PERFORMANCE ANALYSIS AND DESIGN OF BATCH ORDERING POLICIES IN SUPPLY CHAINS

BY ABDULLAH SIDDIK KARAMAN

A dissertation submitted to the
Graduate School—New Brunswick
Rutgers, The State University of New Jersey
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
Graduate Program in Industrial and Systems Engineering

Written under the direction of
Professor Tayfur Altıok
and approved by

New Brunswick, New Jersey
May, 2007

ABSTRACT OF THE DISSERTATION

Performance Analysis and Design of Batch Ordering Policies in Supply Chains

by ABDULLAH SIDDIK KARAMAN

Dissertation Director: Professor Tayfur Altıok

Devising manufacturing/distribution strategies for supply chains and determining their parameter values have been challenging problems. Linking production management to stock keeping processes improves the planning of the supply chain activities, including material management, culminating in improved customer service levels. In this thesis, we investigate a multi-echelon supply chain consisting of a supplier, a plant, a distribution center and a retailer. Material flow between stages is driven by reorder point/order quantity inventory control policies. We develop a model to analyze supply chain behavior using some key performance metrics such as the time averages of inventory and backorder levels, as well as customer service levels at each echelon. The model is validated via simulation, yielding good agreement of robust performance metrics.

The metrics are then used within an optimization framework to help design the supply chain by calculating optimal parameter values minimizing the expected total cost. Optimal design of the material flow system is part of the overall planning and operation of a supply chain. The outcome of the optimization framework specifies not only how much and where to hold inventory but also how to move inventory across the supply chain.

The developed model requires limited computational requirements, which in turn helps frequently update the performance measures and optimal system parameters so as to be more responsive to short-term changes in demand or supply. In addition, it can be used as a decision support system for effective decision making as opposed to using simplistic inventory models, which results in significantly higher operating costs.

In a similar vein, we consider a distribution inventory system with one warehouse and several retailers. The challenge in this system is to describe the demand arrival process at the warehouse. We propose a procedure to characterize the demand arrival process at the warehouse as a superposition of several independent Erlang processes. An important characteristic of the superposed process is that although the individual processes are independent from each other, the superposed process may be no longer independent. We present a methodology to characterize such arrival streams as Markovian processes. We, then, extend the methodology to phase-type arrival streams as well.

Keywords and Phrases: Supply chains; Batch ordering policies; Finite production rate; Stochastic lead-times

Acknowledgements

I would like to express my sincere gratitude to Professor Tayfur Altıok for his supervision, guidance, support, and wisdom throughout the course of dissertation. His encouragement and understanding has brought this thesis come to an end. I am grateful to Professor Benjamin Melamed for his interest, invaluable suggestions and help. I also would like to thank him for being part of my committee, reading and reviewing this thesis. I am indebted to Professors Thomas O. Boucher and Wanpracha Chaovalitwongse for accepting to be part of my committee, reading and reviewing this thesis and for their suggestions.

I would like to take this opportunity to thank Ibrahim Bakır, Peter Bondarenko, Ilker Daştan, Cem Iyigün, Sezgin Kiren, and Ferhan Tunagür for being unique friends during my stay here. I would also like to thank my old roommates Tuğrul Diri and Gökhan Tuna for the long hours we spent together, their friendship, and support.

My special thanks go to my dearest parents for their encouragement, constant support and patience throughout my dissertation.

Finally, I would specially like to thank my wife Rabia for her love, unfailing encouragement, and support in this joyful and happy life.

Dedication

To My Wife: Rabia

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

Improving decision-making practices in a supply chain is a major source of competitive advantage in today's uncertain business environments. For years, different echelons in a supply chain have operated almost independently. However, there is strong evidence of success in supply chain performance in cases with high coordination among echelons. Efforts to link production management to various stock keeping processes result in better planning of the supply chain activities, better management of the materials, culminating in improved customer service levels and lower inventories.

Conflicting objectives often arise among the members of interacting systems fulfilling the customer demand. While, for example, the plant management tries to eliminate frequent setups and produce in large quantities, the distribution management tries to accelerate the flow of finished products to achieve higher flexibility and agility. Devising manufacturing/distribution strategies for supply chains and determining their parameter values have been challenging problems. Efficient and effective management is to produce and distribute at the right quantities, to the right locations, at the right times, while maintaining high customer service levels.

A supply chain is defined as a process that moves goods and information from points-of-origin to points-of-consumption. It includes a set of processes to efficiently link suppliers, manufacturers, distributors, and retailers in order to acquire raw materials, transform them into final products, ship these final products to intermediate storage locations, retailers and customers.

Motivation

A typical supply chain has a topology consisting of a number of retailers where customer demand occurs, distribution centers feeding retailers and other distribution centers, manufacturing plants supplying distribution centers, as well as vendors supplying raw materials to plants. Clearly, a concerted activity is needed across all the nodes for effective material flow in the supply chain. Controlling the material flow in a cost-effective manner has been a major challenge in practice. It depends on how well the demand from a customer or the next stage is forecasted at all levels and integrated into decision and control mechanisms. Inventory control policies are used for this purpose to achieve replenishment at the right quantity and the right time at each level. The more complex the supply chain topology, the more gain is achieved attending to concerted activities.

From a broader perspective, supply chain activities include strategic, tactical and operational decisions. Strategic decisions result in long-run plans. These are closely linked to the corporate strategy, and guide design issues, such as the number, location, and capacity of manufacturing plants and warehouses, and flow of materials in supply and distribution networks, among others. On the other hand, tactical decisions relate to plans and schedules to meet customer demand such as purchasing and production decisions, inventory policies, etc. The operational level focuses on day-to-day activities and executes plans. Tactical and operational level decision-making functions are closely related to each other and are distributed across the supply chain [93].

An important aspect of supply chain management is the establishment and monitoring of well-defined performance measures. A performance measure is used to assess the efficiency of plans and activities across the supply chain, which preferably plays a critical role in determining customer service level, responsiveness to customers. Some critical service measures include but are not limited to fill rate, total order cycle time, total response time to an order, average backorder levels, average lateness or earliness of orders relative to customer due dates, and flexibility. These measures, however, depend exclusively on inherited uncertainties in a supply chain. The sources of uncertainties, on the other hand, are due to supplier lead time and delivery performance, quality of incoming materials, manufacturing process time, transit times and demand, among many others [77]. A prerequisite to determine aforementioned performance measures is to develop models that take uncertainties into account to analyze supply chain behavior.

Scope

There are two important issues in developing a multi-echelon supply chain model, among others. Demand process is considered to be one of them. In general, demand at a facility depends on the decisions and operations of downstream locations. The second one, on the other hand, is the lead time, which depends mainly on the decisions and operations of upstream locations. The performance of the individual facilities depends on both demand process and lead time as well as its own operational rules. The former two processes in a long-term planning horizon is highly uncertain. Dealing with uncertainties, however, requires assumptions on the probability distributions of these values [98].

Over the past two decades, supply chain management has attracted significant attention from researchers and practitioners. This is mainly due to the potential gain achieved by the effective management of the supply chains. In addition, information and communication systems changed the traditional understanding and led to new organizational culture by providing access to data to all components of the supply chain. As a result, decision support systems implementing optimization-based algorithms are needed to account for interaction between all the nodes of the supply chain. Although some novel results have been presented to control material flows, more research needs to be done due to the broad and complex nature of the problem.

Contribution of This Thesis

In this thesis, we study a multi-echelon supply chain and its operational rules. Our aim is to develop a model to analyze supply chain behavior using some key performance metrics such as the time averages of inventory and backorder levels, as well as customer service levels at each echelon. The metrics are then used within an optimization framework. In multi-echelon supply chains, optimal production and inventory control policies have quite complex structures. This is because the control policy for a given echelon has considerable impact on the other echelons. In fact, the general practice is to restrict the control policies to a class of general operating schemes. All echelons, for example, apply reorder point/order quantity inventory control policies. Optimization in this sense is to coordinate such operating schemes in the best possible way. We propose an optimization procedure to help design the supply chain by calculating optimal parameter values minimizing the expected total cost. Optimal design of the material flow system is part of the overall planning and operation of a supply chain. The optimal configuration specifies not only how much and where to hold inventory but also how to move inventory across the supply chain.

The supply chain under consideration consists of a supplier, a plant, a distribution center, and a retailer. Material flow between stages is achieved by reorder point/order quantity inventory control policies. Production rate at the plant is finite and transportation times between stages are stochastic.

The developed model requires limited computational requirements, which in turn helps frequently update the performance measures and optimal system parameters so as to be more responsive to short-term changes in demand or supply. In addition, it can be used as a decision support system for effective decision making as opposed to using simplistic inventory models, which results in higher operating costs.

In a similar vein, we consider a distribution inventory system with one warehouse and several retailers. The challenge in this system is to describe the demand arrival process at the warehouse. We propose a procedure to characterize the demand arrival process at the warehouse as a superposition of several independent Erlang processes. A similar example is also a queue to which the arrival process is the superposition of separate arrival streams, each of whose inter-arrival times is of Erlang distribution. An important characteristic of the superposed process is that although the individual processes are independent from each other, the superposed process may be no longer independent. We present a methodology to characterize such arrival streams as Markovian processes. We, then, extend the methodology to phase-type arrival streams as well.

Forecasting is one of the key ingredients necessary to handle uncertainties in the early stages of planning. It is a crucial driver for procurement, manufacturing and distribution activities in a supply chain. Improving the quality of forecasts has been a challenging problem. Failure to account for large autocorrelations, trend, and seasonality in data sets is key ingredient contributing to lack of accuracy in forecasting. Time series models such as Box-Jenkins auto regressive integrated moving average (ARIMA), and multiple regression have been widely used to account for these type of patterns. Likewise, TES (Transform-Expand-Sample) models were utilized to generate forecasts for correlated data sets [73].

TES is a methodology [71, 72] to model empirical time series. Its merit is to capture both the empirical distribution and autocorrelation function, simultaneously. The analytical formulas of TES processes provide calculation of autocorrelations as well as its transition structure. Forecasts for the future can be calculated by utilizing the known transition structure of TES processes [73]. We also report an experimental study that compares TES process forecasting to traditional Box-Jenkins ARIMA models. Jagerman and Melamed [73] also implement the TES forecasting methodology based on the use of mixture of uniform random variables as the innovation density. Our study contains an extensive computational study of TES forecasting, and exploits phase-type random variables as the innovation density.

Organization

The rest of the thesis is organized as follows. Related literature is reviewed in Chapter 2. In Chapter 3, we describe a decomposition procedure in order to analyze batch ordering policies in a multi-echelon supply chain and obtain important performance measures. In Chapter 4, we propose a methodology illustrating how the performance measures can be used within an optimization framework. Chapter 5 applies the decomposition procedure to a distribution inventory system. Chapter 6 reports on an experimental study that compares TES process forecasting to traditional Box-Jenkins ARIMA models. Finally, Chapter 7 concludes the thesis by mentioning several extensions and future research directions.

Chapter 2
Literature Review

Effectively coordinating activities and decisions across multiple echelons in supply chains has been a challenging problem. From an operational perspective, decisions include but are not limited to deployment strategies (push versus pull), inventory control policies (the determination of the optimal levels of order quantities and reorder points, periodic versus continuous review, echelon and installation stock), and setting safety stock levels at each inventory holding location. These decisions play a critical role in determining the **customer service level**, the most critical measure of performance in a supply chain.

Multi-echelon Inventory Systems

A commonly investigated supply chain network is the multi-echelon, serial inventory system. Customer demand arrives at the last stage, the last stage orders from the next upstream stage, etc., and the first stage orders from a supplier which has unlimited capacity. In the ordering mechanism, either the echelon stock or the installation stock is used. Echelon stock is defined as the stock at any given installation plus stock in transit to or on hand at a lower installation, however installation stock is just the inventory on hand at a given installation.

The research in multi-echelon, serial inventory systems in terms of material handling practices dates back to the classical work of Clark and Scarf [43] where they considered a multi-echelon inventory system with periodic review. They computed the optimal ordering policy for each echelon separately. In order to link the successive echelons to each other, they utilized a penalty function evaluated in the case of stockouts. Federgruen and Zipkin [55] extended the Clark-Scarf approach to infinite horizon, and presented a new solution methodology. De Bodt and Graves [46] considered the multi-echelon model under continuous review and presented inventory control policies that minimize approximate expected cost per unit time. In a similar vein, Badinelli [17] investigated the problem where each facility utilizes an installation stock (R, Q) policy. The motivation behind using installation stock policies was the limited information requirements. Recently, Chen and Zheng [38], Chen [32], and Chen and Song [35] presented optimal policies of the model under different demand processes, in particular, compound Poisson, independent identically distributed, and Markov-modulated demand, respectively. Chiang and Monahan [41] considered a two-echelon inventory system with two channels of demand: a traditional retail store and an Internet-enabled direct channel. Jemai and Karaesmen [75], on the other hand, presented Nash equilibrium inventory strategies in a noncooperative environment.

Distribution Inventory Systems

Another commonly investigated network is the distribution inventory system. In this system, demand arises in the retailers in the form of some stationary stochastic process. An inventory control policy is utilized to maintain inventories at the retailers above certain threshold levels. A central warehouse (distribution center) supplies the retailers, which in turn replenishes its inventory according to a policy from an outside supplier with unlimited inventories. Initially, Sherbrooke [92] considered a depot-base system for repairable items where demand for items follow compound Poisson processes at the bases. An analytical solution was given to determine the optimal base-stock levels for each item subject to a limited system investment. Later, Moinzadeh and Lee [83] investigated the same system where the replenishment is made in batches. They provided a power approximation method to determine the optimal batch sizes and safety stocks.

Deuermeyer and Schwarz [47] and Schwarz *et. al.* [91] examined a single warehouse multi-retailer distribution system where each facility follows a continuous review (R, Q) policy and the identical retailers face stationary Poisson demand. An approximate model was presented to calculate the system service levels in [47], and an optimization framework was developed to maximize the system fill-rate subject to a system safety stock constraint in [91]. The system with one-for-one replenishments was investigated in [10], and a periodic review control policy was used in [30]. Forsberg [56], on the other hand, considered non-identical retailers. Chew and Tang [40] also considered non-identical retailers operating under an (s, S) policy. Recently, Chen *et. al.* [39] presented coordination mechanisms of a centralized system where the demand in each retailer arrives at a constant rate that is a general decreasing function of the retail price in the market.

In the aforementioned studies regarding multi-echelon distribution networks, the main idea has been to decompose the system into smaller subsystems, that is, decompose the system to a warehouse and retailers with their own procurement and demand arrival processes. Effective demand inter-arrival times at the warehouse and effective lead-times at the retailers were characterized. Then, procedures for the single-location models were utilized to obtain desired performance measures.

Svonoros and Zipkin [96], Axsäter [10, 11, 12], and Chen and Zheng [36] considered multi-echelon distribution system with some differences in the solution methodologies. Svonoros and Zipkin, and Axsäter exploited a solution methodology based on the approach to match every supply unit with a demand unit. In other words, they kept track of each supply unit and its sojourn time in the system and calculated the holding and backorder costs accordingly. Chen and Zheng, on the other hand, disaggregated the backorders at the warehouse among the retailers and then computed the long-run inventory levels.

A common characteristic of the above studies related to distribution inventory system is that all assume constant transportation times between the external supplier and the warehouse as well as between the warehouse and retailers. An exception to this was Svonoros and Zipkin [97] where they assumed stochastic transit times under base-stock policies. In addition, Erkip *et. al.* [51] considered the depot-warehouse system with correlated demands at the warehouses, and Nahmias and Smith [85] investigated the system with the partial lost sales assumption. Some recent reviews of the multi-echelon systems were Diks *et. al.* [48], Houtum *et. al.* [69], Thomas and Griffin [100], Beamon [20], and Erenguc *et. al.* [50].

Production Inventory Systems

In a similar vein, there is significant amount of research on modeling, analysis and design of integrated production/inventory systems. Altiok [6] studied a single-product system consisting of a production facility and a finished product warehouse. He used a continuous review (R, r) policy to control the inventory level at the warehouse and presented a procedure to compute cost minimizing values of R and r for both the backorder and lost sales case. Later, Altiok and Ranjan [9] investigated the multi-stage production/inventory systems in series.

This research is a continuation of the study by Gurgur and Altiok [61]. Gurgur and Altiok have further extended the multi-stage production/inventory system where each stage has its own input and output stock keeping activities. In particular, each stage was composed of a machine, an input buffer for the raw materials or semi-finished products, and an output buffer for the finished products. An (R, r) policy was used to control production within a stage and a (Q, R) policy was used to control procurement between stages. The system was decomposed to be able to analyze the performance measures of interest.

Ishii *et. al.* [70] and So and Pinault [94] considered pull type production/distribution systems. A method determining the base-stock levels and lead-times is given in [70], and a method estimating the safety stock is given in [94]. Pyke and Cohen [89] also developed a model to analyze the material flow in an integrated production/distribution system. They considered a single product system that consists of a factory, a finished goods stockpile (FG), and a retailer. A base-stock policy and a (Q, R) policy was used in the retailer and FG, respectively. They assumed constant transportation, processing and set-up times. The presented solution methodology analyzed the FG in isolation and evaluated the steady-state distribution of the stock on hand. The probabilities were then used to link the FG to the factory and to the retailer in order to find the distribution of inventories in these echelons.

Cohen and Lee [44] presented a model framework to measure cost/service tradeoffs for various management strategies. In particular, the framework can be used to assess the impact of various alternative manufacturing and material flow strategies. In a similar vein, Lee and Billington [78] studied a heuristic stochastic model for managing material flows. They considered a pull-type, periodic review, order-up-to inventory system, and determined the review period and order-up-to quantity. Toktay and Wein [101] incorporated forecasting into production/inventory systems. He *et. al.* [63] examined several inventory replenishment policies for a make-to-order inventory-production system and derived an optimal replenishment policy using a Markov decision process approach. Bernstein and DeCroix [21], on the other hand, considered an assembly system using base-stock policies. Boute *et. al.* [24] presented a procedure based on matrix-analytic techniques for computing the replenishment lead time distribution.

Design of Multi-echelon Systems

Optimal design of the material flow system is part of the overall planning and operation of a supply chain. The optimal configuration specifies not only how much and where to hold inventory but also how to move inventory across the supply chain. A prerequisite for such an optimization problem is a descriptive model of system performance as a function of control policies. A viable approach to solve the optimization problem is to employ a cost-minimizing objective function that assigns penalties for holding inventory and shortages. Optimal configuration here constitutes the best trade-off among set-up or ordering, holding, backordering and shortage costs.

Several different models for optimal control of production inventory systems were considered in literature. In particular, Zipkin [108] considered a multi-item single location inventory system and Veatch and Wein [102] considered multi-echelon inventory systems in series. An extension of such models is to consider capacity limits at the production echelons. Vericourt *et. al.* [103] analyzed a make-to-stock system where the supplier has limited production capacity and addressed the optimal stock allocation problem. Song and Yao [95] extended the optimization problem to assemble-to-order systems. Liu *et. al.* [79], on the other hand, studied the production inventory systems under fill rate constraints. A similar model with lost sales was investigated in [67] with at most one replenishment order outstanding.

In multi-echelon supply chains, optimal production and inventory control policies have quite complex structures. In fact, the general practice is to restrict the control policies to a class of general operating schemes. All echelons, for example, apply reorder point/order quantity inventory control policies. Optimization in this sense is to coordinate such operating schemes in the best possible way. Federgruen and Zipkin [54], Zheng and Federgruen [53], and Federgruen and Zheng [53] presented efficient algorithms to compute optimal control parameters. Finding optimal parameters is more difficult under service level constraints [15, 19] though it is a possible alternative to cost minimizing problem where it is difficult to quantify costs explicitly [88]. Yet, the solution procedures should be implementable in day-to-day operations. Upper and lower bounds for the optimal order quantities and reorder levels were derived in [2]. Also, a technique where a high-demand system is approximated by a low-demand system was given in [13].

Glasserman and Wang [60], on the other hand, studied the fill-rate bottlenecks, that is, facilities that most constrain the system-wide fill rate. Kim [76] incorporated lost sales into an optimal inventory model by using a queuing system with finite waiting room. Chiang [42] showed that a base-stock policy is optimal for the backorder case in a periodic review inventory system. A simple procedure for determining order quantities under a fill rate constraint and normally distributed lead-time demand presented in [16]. Adida and Perakis [1], Chen and Simchi-Levi [34], and Chen *et. al.* [33] studied optimal pricing and inventory control policies under general assumptions.

Sensitivity analysis of some standard single location models suggests that system performance is fairly insensitive to stock allocation in the vicinity of optimal solution [37]. Even, in many cases the optimal decisions does not depend on specific form of the demand distributions but on the means and standard deviations of demand [84]. Gallego and Zipkin [57] extended the analysis to multi-echelon systems and showed that similar results hold in case of constant lead times.

Modeling Issues in Multi-echelon Supply Chains

In the analysis of multi-echelon inventory systems, a general practice is to assume constant or independent, identically distributed transportation times. In a similar vein, another practice is to assume them to be phase-type distributed random variables because of their generality and versatility [7, 86]. Phase-type random variables enable to approximate any general distribution. Duri *et. al.* [49], Svonoros and Zipkin [97], and Zipkin [110] utilized phase-type random variables in modeling service and transportation times. Hayya *et. al.* [62] showed the effect of using different forecasting procedures in calculating variance of demand during lead-time.

An important issue in multi-echelon production distribution systems is to investigate the stability of the system. Most studies in this area assume unlimited production capacity. However, real world systems have finite production rate, and an ineffective policy may lead to high backorder levels. An exception to this is Glasserman and Tayur [59], in which they investigated the stability of a multi-echelon system under a base-stock policy and presented conditions for stable inventory and backorder levels.

Superposition of Multiple Arrival Streams

There are several situations in which the arrival process is the superposition of different arrival streams. Such an arrival process arises as the stream of replenishment orders in a distribution inventory system. The inventory system consists of many retailers replenishing their stock from a central warehouse where the retailers face independent, stationary Poisson demand and follow a continuous review (R, Q) inventory control policy. Another example is a queue to which the arrival process is the superposition of separate arrival streams, each of whose inter-arrival times is of Erlang distribution.

An important characteristic of the superposed process is that although the individual processes are independent from each other, the superposed process may no longer be independent. Additionally, exact characterization of the superposed process becomes computationally impractical as the number of the superposed processes increase. For these reasons, most of the work in this area delve into approximations. Typical methods approximate the superposed processes by renewal processes, which may be inadequate in capturing the temporal dependence.

Albin [3] developed a hybrid approximation scheme that combines stationary-interval method and asymptotic method of Whitt [106]. Both methods determine the approximating renewal process by identifying moments for the intervals between successive points and fitting a convenient distribution to the moments. Bitran and Dasu [22] developed an approximation using Super-Erlang chains, which takes into account the local and long-term behavior of the second-order measures of the nonrenewal process being approximated. Bitran and Dasu [23] analyzed a queue in which the arrival process is the superposition of separate arrival streams, each of whose inter-arrival time distributions is phase-type, and the service time distribution is also phase-type. The above approximation methods are based on first order and second order statistics. However, Girish and Hu [58] developed higher order approximations for the single server queue with general inter-arrival and service time distributions. Balcioğlu *et. al.* [18] used a three parameter renewal approximation in predicting the mean waiting time in a queue with deterministic service times. Vuuren and Adan [104], on the other hand, proposed an approximation method based on state space aggregation.

Chapter 3
Analysis of Single-Product, Multi-Echelon Supply Chains

We consider a single-product supply chain consisting of a supplier, a plant, a distribution center (DC), and a retailer arranged in series as illustrated in Figure 3.1. The retailer faces customer demand according to a Poisson process and has its own operating characteristics. It uses a continuous review (R_R, Q_R) installation stock inventory control policy, that is, when the inventory position (inventory on hand plus outstanding orders minus backorders) at the retailer down-crosses R_R, it orders a replenishment batch size of Q_R from the DC. The order arrives after a transportation lead-time delay, if the DC has sufficient on-hand inventory. Otherwise, it experiences additional delays due to stockouts at the DC. Any excess demand at the retailer is backlogged and filled as soon as the replenishment order arrives in a first-in first-out manner.

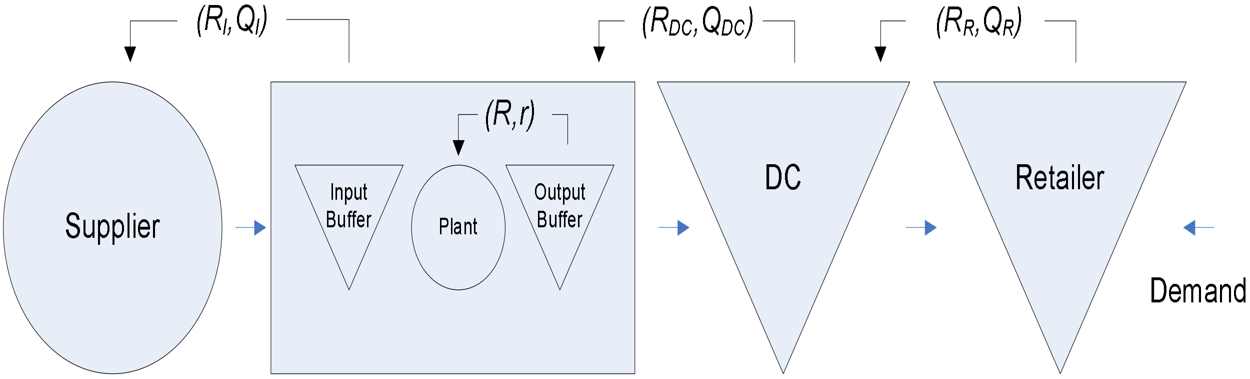


Figure 3.1: A multi-echelon supply chain

We assume that it is possible to have several outstanding backorders at the retailer at any point in time. The effective lead-time between the DC and retailer is the time between the placement and receipt of the order by the retailer. This includes the transportation lead-time as well as the delay in the DC due to the stockouts.

