');
ylabel('value of perforemance measure');
%title('Plot of several perforemance measures','FontSize',12);
legend('E(N1)','E(N2) ','E(D)','E(B)','E(I)');
per=[ EN1;EN2; ED; EB;EC;EI1; EI]
per1=[EI;ED]
```