Demand at the DC are orders from the retailer and satisfied immediately if there is available stock on-hand. The unsatisfied demand is backordered. Similarly, the DC itself replenishes its inventory from the output buffer of the upstream plant, based on an (R_{DC}, Q_{DC}) installation stock inventory policy. The effective lead-time consists of the delay in the plant output buffer and the transportation time.

Note that we use installation stock policies in the replenishment process since they require relatively limited information, that is, only the inventory position at the current installation. On the other hand, echelon stock policies require the inventory position at the current installation and at all the downstream installations.

The plant is the echelon where production in the supply chain takes place. It includes two buffers: one for the raw materials and the other for the finished products, where both have their own stock keeping and production control policies. The plant operates a make-to-order manner, that is, the general aim is to produce as much as needed rather than to produce as much as possible. We assume that the facility produces one at a time, and every time the production activity takes place, one unit of raw material is pulled from the input buffer.

Orders from the DC are satisfied using the available inventory in the plant output buffer. A continuous review (R, r) policy is used to control production in the output buffer. It is an exhaustive production policy in the sense that, whenever the inventory level in the output buffer drops below r, the plant resumes production and continues until the inventory level reaches back R again. Additional stoppages in production may occur due to shortage of raw material in the input buffer.

The input buffer, in turn, orders from an external supplier with unlimited inventories. Again, the inventory control policy is of continuous review, installation stock (R_I, Q_I) type. No additional delays can occur at the supplier and the resulting lead-time includes only the transportation time.

We assume that all transportation times between facilities and production time at the plant are phase-type distributed because of their generality and versatility [7, 86]. However, there are some restrictions on transportation times. We assume that the units are processed sequentially in the transportation system. In other words, replenishment orders do not cross over time, and they are received in the same order they were placed. In contrast, assuming independent, identically distributed random variables represents parallel processing of replenishment orders and allows orders to cross in time. Zipkin [108], and Svoronos and Zipkin [97] utilize same concept of transportation times. We assume, in particular, all transportation times follow a k'th order Erlang (Erlang-k) distribution and the production time at the plant comes from a mixture of generalized Erlang distribution *(MGE-K)*. Erlang-k and *MGE-K* distributions are special cases of phase-type distributions. See Appendix A for a brief introduction to the phase-type random variables.

Performance evaluation of the system above is quite difficult because of its complex nature and large state-space. Indeed, we next present a decomposition procedure, which uses single-location models as building blocks to analyze the entire supply chain. The performance measures of interest are the long-run average number of inventories, the number of backorders, and the customer service levels in each facility. The measures will then be used within an optimization context (cost minimization subject to a given customer service level) to choose among several policy parameters.

3.1 Modeling Approach

It is plausible that the entire system can be modeled using a Markovian approach. However, it is easily seen that exact analysis of the above system is computationally impractical due to the fast growing state space of the underlying Markov chain. Hence, the only viable approach, other than simulation, is approximation. Widely used approximation techniques decomposes the system into several subsystems, which can be analyzed in isolation. Then, an iterative procedure links the subsystems to each other. Here, we will implement a similar procedure.

Let us consider the supply chain shown in Figure 3.1. We will decompose the system in such a way that each subsystem consists of an inventory holding buffer with its own stock keeping policy. Keeping in mind that lower levels of inventory in the upstream facilities will result in longer lead times for downstream stages, and higher levels of inventory in the downstream facilities will cause longer demand inter-arrival times at the upstream stages, we characterize the appropriate effective procurement lead-times and effective demand inter-arrival times at each subsystem. Consequently, we treat each subsystem as a single-location production or inventory system, which can be analyzed with a modest computational effort. Finally, we relate the subsystems to each other by using an iterative scheme. The decomposition method adopted is based on [7, 9, 61]. In summary, it includes constructing each subsystem, deriving a set of equations for the unknown parameters, and linking the subsystems to each other. Now, let us introduce the following notation:

λ : demand rate at the retailer,

$TT_{S,P}$: transportation time between supplier and plant,

MT_P : manufacturing time at the plant,

$TT_{P,DC}$: transportation time between plant and DC,

$TT_{DC,R}$: transportation time between DC and retailer,

$\Omega(i)$: subsystem involving inventory buffer i, $i = I, O, DC, R$

M_i' : node modeling procurement to inventory buffer i, $i = I, O, DC, R$,

M_i'' : node modeling demand arrival process to buffer i, $i = I, O, DC, R$,

U_i' : processing time at M_i', $i = I, O, DC, R$;

U_i'' : processing time at M_i'', $i = I, O, DC, R$;

N_i : inventory level in $\Omega(i)$, $i = I, O, DC, R$.

We propose to develop a decomposition as shown in Figure 3.2. The first subsystem, $\Omega(I)$, includes the input buffer of the plant in the supply chain. An (R_I, Q_I) inventory control policy is used to control replenishment process at the input buffer. Node M_I' models the effective procurement process and M_I'' models the effective demand inter-arrival process at the input buffer. Similarly, the second subsystem, $\Omega(O)$, includes the plant output buffer. An (R, r) policy is used to control production. Node M_O' represents procurement process and node M_O'' represents demand arrival process. Other subsystems are described accordingly. In the following sections, we explain how we construct the nodes M_i' and M_i'''s and their respective processing times U_i' and U_i'''s for $i = I, O, DC, R$.

3.1.1 Analysis of Procurement Times

In this section, we analyze the effective procurement times at each subsystem. In general, in addition to transportation times, the procurement times include possible delays experienced

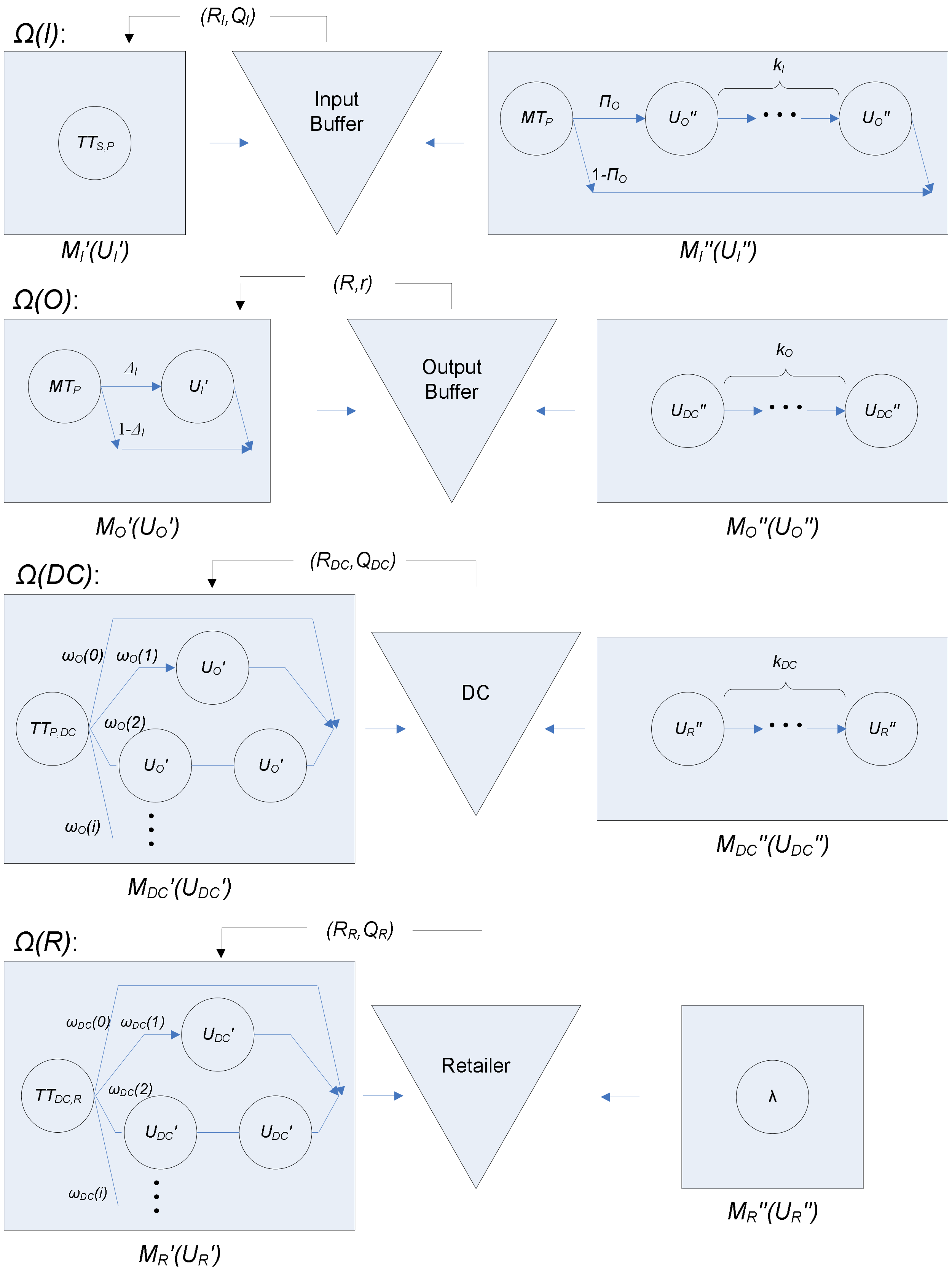


Figure 3.2: Subsystems $\Omega(I)$, $\Omega(O)$, $\Omega(DC)$ and $\Omega(R)$

at the corresponding sources. We start with the subsystem containing the plant input buffer since it is the first upstream facility of the supply chain, and continue with the other subsystems in an orderly manner.

For subsystem $\Omega(I)$, the random variable $U_I^{'}$ represents the effective procurement time at the input buffer. Since, the supplier has always sufficient raw material to replenish the input buffer, the effective procurement time consists only of the transportation lead-time from supplier to the input buffer. That is,

$$U_I^{'} = TT_{S,P}.$$

The first node in the second subsystem, $M_O^{'}$, represents the procurement process of the output buffer. The procurement time is simply the manufacturing time, MT_P, at the plant when there is available inventory in the input buffer. However, when the input buffer is out-of-stock, we have to wait for a replenishment order to arrive at the input buffer. More rigorously, let Δ_I be the conditional probability that there are no units in the input buffer given that a unit is about to finish processing at the plant. Then, the effective procurement time is given by:

$$U_O^{'} = \begin{cases} MT_P & \text{w.p.} \quad 1 - \Delta_I, \\ MT_P + TT_{S,PL} & \text{w.p.} \quad \Delta_I. \end{cases}$$

The procurement time at the DC, $U_{DC}^{'}$, however, is more involved. The DC replenishment request is filled as soon as it is received, if the output buffer has ample stock on hand. Otherwise, a delay occurs until sufficient number of units accumulate in the output buffer since no partial shipment is allowed between facilities. Let $\omega_O(i)$ denote the conditional probability that there are i units missing $(i = 0, 1, 2, \ldots)$ in the output buffer at the time a procurement order is received from the DC. Then, the effective procurement time will be:

$$U_{DC}^{'} = \begin{cases} TT_{P,DC} & \text{w.p.} \quad \omega_O(0), \\ TT_{P,DC} + \sum_{j=1}^{i} U_O^{'} & \text{w.p.} \quad \omega_O(i). \end{cases}$$

Finally, let $\omega_{DC}(0)$ be the conditional probability that there are no units missing in the DC given that a demand arrives from the retailer, and let $\omega_{DC}(i)$, $i = 1, 2, \ldots$ be the conditional probability that there are, for any i, $(i-1) * Q_{DC} + 1, (i-1) * Q_{DC} + 2, \ldots, i * Q_{DC}$ units missing in the DC given that a demand arrives from the retailer. Then, the effective lead time to the retailer is given by:

$$U_R^{'} = \begin{cases} TT_{DC,R} & \text{w.p.} \quad \omega_{DC}(0), \\ TT_{DC,R} + \sum_{j=1}^{i} U_{DC}^{'} & \text{w.p.} \quad \omega_{DC}(i). \end{cases}$$

It is clear that, with probability $\omega_{DC}(0)$, there is enough stock in the DC and the order experiences no delays. On the other hand, with probability $\omega_{DC}(i)$, the DC does not have sufficient inventories resulting in delay in the replenishment process. Note that, this delay is approximately i procurement lead times from the output buffer to the DC.

3.1.2 Analysis of Demand Inter-Arrival Times

In this section, we analyze the effective demand inter-arrival times at subsystems. These are generally simpler than the analysis of the effective procurement times. Here, we start with the subsystem including the retailer, and continue with the rest in an orderly manner.

The retailer faces customer demand according to a Poisson process with constant rate λ. Equivalently, the effective demand inter-arrival times are independent and follow an exponential distribution.

Demand to the DC arrives from the retailer that uses an (R_R, Q_R) stock keeping policy to place orders. That is, when the inventory position in the retailer down-crosses R_R, the retailer places a replenishment order and the inventory position is immediately updated to $R_R + Q_R$. The next replenishment order from the retailer is triggered when the inventory position again drops below R_R. So, every time the retailer receives $k_{DC} = Q_R$ orders, it places a replenishment request to DC. As a result, the orders to the DC follow an Erlang distribution with phase rate λ and k_{DC} phases due to the fact that orders to the retailer follow a Poisson process with intensity λ.

In order to characterize the effective demand inter-arrival time at the output buffer, let $k_O = \lceil Q_{DC}/Q_R \rceil$ with the operator, $\lceil\rceil$, denoting the ceiling function. A procurement order is placed by the DC to the output buffer every time there are k_O orders from the retailer. This is because, with every k_O orders to the DC, its inventory position drops to R_{DC} again, and a replenishment order is placed by the DC. Consequently, the effective demand inter-arrival time to the output buffer follows an Erlang distribution with k_O phases, with every phase being an Erlang random variable with rate λ and k_{DC} phases.

Effective demand inter-arrival times to the input buffer are more involved. Manufacturing in the supply chain takes place at the plant. Every time the plant produces one product, a unit of raw material is withdrawn from the input buffer. So, the random variable, U_I'', includes the manufacturing lead-time, MT_P, at the plant. On the other hand, there are stoppages due to the production control policy used in the plant. As we mentioned before, an (R, r) policy is in effect at the plant. That is, when the inventory level drops below r, the plant resumes production and continues until the inventory level reaches back R again. When the target value R is attained, the plant goes into an idle period and remains there for $R - r$ departures to occur from the output buffer. Rigorously, let Π_O be the conditional probability that there is only one space available in the output buffer at the time a unit is about to finish processing at the plant. Accordingly, the effective demand inter-arrival time is as follows:

$$U_I'' = \begin{cases} MT_P & \text{w.p.} \quad 1 - \Pi_O \\ MT_P + \sum_{j=1}^{k_I} U_O'' & \text{w.p.} \quad \Pi_O \end{cases}$$

Thus, with probability $1 - \Pi_O$, the inter-arrival time consists only the manufacturing lead-time at the plant, and with probability Π_O, it includes both the production time and the time it remains blocked where $k_I = \lceil (R - r)/Q_{DC} \rceil$.

3.1.3 Steady-State Analysis of the Subsystems

In this section, we calculate the steady-state probabilities of the underlying Markovian process in each subsystem. Each of the subsystems, $\Omega(i)$ for inventory holding buffers $i = I, O, DC, R$, is a two-node subsystem with its own stock keeping policy, and phase-type procurement and demand inter-arrival times. The use of the phase-type random variables gives rise to a Markovian analysis, and matrix-recursive procedures based on [29, 64, 86] are used to obtain steady-state probabilities and the measures of interest. We assume all transportation times follow a second order Erlang distribution (Erlang-2) and the production time at the plant comes from a mixture of generalized Erlang distribution *(MGE-2)* for numerical convenience. The following notation is needed:

β_i : transportation time (rate of Erlang-2), $i = S, P, DC$,

$\beta_i^{'}$: processing rate of $U_i^{'}$, $i = I, O, DC, R$,

$\beta_i^{''}$: processing rate of $U_i^{''}$, $i = I, O, DC, R$,

$\mu_i^{'}$: rate of *MGE-2* (processing time) at the plant, $i = 1, 2$,

a : prob. of moving from first phase to the second phase of *MGE-2*.

Analysis of Subsystem Involving Plant Input Buffer

Let us start with subsystem $\Omega(I)$, the subsystem involving the plant input buffer. Let $\{I_t, J_t, N_t, t \geq 0\}$ be a stochastic process where I_t represents the phase of $U_I^{'}$, J_t represents the phase of $U_I^{''}$, N_t denotes the number of inventories in the input buffer where

$$I_t = \begin{cases} i, & U_I^{'} \text{ is in phase } i, \; i = 1, 2, \\ B, & U_I^{'} \text{ is blocked,} \end{cases}$$

$$J_t = \begin{cases} 0, & U_I^{''} \text{ is starving,} \\ i, & U_I^{''} \text{ is in phase } i, \; i = 1, 2, \ldots, k_I + 2, \end{cases}$$

$$N_t = \; 0, 1, 2, \ldots, R_I + Q_I,$$

making $\{I_t, J_t, N_t, t \geq 0\}$ a Markov chain with a finite number of states. Here, demand to the input buffer arrives singly, however, the supply comes in batches of Q_I. The state-space and the transitions of the Markov chain are presented in Figure 3.3. Let us define the following steady-state probabilities:

$$\tilde{\mathbf{P}}(i,0,0) = \begin{bmatrix} P(i,0,0) \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(k_I+2)\times 1}, \quad \tilde{\mathbf{P}}(i,j,n)|_{n=0}^{R_I} = \begin{bmatrix} P(i,1,n) \\ P(i,2,n) \\ \vdots \\ P(i,k_I+2,n) \end{bmatrix}_{(k_I+2)\times 1},$$

for $i = 1, 2$, and

$$\tilde{\mathbf{P}}(B,j,n)|_{n=R_I+1}^{R_I+Q_I} = \begin{bmatrix} P(B,1,n) \\ P(B,2,n) \\ \vdots \\ P(B,k_I+2,n) \end{bmatrix}_{(k_I+2)\times 1}.$$

In fact, $\tilde{\mathbf{P}}(i,j,n)$ denotes the steady-state probability vector that the effective procurement time is in phase i, the demand inter-arrival time is in phase j, and the input buffer contains

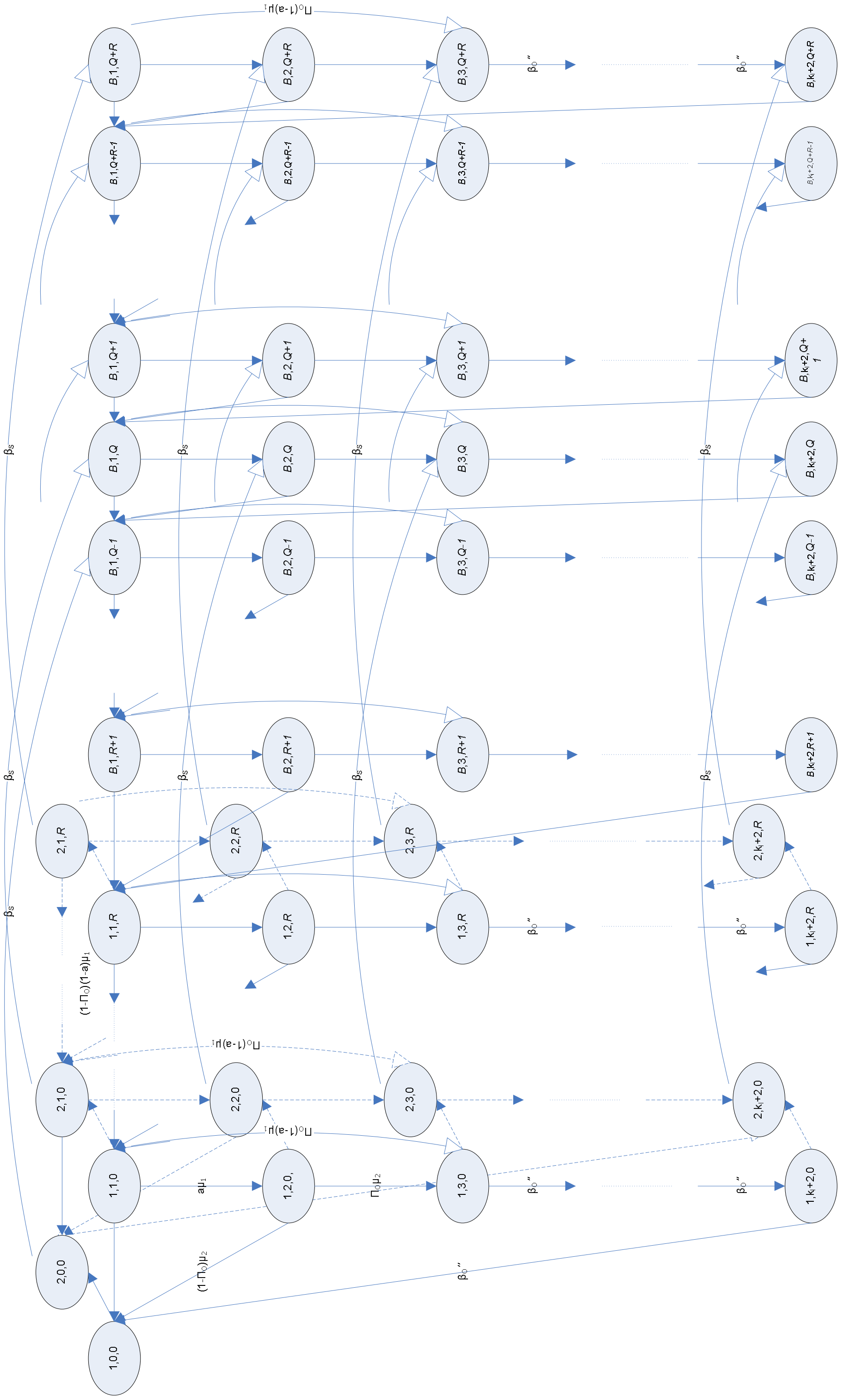


Figure 3.3: Transition diagram for subsystem $\Omega(I)$ (backordering case)

n units of raw material. Similarly, $\tilde{\mathbf{P}}(B, j, n)$ represents the steady-state probability that the procurement process is blocked, the effective demand inter-arrival time is in phase j, and there are n units of raw material in the input buffer. The rest of the probabilities are defined in the same manner.

We use the following sets of flow-balance equations in order to obtain the long-run probabilities of the subsystem:

$$\begin{aligned}
\beta_S\tilde{\mathbf{P}}(1,0,0) &= \mathbf{B}\tilde{\mathbf{P}}(1,j,0), \\
\mathbf{A}\tilde{\mathbf{P}}(1,j,n) &= \mathbf{B}\tilde{\mathbf{P}}(1,j,n+1), \quad n = 0,1,2,\ldots,R_I \\
\beta_S\tilde{\mathbf{P}}(2,0,0) &= \mathbf{B}\tilde{\mathbf{P}}(2,j,0) + \beta_S\tilde{\mathbf{P}}(1,0,0), \\
\mathbf{A}\tilde{\mathbf{P}}(2,j,n) &= \mathbf{B}\tilde{\mathbf{P}}(2,j,n+1) + \beta_S\tilde{\mathbf{P}}(1,j,n), \quad n = 0,1,2,\ldots,R_I - 1 \\
\mathbf{A}\tilde{\mathbf{P}}(2,j,R) &= \beta_S\tilde{\mathbf{P}}(1,j,R), \\
\mathbf{C}\tilde{\mathbf{P}}(B,j,n) &= \mathbf{B}\tilde{\mathbf{P}}(B,j,n+1), \quad n = R_I+1, R_I+2, \ldots, Q_I - 2 \\
\mathbf{C}\tilde{\mathbf{P}}(B,j,Q_I-1) &= \mathbf{B}\tilde{\mathbf{P}}(B,j,Q_I) + \beta_S\tilde{\mathbf{P}}(2,0,0), \qquad (3.1) \\
\mathbf{C}\tilde{\mathbf{P}}(B,j,n) &= \mathbf{B}\tilde{\mathbf{P}}(B,j,n+1) + \beta_S\tilde{\mathbf{P}}(2,j,n-Q_I), \quad n = Q_I, Q_I+1, \ldots, Q_I+R_I-1 \\
\mathbf{C}\tilde{\mathbf{P}}(B,j,Q_I+R_I) &= \beta_S\tilde{\mathbf{P}}(2,j,R_I),
\end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix}
\mu_1+\beta_S & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\
-a\mu_1 & \mu_2+\beta_S & 0 & 0 & 0 & \ldots & 0 & 0 \\
-\Pi_O(1-a)\mu_1 & -\Pi_O\mu_2 & \beta_O^{''}+\beta_S & 0 & 0 & \ldots & 0 & 0 \\
0 & 0 & -\beta_O^{''} & \beta_O^{''}+\beta_S & 0 & \ldots & 0 & 0 \\
0 & 0 & 0 & -\beta_O^{''} & \beta_O^{''}+\beta_S & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \ldots & -\beta_O^{''} & \beta_O^{''}+\beta_S
\end{bmatrix}_{(k_I+2)\times(k_I+2)},$$

$$\mathbf{B} = \begin{bmatrix}
(1-\Pi_O)(1-a)\mu_1 & (1-\Pi_O)\mu_2 & 0 & \ldots & 0 & \beta_0^{''} \\
0 & 0 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}_{(k_I+2)\times(k_I+2)},$$

$$\mathbf{C} = \begin{bmatrix} \mu_1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -a\mu_1 & \mu_2 & 0 & 0 & 0 & \dots & 0 & 0 \\ -\Pi_O(1-a)\mu_1 & -\Pi_O\mu_2 & \beta_O^{''} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -\beta_O^{''} & \beta_O^{''} & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & -\beta_O^{''} & \beta_O^{''} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & -\beta_O^{''} & \beta_O^{''} \end{bmatrix}_{(k_I+2)\times(k_I+2)} .$$

In total, we have $(k_I+2)\times(Q_I+2R_I+2)$ unknowns. Representing $\tilde{\mathbf{P}}(B,j,Q_I)$ in terms of $\tilde{\mathbf{P}}(B,j,Q_I-1)$, and utilizing the Equation 3.1, we obtain

$$\tilde{\mathbf{P}}(B,j,Q_I-1) \quad = \quad \beta_S(\mathbf{C}-\beta_S^2\mathbf{B}\mathbf{W}(\mathbf{C}^{-1}\mathbf{B})^{Q_I-R_I-2})^{-1}\tilde{\mathbf{P}}(2,0,0),$$

where

$$\begin{aligned} \mathbf{W} \;=\; & ((\mathbf{C}^{-1}\mathbf{B})^{R_I}\mathbf{C}^{-1}\mathbf{A}^{-1}+(\mathbf{C}^{-1}\mathbf{B})^{R_I-1}\mathbf{C}^{-1}(\mathbf{A}^{-1}\mathbf{B})\mathbf{A}^{-1}+\cdots+\mathbf{C}^{-1}(\mathbf{A}^{-1}\mathbf{B})^{R_I}\mathbf{A}^{-1})(\mathbf{A}^{-1}\mathbf{B}) \\ & +((\mathbf{C}^{-1}\mathbf{B})^{R_I-1}\mathbf{C}^{-1}\mathbf{A}^{-1}+\cdots+\mathbf{C}^{-1}(\mathbf{A}^{-1}\mathbf{B})^{R_I-1}\mathbf{A}^{-1})(\mathbf{A}^{-1}\mathbf{B})^2 \\ & \vdots \\ & +(\mathbf{C}^{-1}\mathbf{A}^{-1})(\mathbf{A}^{-1}\mathbf{B})^{R_I+1}. \end{aligned}$$

Letting

$$\tilde{\mathbf{P}}(2,0,0) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(k_I+2)\times 1} ,$$

we solve for probabilities $\tilde{\mathbf{P}}(B,j,Q_I-1)$; from which we can obtain the rest of the probabilities as shown below:

$$\begin{aligned}
\tilde{\mathbf{P}}(1,0,0) &= (1/\beta_S)\mathbf{B}(\mathbf{A}^{-1}\mathbf{B})^{R_I+1}(\mathbf{C}^{-1}\mathbf{B})^{Q_I-R_I-2}\tilde{\mathbf{P}}(B,j,Q_I-1),\\
\tilde{\mathbf{P}}(1,j,n) &= (\mathbf{A}^{-1}\mathbf{B})^{R_I+1-n}(\mathbf{C}^{-1}\mathbf{B})^{Q_I-R_I-2}\tilde{\mathbf{P}}(B,j,Q_I-1), \quad n=0,1,2,\ldots,R_I\\
\tilde{\mathbf{P}}(2,j,0) &= \beta_S((\mathbf{A}^{-1}\mathbf{B})^{R_I}\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,R_I)+(\mathbf{A}^{-1}\mathbf{B})^{R_I-1}\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,R_I-1)+\\
&\quad \ldots+(\mathbf{A}^{-1}\mathbf{B})\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,1)+\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,0)),\\
\tilde{\mathbf{P}}(2,j,n) &= \beta_S((\mathbf{A}^{-1}\mathbf{B})^{R_I-n}\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,R_I)+(\mathbf{A}^{-1}\mathbf{B})^{R_I-n-1}\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,R_I-1)+\\
&\quad \ldots+(\mathbf{A}^{-1}\mathbf{B})\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,n+1)+\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,n)), \quad n=1,2,\ldots,R_I-1\\
\tilde{\mathbf{P}}(2,j,R_I) &= \beta_S\mathbf{A}^{-1}\tilde{\mathbf{P}}(1,j,R_I),\\
\tilde{\mathbf{P}}(B,j,n) &= (\mathbf{C}^{-1}\mathbf{B})^{Q_I-n-1}\tilde{\mathbf{P}}(B,j,Q_I-1), \quad n=R_I+1,R_I+2,\ldots,Q_I-2\\
\tilde{\mathbf{P}}(B,j,Q_I) &= \beta_S((\mathbf{C}^{-1}\mathbf{B})^{R_I}\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,R_I)+(\mathbf{C}^{-1}\mathbf{B})^{R_I-1}\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,R_I-1)+\\
&\quad \cdots+(\mathbf{C}^{-1}\mathbf{B})\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,1)+\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,0)),\\
\tilde{\mathbf{P}}(B,j,Q_I+n) &= \beta_S((\mathbf{C}^{-1}\mathbf{B})^{R_I-n}\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,R_I)+(\mathbf{C}^{-1}\mathbf{B})^{R_I-n-1}\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,R_I-1)+\\
&\quad \cdots+(\mathbf{C}^{-1}\mathbf{B})\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,n+1)+\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,n)), \quad n=1,2,\ldots,R_I-1\\
\tilde{\mathbf{P}}(B,j,Q_I+R_I) &= \beta_S\mathbf{C}^{-1}\tilde{\mathbf{P}}(2,j,R_I)).
\end{aligned}$$

Finally, all the probabilities are normalized.

Analysis of Subsystem Involving Plant Output Buffer

Our second subsystem is $\Omega(O)$, the subsystem involving the plant output buffer. Here, again $\{I_t, J_t, N_t, t \geq 0\}$ is a Markov chain where I_t represents the phase of $U_O^{'}$, J_t represents the phase of $U_O^{''}$, N_t denotes the number of inventories in the input buffer where $I_t = 1, 2, B$, $J_t = 1, 2, \ldots, k_O$, and $N_t = R, R-1, R-2, \ldots$. We use a three-moment *MGE-2* approximation for the 4-phase procurement time (the parameters are γ_1, γ_2, and b). This approximation has practically no effect on the accuracy of the results [7]. The state-space and the transitions of the Markov chain are presented in Figure 3.4. Let us denote the steady-state probabilities

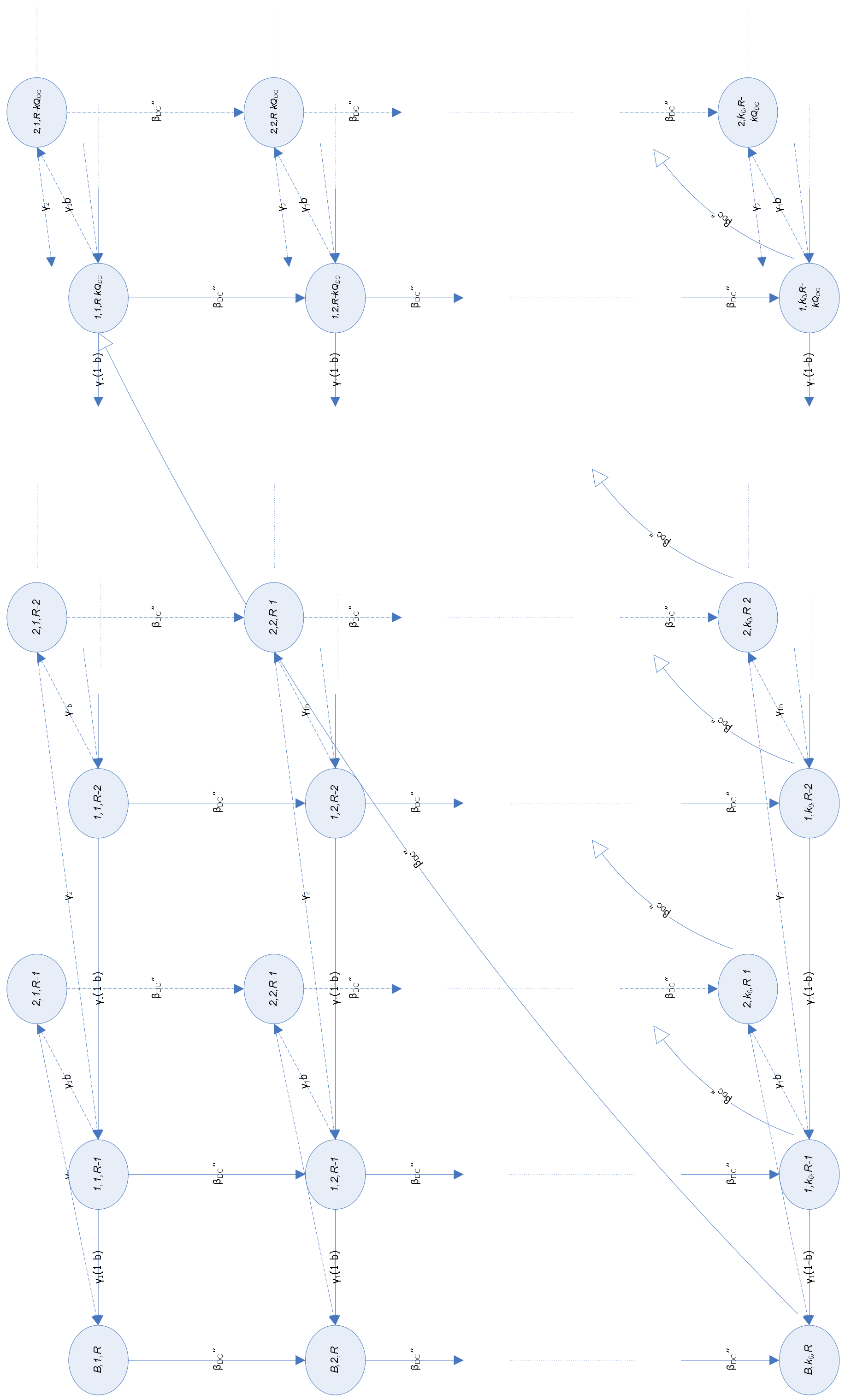


Figure 3.4: Transition diagram for subsystem $\Omega(O)$ (backordering case)

of the Markov chain by:

$$\tilde{\mathbf{P}}_w(n)|_{n=1}^{k_O} = \begin{bmatrix} P(1,n,-kQ_{DC}) \\ P(2,n,-kQ_{DC}) \\ \vdots \\ P(1,n,0) \\ P(2,n,0) \\ \vdots \\ P(1,n,R-1) \\ P(2,n,R-1) \end{bmatrix}_{(2R+2kQ_{DC})\times 1}, \quad \tilde{\mathbf{P}}_B(i) = \begin{bmatrix} P(B,1,i) \\ P(B,2,i) \\ \vdots \\ P(B,k_O,i) \end{bmatrix}_{k_O\times 1}$$

where $i = R, R-Q_{DC}, \ldots, R-(k_{min})Q_{DC}$, k is a large enough number that ensures the remaining probabilities are zero, and k_{min} is defined as $k_{min} = \min\{i : R - iQ_{DC} \leq r,\ i \text{ integer},\ i \geq 0\}$. Then, the flow-balance equations are given below:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}_w(1) &= \mathbf{B}\tilde{\mathbf{P}}_w(k_O) + \mathbf{C}\tilde{\mathbf{P}}_B(R), \qquad (3.2) \\ \mathbf{A}\tilde{\mathbf{P}}_w(n+1) &= \beta_{DC}^{''}\tilde{\mathbf{P}}_w(n), \quad n = 1,2,\ldots,k_O-1, \\ \mathbf{D}\tilde{\mathbf{P}}_B(R) &= \mathbf{E}_1\tilde{P}_w(1) + \mathbf{E}_2\tilde{P}_w(2) + \ldots + \mathbf{E}_{k_O}\tilde{\mathbf{P}}_w(k_O), \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \gamma_1+\beta_{DC}^{''} & 0 & 0 & 0 & 0 & 0 & \ldots \\ -b\gamma_1 & \gamma_2+\beta_{DC}^{''} & 0 & 0 & 0 & 0 & \ldots \\ -(1-b)\gamma_1 & -\gamma_2 & \gamma_1+\beta_{DC}^{''} & 0 & 0 & 0 & \ldots \\ 0 & 0 & -b\gamma_1 & \gamma_2+\beta_{DC}^{''} & 0 & 0 & \ldots \\ 0 & 0 & -(1-b)\gamma_1 & -\gamma_2 & \gamma_1+\beta_{DC}^{''} & 0 & \ldots \\ 0 & 0 & 0 & 0 & -b\gamma_1 & \gamma_2+\beta_{DC}^{''} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \beta_{DC}^{''} & 0 & \ldots & \beta_{DC}^{''} & 0 & \beta_{DC}^{''} & 0 & 0 & \ldots \\ 0 & \beta_{DC}^{''} & \ldots & 0 & \beta_{DC}^{''} & 0 & \beta_{DC}^{''} & 0 & \ldots \\ 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \beta_{DC}^{''} & \ldots \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

are $(2R+2kQ_{DC}) \times (2R+2kQ_{DC})$ matrices,

$$\mathbf{C} = [c_{ij}] = \begin{cases} \beta_{DC}^{''}, & \text{if } i = 2kQ_{DC} + 2R - 2k_{min}Q_{DC} + 1, j = k_O \\ 0, & \text{otherwise,} \end{cases}$$

is a $(2R + 2kQ_{DC}) \times k_O$ matrix,

$$\mathbf{D} = \begin{bmatrix} \beta''_{DC} & 0 & \dots & 0 & 0 \\ -\beta''_{DC} & \beta''_{DC} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -\beta''_{DC} & \beta''_{DC} \end{bmatrix}_{k_O \times k_O},$$

and

$$\mathbf{E}_t = [e_{ij}] = \begin{cases} (1-b)\gamma_1, & \text{if } i = t, j = 2kQ_{DC} + 2R - 1 \\ \gamma_2, & \text{if } i = t, j = 2kQ_{DC} + 2R \\ 0, & \text{otherwise,} \end{cases}$$

are $k_O \times (2R + 2kQ_{DC})$ matrices for $t = 1, 2, \dots k_O$. By utilizing Equation 3.2, we get $\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = 0$ and $\mathbf{P}$ is given

$$\mathbf{P} = \mathbf{A} - \mathbf{B}(\beta''_{DC}\mathbf{A}^{-1})^{k_O-1} - \mathbf{C}\mathbf{D}^{-1}\mathbf{E},$$

and

$$\mathbf{E} = \mathbf{E}_1 + \mathbf{E}_2(\beta''_{DC}\mathbf{A}^{-1})^1 + \dots + \mathbf{E}_{k_O}(\beta''_{DC}\mathbf{A}^{-1})^{k_O-1}.$$

In addition, we have the normalization equation:

$$\begin{aligned} \mathbf{p} \;&=\; \mathbf{e}_{(1\times(2R+2kQ_{DC}))} + \mathbf{e}_{(1\times(2R+2kQ_{DC}))}(\beta''_{DC}\mathbf{A}^{-1})^1 + \dots + \mathbf{e}_{(1\times(2R+2kQ_{DC}))}(\beta''_{DC}A^{-1})^{k_0-1} \\ &\quad + \mathbf{e}_{(1\times k_O)}\mathbf{D}^{-1}\mathbf{E} + (k_{min} - 1)k_O(0, \dots, 0, 1)_{(1\times k_O)}\mathbf{D}^{-1}\mathbf{E}. \end{aligned}$$

After replacing the first row of matrix $\mathbf{P}$ with row vector $\mathbf{p}$, we solve for $\tilde{\mathbf{P}}_w(1)$ from the equation $\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = [1, 0, \dots, 0]^T_{2R+2kQ_{DC}}$. The rest of the probabilities are obtained using

$$\begin{aligned} \tilde{\mathbf{P}}_w(n) &= (\beta''_{DC}\mathbf{A}^{-1})^{n-1}\tilde{\mathbf{P}}_w(1), \quad n = 2, \dots, k_O, \\ \tilde{\mathbf{P}}_B(R) &= \mathbf{D}^{-1}\mathbf{E}\tilde{\mathbf{P}}_w(1). \end{aligned}$$

Note that, N_t may take values in $(-\infty, R]$. We use truncation at a reasonable backorder level to deal with finite number of states.

Analysis of Subsystem Involving Distribution Center

On the other hand, solving the probabilities of the subsystem $\Omega(DC)$, the subsystem involving the distribution center, is more involved. First, the effective procurement time has a complex phase structure. However, a three-moment *MGE-2* approximation is utilized based on [7] (the parameters of the *MGE-2* are γ_1, γ_2 and b). This approximation simplifies the solution procedure. Second, the procurement orders and demand arrivals to this system are both in batches. Still, we are able to utilize the matrix recursive scheme used in the previous subsystems and solve for the probabilities. Again, $\{I_t, J_t, N_t, t \geq 0\}$ is a Markov chain where I_t represents the phase of $U_{DC}^{'}$, J_t represents the phase of $U_{DC}^{''}$, N_t denotes the number of inventories in the DC where $I_t = 1, 2, B$, $J_t = 1, 2, \ldots, k_{DC}$, and $N_t = Q_{DC} + R_{DC}, Q_{DC} + R_{DC} - 1, \ldots$. The Markov chain has infinite number of states, and yet we again truncate the state-space at a state with negligible holding probability. The state-space and the transitions of the Markov chain are presented in Figure 3.5. Let the probabilities of the subsystem be:

$$\tilde{\mathbf{P}}(n)|_{n=1}^{k_{DC}} = \begin{bmatrix} P(1,n,R_{DC}-kQ_R) \\ P(2,n,R_{DC}-kQ_R) \\ \vdots \\ P(1,n,R_{DC}) \\ P(2,n,R_{DC}) \\ P(B,n,R_{DC}+1) \\ \vdots \\ P(B,n,R_{DC}+Q_{DC}) \end{bmatrix}_{(Q_{DC}+2kQ_R+2)\times 1}.$$

The flow-balance equations of the system in compact form are:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}(1) &= \mathbf{B}\tilde{\mathbf{P}}(k_{DC}), \\ \mathbf{A}\tilde{\mathbf{P}}(n+1) &= \beta_R^{''}\tilde{\mathbf{P}}(n), \quad n = 1, 2, \ldots, k_{DC}-1 \end{aligned} \tag{3.3}$$

The matrices **A** and **B** are given as:

$$\mathbf{A} = \begin{bmatrix} \gamma_1+\beta_R^{''} & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ -b\gamma_1 & \gamma_2+\beta_R^{''} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \gamma_1+\beta_R^{''} & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & -b\gamma_1 & \gamma_2+\beta_R^{''} & 0 & 0 & \cdots \\ -(1-b)\gamma_1 & \gamma_2 & 0 & 0 & 0 & \gamma_1+\beta_R^{''} & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & -b\gamma_1 & \gamma_2+\beta_R^{''} & \cdots \\ 0 & 0 & -(1-b)\gamma_1 & \gamma_2 & 0 & 0 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots \end{bmatrix},$$

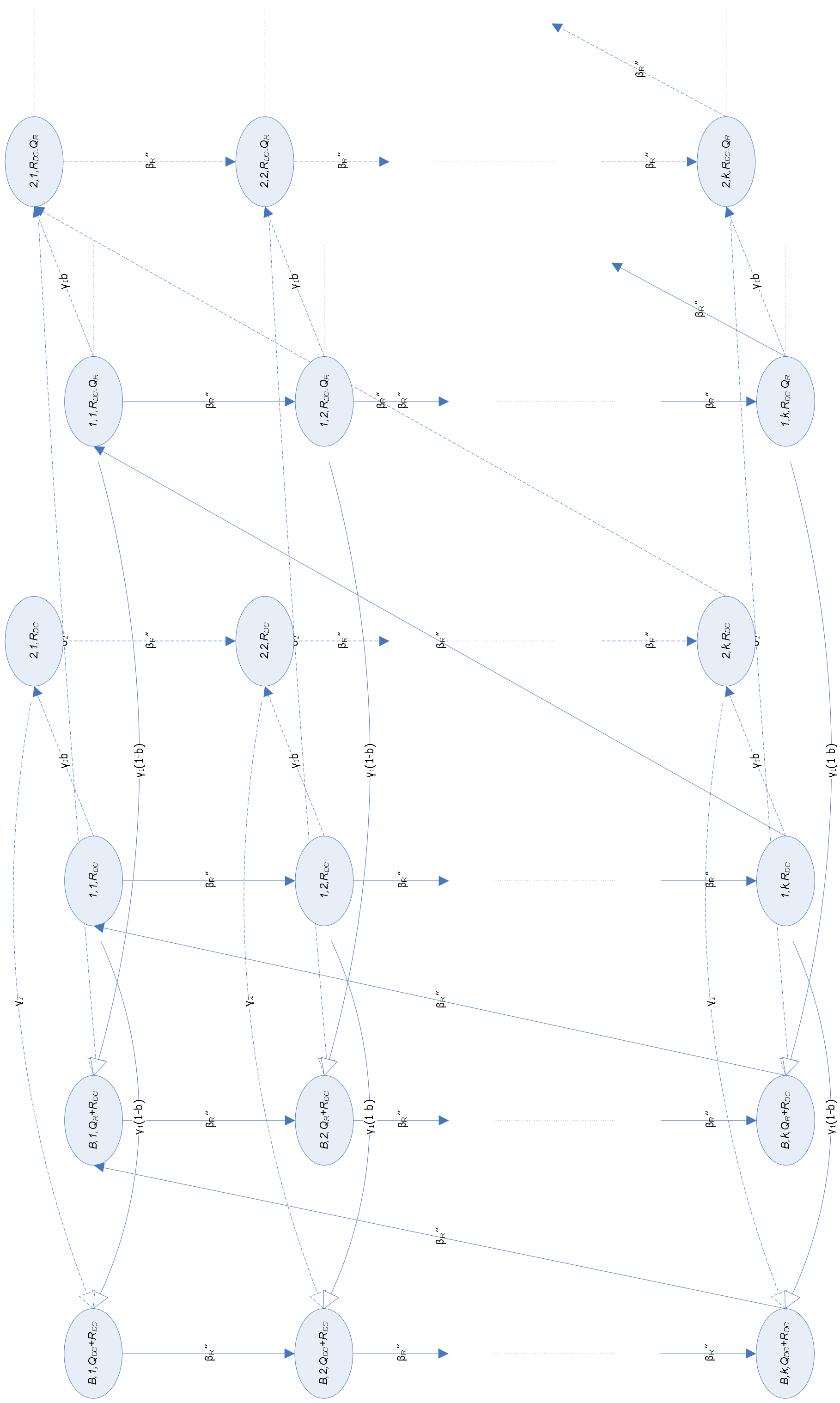


Figure 3.5: Transition diagram for subsystem $\Omega(DC)$ (backordering case)

$$\mathbf{B} = \begin{bmatrix} \beta_R^{''} & 0 & \beta_R^{''} & 0 & 0 & \dots \\ 0 & \beta_R^{''} & 0 & \beta_R^{''} & 0 & \dots \\ 0 & 0 & 0 & 0 & \beta_R^{''} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Utilizing the Equation 3.3, we get $\mathbf{P} \times \tilde{\mathbf{P}}(1) = 0$ where

$$\mathbf{P} = \mathbf{A} - \mathbf{B}(\beta_R^{''}\mathbf{A}^{-1})^{k_{DC}-1}.$$

Finally, normalization is achieved by:

$$\mathbf{p} = \mathbf{e}_{(1\times(2R_{DC}+Q_{DC}+2+2kQ_R))} \times (\mathbf{I} + (\beta_R^{''}\mathbf{A}^{-1}) + \ldots + (\beta_R^{''}\mathbf{A}^{-1})^{k_{DC}-1}).$$

Replacing the first row of matrix $\mathbf{P}$ with row vector $\mathbf{p}$, we solve for

$$\mathbf{P} \times \tilde{\mathbf{P}}(1) = [1, 0, \ldots, 0]^T_{(2R_{DC}+Q_{DC}+2+2kQ_R)}.$$

Rest of the probabilities are given by:

$$\tilde{\mathbf{P}}(n) = (\beta_R^{''}\mathbf{A}^{-1})^{n-1}\tilde{\mathbf{P}}(1), \quad n = 2, \ldots, k_{DC}.$$

Analysis of Subsystem Involving Retailer

Finally, the subsystem $\Omega(R)$ models the behavior of the retailer where demand arrives singly and according to a Poisson process, and the replenishment process takes place in batches. A queuing analogy of the above model is the system $M/PH^k/1$ where arrivals are from a Poisson process, and the service time distribution is of phase-type and in exact batches of k. Although general solution procedures for the above queuing system are given in [31], we will again use the matrix-recursive algorithms utilized in the previous subsystems. Typical approaches use the generating function of the steady-state distribution. Inverting this function to compute the probabilities may be problematic and may require more computational effort than our approach.

Let $\{I_t, N_t, t \geq 0\}$ be a Markov chain where I_t represents the phase of $U_R^{'}$, and N_t denotes the level of inventories at the retailer where $I_t = 1, 2, B$, and $N_t = Q_R + R_R, Q_R + R_R - 1, \ldots$. As in the subsystem $\Omega(DC)$, the effective procurement time has a complex phase structure. We again use a three-moment *MGE-2* approximation (the parameters are γ_1, γ_2 and b). The Markov chain has infinite number of states, however we truncate the state-space. The

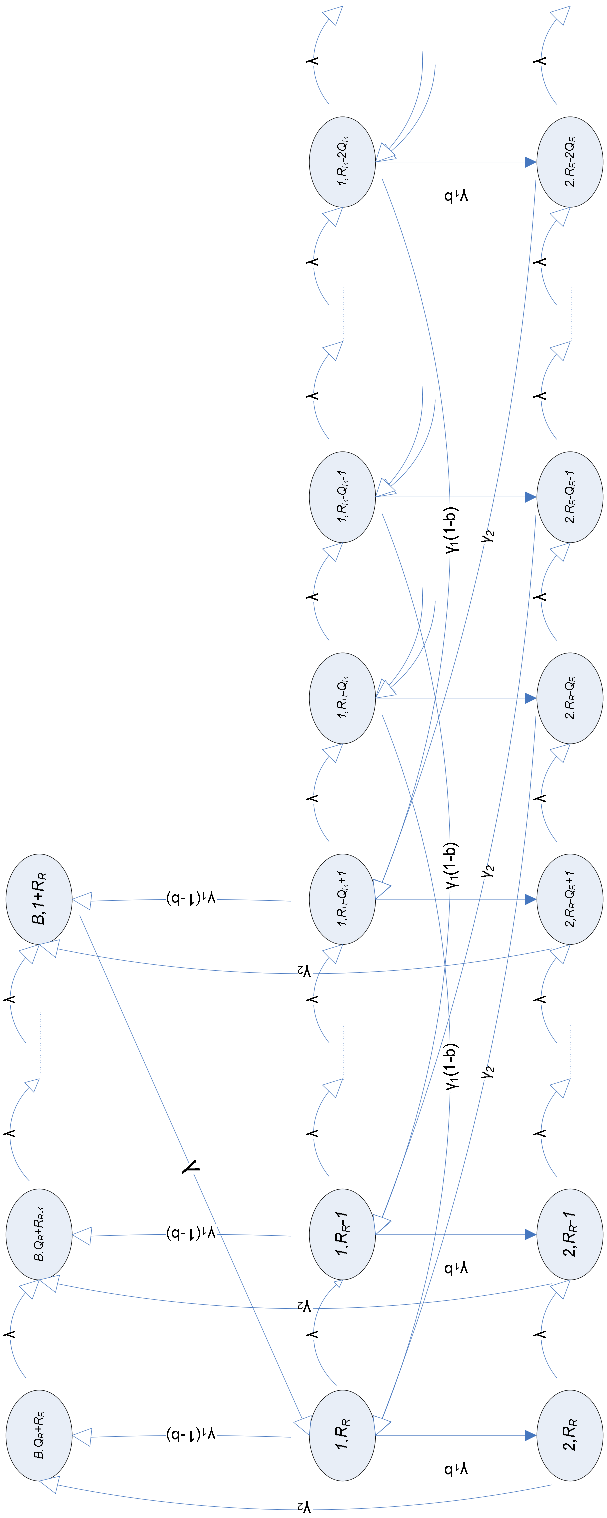


Figure 3.6: Transition diagram for subsystem $\Omega(R)$ (backordering case)

state-space and the transitions of the Markov chain are presented in Figure 3.6. Let the probabilities of the subsystem be:

$$\tilde{\mathbf{P}}_w(n)|_{n=1}^{2} = \begin{bmatrix} P(n,R_R) \\ P(n,R_R-1) \\ \vdots \\ P(n,R_R-kQ_R) \end{bmatrix}_{(kQ_R+1)\times 1}, \quad \tilde{\mathbf{P}}(B) = \begin{bmatrix} P(B,R_R+Q_R) \\ \vdots \\ P(B,R_R+1) \end{bmatrix}_{Q_R\times 1}.$$

Corresponding flow-balance equations are:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}(B) &= \mathbf{B}\tilde{\mathbf{P}}_w(1) + \mathbf{C}\tilde{\mathbf{P}}_w(2), \\ \mathbf{D}\tilde{\mathbf{P}}_w(1) &= \mathbf{E}\tilde{\mathbf{P}}_w(2) + \mathbf{F}\tilde{\mathbf{P}}(B), \\ \mathbf{G}\tilde{\mathbf{P}}_w(2) &= \gamma_1 b\tilde{\mathbf{P}}_w(1), \end{aligned} \tag{3.4}$$

where

$$\mathbf{A} = \begin{bmatrix} \lambda & & & \\ -\lambda & \lambda & & \\ & & \ddots & \\ & & -\lambda & \lambda \end{bmatrix}_{Q_R\times Q_R}, \mathbf{B} = \begin{bmatrix} \gamma_1(1-b) & & & \\ & \gamma_1(1-b) & & \\ & & \ddots & \dots \\ & & \gamma_1(1-b) & \end{bmatrix}_{Q_R\times(kQ_R+1)},$$

$$\mathbf{C} = \begin{bmatrix} \gamma_2 & & & \\ & \gamma_2 & & \\ & & \ddots & \dots \\ & & \gamma_2 & \end{bmatrix}_{Q_R\times(kQ_R+1)}, \mathbf{D} = \begin{bmatrix} \lambda+\gamma_1 & \dots & -\gamma_1(1-b) & & & \\ -\lambda & \lambda+\gamma_1 & & \ddots & & \\ & & \ddots & & & \\ & & & -\lambda & \lambda+\gamma_1 & \\ & & & & -\lambda & \gamma_1 \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} & -\gamma_1(1-b) & & \\ & & & \ddots \\ & & & \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \lambda+\gamma_2 & & & \\ -\lambda & \lambda+\gamma_2 & & \\ & & \ddots & \\ & & -\lambda & \lambda+\gamma_2 \\ & & & -\lambda & \gamma_2 \end{bmatrix},$$

and $\mathbf{D}$, $\mathbf{E}$, $\mathbf{G}$ are $(kQ_R+1)\times(kQ_R+1)$ matrices. Additionally,

$$\mathbf{F} = [f_{ij}] = \begin{cases} \lambda, & \text{if } i=1, j=Q_R \\ 0, & \text{otherwise,} \end{cases}$$

is a $(kQ_R+1)\times(Q_R)$ matrix.

After, representing $\tilde{\mathbf{P}}(B)$ and $\tilde{\mathbf{P}}_w(2)$ in terms of $\tilde{\mathbf{P}}_w(1)$, and by utilizing Equation 3.4, we obtain $\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = 0$ where

$$\mathbf{P} = \mathbf{D} - \gamma_1 b\mathbf{E}\mathbf{G}^{-1} - \mathbf{F}\mathbf{A}^{-1}\mathbf{B} - \gamma_1 b\mathbf{F}\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1}.$$

The normalization condition is given as:

$$\mathbf{p} = \mathbf{e}_{(1\times(kQ_R+1))} \times (\mathbf{I} + \gamma_1 b\mathbf{G}^{-1}) + \mathbf{e}_{(1\times(Q_R))} \times (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1}).$$

Replacing the first row of matrix $\mathbf{P}$ by the row vector $\mathbf{p}$, we solve for

$$\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = [1, 0, \ldots, 0]^T_{(kQ_R+1)\times 1}.$$

The rest of the probabilities are given by:

$$\begin{aligned} \tilde{\mathbf{P}}_w(2) &= \gamma_1 b\mathbf{G}^{-1}\tilde{\mathbf{P}}_w(1), \\ \tilde{\mathbf{P}}(B) &= (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1})\tilde{\mathbf{P}}_w(1). \end{aligned}$$

Thus, we have analyzed each of the subsystems with its own stock keeping policies and with phase-type procurement and demand inter-arrival times. We present matrix-recursive procedures in order to compute the steady-state probabilities of the subsystems. We assume the parameters of the procurement and demand inter-arrival times are unknown. Consequently, we are still in need of a way to relate the parameters of the subsystems to each other. As an example, consider the conditional probability Δ_I. It is a key ingredient to determine the effective procurement time of the subsystem $\Omega(O)$ and is required before the solution procedure started. To circumvent the situation an iterative algorithm that links the subsystems to each other is presented in the next section.

3.1.4 An Aggregation Algorithm

The nature of the decomposition algorithm requires subsystems to supply information to each other. This is achieved by utilizing a fixed-point iteration algorithm. The unknown parameters of the subsystems are Π_O, Δ_I, $\omega_O(i),\ i = 0, 1, 2, \ldots$, and $\omega_{DC}(i),\ i = 0, 1, 2, \ldots$. As part of the algorithm, Π_O is used in the analysis of $\Omega(I)$ and updated in the analysis of $\Omega(O)$. Similarly, Δ_I is used in the analysis of $\Omega(O)$ and updated in the analysis of $\Omega(I)$. $\omega_O(i)$'s, on the other hand, are obtained from the analysis of $\Omega(O)$ and used only in the analysis of $\Omega(DC)$. Similarly, $\omega_{DC}(i)$'s are obtained from the analysis of $\Omega(DC)$ and used only in the analysis of $\Omega(R)$. Yet we have to assign values to these unknown probabilities. Using the formula presented by Altiok and Ranjan [9], and Gurgur and Altiok [61] based on the Little's Law, the conditional probabilities Δ_I and Π_O are computed using the steady-state probabilities of the subsystems $\Omega(I)$ and $\Omega(O)$:

$$\Delta_I = \frac{\tilde{P}_I(0)}{\bar{\xi}_I E[U_I']},$$

where $\tilde{P}_I(0)$ is the arbitrary-time probability that there are no units in the input buffer, $\bar{\xi}_I$ is the throughput of the first subsystem, and $E[U_I']$ is the expected value of the starvation period. Similarly,

$$\Pi_O = \frac{\tilde{P}_O(B)}{\bar{\xi}_O E[U_O'']}$$

where $\tilde{P}_O(B)$ is the probability that the output buffer is blocked, $\bar{\xi}_O$ is the throughput of the second subsystem, and $E[U_O'']$ is the expected value of the blocking period. In addition, $\omega_O(i)$'s are evaluated

$$\begin{aligned} \omega_O(0) &= Pr(N_O \geq Q_{DC} \backslash N_{DC} = R_{DC}), \\ \omega_O(i) &= Pr(N_O = Q_{DC} - i \backslash N_{DC} = R_{DC}),\ i = 1, 2, \ldots . \end{aligned}$$

where N_O and N_{DC} represent the inventory level in the output buffer and the DC, respectively. Finally, the conditional probabilities $\omega_{DC}(i)$'s are obtained by

$$\begin{aligned} \omega_{DC}(0) &= Pr(N_{DC} \geq Q_R \backslash N_R = R_R), \\ \omega_{DC}(i) &= \sum_{j=(i-1)Q_{DC}+1}^{iQ_{DC}} Pr(N_{DC} = Q_R - j \backslash N_R = R_R),\ i = 1, 2, \ldots . \end{aligned}$$

The ω_O's and ω_{DC}'s are arrival-point probabilities. The throughput of the subsystems are obtained using

$$\bar{\xi}_j = \frac{\text{Utilization of } M_j''}{E[U_j'']},\ j = I, O, DC, R.$$

Due to backordering practice in the system, the throughput of the system is known to be λ. Thus, the algorithm stops when all the subsystems' throughputs converge to the actual throughput, λ. As a result, the algorithm starts by assuming some initial values for the unknown parameters. It iterates back and forth between the subsystems $\Omega(I)$ and $\Omega(O)$. After all the throughputs are sufficiently close to λ, it analyzes subsystems $\Omega(DC)$ and $\Omega(R)$ and stops. A summary of the algorithm is given in Table 3.1.

1. Initialize: $\Pi_O = \Delta_I = \omega_O(i) = \omega_{DC}(i) = 0,$ for all $i = 0, 1, 2, \ldots, \epsilon = 10^{-4}$.
2. Analyze $\Omega(I)$, obtain its steady-state probabilities, update Δ_I and $\bar{\xi}_I$.
3. Analyze $\Omega(O)$, obtain its steady-state probabilities, update Π_O and $\bar{\xi}_O$.
4. If $\max\{|\bar{\xi}_I - \lambda|, |\bar{\xi}_O - \lambda|\} \leq \epsilon$, obtain $\omega_O(i)$ and go to step 5; else go to step 2.
5. Analyze $\Omega(DC)$, obtain its steady-state probabilities, obtain $\omega_{DC}(i)$ and $\bar{\xi}_{DC}$.
6. Analyze $\Omega(R)$, obtain its steady-state probabilities and $\bar{\xi}_R$.

Table 3.1: The aggregation algorithm for multi-echelon supply chains with backordering

3.2 Computational Accuracy

We test the accuracy of our disaggregation/aggregation approximation by comparing its results against simulation in a number of examples. The purpose of numerical examples is to see the ranges of the system parameters where the approximation is accurate and where it is not. The approximation procedure described above and the discrete-event simulation model runs are implemented on a Pentium IV PC operating at 2.80 GHz. The simulation model is developed using the Arena[1] simulation software. Each simulation run consists of 50,000,000 job departures to provide point estimates and 95% confidence intervals for key performance measures. The convergence criterion is chosen to be $\epsilon = 10^{-4}$. In most of the cases, the convergence is achieved in three iterations. The approximation and the simulation results are given in Tables 3.2, 3.3, and 3.4 for different traffic intensities.

In this study, we focus on average inventory levels, average backorder levels, and customer service levels. Here, we define the customer service level as the probability of fully satisfying the demand of an arriving customer.

We have three plant-related scenarios: low production rate (Table 3.2), medium production rate (Table 3.3),and high production rate (Table 3.4). DC buffer capacities are chosen proportional to the retailer buffer capacities. In each experiment, demand rate is varied while keeping other parameters constant. The relative error of the performance estimates varies from -12.32% to 0.20% for the average inventory levels, and -9.49% to 0.24% for the customer service levels. It is clear from the results that the percentage deviation gradually increases as the demand rate (system load) increases. On the other hand, the accuracy in

[1] Arena is a trademark of Rockwell Software.

		Plant		DC	Retailer
	$R_I = 10$	$R = 30$	$\mu_1 = 1$	$R_{DC} = 10$	$R_R = 5$
Parameters:	$Q_I = 13$	$r = 10$	$\mu_2 = 1$	$Q_{DC} = 20$	$Q_R = 10$
	$\beta_S = 1$	$\beta_P = 1$	$a = 0.1$	$\beta_{DC} = 1$	

		λ=0.6			λ=0.65		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Input Buffer	**Analytic**	15.7997	N/A	99.99%	15.6996	N/A	99.99%
	Simulation	15.7994	N/A	99.99%	15.6997	N/A	99.99%
	Rel. Error	0.00%	N/A	0.00%	0.00%	N/A	0.00%
Output Buffer	**Analytic**	22.7858	0.0010	99.64%	21.9288	0.0043	98.97%
	Simulation	22.7823	0.0010	99.64%	21.9227	0.0044	98.96%
	Rel. Error	0.02%	0.10%	0.00%	0.03%	-2.87%	0.01%
DC	**Analytic**	23.7933	0.0000	100.00%	23.6748	0.0000	100.00%
	Simulation	23.7923	0.0000	100.00%	23.6740	0.0000	100.00%
	Rel. Error	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Retailer	**Analytic**	9.3014	0.0016	99.77%	9.2019	0.0024	99.69%
	Simulation	9.3023	0.0016	99.77%	9.2027	0.0024	99.69%
	Rel. Error	-0.01%	0.00%	0.00%	-0.01%	0.00%	0.00%

		λ=0.7			λ=0.75		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Input Buffer	**Analytic**	15.5996	N/A	99.99%	15.4995	N/A	99.99%
	Simulation	15.5988	N/A	99.99%	15.4998	N/A	99.99%
	Rel. Error	0.01%	N/A	0.00%	0.00%	N/A	0.00%
Output Buffer	**Analytic**	20.8371	0.0196	97.22%	19.2997	0.0918	92.90%
	Simulation	20.8356	0.0192	97.23%	19.2980	0.0910	92.91%
	Rel. Error	0.01%	1.99%	-0.01%	0.01%	0.88%	-0.01%
DC	**Analytic**	23.5084	0.0000	100.00%	23.1657	0.0082	99.92%
	Simulation	23.5087	0.0003	99.99%	23.1769	0.0055	99.88%
	Rel. Error	0.00%	-100.00%	0.01%	-0.05%	48.82%	0.04%
Retailer	**Analytic**	9.1027	0.0035	99.59%	9.0019	0.0048	99.47%
	Simulation	9.1028	0.0034	99.59%	9.0004	0.0059	99.45%
	Rel. Error	0.00%	1.72%	0.00%	0.02%	-18.51%	0.02%

		λ=0.8			λ=0.85		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Input Buffer	**Analytic**	15.3995	N/A	99.99%	15.2995	N/A	99.98%
	Simulation	15.3996	N/A	99.99%	15.2996	N/A	99.99%
	Rel. Error	0.00%	N/A	0.00%	0.00%	N/A	-0.01%
Output Buffer	**Analytic**	16.8501	0.4775	82.80%	12.3514	3.1976	60.38%
	Simulation	16.8410	0.4806	82.78%	12.3409	3.2264	60.38%
	Rel. Error	0.05%	-0.65%	0.02%	0.09%	-0.89%	0.00%
DC	**Analytic**	22.1015	0.0917	98.86%	16.9652	2.4833	85.58%
	Simulation	22.2227	0.0781	98.94%	18.8766	1.2556	91.18%
	Rel. Error	-0.55%	17.35%	-0.08%	-10.13%	97.78%	-6.14%
Retailer	**Analytic**	8.8698	0.0086	99.19%	7.2189	1.7709	85.11%
	Simulation	8.8524	0.0357	98.95%	8.2332	0.7466	94.03%
	Rel. Error	0.20%	-75.88%	0.24%	-12.32%	137.20%	-9.49%

Table 3.2: Accuracy of the approximation algorithm for a low production rate with backordering

		Plant		DC	Retailer
	R_I = 10	R = 30	μ_1 = 2	R_{DC} = 10	R_R = 5
Parameters:	Q_I = 13	r = 10	μ_2 = 1	Q_{DC} = 20	Q_R = 10
	β_S = 1	β_P = 1	a = 0.1	β_{DC} = 1	

		λ=1.0			**λ=1.1**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.9903	N/A	99.86%	14.7890	N/A	99.85%
Input Buffer	**Simulation**	14.9924	N/A	99.88%	14.7922	N/A	99.87%
	Rel. Error	-0.01%	N/A	-0.02%	-0.02%	N/A	-0.02%
	Analytic	23.5033	0.0006	99.77%	22.6686	0.0025	99.33%
Output Buffer	**Simulation**	23.5246	0.0005	99.79%	22.6955	0.0023	99.36%
	Rel. Error	-0.09%	20.00%	-0.02%	-0.12%	8.70%	-0.03%
	Analytic	22.9963	0.0000	100.00%	22.7856	0.0000	100.00%
DC	**Simulation**	22.9959	0.0000	100.00%	22.7839	0.0000	100.00%
	Rel. Error	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%
	Analytic	8.5104	0.0174	98.58%	8.3143	0.0263	98.07%
Retailer	**Simulation**	8.5109	0.0175	98.57%	8.3146	0.0262	98.06%
	Rel. Error	-0.01%	-0.57%	0.01%	0.00%	0.38%	0.01%

		λ=1.2			**λ=1.3**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.5876	N/A	99.85%	14.3861	N/A	99.84%
Input Buffer	**Simulation**	14.5916	N/A	99.86%	14.3896	N/A	99.84%
	Rel. Error	-0.03%	N/A	-0.01%	-0.02%	N/A	0.00%
	Analytic	21.6497	0.0112	98.15%	20.2854	0.0512	95.16%
Output Buffer	**Simulation**	21.6821	0.0102	98.24%	20.3260	0.0495	95.30%
	Rel. Error	-0.15%	9.80%	-0.09%	-0.20%	3.43%	-0.15%
	Analytic	22.5465	0.0001	100.00%	22.2047	0.0004	99.98%
DC	**Simulation**	22.5466	0.0002	99.99%	22.2087	0.0035	99.92%
	Rel. Error	0.00%	-50.00%	0.01%	-0.02%	-88.57%	0.06%
	Analytic	8.1185	0.0379	97.47%	7.9228	0.0528	96.78%
Retailer	**Simulation**	8.1192	0.0378	97.47%	7.9205	0.0537	96.75%
	Rel. Error	-0.01%	0.26%	0.00%	0.03%	-1.68%	0.03%

		λ=1.4			**λ=1.5**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.1845	N/A	99.84%	13.9829	N/A	99.83%
Input Buffer	**Simulation**	14.1882	N/A	99.83%	13.9857	N/A	99.80%
	Rel. Error	-0.03%	N/A	0.01%	-0.02%	N/A	0.03%
	Analytic	18.2354	0.2517	87.99%	14.7063	1.4689	71.70%
Output Buffer	**Simulation**	18.3063	0.2438	88.30%	14.7971	1.3927	72.17%
	Rel. Error	-0.39%	3.24%	-0.35%	-0.61%	5.47%	-0.65%
	Analytic	21.4653	0.0124	99.64%	18.7505	0.7023	94.12%
DC	**Simulation**	21.5210	0.0368	99.40%	19.5314	0.4381	95.71%
	Rel. Error	-0.26%	-66.30%	0.24%	-4.00%	60.31%	-1.66%
	Analytic	7.7162	0.0737	95.91%	7.0983	0.3662	89.71%
Retailer	**Simulation**	7.7027	0.0862	95.78%	7.3054	0.3470	92.87%
	Rel. Error	0.18%	-14.50%	0.14%	-2.83%	5.53%	-3.40%

Table 3.3: Accuracy of the approximation algorithm for a medium production rate with backordering

	Plant			DC	Retailer
	R_I = 10	R = 30	μ_1 = 3	R_{DC} = 10	R_R = 5
Parameters:	Q_I = 13	r = 10	μ_2 = 1	Q_{DC} = 20	Q_R = 10
	β_S = 1	β_P = 1	a = 0.1	β_{DC} = 1	

		λ=1.5			**λ=1.6**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	13.9536	N/A	99.60%	13.7493	N/A	99.59%
Input Buffer	**Simulation**	13.9634	N/A	99.59%	13.7603	N/A	99.57%
	Rel. Error	-0.07%	N/A	0.01%	-0.08%	N/A	0.02%
	Analytic	22.5378	0.0077	98.67%	21.7468	0.0203	97.48%
Output Buffer	**Simulation**	22.6589	0.0060	98.89%	21.8943	0.0162	97.82%
	Rel. Error	-0.53%	28.33%	-0.22%	-0.67%	25.31%	-0.35%
	Analytic	21.9653	0.0003	99.98%	21.7186	0.0007	99.96%
DC	**Simulation**	21.9677	0.0005	99.98%	21.7228	0.0014	99.50%
	Rel. Error	-0.01%	-40.00%	0.00%	-0.02%	-50.00%	0.46%
	Analytic	7.5330	0.0942	95.12%	7.3375	0.1220	94.15%
Retailer	**Simulation**	7.5329	0.0941	95.11%	7.3379	0.1222	94.13%
	Rel. Error	0.00%	0.11%	0.01%	-0.01%	-0.16%	0.02%

		λ=1.7			**λ=1.8**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	13.5448	N/A	99.58%	13.3401	N/A	99.57%
Input Buffer	**Simulation**	13.5553	N/A	99.54%	13.3516	N/A	99.50%
	Rel. Error	-0.08%	N/A	0.04%	-0.09%	N/A	0.07%
	Analytic	20.7776	0.0542	95.32%	19.5297	0.1485	91.52%
Output Buffer	**Simulation**	20.9550	0.0438	95.89%	19.7482	0.1219	92.38%
	Rel. Error	-0.85%	23.74%	-0.59%	-1.11%	21.82%	-0.93%
	Analytic	21.4102	0.0018	99.92%	20.9534	0.0071	99.74%
DC	**Simulation**	21.4294	0.0046	99.87%	21.0112	0.0180	99.63%
	Rel. Error	-0.09%	-60.87%	0.05%	-0.28%	-60.56%	0.11%
	Analytic	7.1409	0.1558	93.07%	6.9384	0.1981	91.84%
Retailer	**Simulation**	7.1417	0.1563	93.05%	6.9361	0.2016	91.81%
	Rel. Error	-0.01%	-0.32%	0.02%	0.03%	-1.74%	0.03%

		λ=1.9			**λ=2.0**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	13.1352	N/A	99.56%	12.9301	N/A	99.55%
Input Buffer	**Simulation**	13.1474	N/A	99.46%	12.9402	N/A	99.41%
	Rel. Error	-0.09%	N/A	0.10%	-0.08%	N/A	0.14%
	Analytic	17.8320	0.4265	84.97%	15.3820	1.3327	73.90%
Output Buffer	**Simulation**	18.1074	0.3495	86.29%	15.6731	1.1521	75.37%
	Rel. Error	-1.52%	22.03%	-1.53%	-1.86%	15.68%	-1.95%
	Analytic	20.1064	0.0526	98.81%	18.0850	0.6440	94.24%
DC	**Simulation**	20.3249	0.0770	98.83%	18.9284	0.3970	95.36%
	Rel. Error	-1.08%	-31.69%	-0.02%	-4.46%	62.22%	-1.17%
	Analytic	6.6954	0.2689	90.06%	6.0693	0.7905	82.38%
Retailer	**Simulation**	6.7066	0.2815	90.20%	6.3649	0.5566	87.19%
	Rel. Error	-0.17%	-4.48%	-0.16%	-4.64%	42.02%	-5.52%

Table 3.4: Accuracy of the approximation algorithm for a high production rate with backordering

the backorder levels is somehow surprising (ranging from -100% to 137.24%). This is because backorder levels are very low and approximating very small probabilities does not seem to be quite successful.

3.3 Analysis of Single-Product Supply Chains with Lost Sales

In this section, we consider the multi-echelon supply chain model with lost sales. Customer demand arrives to the retailer according to a Poisson process and any excess demand that is not immediately satisfied from the on-hand inventory is lost. Compared to inventory models with backordering, models with lost sales has received less attention from the researchers.

Performance evaluation of the multi-echelon supply chain with the lost sales assumption is quite difficult because of its complex nature and large state-space. Indeed, we next present a decomposition procedure, which is similar to the decomposition procedure with backordering. The performance measures of interest are the time averages of inventories and backorders, and the customer service levels in each facility.

Let us consider the supply chain shown in Figure 3.1. We will decompose the system in such a way that each subsystem consists of an inventory holding buffer with its own stock keeping policy. Consequently, we treat each subsystem as a single-location production or inventory system, which can be analyzed with a modest computational effort. Finally, we relate the subsystems to each other by using an iterative scheme. In summary, it includes constructing each subsystem, deriving a set of equations for the unknown parameters, and linking the subsystems to each other.

We propose to develop a decomposition as shown in Figure 3.2. The first subsystem, $\Omega(I)$, includes the input buffer of the plant in the supply chain. An (R_I, Q_I) inventory control policy is used to control replenishment process at the input buffer. Node $M_I^{'}$ models the effective procurement process and $M_I^{''}$ models the effective demand inter-arrival process at the input buffer. Next, we explain how we construct the nodes $M_i^{'}$ and $M_i^{''}$'s and their respective processing times $U_i^{'}$ and $U_i^{''}$'s for $i = I, O, DC, R$.

In this part, we analyze the effective demand inter-arrival times at subsystems. We start with the subsystem including the retailer, and continue with the rest in an orderly manner. The retailer faces customer demand according to a Poisson process with constant rate λ. Equivalently, the effective demand inter-arrival times are independent and follow an exponential distribution.

Due to the lost-sales practice in the supply chain, some portion of the demand at the retailer is lost. Let λ_e be the effective demand arrival rate. We compute λ_e as $\lambda_e = \lambda \times (1 - Pr(N_R = 0))$ where N_R represents the number of inventories at the retailer.

Demand to the DC arrives from the retailer. Every time the retailer receives $k_{DC} = Q_R$ orders, it places a replenishment request to DC. As a result, the orders to the DC follow an Erlang distribution with phase rate λ_e and k_{DC} phases due to the fact that orders to the retailer follow a Poisson process with rate λ.

The effective demand inter-arrival times to the output buffer and input buffer are analyzed accordingly.

The effective procurement times at each subsystem, in general, includes in addition to transportation times, possible delays experienced at the corresponding supplying echelons. The effective procurement times in lost sales case are identical to effective procurement times in backordering case and are not described here.

We calculate the steady-state probabilities of the underlying Markovian process in each subsystem. As before, each of the subsystems, $\Omega(i)$ for inventory holding buffers $i = I, O, DC, R$, is a two-node subsystem with its own stock keeping policy, and phase-type procurement and demand inter-arrival times.

Steady-state analysis of the subsystems $\Omega(I)$, $\Omega(O)$, and $\Omega(DC)$ are identical to the steady-state analysis in the backordering case and are not discussed here. The only difference is in the steady-state analysis of the subsystem $\Omega(R)$. Hence, we next present analysis of $\Omega(R)$.

Let $\{I_t, N_t, t \geq 0\}$ be a Markov chain where I_t represents the phase of $U_R^{'}$, and N_t denotes the level of inventories at the retailer where $I_t = 1, 2, B$, and $N_t = Q_R{+}R_R, Q_R{+}R_R{-}1, \ldots$,0. As in the subsystem $\Omega(DC)$ in backordering case, the effective procurement time has a complex phase structure. We again use a the three-moment *MGE-2* approximation (the parameters are γ_1, γ_2 and b). The Markov chain has finite number of states. The state-space and the transitions of the Markov chain are presented in Figure 3.7. Let the probabilities of

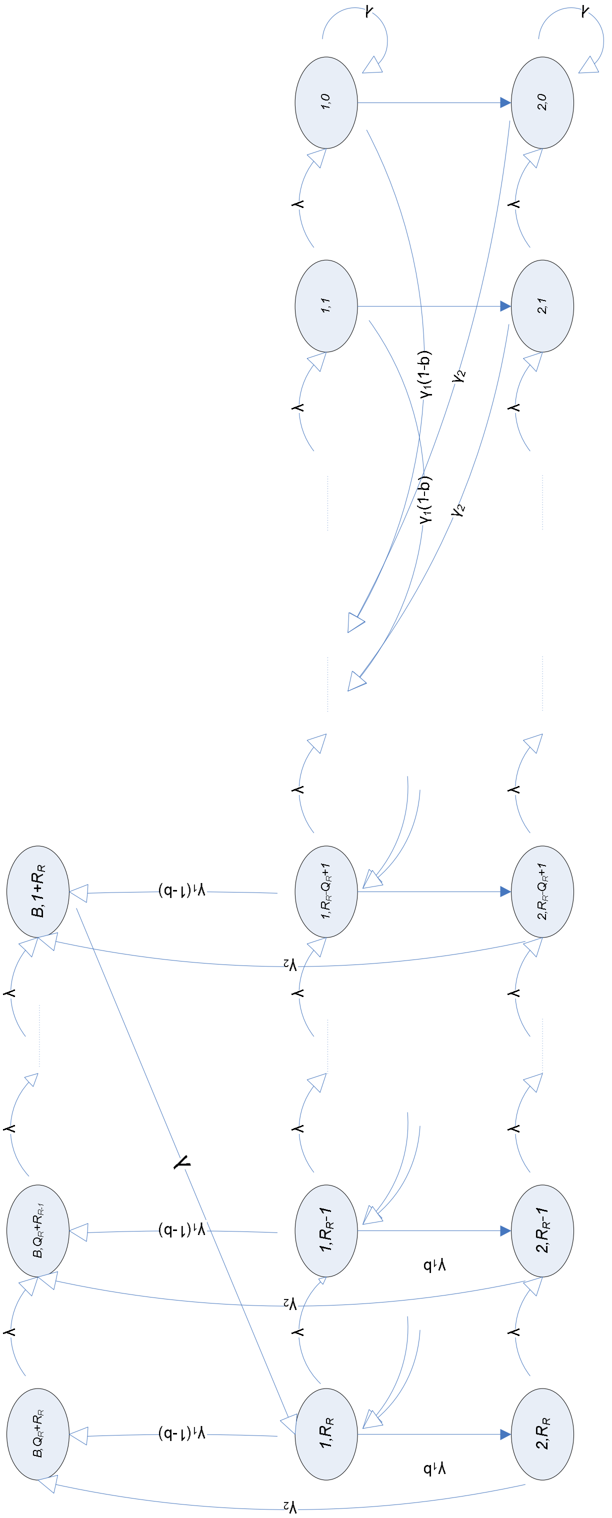


Figure 3.7: Transition diagram for subsystem $\Omega(R)$ (lost-sales case)

the subsystem be:

$$\tilde{\mathbf{P}}_w(n)|_{n=1}^2 = \begin{bmatrix} P(n,R_R) \\ P(n,R_R-1) \\ \vdots \\ P(n,0) \end{bmatrix}_{(R_R+1)\times 1}, \quad \tilde{\mathbf{P}}(B) = \begin{bmatrix} P(B,R_R+Q_R) \\ \vdots \\ P(B,R_R+1) \end{bmatrix}_{Q_R\times 1}.$$

Corresponding flow-balance equations are:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}(B) &= \mathbf{B}\tilde{\mathbf{P}}_w(1) + \mathbf{C}\tilde{\mathbf{P}}_w(2), \\ \mathbf{D}\tilde{\mathbf{P}}_w(1) &= \mathbf{E}\tilde{\mathbf{P}}(B), \\ \mathbf{F}\tilde{\mathbf{P}}_w(2) &= \gamma_1 b\tilde{\mathbf{P}}_w(1), \end{aligned} \tag{3.5}$$

where

$$\mathbf{A} = \begin{bmatrix} \lambda & & & \\ -\lambda & \lambda & & \\ & & \ddots & \\ & & -\lambda & \lambda \end{bmatrix}_{Q_R\times Q_R}, \quad \mathbf{B} = \begin{bmatrix} \gamma_1(1-b) & & \\ & \ddots & \\ & & \gamma_1(1-b) \\ & \cdots & \end{bmatrix}_{Q_R\times(R_R+1)},$$

$$\mathbf{C} = \begin{bmatrix} \gamma_2 & & \\ & \ddots & \\ & & \gamma_2 \\ & \cdots & \end{bmatrix}_{Q_R\times(R_R+1)}, \quad \mathbf{D} = \begin{bmatrix} \lambda+\gamma_1 & & & & \\ -\lambda & \lambda+\gamma_1 & & & \\ & & \ddots & & \\ & & -\lambda & \lambda+\gamma_1 & \\ & & & -\lambda & \gamma_1 \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \lambda+\gamma_2 & & & & \\ -\lambda & \lambda+\gamma_2 & & & \\ & & \ddots & & \\ & & -\lambda & \lambda+\gamma_2 & \\ & & & -\lambda & \gamma_2 \end{bmatrix},$$

and $\mathbf{D}$, $\mathbf{F}$ are $(R_R+1)\times(R_R+1)$ matrices. Additionally,

$$\mathbf{E} = [e_{ij}] = \begin{cases} \lambda, & \text{if } i=1, j=Q_R \\ 0, & \text{otherwise,} \end{cases}$$

is a $(R_R+1)\times(Q_R)$ matrix.

After, representing $\tilde{\mathbf{P}}(B)$ and $\tilde{\mathbf{P}}_w(2)$ in terms of $\tilde{\mathbf{P}}_w(1)$, and by utilizing Equation 3.5, we obtain $\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = 0$ where

$$\mathbf{P} = \mathbf{D} - \mathbf{E}\mathbf{A}^{-1}\mathbf{B} - \gamma_1 b\mathbf{E}\mathbf{A}^{-1}\mathbf{C}\mathbf{F}^{-1}.$$

The normalization condition is given as:

$$\mathbf{p} = \mathbf{e}_{(1\times(R_R+1))} \times (\mathbf{I} + \gamma_1 b\mathbf{F}^{-1}) + \mathbf{e}_{(1\times(Q_R))} \times (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{F}^{-1}).$$

Replacing the first row of matrix $\mathbf{P}$ by the row vector $\mathbf{p}$, we solve for

$$\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = [1, 0, \ldots, 0]^T_{(R_R+1)\times 1}.$$

The rest of the probabilities are given by:

$$\begin{aligned} \tilde{\mathbf{P}}_w(2) &= \gamma_1 b\mathbf{F}^{-1}\tilde{\mathbf{P}}_w(1), \\ \tilde{\mathbf{P}}(B) &= (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{F}^{-1})\tilde{\mathbf{P}}_w(1). \end{aligned}$$

Thus, we have analyzed each of the subsystems with its own stock keeping policies and with phase-type procurement and demand inter-arrival times. We present matrix-recursive procedures in order to compute the steady-state probabilities of the subsystems. An iterative algorithm that links the subsystems to each other is presented in the next paragraph.

This is again a fixed-point algorithm that subsystems supply information to each other. The unknown parameters of the subsystems are Π_O, Δ_I, $\omega_O(i),\ i = 0, 1, 2, \ldots$, $\omega_{DC}(i),\ i = 0, 1, 2, \ldots$, and U_i'', $i = O, DC, R$. As part of the algorithm, Δ_I is obtained from the analysis of $\Omega(I)$ and used in the analysis of $\Omega(O)$. Similarly, $\omega_O(i)$'s are obtained from the analysis of $\Omega(O)$ and used in the analysis of $\Omega(DC)$, and $\omega_{DC}(i)$'s are obtained from the analysis of $\Omega(DC)$ and used in the analysis of $\Omega(R)$. On the other hand, U_R'' is obtained from the analysis of $\Omega(R)$ and used in the analysis of $\Omega(DC)$. U_{DC}'', U_O'', and Π_O are exploited similarly.

As a result, the algorithm starts by assuming some initial values for the unknown parameters. It iterates back and forth between all the subsystems. After all the throughputs are sufficiently close to each other it stops. A summary of the algorithm is given in Table 3.5.

We test the accuracy of our disaggregation/aggregation approximation by comparing its results against simulation in a number of examples. The purpose of numerical examples is

```
1. Initialize: k=1,$\Pi_O = \Delta_I = \omega_O(i) = \omega_{DC}(i) = 0,$ for all $i = 0, 1, 2, \ldots,$ $U_i'' = \lambda,$
   $i = O, DC, R,$ and $\epsilon = 10^{-4}$.
2. Analyze $\Omega(I)$, obtain steady-state probabilities, update $\Delta_I$ and $\bar{\xi}_I$.
3. Analyze $\Omega(O)$, obtain steady-state probabilities, update $\Pi_O$, $\bar{\xi}_O$, and $\omega_O(i)$.
4. Analyze $\Omega(DC)$, obtain steady-state probabilities, update $\bar{\xi}_{DC}$ and $\omega_{DC}(i)$.
5. Analyze $\Omega(R)$, obtain steady-state probabilities, update $\bar{\xi}_R$ and $U_R''$.
6. Analyze $\Omega(DC)$, obtain steady-state probabilities, update $\bar{\xi}_{DC}$, and $U_{DC}''$.
7. Analyze $\Omega(O)$, obtain steady-state probabilities, update $\bar{\xi}_O$, and $U_O''$.
8. If max $\{|\bar{\xi}_I^k - \bar{\xi}_I^{k-1}|, |\bar{\xi}_O^k - \bar{\xi}_O^{k-1}|, |\bar{\xi}_{DC}^k - \bar{\xi}_{DC}^{k-1}|, |\bar{\xi}_R^k - \bar{\xi}_R^{k-1}|\} \leq \epsilon$, stop;
   else k = k+1, go to step 2.
```

Table 3.5: The aggregation algorithm for multi-echelon supply chains with lost sales

to see the ranges of the system parameters where the approximation is accurate and where it is not. The approximation and the simulation results are given in Tables 3.6, 3.7, and 3.8 for different traffic intensities.

We have three plant-related scenarios: low production rate (Table 3.6), medium production rate (Table 3.7),and high production rate (Table 3.8). It is clear from the results that the percentage deviation gradually increases as the demand rate (system load) increases.

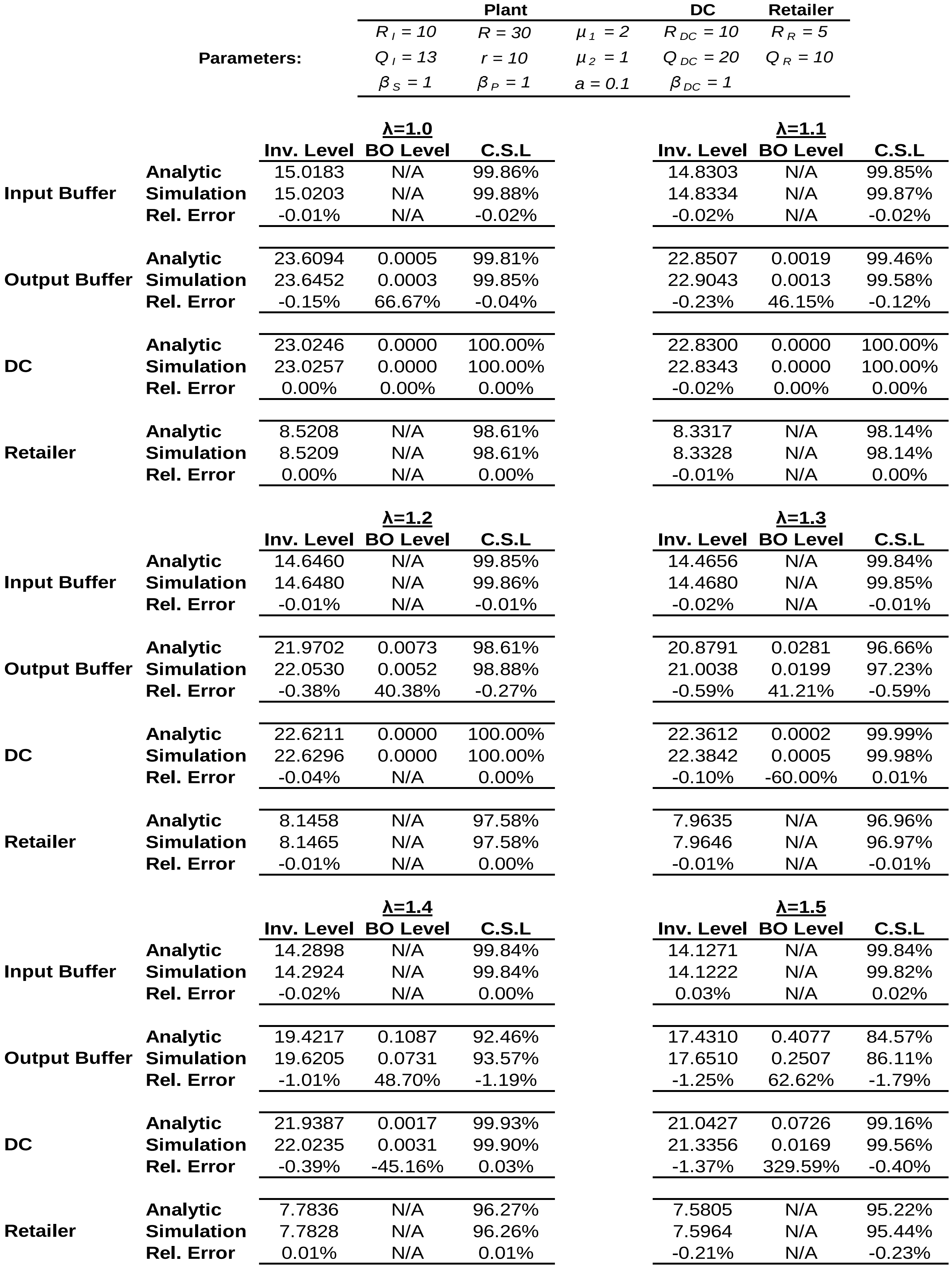
		Plant		DC	Retailer
	R_I = 10	R = 30	μ_1 = 1	R_{DC} = 10	R_R = 5
Parameters:	Q_I = 13	r = 10	μ_2 = 1	Q_{DC} = 20	Q_R = 10
	β_S = 1	β_P = 1	a = 0.1	β_{DC} = 1	

		λ=0.6			**λ=0.65**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	15.8024	N/A	99.99%	15.7037	N/A	99.99%
Input Buffer	**Simulation**	15.8021	N/A	99.99%	15.7046	N/A	99.99%
	Rel. Error	0.00%	N/A	0.00%	-0.01%	N/A	0.00%
	Analytic	22.8070	0.0009	99.65%	21.9669	0.0041	99.02%
Output Buffer	**Simulation**	22.8147	0.0009	99.68%	21.9764	0.0037	99.06%
	Rel. Error	-0.03%	0.00%	-0.03%	-0.04%	10.59%	-0.04%
	Analytic	23.7963	0.0000	100.00%	23.6801	0.0000	100.00%
DC	**Simulation**	23.7969	0.0000	100.00%	23.6821	0.0000	100.00%
	Rel. Error	0.00%	0.00%	0.00%	-0.01%	N/A	0.00%
	Analytic	9.3016	N/A	99.77%	9.2025	N/A	99.69%
Retailer	**Simulation**	9.3017	N/A	99.77%	9.2023	N/A	99.69%
	Rel. Error	0.00%	N/A	0.00%	0.00%	N/A	0.00%

		λ=0.7			**λ=0.75**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	15.6053	N/A	99.99%	15.5075	N/A	99.99%
Input Buffer	**Simulation**	15.6059	N/A	99.99%	15.5069	N/A	99.99%
	Rel. Error	0.00%	N/A	0.00%	0.00%	N/A	0.00%
	Analytic	20.9086	0.0180	97.37%	19.4458	0.0810	93.39%
Output Buffer	**Simulation**	20.9304	0.0163	97.49%	19.4962	0.0686	93.73%
	Rel. Error	-0.10%	10.73%	-0.12%	-0.26%	18.11%	-0.36%
	Analytic	23.5206	0.0000	100.00%	23.2068	0.0067	99.95%
DC	**Simulation**	23.5270	0.0002	99.99%	23.2404	0.0018	99.94%
	Rel. Error	-0.03%	-100.00%	0.01%	-0.14%	268.62%	0.01%
	Analytic	9.1037	N/A	99.59%	9.0040	N/A	99.47%
Retailer	**Simulation**	9.1034	N/A	99.59%	9.0036	N/A	99.47%
	Rel. Error	0.00%	N/A	0.00%	0.00%	N/A	0.00%

		λ=0.8			**λ=0.85**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	15.4115	N/A	99.99%	15.3948	N/A	99.99%
Input Buffer	**Simulation**	15.4093	N/A	99.99%	15.3191	N/A	99.99%
	Rel. Error	0.01%	N/A	0.00%	0.49%	N/A	0.00%
	Analytic	17.2192	0.3884	84.49%	16.6965	0.5187	82.09%
Output Buffer	**Simulation**	17.3397	0.2784	85.31%	13.9497	1.0050	68.87%
	Rel. Error	-0.69%	39.51%	-0.96%	19.69%	-48.39%	19.20%
	Analytic	22.3156	0.0661	99.14%	22.0056	0.1048	98.72%
DC	**Simulation**	22.5380	0.0136	99.63%	20.7545	0.0776	98.18%
	Rel. Error	-0.99%	385.11%	-0.49%	6.03%	35.10%	0.55%
	Analytic	8.8834	N/A	99.25%	8.7683	N/A	98.99%
Retailer	**Simulation**	8.8937	N/A	99.29%	8.7363	N/A	98.81%
	Rel. Error	-0.12%	N/A	-0.04%	0.37%	N/A	0.18%

Table 3.6: Accuracy of the approximation algorithm for a low production rate with lost sales



	Plant			DC	Retailer
	$R_I = 10$	$R = 30$	$\mu_1 = 2$	$R_{DC} = 10$	$R_R = 5$
Parameters:	$Q_I = 13$	$r = 10$	$\mu_2 = 1$	$Q_{DC} = 20$	$Q_R = 10$
	$\beta_S = 1$	$\beta_P = 1$	$a = 0.1$	$\beta_{DC} = 1$	

		λ=1.0			**λ=1.1**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	15.0183	N/A	99.86%	14.8303	N/A	99.85%
Input Buffer	**Simulation**	15.0203	N/A	99.88%	14.8334	N/A	99.87%
	Rel. Error	-0.01%	N/A	-0.02%	-0.02%	N/A	-0.02%
	Analytic	23.6094	0.0005	99.81%	22.8507	0.0019	99.46%
Output Buffer	**Simulation**	23.6452	0.0003	99.85%	22.9043	0.0013	99.58%
	Rel. Error	-0.15%	66.67%	-0.04%	-0.23%	46.15%	-0.12%
	Analytic	23.0246	0.0000	100.00%	22.8300	0.0000	100.00%
DC	**Simulation**	23.0257	0.0000	100.00%	22.8343	0.0000	100.00%
	Rel. Error	0.00%	0.00%	0.00%	-0.02%	0.00%	0.00%
	Analytic	8.5208	N/A	98.61%	8.3317	N/A	98.14%
Retailer	**Simulation**	8.5209	N/A	98.61%	8.3328	N/A	98.14%
	Rel. Error	0.00%	N/A	0.00%	-0.01%	N/A	0.00%

		λ=1.2			**λ=1.3**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.6460	N/A	99.85%	14.4656	N/A	99.84%
Input Buffer	**Simulation**	14.6480	N/A	99.86%	14.4680	N/A	99.85%
	Rel. Error	-0.01%	N/A	-0.01%	-0.02%	N/A	-0.01%
	Analytic	21.9702	0.0073	98.61%	20.8791	0.0281	96.66%
Output Buffer	**Simulation**	22.0530	0.0052	98.88%	21.0038	0.0199	97.23%
	Rel. Error	-0.38%	40.38%	-0.27%	-0.59%	41.21%	-0.59%
	Analytic	22.6211	0.0000	100.00%	22.3612	0.0002	99.99%
DC	**Simulation**	22.6296	0.0000	100.00%	22.3842	0.0005	99.98%
	Rel. Error	-0.04%	N/A	0.00%	-0.10%	-60.00%	0.01%
	Analytic	8.1458	N/A	97.58%	7.9635	N/A	96.96%
Retailer	**Simulation**	8.1465	N/A	97.58%	7.9646	N/A	96.97%
	Rel. Error	-0.01%	N/A	0.00%	-0.01%	N/A	-0.01%

		λ=1.4			**λ=1.5**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.2898	N/A	99.84%	14.1271	N/A	99.84%
Input Buffer	**Simulation**	14.2924	N/A	99.84%	14.1222	N/A	99.82%
	Rel. Error	-0.02%	N/A	0.00%	0.03%	N/A	0.02%
	Analytic	19.4217	0.1087	92.46%	17.4310	0.4077	84.57%
Output Buffer	**Simulation**	19.6205	0.0731	93.57%	17.6510	0.2507	86.11%
	Rel. Error	-1.01%	48.70%	-1.19%	-1.25%	62.62%	-1.79%
	Analytic	21.9387	0.0017	99.93%	21.0427	0.0726	99.16%
DC	**Simulation**	22.0235	0.0031	99.90%	21.3356	0.0169	99.56%
	Rel. Error	-0.39%	-45.16%	0.03%	-1.37%	329.59%	-0.40%
	Analytic	7.7836	N/A	96.27%	7.5805	N/A	95.22%
Retailer	**Simulation**	7.7828	N/A	96.26%	7.5964	N/A	95.44%
	Rel. Error	0.01%	N/A	0.01%	-0.21%	N/A	-0.23%

Table 3.7: Accuracy of the approximation algorithm for a medium production rate with lost sales

		Plant		DC	Retailer
	R_I = 10	*R = 30*	*μ_1 = 3*	*R_{DC} = 10*	*R_R = 5*
Parameters:	*Q_I = 13*	*r = 10*	*μ_2 = 1*	*Q_{DC} = 20*	*Q_R = 10*
	β_S = 1	*β_P = 1*	*a = 0.1*	*β_{DC} = 1*	

		λ=1.5			**λ=1.6**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	14.0903	N/A	99.61%	13.9211	N/A	99.60%
Input Buffer	**Simulation**	14.1013	N/A	99.61%	13.9328	N/A	99.60%
	Rel. Error	-0.08%	N/A	0.00%	-0.08%	N/A	0.00%
	Analytic	22.9957	0.0042	99.13%	22.4157	0.0093	98.50%
Output Buffer	**Simulation**	23.1407	0.0024	99.41%	22.5938	0.0053	98.95%
	Rel. Error	-0.63%	75.00%	-0.28%	-0.79%	75.47%	-0.45%
	Analytic	22.1138	0.0002	99.99%	21.9274	0.0004	99.98%
DC	**Simulation**	22.1203	0.0002	99.99%	21.9413	0.0004	99.98%
	Rel. Error	-0.03%	0.00%	0.00%	-0.06%	0.00%	0.00%
	Analytic	7.6111	N/A	95.54%	7.4412	N/A	94.75%
Retailer	**Simulation**	7.6107	N/A	95.54%	7.4415	N/A	94.74%
	Rel. Error	0.01%	N/A	0.00%	0.00%	N/A	0.01%

		λ=1.7			**λ=1.8**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	13.7567	N/A	99.59%	13.5972	N/A	99.59%
Input Buffer	**Simulation**	13.7718	N/A	99.58%	13.6082	N/A	99.55%
	Rel. Error	-0.11%	N/A	0.01%	-0.08%	N/A	0.04%
	Analytic	21.7709	0.0202	97.49%	21.0382	0.0432	95.94%
Output Buffer	**Simulation**	21.9873	0.0122	98.18%	21.2934	0.0259	96.92%
	Rel. Error	-0.98%	65.57%	-0.70%	-1.20%	66.80%	-1.01%
	Analytic	21.7265	0.0007	99.96%	21.4954	0.0014	99.93%
DC	**Simulation**	21.7479	0.0008	99.97%	21.5380	0.0017	99.94%
	Rel. Error	-0.10%	-12.50%	-0.01%	-0.20%	-17.65%	-0.01%
	Analytic	7.2755	N/A	93.91%	7.1140	N/A	93.02%
Retailer	**Simulation**	7.2762	N/A	93.90%	7.1159	N/A	93.03%
	Rel. Error	-0.01%	N/A	0.01%	-0.03%	N/A	-0.01%

		λ=1.9			**λ=2.0**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	13.4430	N/A	99.58%	13.2960	N/A	99.57%
Input Buffer	**Simulation**	13.4545	N/A	99.52%	13.3035	N/A	99.49%
	Rel. Error	-0.09%	N/A	0.06%	-0.06%	N/A	0.08%
	Analytic	20.1887	0.0912	93.62%	19.1951	0.1894	90.30%
Output Buffer	**Simulation**	20.4906	0.0540	94.96%	19.5549	0.1088	92.11%
	Rel. Error	-1.47%	68.89%	-1.41%	-1.84%	74.08%	-1.97%
	Analytic	21.2057	0.0034	99.86%	20.8094	0.0108	99.65%
DC	**Simulation**	21.2845	0.0039	99.87%	20.9692	0.0092	99.74%
	Rel. Error	-0.37%	-12.82%	-0.01%	-0.76%	17.39%	-0.09%
	Analytic	6.9557	N/A	92.09%	6.7974	N/A	91.08%
Retailer	**Simulation**	6.9584	N/A	92.12%	6.8028	N/A	91.13%
	Rel. Error	-0.04%	N/A	-0.03%	-0.08%	N/A	-0.05%

Table 3.8: Accuracy of the approximation algorithm for a high production rate with lost sales

Chapter 4
Designing the Supply Chain

In this chapter, we propose an optimization procedure to help design the supply chain by calculating optimal parameter values minimizing the expected total cost. Optimal design of the material flow system is part of the overall planning and operation of a supply chain. The optimal configuration specifies not only how much and where to hold inventory but also how to move inventory across the supply chain. Following this, we examine the attributes that drive the overall performance of the supply chain.

In multi-echelon supply chains, optimal production and inventory control policies have quite complex structures. This is because the control policy for a given echelon has a considerable impact on the other echelons. In fact, the general practice is to restrict the control policies to a class of general operating schemes. All echelons, for example, apply reorder point/order quantity inventory control policies. Optimization in this sense is to coordinate such operating schemes in the best possible way.

The focus of this chapter is on the multi-echelon supply chain illustrated in Figure 3.1. Production at the manufacturing plant is controlled by a continuous review (R, r) policy. Material flow between stages is achieved by reorder point/order quantity inventory control policies. So far, we have achieved a fast aggregation/disaggregation approximation method that provides us with a set of key performance measures in the supply chain such as the time averages of inventory and backorder levels, as well as the customer service levels. Here, we use these measures to construct an optimization framework that effectively address the possible configuration of control policies.

4.1 Problem Formulation

The objective of optimization in our problem is to determine appropriate production and inventory policy parameters. A viable approach to solve the optimization problem is to employ a cost-minimizing objective function that assigns penalties for holding inventory and shortages. In addition, a penalty per set-up or ordering is charged to avoid excessive set-ups or replenishment orders, respectively.

Let us introduce the following notation:

λ : demand rate at the retailer,

K_i : set-up or ordering cost per replenishment order at echelon i, $i = I, O, DC, R$

h_i : unit holding cost per unit time at echelon i, $i = I, O, DC, R$

g_i : unit backordering cost per unit time at echelon i, $i = O, DC, R$

p_i : shortage cost per unit short at echelon i, $i = I, O, DC, R$

TC_i steady-state expected total cost per unit time at echelon i, $i = I, O, DC, R$

TC : $TC_I(R_I, Q_I) + TC_O(R, r) + TC_{DC}(R_{DC}, Q_{DC}) + TC_R(R_R, Q_R)$.

The expected total cost for subsystem i includes set-up or ordering cost, holding and backordering costs per unit time, and shortage cost per unit short. Thus, the expected total cost per unit time can be written as

$$TC_i = \frac{K_i \lambda}{Q_i} + h_i \bar{N}_i + g_i \bar{B}_i + p_i \lambda \mathrm{Pr}_i(backorder),$$

where $\bar{N}_i$ denotes long-run average number of inventories at echelon i, $i = I, O, DC, R$, $\bar{B}_i$ denotes the long-run average number of backorders at echelon i, $i = O, DC, R$, and Pr_i denotes the probability of encountering a shortage upon order arrival at echelon $i = I, O, DC, R$.

Optimal configuration here constitutes the best trade-off among set-up or ordering, holding, backordering and shortage costs. The overall goal is to minimize the total expected system-wide costs, TC, throughout the supply chain, and find the corresponding decision variables (R_I, Q_I), (R, r), (R_{DC}, Q_{DC}), (R_R, Q_R).

We can construct other optimization problems as well. One of them is an optimization framework that minimizes total system-wide costs (ordering and holding costs) while conforming a prescribed customer service level. Boyaci and Gallego [26] relates this service-constrained model to the traditional model with backorder costs and shows that it is possible to prespecify backorder costs to achieve desired service levels. Another problem is an optimization scheme that maximizes customer service level subject to a given system capacity constraint, among others. However, we address only the first problem in this thesis.

4.2 An Approximate Optimization Procedure

In this section, we propose an iterative optimization procedure, which iterates back and forth among the subsystems. In each iteration, it uses the approach developed in the previous sections to evaluate the system performance. The basic idea of the optimization procedure is that we optimize the original system as we optimize the subsystems while the iterative procedure continues. That is, the total system cost is reduced by reducing the individual subsystem costs.

In fact, the optimization procedure both updates the values of control parameters and the unknown parameters of the decomposition approach. To be more precise, as the procedure passes through the first subsystem, $\Omega(I)$, it updates the unknown parameter Δ_I as well as control parameters (R_I, Q_I) so that the first subsystem cost is reduced. Then, the procedure proceeds with the second subsystem, $\Omega(O)$. In a similar vein, it updates Π_O, $\omega_O(i)$'s, and the policy values (R, r) so that the second subsystem cost is also decreased, and so on. The algorithm continues in this way until the convergence criterion is achieved and there is no further improvement in the subsystem costs.

In each subsystem, we use a direct search method to improve the subsystem cost. These methods are heuristic techniques and use only function values to improve the current solution. Since our procedure provides us the function values (subsystem costs) in a fast pace, direct search methods are suitable for our problem. In particular, we use a modified Hooke-Jeeves pattern search method. The Hooke-Jeeves method has been extensively used to incorporate the history of a sequence of iterations into the generation of a new search direction [68, 90].

The Hooke-Jeeves method performs two types of search. An *exploratory* search examines the local behavior of the function being optimized. Then, a *pattern* move uses the information generated in the exploration search to accelerate the convergence of the method.

The exploratory search proceeds from an initial point to each coordinate direction by a specified step size. If the function value improves, the move is considered successful and the current point is retained. Otherwise, the step is replaced by a step in the opposite direction and the resulting point is retained depending upon whether it succeeds or fails. The exploratory search continues until all coordinate directions are investigated. The resulting point is termed as a *base point.* Then, the pattern move starts from the current base point and moves along the direction from the previous to the current base point. If the function value improves, this new point is termed as the temporary base point. An exploratory search is conducted starting from the temporary base point. If the exploratory search finds a point with an improved function value, the temporary base point is accepted as the new base point. If not, the search resumes to the previous base point for a new exploratory search. The overall search terminates whenever this exploratory search fails.

The search procedure can be made more efficient if we consider the special property of the underlying Markov chain of the subsystems. We have eight parameters in the search space, namely (R_I, Q_I), (R, r), (R_{DC}, Q_{DC}), and (R_R, Q_R). In our case, the second subsystem, $\Omega(O)$, possesses the following important property: probabilities remain the same for fixed $Q = R - r$ as explained in [6]. For fixed $Q = R - r$, the transition rates of the Markov chain remain the same no matter what the values R and r are. So, a one-dimensional search procedure is sufficient for each Q and R. This property also holds for subsystems $\Omega(DC)$ and $\Omega(R)$. That is, for fixed Q_{DC} and Q_R the long-run probabilities continue remain the same for subsystems $\Omega(DC)$ and $\Omega(R)$, respectively. Therefore, once the probabilities obtained for a given Q_{DC} and Q_R, we can evaluate the cost function for all pairs of (R_{DC}, Q_{DC}) and (R_R, Q_R), no matter what the values of R_{DC} and R_R are. These special characteristics of the subsystems substantially reduce the computational effort required for the optimization procedure.

Following this, we develop two different search schemes. First one will use only modified exploratory moves of the Hooke-Jeeves method, which we call *single step* search. Second one will use both modified exploratory and pattern moves all the way to the optimal solution, which we call *optimized steps* search. We investigate the convergence and stability issues of both search schemes.

To sum up, the optimization procedure starts from an initial point in the feasible region. As the iterative procedure goes back and forth among the subsystems, modified Hooke-Jeeves method perturbs the production and inventory control parameters of the subsystems so that the respective costs are improved. During these steps, we exploit the special property of the underlying Markov chains of the subsystems to reduce the required computational effort in the search procedure. In the next sections, we give a detailed description of the optimization procedure both with single step search and optimized step search.

4.2.1 Optimal Configuration with Single Step Search

In this section, we consider the optimization procedure only exploiting the modified exploratory moves of the Hooke-Jeeves method. The procedure starts from an initial point in the feasible region. As it iterates throughout the subsystems, modified exploratory moves of the Hooke-Jeeves method perturbs the production and inventory control parameters.

In a given subsystem, the modified exploratory moves search all the directions around a given point and chooses the one that has the minimum subsystem cost. Our procedure starts from the first subsystem, $\Omega(I)$. As it passes through the first subsystem, it updates the unknown parameter Δ_I as well as the control parameters, (R_I, Q_I), so that the cost of the first subsystem is decreased. The exploratory search of the Hooke-Jeeves method uses the directions $d_1 = (1, 0)$ and $d_2 = (0, 1)$. The initial base point is (R_I, Q_I), and its function value is its corresponding subsystem cost. The exploratory search first checks the cost of the subsystem at $(R_I, Q_I) + d_1$. If the cost is lower, it continues with the other direction, d_2, proceeding from the point $(R_I, Q_I) + d_1$ and its corresponding cost. If not, it considers the opposite direction and checks the cost at $(R_I, Q_I) - d_1$. Again, if the cost is lower, it continues with the direction, d_2, proceeding from the point $(R_I, Q_I) - d_1$ and its corresponding cost. If not, the method continues with the direction, d_2, proceeding from the point (R_I, Q_I) and its corresponding function value, etc. In general, the method continues to search other directions until all of them are exhausted. On the whole, the exploratory search fails to consider all the adjacent points of (R_I, Q_I).

The exploratory search is modified so that it searches all the adjacent points of (R_I, Q_I)

and selects the one that has the minimum subsystem cost. The modified exploratory search uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The initial base point is (R_I, Q_I) and its function value is its corresponding subsystem cost. The search checks the cost at $(R_I, Q_I) + d_1$, $(R_I, Q_I) - d_1$, $(R_I, Q_I) + d_2$, $\ldots$,$(R_I, Q_I) - d_4$. The one with the minimum subsystem cost is selected. In fact, there are eight adjacent points of (R_I, Q_I) and the underlying Markov chain need to be solved eight times for the long-run probabilities. Note that, the modified exploratory search is utilized only once, not all the way to the optimal solution.

The optimization procedure continues with the other subsystems. As the procedure passes through the second subsystem, $\Omega(O)$, it updates the unknown parameters Π_O and $\omega_O(i)$'s as well as the control parameters, (R, r), so that the cost of the second subsystem is decreased. Again, it uses the modified exploratory moves of the Hooke-Jeeves method. The exploratory search again uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The initial point is (R, r), and the function value is its corresponding subsystem cost. The search checks the costs of the subsystem at $(R, r) + d_1$, $(R, r) - d_1$, $(R, r) + d_2$, $\ldots$,$(R, r) - d_4$. It chooses the one with the minimum subsystem cost.

In the second subsystem, the exploratory search is more efficient due to the special property of the underlying Markov chain. Since, all the probabilities remain same for fixed $Q = R - r$, the underlying Markov chain need to be solved only once for $(R + 1, r + 1)$, (R, r) and $(R - 1, r - 1)$, once for $(R + 1, r)$, $(R, r - 1)$, and once for $(R, r + 1)$, $(R - 1, r)$. There are two more cases to be considered $(R - 1, r + 1)$ and $(R + 1, r - 1)$. As a result, the underlying Markov chain need to be solved only five times for the long-run probabilities. Next, the optimization procedure proceeds with the third subsystem.

As the optimization procedure passes through the third subsystem, $\Omega(DC)$, the modified exploratory search uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The starting base point is (R_{DC}, Q_{DC}), and the function value is its corresponding subsystem cost. The search checks the cost at $(R_{DC}, Q_{DC}) + d_1$, $(R_{DC}, Q_{DC}) - d_1$, $(R_{DC}, Q_{DC}) + d_2$, $\ldots$,$(R_{DC}, Q_{DC}) - d_4$. The point with the minimal subsystem cost is selected.

In a similar vein, the special property of the underlying Markov chain makes the exploratory search much more efficient. The underlying Markov chain need to be solved only once for the $(R_{DC}+1, Q_{DC}+1)$, $(R_{DC}, Q_{DC}+1)$ and $(R_{DC}-1, Q_{DC}+1)$, once for $(R_{DC}+1, Q_{DC})$, (R_{DC}, Q_{DC}), $(R_{DC}-1, Q_{DC})$, and once for $(R_{DC}+1, Q_{DC}-1)$, $(R_{DC}, Q_{DC}-1)$, $(R_{DC}-1, Q_{DC}-1)$. Consequently, the underlying Markov chain need to be solved only three times for the long-run probabilities.

Finally, the optimization procedure passes through the fourth subsystem, $\Omega(R)$. The exploratory search checks the cost of the subsystem at $(R_R, Q_R)+d_1$, $(R_R, Q_R)-d_1$, $(R_R, Q_R)+d_2$, ...,$(R_R, Q_R)-d_4$ where $d_1=(1,0)$, $d_2=(0,1)$, $d_3=(1,1)$, and $d_4=(1,-1)$. It selects the one with the minimum subsystem cost.

Again, the underlying Markov chain need to be solved only three times for the long-run probabilities due to the special property of the underlying Markov chain. The underlying Markov chain need to be solved only once for (R_R+1, Q_R+1), (R_R, Q_R+1), (R_R-1, Q_R+1), once for (R_R+1, Q_R), (R_R, Q_R), (R_R-1, Q_R), and once for (R_R+1, Q_R-1), (R_R, Q_R-1), (R_R-1, Q_R-1). In the forward iteration, the modified exploratory search has been utilized only once, not all the way to the optimal solution.

When the optimization procedure completes a forward iteration, it starts a backward iteration as well. In the backward iteration, it passes only through the third subsystem, $\Omega(DC)$, and the second subsystem, $\Omega(O)$. As the procedure passes through the third subsystem, it utilizes both the modified exploratory and the pattern moves of the Hooke-Jeeves method in contrast to the forward iteration. The underlying reason of using both moves is to extend the search space. The procedure needs this extension because if the value of Q_R is changed in the forward iteration, the usual modified exploratory search does not check the values of Q_{DC} that are multiples of Q_R. If, for example, the optimal value of Q_{DC} is equal to $2Q_R$, the regular search space does not include the point $2Q_R$. In a way, the backward iteration is a procedure that adopts the changes in the forward iteration.

The exploratory search uses the directions $d_1=(1,0)$, $d_2=(0,1)$, $d_3=(1,1)$, and $d_4=(1,-1)$. The initial base point is (R_{DC}, Q_{DC}) with its corresponding subsystem cost. The search checks the cost of the subsystem at $(R_{DC}, Q_{DC})+d_1$, $(R_{DC}, Q_{DC})-d_1$, $(R_{DC}, Q_{DC})+d_2$, ...,$(R_{DC}, Q_{DC})-d_4$. It selects the one with the minimum subsystem cost. Let us suppose that the point $(R_{DC}+1, Q_{DC}+1)$ has the minimal value. Now, this point is termed as the current base point. The pattern move starts from the current base point

$(R_{DC}+1, Q_{DC}+1)$ and moves along the direction $(R_{DC}+1, Q_{DC}+1) - (R_{DC}, Q_{DC})$. If the resulting point, $(R_{DC}+2, Q_{DC}+2)$, has an improved function value, this new point is termed as the temporary base point. An exploratory search is conducted around the temporary base point $(R_{DC}+2, Q_{DC}+2)$. If the exploratory search finds a point with an improved function value, the temporary base point, $(R_{DC}+2, Q_{DC}+2)$, is accepted as the new base point. If not, the search resumes to the previous base point, $(R_{DC}+1, Q_{DC}+1)$, for a new exploratory search. The overall search terminates whenever this exploratory search fails. Note that, the exploratory search utilizes both searches all the way to the optimal solution.

As the procedure passes through the second subsystem, $\Omega(O)$, in the backward iteration, it utilizes both the modified exploratory and the pattern moves of the Hooke-Jeeves method as well. Again, the reason for using both searches is to be adaptive to the parameter changes in the previous steps. The exploratory search again uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The initial base point is (R, r), and the function value is its corresponding subsystem cost. The search checks the costs of the subsystem at $(R, r) + d_1$, $(R, r) - d_1$, $(R, r) + d_2$, $\ldots$,$(R, r) - d_4$. It chooses the one with the minimum subsystem cost. Let us suppose that the point $(R+1, r+1)$ has the minimum cost, which makes it the next base point. The pattern move starts from $(R+1, r+1)$ and moves along the direction $(R+1, r+1) - (R, r)$. So, if the resulting point, say $(R+2, r+2)$, has an improved cost, this new point is termed as the temporary base point. An exploratory search is conducted around the temporary base point $(R+2, r+2)$. If the exploratory search finds a point with an improved cost, the temporary base point, $(R+2, r+2)$, is accepted as the new base point. If not, the search resumes at the previous base point, $(R+1, r+1)$, for a new exploratory search. The overall search ends whenever the exploratory search fails.

The iterative optimization procedure stops whenever all the production and inventory control parameters converge to their final values. In case there is cyclical behavior, we stop the algorithm after two identical cycles accepting the current solution. A summary of the algorithm is given in Table 4.1.

1.	Initialize: k=1, $\Pi_O = \Delta_I = 0, \omega_O(i) = \omega_{DC}(i) = 0,$ for all $i = 0, 1, 2, \ldots$, $\epsilon = 10^{-4}$, and $(R_I^k, Q_I^k), (R^k, r^k), (R_{DC}^k, Q_{DC}^k), (R_R^k, Q_R^k)$.
2.	Iteration k
i.	Perform exploratory moves on $\Omega(I)$, update (R_I, Q_I), Δ_I and $\bar{\xi}_I$.
ii.	Perform exploratory moves on $\Omega(O)$, update (R, r), $\omega_O(i)$.
iii.	Perform exploratory moves on $\Omega(DC)$, update (R_{DC}, Q_{DC}), $\omega_{DC}(i)$.
iv.	Perform exploratory moves on $\Omega(R)$, update (R_R, Q_R), $\bar{\xi}_R$.
v.	Perform exploratory and pattern moves on $\Omega(DC)$, update (R_{DC}, Q_{DC}), $\bar{\xi}_{DC}$.
vi.	Perform exploratory and pattern moves on $\Omega(O)$, update (R, r), Π_O and $\bar{\xi}_O$.
3.	If $\max\{|\bar{\xi}_I^k - \bar{\xi}_I^{k-1}|, |\bar{\xi}_O^k - \bar{\xi}_O^{k-1}|, |\bar{\xi}_{DC}^k - \bar{\xi}_{DC}^{k-1}|, |\bar{\xi}_R^k - \bar{\xi}_R^{k-1}|\} \le \epsilon$, and $\max\{|(R_i^k, Q_i^k) - (R_i^{k-1}, Q_i^{k-1})|, |(R^k, r^k) - (R^{k-1}, r^{k-1})|\} \le 0$ for all i, $i = I, DC, R$, stop; else let k = k+1, and go to step 2.

Table 4.1: The single step optimization procedure for the multi-echelon supply chain

4.2.2 Optimal Configuration with the Optimized Step Search

In this section, we consider the optimization procedure exploiting both the modified exploratory moves and pattern moves of the Hooke-Jeeves method all the way to the optimal solution. The procedure starts from an initial point in the feasible region. As it iterates throughout the subsystems, modified exploratory search and pattern moves of the Hooke-Jeeves method perturbs the production and inventory control parameters of the subsystems.

The procedure starts from the first subsystem, $\Omega(I)$. As the procedure passes through the first subsystem, it updates the unknown parameter Δ_I as well as the policy values, (R_I, Q_I), so that the cost of the subsystem is decreased. It uses both the modified exploratory search and pattern moves of the Hooke-Jeeves method to check for possible improvement directions and chooses the one with the lowest subsystem cost.

The modified exploratory search uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The initial point is (R_I, Q_I), and its function value is its corresponding subsystem cost. The search checks the cost of the subsystem at $(R_I, Q_I) + d_1$, $(R_I, Q_I) - d_1$, $(R_I, Q_I) + d_2$, $\ldots$,$(R_I, Q_I) - d_4$. It selects the one with the minimum subsystem cost.

Let us assume that the point $(R_I + 1, Q_I + 1)$ has the minimum function value. This point is termed as the base point. The pattern move starts from $(R_I + 1, Q_I + 1)$ and moves along the direction $(R_I + 1, Q_I + 1) - (R_I, Q_I)$. If the resulting point, $(R_I + 2, Q_I + 2)$, has an improved function value, this new point is termed as the temporary base point. An exploratory search is conducted around the temporary base point $(R_I + 2, Q_I + 2)$. If the exploratory search finds a point with an improved function value, the temporary base point, $(R_I + 2, Q_I + 2)$, is accepted as the new base point. If not, the search resumes to the previous base point, $(R_I + 1, Q_I + 1)$, for a new exploratory search. The overall search terminates whenever this exploratory search fails.

The optimization procedure continues with the other subsystems. As the iterative procedure passes through the second subsystem, $\Omega(O)$, it utilizes both the exploratory and the pattern moves all the way to the optimal solution. The exploratory search again uses the directions $d_1 = (1, 0)$, $d_2 = (0, 1)$, $d_3 = (1, 1)$, and $d_4 = (1, -1)$. The initial point is (R, r), and the function value is its corresponding subsystem cost. The search checks the costs of the subsystem at $(R, r) + d_1$, $(R, r) - d_1$, $(R, r) + d_2$, $\ldots$,$(R, r) - d_4$. It selects the one with the minimum subsystem cost. Let us suppose that the point $(R+1, r+1)$ has the minimum cost, which makes it the next base point. The pattern move starts from $(R + 1, r + 1)$ and moves along the direction $(R+1, r+1) - (R, r)$. So, if the resulting point, say $(R+2, r+2)$, has an improved cost, this new point is termed as the temporary base point. An exploratory search is conducted around the temporary base point $(R+2, r+2)$. If the exploratory search finds a point with an improved cost, the temporary base point, $(R + 2, r + 2)$, is accepted as the new base point. If not, the search resumes at the previous base point, $(R + 1, r + 1)$, for a new exploratory search. The overall search ends whenever the exploratory search fails.

In a similar vein, the procedure passes through the third and fourth subsystems. It utilizes both the exploratory and the pattern moves in the third subsystem. However, it utilizes only the exploratory moves in the fourth subsystem. This is because, if the procedure employs both the exploratory and pattern moves in the fourth subsystem, the policy values in the fourth subsystem, (R_R, Q_R), can change considerably. Consequently, the search procedure employed in the backward iteration can not adopt the changes in the forward iteration.

When the optimization procedure completes a forward iteration, it starts a backward iteration. In the backward iteration, it passes only through the third subsystem, $\Omega(DC)$, and the second subsystem, $\Omega(O)$. Again, as the procedure passes through the third subsystem, $\Omega(DC)$, it utilizes both the modified exploratory and the pattern moves of the Hooke-Jeeves method.

As the procedure passes through the second subsystem, $\Omega(O)$, in the backward iteration, again it utilizes both the exploratory and the pattern moves. It uses both to be adaptive to the parameter changes in the previous steps.

The iterative optimization procedure stops whenever all the production and inventory control parameters converge to their final values. In case there is cyclical behavior, we stop the algorithm after two identical cycles accepting the current solution. A summary of the algorithm is given in Table 4.2.

1.	Initialize: k=1, $\Pi_O = \Delta_I = 0, \omega_O(i) = \omega_{DC}(i) = 0,$ for all $i = 0, 1, 2, \ldots$, $\epsilon = 10^{-4}$, and $(R_I^k, Q_I^k), (R^k, r^k), (R_{DC}^k, Q_{DC}^k), (R_R^k, Q_R^k)$.
2.	Iteration k
i.	Perform exploratory and pattern moves on $\Omega(I)$, update (R_I, Q_I), Δ_I and $\bar{\xi}_I$.
ii.	Perform exploratory pattern moves on $\Omega(O)$, update (R, r), $\omega_O(i)$.
iii.	Perform exploratory pattern moves on $\Omega(DC)$, update (R_{DC}, Q_{DC}), $\omega_{DC}(i)$.
iv.	Perform exploratory moves on $\Omega(R)$, update (R_R, Q_R), $\bar{\xi}_R$.
v.	Perform exploratory and pattern moves on $\Omega(DC)$, update (R_{DC}, Q_{DC}), $\bar{\xi}_{DC}$.
vi.	Perform exploratory and pattern moves on $\Omega(O)$, update (R, r), Π_O and $\bar{\xi}_O$.
3.	If $\max\{\lvert\bar{\xi}_I^k - \bar{\xi}_I^{k-1}\rvert, \lvert\bar{\xi}_O^k - \bar{\xi}_O^{k-1}\rvert, \lvert\bar{\xi}_{DC}^k - \bar{\xi}_{DC}^{k-1}\rvert, \lvert\bar{\xi}_R^k - \bar{\xi}_R^{k-1}\rvert\} \le \epsilon$, and $\max\{\lvert(R_i^k, Q_i^k) - (R_i^{k-1}, Q_i^{k-1})\rvert, \lvert(R^k, r^k) - (R^{k-1}, r^{k-1})\rvert\} \le 0$ for all i, $i = I, DC, R$, stop; else let k = k+1, and go to step 2.

Table 4.2: The optimized step optimization procedure for the multi-echelon supply chain

4.3 Numerical Experience

In this section, we address a number of planning and control issues in the multi-echelon supply chain considered in this chapter. We present the path of convergence on some of the numerical examples to provide insight into both the single step (Tables 4.3, 4.4) and the optimized step (Tables 4.5, 4.6) optimization procedures, and the optimal configuration of production and inventory control policies. For instance, in Table 4.3, the upper part shows the demand rate, production capacity at the plant, transportation times, and cost parameters at the input buffer, output buffer, DC and retailer. In the following part, the first row shows the initial values of the inventory and production control parameters and their respective subsystem costs. For example, the value of (Q_I, R_I) is equal to $(13, 10)$ in the input buffer, and the value of (R, r) is equal to $(30, 10)$ in the output buffer, etc. In addition, the subsystem

				I.B.	O.B.	DC	Retailer
	μ_1 = 2	β_S = 1	***K***	0	25	20	15
λ=1.5	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	*10*	*30*	*10*	*20*	*10*	*10*	*5*	*2.0495*	*12.8896*	*17.4546*	*11.8722*	48.6952	89.71%
12	*9*	*31*	*11*	*20*	*11*	*11*	*6*	*2.9038*	*12.9424*	*15.6534*	*11.637*	43.1367	
		35	*13*	*22*	*12*				*13.0352*	*16.3958*		43.9718	
11	*9*	*35*	*13*	*22*	*12*	*12*	*5*	*2.8447*	*13.0659*	*16.4209*	*10.9298*	43.2613	
		36	*12*	*24*	*1*				*13.2773*	*12.2116*		39.2634	
10	*8*	*37*	*13*	*24*	*1*	*12*	*5*	*2.8257*	*13.5157*	*12.7513*	*10.9966*	40.0893	
		37	*13*	*24*	*1*				*13.5157*	*12.7513*		40.0893	
10	*8*	*37*	*13*	*24*	*1*	*12*	*5*	*2.8213*	*13.519*	*12.7597*	*11.0009*	40.1009	
		37	*13*	*24*	*1*				*13.519*	*12.7597*		40.1009	
10	*8*	*37*	*13*	*24*	*1*	*12*	*5*	*2.8213*	*13.519*	*12.7597*	*11.0009*	40.1009	
		37	*13*	*24*	*1*				*13.519*	*12.7597*		40.1009	
													C.S.L
10	*8*	*37*	*13*	*24*	*1*	*12*	*5*	*2.8213*	*13.519*	*12.7597*	*11.0009*	40.1009	93.51%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table 4.3: Convergence path of the optimization procedure with single steps (medium production rate)

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 2$	$\beta_S = 1$	***K***	30	25	20	15
λ=1.5	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	$a = 0.1$	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	6.5111	12.8839	17.4397	11.8606	48.6952	89.71%
14	9	31	11	20	11	11	6	6.2612	12.8844	15.5804	11.5368	46.2627	
		35	13	22	12				12.9888	16.3566		47.1433	
15	8	35	13	22	12	12	5	6.0874	13.0987	16.4442	10.9708	46.6011	
		37	13	24	1				13.3082	12.2814		42.6478	
16	8	37	13	24	1	11	5	5.9705	13.2861	12.204	10.7301	42.1907	
		35	13	22	1				13.0752	12.1223		41.8981	
17	8	35	13	22	1	12	5	5.8708	13.0541	12.104	10.9154	41.9443	
		36	12	24	1				13.2652	12.1867		42.2382	
18	7	37	13	24	1	12	5	5.8029	13.4016	12.4706	10.8595	42.5346	
		37	13	24	1				13.4016	12.4706		42.5346	
19	7	37	13	24	1	12	5	5.7361	13.3771	12.4125	10.8323	42.358	
		37	13	24	1				13.3771	12.4125		42.358	
20	7	37	13	24	1	12	5	5.6901	13.3536	12.3577	10.807	42.2085	
		37	13	24	1				13.3536	12.3577		42.2085	
21	7	37	13	24	1	11	5	5.6581	13.3327	12.3094	10.7834	42.0835	
		35	13	22	1				13.1249	12.2604		41.8267	
22	7	35	13	22	1	12	5	5.6335	13.1049	12.2456	10.9786	41.9625	
		37	13	24	1				13.3139	12.2946		42.2207	
23	7	37	13	24	1	11	5	5.632	13.296	12.2261	10.7411	41.8952	
		35	13	22	1				13.0857	12.1513		41.6101	
24	7	35	13	22	1	12	5	5.6234	13.0705	12.1492	10.9354	41.7784	
		37	13	24	1				13.2817	12.2212		42.0616	
24	7	37	13	24	1	11	5	5.6318	13.2807	12.1918	10.724	41.8283	
		35	13	22	1				13.0694	12.1063		41.5314	
24	7	35	13	22	1	12	5	5.6235	13.0704	12.1491	10.9353	41.7783	
		37	13	24	1				13.2816	12.2211		42.0616	
24	7	37	13	24	1	11	5	5.6318	13.2807	12.1918	10.724	41.8283	
		35	13	22	1				13.0694	12.1063		41.5314	
													C.S.L
24	7	37	13	24	1	12	5	5.6281	13.2817	12.194	10.7338	41.8376	94.34%
24	7	35	13	22	1	11	5	5.6268	13.0694	12.1462	10.9535	41.7959	93.01%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table 4.4: Convergence path of the optimization procedure with single steps (medium production rate)

cost of the input buffer is 3.0495 and the subsystem cost of the output buffer is 12.8896, etc. The initial expected total system cost is 48.6952 and the customer service level at the retailer is 89.71%. Following rows show the revised values as the optimization algorithm passes through the subsystems. In particular, the second row shows the updated values during the first forward iteration and their respective subsystem costs. As the optimization procedure passes through the first subsystem, $\Omega(I)$, (Q_I, R_I) is updated to $(12, 9)$ in the input buffer, and (R, r) is updated to $(31, 11)$ in the output buffer, etc. The respective subsystem cost of the input buffer is 2.9038 and the subsystem cost of the output buffer is 12.9424, etc. The resulting expected total system cost is 43.1367. Note that, the procedure utilized a single step towards the optimal solution through the forward iteration. The third row shows the updated values during the first backward iteration. As the iterative procedure passes through the third subsystem, $\Omega(DC)$, (Q_{DC}, R_{DC}) is updated to $(22, 12)$ in the DC, and (R, r) is updated to $(35, 13)$ in the output buffer. The respective subsystem cost of the DC is 16.3958 and the subsystem cost of the output buffer is 13.0352. The resulting expected total system cost is 43.9718. Note that, the procedure utilized the single step optimization procedure through the forward iteration and optimized step procedure all the way to the optimal solution through the backward iteration. Other rows are interpreted accordingly.

As can be seen from Table 4.3, the expected total system cost has been decreased by almost 18% and the customer service level has been increased by more than 4%. A similar conclusion can be drawn from Table 4.4. We get the same results by using the optimized step search procedure 4.5 in a fewer number of steps. However, optimized step search has failed to converge in a number of examples concluding that the single step procedure, at the expense of more iterations, is a more robust method than the optimized step procedure. Additionally, we have observed much higher gain if we started the optimization procedure from other initial points 4.6. Also, for different production rates, we include the paths of convergence both using single step search as well as optimized step search in Appendix B. We include additional paths of convergence for varying demand levels.

Within the given input settings, our results show that the optimal Q_{DC} value is always a multiple of Q_R value. This is intuitive and consistent with existing results. If, on the other hand, the reverse is true, the DC will carry unnecessary inventories resulting in excessive

				I.B.	O.B.	DC	Retailer
	μ_1 = 2	β_S = 1	***K***	30	25	20	15
λ=1.5	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	6.5111	12.8839	17.4397	11.8606	48.6952	89.71%
24	7	33	13	20	11	11	5	5.6266	12.8952	15.5682	11.294	45.3841	
		35	13	22	12				13.0677	16.4189		46.4072	
24	7	35	13	22	12	12	5	5.6308	13.0681	16.4225	10.9325	46.0538	
		37	13	24	1				13.2795	12.2162		42.0589	
24	7	37	13	24	1	11	5	5.6318	13.2807	12.1918	10.724	41.8283	
		35	13	22	1				13.0694	12.1063		41.5314	
24	7	35	13	22	1	12	5	5.6235	13.0704	12.1491	10.9353	41.7783	
		37	13	24	1				13.2816	12.2211		42.0616	
24	7	37	13	24	1	11	5	5.6318	13.2806	12.1918	10.724	41.8282	
		35	13	22	1				13.0694	12.1063		41.5314	
24	7	35	13	22	1	12	5	5.6235	13.0704	12.1491	10.9353	41.7783	
		37	13	24	1				13.2817	12.2211		42.0616	
24	7	37	13	24	1	11	5	5.6318	13.2807	12.1918	10.724	41.8283	
		35	13	22	1				13.0694	12.1063		41.5314	
													C.S.L
24	7	37	13	24	1	12	5	5.6281	13.2817	12.194	10.7338	41.8376	94.34%
24	7	35	13	22	1	11	5	5.6268	13.0694	12.1462	10.9535	41.7959	93.01%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table 4.5: Convergence path of the optimization procedure with optimized steps (medium production rate)

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 2$	$\beta_S = 1$	***K***	30	25	20	15
λ=1.5	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	a = 0.1	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		**O.B.**		**DC**		**Retailer**				**Cost**			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	**I.B.**	**O.B.**	**DC**	**Retailer**	**TC**	**C.S.L**
22	20	15	12	10	15	5	20	7.7476	23.7671	39.2231	137.6181	208.3559	0.00%
23	7	25	16	10	16	6	21	5.6107	13.5541	19.2399	42.8631	81.2678	
		27	16	12	14				13.1224	17.2453		78.8416	
23	7	27	16	12	13	7	20	5.6195	13.1248	16.9384	20.0623	55.7451	
		28	15	14	11				12.9241	16.4639		55.07	
24	7	28	15	14	8	8	19	5.6233	12.911	15.3226	18.5828	52.4397	
		30	15	16	9				12.847	14.7016		51.7547	
24	7	30	15	16	9	7	18	5.626	12.8504	14.7007	17.6729	50.85	
		28	14	14	8				12.7086	15.2882		51.2957	
24	7	28	14	14	8	8	17	5.6143	12.7115	15.352	17.3769	51.0546	
		30	14	16	9				12.6969	14.7198		50.4079	
24	7	30	14	16	9	8	16	5.6259	12.6942	14.7008	16.4023	49.4232	
		30	14	16	9				12.6942	14.7008		49.4232	
24	7	30	14	16	9	8	15	5.6212	12.6969	14.7087	15.784	48.8108	
		30	14	16	9				12.6969	14.7087		48.8108	
24	7	30	14	16	9	8	14	5.6212	12.6969	14.7088	15.2029	48.2298	
		30	14	16	9				12.6969	14.7088		48.2298	
24	7	30	14	16	9	9	13	5.6212	12.6969	14.7088	14.6175	47.6444	
		32	14	18	10				12.7686	14.8879		47.8952	
24	7	32	14	18	10	9	12	5.6279	12.7686	14.8786	13.8527	47.1278	
		32	14	18	10				12.7686	14.8786		47.1278	
24	7	32	14	18	10	9	11	5.6235	12.7706	14.8824	13.2955	46.572	
		32	14	18	10				12.7706	14.8824		46.572	
24	7	32	14	18	10	9	10	5.6235	12.7706	14.8824	12.8041	46.0806	
		32	14	18	10				12.7706	14.8824		46.0806	
24	7	32	14	18	10	10	9	5.6235	12.7706	14.8824	12.3552	45.6317	
		33	13	20	11				12.8941	15.5159		46.3887	
24	7	33	13	20	11	10	8	5.6295	12.8941	15.5668	11.7954	45.8858	
		33	13	20	11				12.8941	15.5668		45.8858	
24	7	33	13	20	11	11	7	5.6253	12.8957	15.5688	11.4978	45.5877	
		35	13	22	12				13.0681	16.4192		46.6105	
24	7	35	13	22	12	11	6	5.6308	13.0681	16.4225	11.0076	46.129	
		35	13	22	12				13.0681	16.4225		46.129	
24	7	35	13	22	12	12	5	5.6269	13.0694	16.4233	10.934	46.0536	
		37	13	24	10				13.2807	17.1542		46.9957	
24	7	37	13	24	1	11	5	5.6318	13.2806	12.1918	10.724	41.8282	
		35	13	22	1				13.0694	12.1063		41.5314	
24	7	35	13	22	1	12	5	5.6235	13.0704	12.1491	10.9353	41.7783	
		37	13	24	1				13.2816	12.2211		42.0616	
													C.S.L
24	7	37	13	24	1	12	5	5.6281	13.2817	12.194	10.7338	41.8376	94.34%
24	7	35	13	22	1	11	5	5.6268	13.0694	12.1462	10.9535	41.7959	93.01%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table 4.6: Convergence path of the optimization procedure with optimized steps (medium production rate)

holding costs. In addition, the optimal R_{DC} value is always close to zero. Consequently, it is suggested to operate the DC with low safety stock levels resulting in lower operational costs.

Similarly, our results indicate that $Q = R - r$ is always a multiple of Q_{DC}. At first glance, this may be surprising. However, while the production set-up cost tries to increase the value of $Q = R - r$, holding cost tries to reduce it. A trade-off is achieved when $Q = R - r$ is a multiple of Q_{DC}. If $Q = R - r$ is not a multiple of Q_{DC}, the output buffer will incur unnecessary holding costs.

We have observed that parameters R_R and r mainly depend on local cost parameters. For instance, R_R decreases as the holding cost at the retailer increases, and increases as the backorder and shortage cost at the retailer increases. The value of r shows a similar behavior. Although the above results guide planning and operational issues, further modeling, analysis and numerical studies required to set-up solid rules of thumb. For instance it is possible to restructure the optimization procedure to exploit the above observations.

4.4 Impact of Cost on System Parameters

In this section, we examine the attributes that drive the overall performance of the supply chain. The attributes are mainly the cost parameters, that is, set-up, holding, backorder, and shortage costs in different echelons.

Figure 4.1 shows the impact of cost parameters on the overall system performance, that is, on customer service level at the retailer. In the upper left part of the graph, the impact of input buffer's ordering cost on the customer service level can be seen. As the input buffer's ordering cost increases, the customer service level decreases. This is because higher ordering cost results in larger order quantities and lower reorder points in the input buffer, which in turn results in lower customer service level at the retailer. In the upper right part of the graph, the impact of plant's shortage cost on the customer service level can be seen. The graph indicates that increasing shortage cost has a positive effect on the customer service level. In fact, this is true for the other echelon's shortage cost as well. Interestingly, there is no effect of DC's backorder cost on the customer service level (middle left part of the graph). Even, we see that the DC's cost values only effect its own operating characteristics.

An expected impact of retailer's holding cost on the customer service level can be seen in the middle right part of the graph. Finally, it is seen that as the demand rate increases, the customer service level decreases (lower part of the graph). Appendix B includes the impact of individual echelons' cost parameters on the overall system performance.

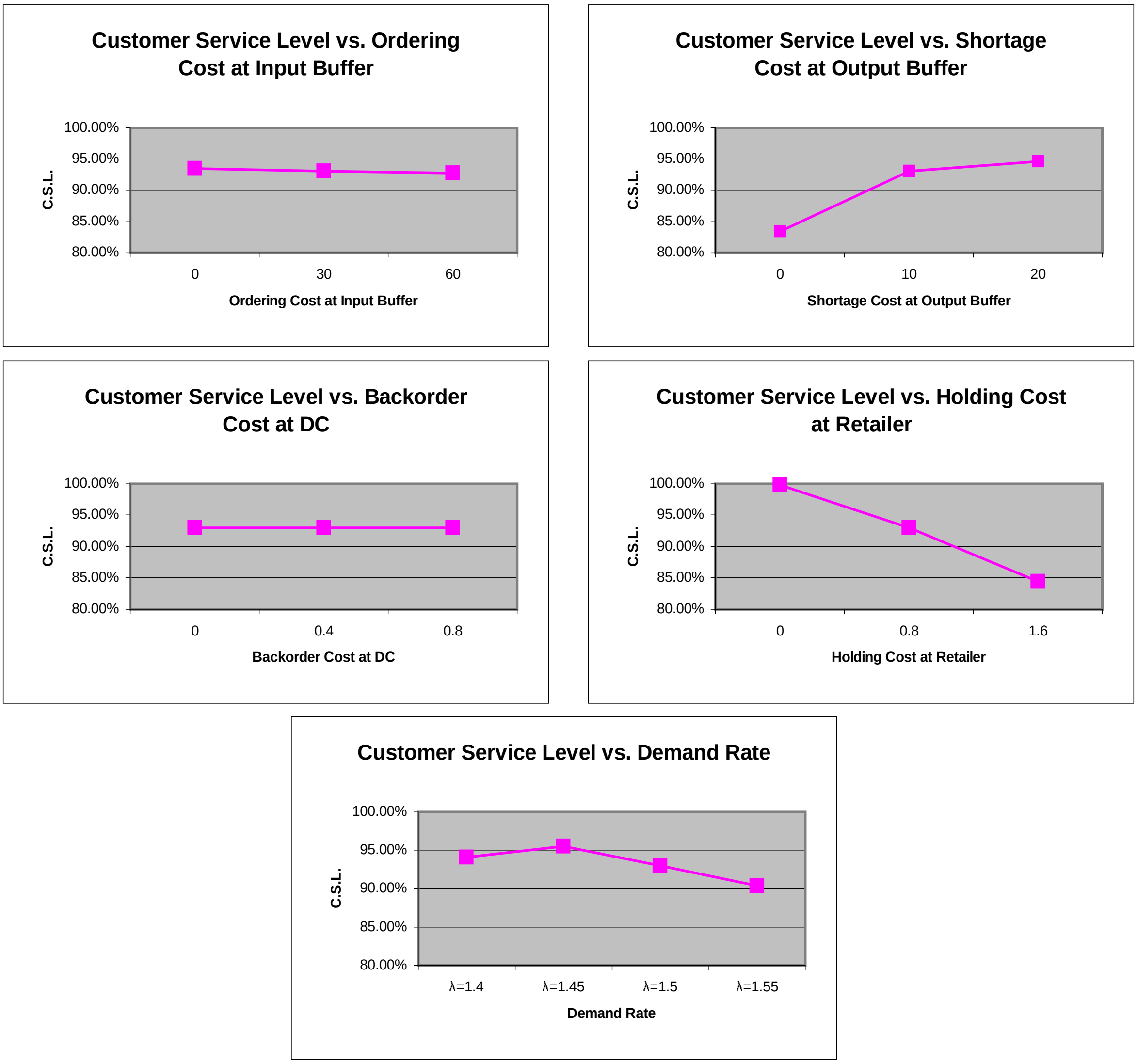


Figure 4.1: Attributes that drive the overall performance of the supply chain

Chapter 5
Analysis of Multi-Echelon Distribution Inventory Systems

In this chapter, we consider a distribution inventory system with one warehouse (W) and N retailers, as illustrated in Figure 5.1. The retailers face independent, stationary unit Poisson demand and have their own operating characteristics. They follow a continuous review (R, Q) inventory control policies, that is, when the inventory position (inventory on hand plus outstanding orders minus backorders) at a retailer down-crosses R, it orders a replenishment batch size of Q from the central warehouse. The order arrives after a transportation lead-time delay, if the warehouse has sufficient on-hand inventory. Otherwise, it experiences additional delays due to stockouts at the warehouse. Any excess demand at a retailer is backlogged and filled as soon as the replenishment orders arrive in a first-in first-out manner.

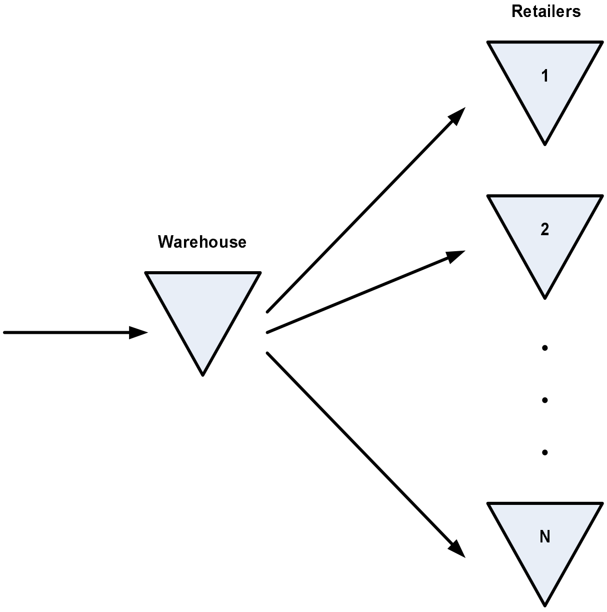


Figure 5.1: A two-echelon distribution inventory system

We assume that it is possible to have several outstanding backorders at a retailer at any point in time. The effective lead-time between the warehouse and a retailer is the time between the placement and receipt of the order by the retailer. This includes the transportation lead-time as well as the delay in the warehouse due to the stockouts.

Demand at the warehouse are orders from the retailers and satisfied immediately if there is available stock on-hand. The unsatisfied demand is backordered. The warehouse, in turn, orders from an outside supplier with infinite inventories based on an (R, Q) inventory control policy. The effective lead-time includes only the transportation time.

We assume that all replenishment batch quantities are multiples of a batch size q. In addition, we assume that all transportation times between facilities are phase-type distributed. In fact, we assume that the units are processed sequentially in the transportation system. In other words, no overtaking is possible and orders are received in the same order they were placed. In contrast, assuming independent, identically distributed random variables represents parallel processing of replenishment orders and allows orders to cross in time. Zipkin [108], and Svoronos and Zipkin [97] utilize same concept of transportation times. We assume, in particular, all transportation times follow a k'th order Erlang distribution. Erlang distribution is a special case of phase-type distributions. See Appendix A for a brief introduction to the phase-type distributions.

Performance evaluation of the system above is quite difficult because of the underlying complexities and large state-space. Indeed, we next present a decomposition procedure, which uses single-location models as building blocks to analyze the entire distribution system. The performance measures of interest are the long-run average number of inventories, the number of backorders, and the customer service levels in each facility.

Note that, the distribution system with one-for-one replenishment policies is a special case of this system and easily solved, since the demand process at the warehouse is a superposition of N independent Poisson processes and still a Poisson process. General solution procedures for this system are given in [31, 97]. On the other hand, the distribution system with (R, Q) inventory control policies is quite difficult to solve, because the demand process at the warehouse is a superposition of N Erlang processes. In the following sections, we also present a characterization of the demand process at the warehouse.

5.1 Modeling Approach

It is possible that the entire system can be modeled using a Markovian approach. However, it is easily seen that exact analysis of the above system is computationally impractical due to the fast growing state space of the underlying Markov chain. Hence, the only viable approach, other than simulation, is approximation. Widely used approximation techniques decomposes the system into several subsystems, which can be analyzed in isolation. Then, the subsystems are linked to each other. Here, we will implement a similar procedure.

Let us consider the distribution inventory system shown in Figure 5.1. We will decompose the system in such a way that each subsystem consists of an inventory holding buffer with its own stock keeping policy. Consequently, we treat each subsystem as a single-location inventory system, which can be analyzed with a modest computational effort. Finally, we relate the subsystems to each other. In summary, it includes constructing each subsystem, deriving input parameters to link the subsystems to each other. Now, let us introduce the following notation:

N number of retailers,

Q_W batch size at warehouse,

R_W reorder level at warehouse,

λ_i : demand rate at retailer i, $i = 1, 2, \ldots, N$,

Q_i batch size at retailer i, $i = 1, 2, \ldots, N$,

R_i reorder level at retailer i, $i = 1, 2, \ldots, N$,

q largest common factor of $Q_W, Q_1, Q_2, \ldots, Q_N$,

TT_W : transportation time between supplier and warehouse,

TT_i : transportation time between warehouse and retailer i, $i = 1, 2, \ldots, N$,

$\Omega(W)$: subsystem involving warehouse,

$\Omega(i)$: subsystem involving retailer i, $i = 1, 2, \ldots, N$,

$M_j^{'}$: node modeling procurement to facility j, $j = W, 1, 2, \ldots, N$,

$M_j^{''}$: node modeling demand arrival process to facility j, $j = W, 1, 2, \ldots, N$,

N_j : inventory level in $\Omega(j)$, $j = W, 1, 2, \ldots, N$.

We develop a decomposition as shown in Figure 5.2. The first subsystem, $\Omega(W)$, includes the warehouse in the distribution system. An (R_W, Q_W) inventory control policy is used to control replenishment process at the warehouse. Node M_W' models the effective procurement process and M_W'' models the effective demand inter-arrival process at the warehouse. Similarly, the subsystems, $\Omega(i)$, include retailer i, $i = 1, 2, \ldots, N$. An (R_i, Q_i) policy is used to control inventory level. Node M_i' represents the procurement process and node M_i'' represents the demand arrival process. In the following sections, we explain how we construct the nodes M_j' and M_j'''s and their respective processing times U_j' and U_j'''s for $j = W, 1, 2, \ldots, N$.

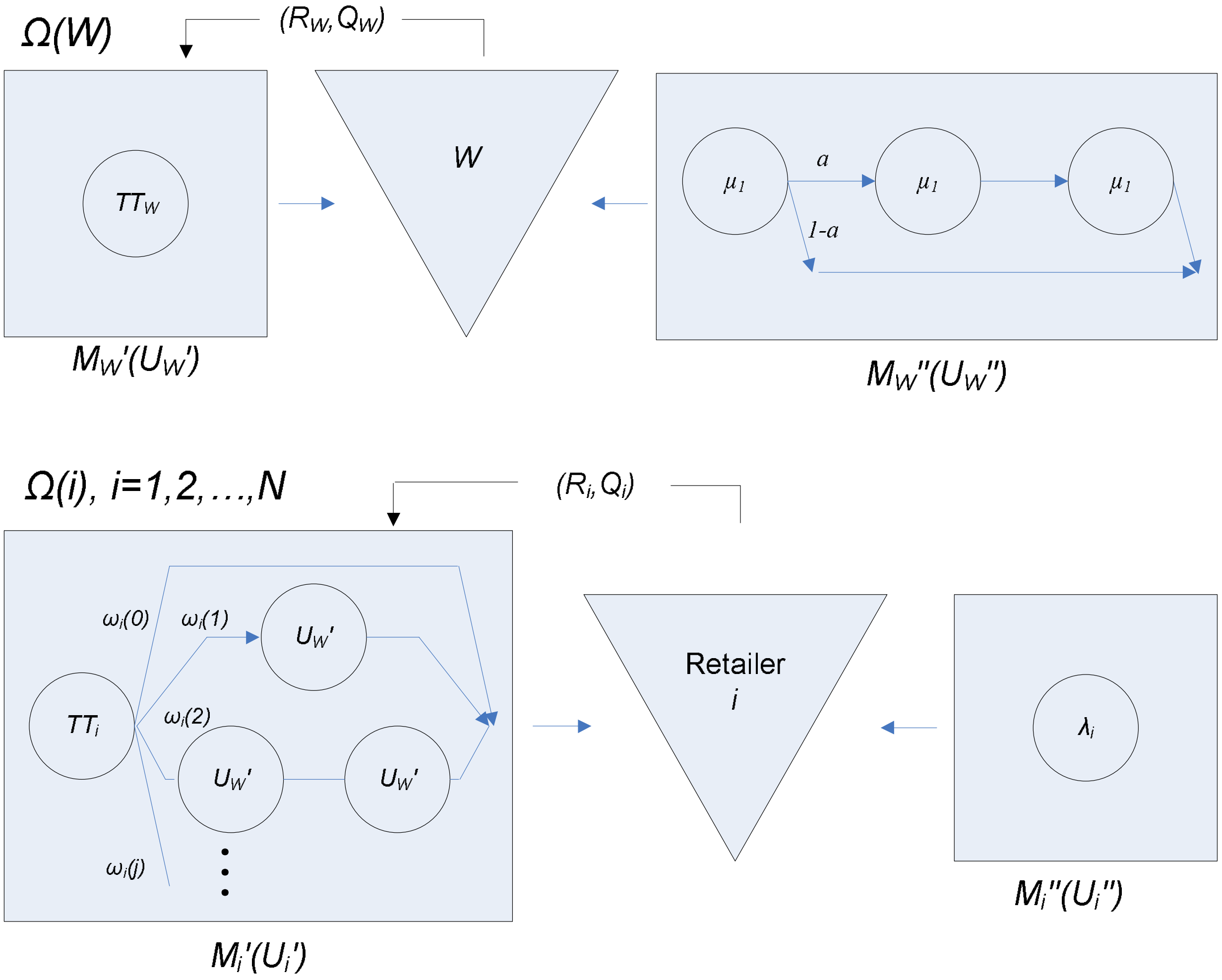


Figure 5.2: Subsystems $\Omega(W)$ and $\Omega(i)$, $i = 1, 2, \ldots, N$

5.1.1 Analysis of Procurement Times

In this section, we analyze the effective procurement times at each subsystem. For subsystem $\Omega(W)$, the random variable U_W' represents the effective procurement time at the warehouse.

Since, the supplier has always sufficient raw material to replenish the warehouse, the effective procurement time consists only of the transportation lead-time from supplier to the warehouse. That is,

$$U'_W = TT_W.$$

For subsystems $\Omega(i)$, the random variable U'_i represents the effective procurement time at retailer i, $i = 1, 2, \ldots, N$. The retailer order is filled as soon as it is received, if the warehouse has sufficient stock on hand. Otherwise, it is delayed until sufficient number of units arrive in the warehouse. Let $\omega_i(0)$ be the conditional probability that there are no units missing in the warehouse at the point a replenishment order arrives from retailer i. Similarly, let $\omega_i(j)$, $j = 1, 2, \ldots$ be the conditional probability that there are, for any j, $(j-1) * Q_W + 1, (j-1) * Q_W + 2, \ldots, j * Q_W$ units missing in the warehouse at the point a demand arrives from the retailer. Then, the effective lead time to the retailer is given by:

$$U'_i = \begin{cases} TT_i & \text{w.p.} \quad \omega_i(0), \\ TT_i + \sum_{k=1}^{j} U'_W & \text{w.p.} \quad \omega_i(j). \end{cases}$$

It is clear that, with probability $\omega_i(0)$, there is enough stock at the warehouse and retailer i's order experiences no delays. On the other hand, with probability $\omega_i(j)$, the warehouse does not have sufficient inventories resulting in delay in the replenishment process. This delay, however, is approximately j procurement lead times from the supplier to the warehouse.

5.1.2 Analysis of Demand Inter-Arrival Times

In this section, we analyze the effective demand inter-arrival times at each subsystem. The retailers face customer demand according to a Poisson process with rate λ_i, $i = 1, 2, \ldots, N$. Equivalently, the effective demand inter-arrival times at retailer i are independent and follow an exponential distribution with rate λ_i, $i = 1, 2, \ldots, N$.

Demand to the DC are replenishment orders from the retailers. Since the retailers replenish their stock according to an (R, Q) policy, the inter-arrival time of the orders from the retailers follow an Erlang distribution. As a result, the demand arrival process at the warehouse is a superposition of N independent Erlang processes.

5.1.3 Superposition of Erlang Processes

In this section, we consider an arrival process that is the superposition of N independent Erlang processes. Such an arrival process arises as the stream of replenishment orders in a distribution inventory system. The inventory system consists of many retailers replenishing their stock from a central warehouse where the retailers face independent, stationary Poisson demand and follow a continuous review (R, Q) inventory control policy. Another example is a queue to which the arrival process is the superposition of separate arrival streams, each of whose inter-arrival times is of Erlang distribution.

An important characteristic of the superposed process is that although the individual processes are independent from each other, the superposed process may be no longer independent. Here, we present a methodology to characterize such arrival streams as Markovian processes. We, then, extend the methodology to phase-type arrival streams as well. Our methodology exactly describes the superposed process, however the state-space of the proposed Markovian process increases considerably. We, in addition, develop a three-moment approximation scheme to efficiently use the methodology in practice. We illustrate the accuracy of the methodology in a number of test problems.

Preliminaries

In this section, we give some definitions and theorems that are repeatedly used in the sequel. A k-phase Erlang (Erlang-k) distribution is the sum of k exponential random variables. A phase diagram of the Erlang-k distribution with rate λ is shown in Figure 5.3. The Erlang-k distribution has also the following (α, T) phase-type distribution representation:

$$\alpha^T = (1, 0, \ldots, 0), \quad T = \begin{bmatrix} -\lambda & \lambda & & & \\ & -\lambda & \lambda & & \\ & & -\lambda & \ddots & \\ & & & \ddots & \lambda \\ & & & & -\lambda \end{bmatrix}.$$

An important property of the Erlang-k distribution is that the residual, or remaining time has a mixture of generalized Erlang-k (*MGE-k*) distribution. This is due to the following

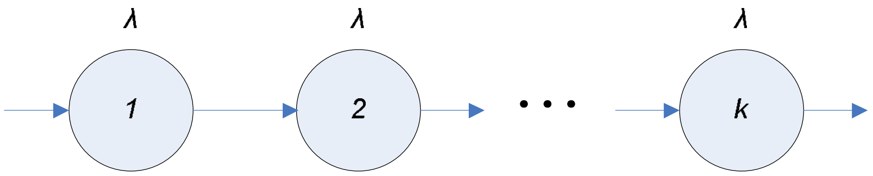


Figure 5.3: Phase diagram of the Erlang-k distribution

arguments. At any point in time, the Erlang-k distribution, with probability $1/k$, is in any one of its exponential phases. Hence, the residual time has one exponential phase with probability $1/k$. Similarly, the residual time has two exponential phases with probability $1/k$, and so on. The resulting *MGE-k* distribution has a graphical representation shown in Figure 5.4 with corresponding probabilities. The *MGE-k* distribution has also the following (α, T^*) phase-type distribution representation:

$$\alpha^T = (1, 0, \ldots, 0), \quad T^* = \begin{bmatrix} -\lambda & \frac{k-1}{k}\lambda & & & \\ & -\lambda & \frac{k-2}{k-1}\lambda & & \\ & & -\lambda & \ddots & \\ & & & \ddots & \frac{1}{2}\lambda \\ & & & & -\lambda \end{bmatrix}.$$

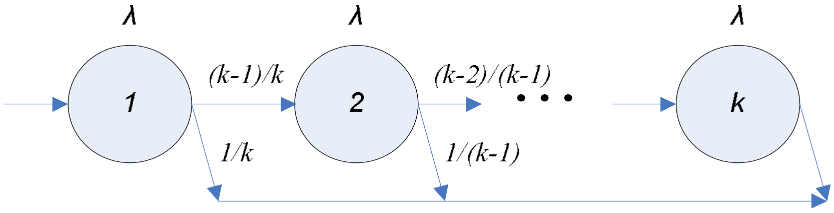


Figure 5.4: Phase diagram of remaining time of an Erlang-k distribution

We borrow the following definition and theorems from Neuts [86].

Definition 5.1 *If L and M are rectangular matrices of dimensions $k_1 \times k_2$ and $k_1' \times k_2'$, their Kronecker product $L \otimes M$ is the matrix of dimensions $k_1k_1' \times k_2k_2'$, written in block-partitioned form as*

$$\begin{bmatrix} L_{11}M & L_{12}M & \ldots & L_{1k_2}M \\ \vdots & \vdots & & \vdots \\ L_{k_11}M & L_{k_12}M & \ldots & L_{k_1k_2}M \end{bmatrix}.$$

If X and Y are independent random variables with phase-type distributions $F(\cdot)$ and $G(\cdot)$, then the distribution $H = 1 - [1 - F(\cdot)][1 - G(\cdot)]$, corresponding to $\min(X, Y)$, is also phase-type.

Theorem 5.1 *Let $F(\cdot)$ and $G(\cdot)$ have representations (α, T) and (β, S) of orders m and n respectively, then $H(\cdot)$ has the representation $[\alpha \otimes \beta, T \otimes I + I \otimes S]$.*

Theorem 5.2 *A finite mixture of phase-type distributions is a phase-type distribution. If $(p_1, \ldots, p_k)$ is the mixing density and $F_j(\cdot)$ has representation $[\alpha(j), T(j)]$, $1 \le j \le k$, then the mixture has the representation $\alpha = [p_1\alpha(1), \ldots, p_k\alpha(k)]$, and*

$$T = \begin{bmatrix} T(1) & 0 & \ldots & 0 \\ 0 & T(2) & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & T(k) \end{bmatrix}.$$

Superposition of Two Erlang Processes

We start from the simplest case, an arrival process that is the superposition of two independent Erlang processes. We characterize the arrival stream of two and then generalize to N independent Erlang processes. Let us denote by $F(\cdot)$ and $G(\cdot)$ two Erlang distributions with representations (α, T) and (β, S) of orders m and n, respectively.

Consider the superposed process at an arrival instance, that is, an instance at which an arrival just happened. Without loss of generality, let us assume that the arrival is from the first process. The amount of time for the next arrival from the first process follows an Erlang-m distribution. On the other hand, the amount of time for the next arrival from the second process follows an *MGE-n* distribution, since the remaining time of an Erlang-n distribution is an *MGE-n* distribution. In fact, the amount of time for the next arrival is distributed as the minimum of Erlang-m and *MGE-n* distributions. From Theorem 5.1, this distribution has a representation $[\alpha \otimes \beta, T \otimes I + I \otimes S^*]$ where (β, S^*) is the corresponding representation of *MGE-n* distribution. In a similar vein, if we assume that the arrival is from the second process, the amount of time for the next arrival from the first process follows an *MGE-m* distribution, and the amount of time for the next arrival from the second process follows an Erlang-n distribution. Consequently, the amount of time for the next arrival is distributed as the minimum of *MGE-m* and Erlang-n distributions, and has a representation $[\alpha \otimes \beta, T^* \otimes I + I \otimes S]$ where (α, T^*) is the corresponding representation of *MGE-m* distribution.

If we denote by $p(1)$ the probability of an arrival from the first stream and by $p(2)$ the probability of an arrival from the second stream, the superposed process is going to be a mixture of phase-type distributions. By Theorem 5.2, it is again a phase-type distribution with corresponding representation, $\alpha = [p(1)(\alpha \otimes \beta), p(2)(\alpha \otimes \beta)]$, and

$$T = \begin{bmatrix} T \otimes I + I \otimes S^* & \underline{0} \\ \underline{0} & T^* \otimes I + I \otimes S \end{bmatrix}.$$

Example 5.1 *Consider an arrival process that is the superposition of an Erlang-2 and an Erlang-3 processes with respective rates λ_1 and λ_2. Let us denote by $F(\cdot)$ and $G(\cdot)$ the respective phase-type distributions with representations (α, T) and (β, S) of orders 2 and 3, respectively. The (α, T) and (β, S) are given as $\alpha = (1, 0)$, $\beta = (1, 0, 0)$, and*

$$T = \begin{bmatrix} -\lambda_1 & \lambda_1 \\ 0 & -\lambda_1 \end{bmatrix}, \quad S = \begin{bmatrix} -\lambda_2 & \lambda_2 & 0 \\ 0 & -\lambda_2 & \lambda_2 \\ 0 & 0 & -\lambda_2 \end{bmatrix}.$$

Let us assume that an arrival just happened from the first process. The amount of time for the next arrival from the first process follows an Erlang-2 distribution with (α, T) representation. On the other hand, the amount of time for the next arrival from the second process follows an MGE-3 distribution with the following (β, S^*), $\beta = (1, 0, 0)$, and

$$S^* = \begin{bmatrix} -\lambda_2 & \frac{2}{3}\lambda_2 & 0 \\ 0 & -\lambda_2 & \frac{1}{2}\lambda_2 \\ 0 & 0 & -\lambda_2 \end{bmatrix}.$$

In fact, the amount of time for the next arrival has a representation $\alpha(1) = (\alpha \otimes \beta) =$

$(1,0,0,0,0,0)$*, and* $T(1) = T \otimes I + I \otimes S^*$ *is given as*

$$\begin{bmatrix} -\lambda_1-\lambda_2 & \frac{2}{3}\lambda_2 & 0 & \lambda_1 & 0 & 0 \\ 0 & -\lambda_1-\lambda_2 & \frac{1}{2}\lambda_2 & 0 & \lambda_1 & 0 \\ 0 & 0 & -\lambda_1-\lambda_2 & 0 & 0 & \lambda_1 \\ 0 & 0 & 0 & -\lambda_1-\lambda_2 & \frac{2}{3}\lambda_2 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 & \frac{1}{2}\lambda_2 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 \end{bmatrix}.$$

*If we assume that the arrival is from the second process, the amount of time for the next arrival from the first process follows an MGE-*2 *distribution with the following* (α, T^*) *representation,* $\alpha = (1,0)$*, and*

$$T^* = \begin{bmatrix} -\lambda_1 & \frac{1}{2}\lambda_1 \\ 0 & \lambda_1 \end{bmatrix}.$$

*The amount of time for the next arrival from the second process follows an Erlang-*3 *distribution with* (β, S) *representation. Consequently, the amount of time for the next arrival has a representation* $\alpha(2) = (\alpha \otimes \beta) = (1,0,0,0,0,0)$*, and* $T(2) = T^* \otimes I + I \otimes S$ *is given as*

$$\begin{bmatrix} -\lambda_1-\lambda_2 & \lambda_2 & 0 & \frac{1}{2}\lambda_1 & 0 & 0 \\ 0 & -\lambda_1-\lambda_2 & \lambda_2 & 0 & \frac{1}{2}\lambda_1 & 0 \\ 0 & 0 & -\lambda_1-\lambda_2 & 0 & 0 & \frac{1}{2}\lambda_1 \\ 0 & 0 & 0 & -\lambda_1-\lambda_2 & \lambda_2 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 & \lambda_2 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 \end{bmatrix}.$$

It remains to calculate the probabilities $p(1)$ *and* $p(2)$*, the probabilities of an arrival from the first and second streams, respectively. The total rate of the superposed process is given by* $\frac{n\lambda_1+m\lambda_2}{mn}$*, and* $p(1)$*,* $p(2)$ *are given by* $p(1) = \frac{n\lambda_1}{n\lambda_1+m\lambda_2}$*,* $p(2) = \frac{m\lambda_2}{n\lambda_1+m\lambda_2}$*. The superposed process, By Theorem 5.2, has a* (α, T) *representation* $\alpha = [p(1),0,0,0,0,0,p(2),0,0,0,0,0]$*, and*

$$T = \begin{bmatrix} T(1) & \underline{0} \\ \underline{0} & T(2) \end{bmatrix}.$$

The moments of the superposed process is calculated by

$$E[X^n] = (-1)^n n!(\alpha^T T^{-n} e), \quad n \geq 1. \tag{5.1}$$

If $\lambda_1 = 1.5$ and $\lambda_2 = 1$, first three moments of the superposed process are $E[X^1] = 0.9231$, $E[X^2] = 1.3499$, and $E[X^3] = 2.6277$. In addition, its squared coefficient of variation is $Cv^2 = 0.5843$.

Superposition of N Erlang Processes

In this section, we generalize the methodology presented in the previous section to N independent Erlang processes. We first extend Theorem 5.1 to accommodate N phase-type distributions. If X_1, X_2, …, X_N are independent random variables with phase-type distributions $F_1(\cdot)$,$F_2(\cdot)$, …, $F_N(\cdot)$, then the distribution $H = 1-[1-F_1(\cdot)][1-F_2(\cdot)]\dots[1-F_N(\cdot)]$, corresponding to $\min(X_1, X_2 \dots X_N)$, is also phase-type.

Theorem 5.3 *Let $F_1(\cdot)$, $F_2(\cdot)$, …, $F_N(\cdot)$ have representations (α_1, T_1), (α_2, T_2), …, (α_N, T_N) of orders n_1, n_2, …, n_N, respectively. Then, $H(\cdot)$ has the representation $[\alpha_1 \otimes \alpha_2 \otimes \dots \otimes \alpha_N, T_1 \otimes I_2 \otimes \dots \otimes I_N + I_1 \otimes T_2 \otimes I_3 \otimes \dots \otimes I_N + \dots + I_1 \otimes I_2 \otimes \dots \otimes I_{N-1} \otimes T_N]$.*

Now, consider the superposed process at an arrival instance, that is, an instance at which an arrival just happened. Without loss of generality, let us assume that the arrival is from the first process. The amount of time for the next arrival from the first process follows an Erlang-n_1 distribution. On the other hand, the amount of time for the next arrival from the second process follows an *MGE-*n_2 distribution, the amount of time for the next arrival from the third process follows an *MGE-*n_3 distribution, and so on. In fact, the amount of time for the next arrival is distributed as the minimum of Erlang-n_1, *MGE-*n_2, …, *MGE-*n_N distributions. The distribution is defined by Theorem 5.3 and has the representation $[\alpha_1 \otimes \alpha_2 \otimes \dots \otimes \alpha_N, T_1 \otimes I_2 \otimes \dots \otimes I_N + I_1 \otimes T_2^* \otimes I_3 \otimes \dots \otimes I_N + \dots + I_1 \otimes I_2 \otimes \dots \otimes I_{N-1} \otimes T_N^*]$ where (α_i, T_i^*) is the corresponding representation of *MGE-*n_i distribution.

Similarly, if we assume that the arrival is from the second process, the amount of time for the next arrival from the first process follows an *MGE-*n_1 distribution, the amount of time for the next arrival from the second process follows an Erlang-n_2 distribution, the amount of time for the next arrival from the third process follows an *MGE-*n_3 distribution, and so on. Consequently, the amount of time for the next arrival is distributed as the minimum of *MGE-*n_1, Eralng-n_2, …, *MGE-*n_N distributions. The distribution is defined by Theorem 5.3

and has the representation $[\alpha_1 \otimes \alpha_2 \otimes \ldots \otimes \alpha_N, T_1^* \otimes I_2 \otimes \ldots \otimes I_N + I_1 \otimes T_2 \otimes I_3 \otimes \ldots \otimes I_N + \ldots + I_1 \otimes I_2 \otimes \ldots \otimes I_{N-1} \otimes T_N^*]$. We continue this analysis for the remaining streams as well.

If we denote by $p(1)$ the probability of an arrival from the first stream, by $p(2)$ the probability of an arrival from the second stream, and so on, the superposed process is going to be a mixture of phase-type distributions. By Theorem 5.2, it is again a phase-type distribution with corresponding representation, $\alpha = [p(1)\alpha(1) \otimes p(2)\alpha(2) \otimes \ldots \otimes p(N)\alpha(N)]$, and

$$T = \begin{bmatrix} T(1) & 0 & \ldots & 0 \\ 0 & T(2) & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & T(N) \end{bmatrix}.$$

Although the above methodology exactly characterizes the superposed process, it has limited practical utility because of the fast growing state-space. In the next sections, we present an approximation scheme to use it in practice.

Superposition of *MGE* Random Variables

We can easily extend the above methodology to *MGE* random variables. We just need to substitute (α, T) of an Erlang random variable appropriately with (α, T) of an *MGE* random variable. We only need to present a way to find residual time of an *MGE* random variable. Let X be an independent random variable with *MGE* distribution $F(\cdot)$. Let $F(\cdot)$ has (α, T) representation of order m. The long-run probabilities are defined as the limiting probabilities of being in a state at any point in time. Let us illustrate the residual time analysis in an example.

Example 5.2 *Consider the MGE-*2 *distribution illustrated in Figure 5.5. The long-run probabilities, limiting probabilities of being in state one or two, are obtained using*

$$\pi_1 = (1 - \lambda_1 a)\pi_1 + \lambda_2 \pi_2,$$
$$\pi_2 = \lambda_1 a \pi_1 + (1 - \lambda_2)\pi_2,$$
$$\pi_1 + \pi_2 = 1.$$

From these $\pi_1 = \lambda_2/(\lambda_2 + \lambda_1 a)$ *and* $\pi_2 = \lambda_1 a/(\lambda_2 + \lambda_1 a)$*. So, with probability* $\pi_1 = \lambda_2/(\lambda_2 +$

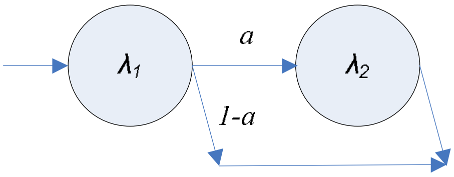


Figure 5.5: Phase diagram of an MGE-2 distribution

$\lambda_1 a$), the MGE-2 distribution is in phase 1 and the remaining time is same as the initial MGE-2 distribution. On the other hand, with probability $\pi_2 = \lambda_1 a/(\lambda_2 + \lambda_1 a)$, the process is in phase 2 and the remaining time includes only the second exponential phase. The resulting phase-type distribution is illustrated in Figure 5.6. The (α, T) representation of the residual time of an MGE-2 distribution is given by $\alpha = (\lambda_2/(\lambda_2 + \lambda_1 a), \lambda_1 a/(\lambda_2 + \lambda_1 a))$ and

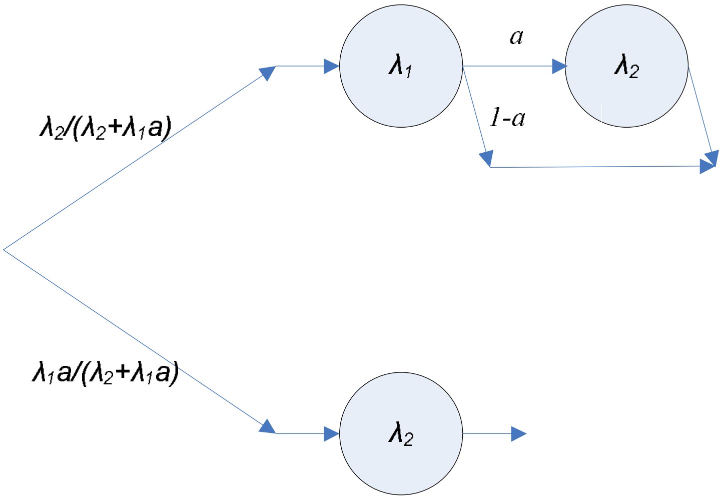


Figure 5.6: Phase diagram of remaining time of an MGE-2 distribution

$$T = \begin{bmatrix} -\lambda_1 & \lambda_1 a & 0 \\ 0 & -\lambda_2 & 0 \\ 0 & 0 & -\lambda_2 \end{bmatrix}.$$

We calculate the moments of the process by using Equation 5.1.

Approximating the Superposition Process

The methodology we described in the previous section exactly characterizes the superposed process. However, it has limited practical utility because of the fast growing state-space. In this section, we are mainly concerned with approximating the superposed process. We identify a subset of phase-type random variables that are used in the development of approximations. We facilitate both two-moment and three-moment approximation schemes in our procedures.

The idea of the approximation procedure is that we superpose individual arrival steams one by one avoiding the state-space getting larger. Initially, we superpose first two individual arrival streams. We approximate the resulting stream by using two-moment or three-moment approximation schemes. Then, we superpose the resulting arrival stream with the third arrival stream. Again, we use two-moments or three-moments to approximate the resulting process. We continue this way until all the arrival streams exhausted. In fact, we avoid state-space getting larger at the expense of loosing some degree of accuracy.

In general, the squared coefficient of variation, Cv^2, of the arrival process is less than one. This is due to the fact that individual superposed processes are Erlang distributions with low variability. We facilitate both two-moment and three-moment approximation schemes in our procedures. The two-moment approximation scheme is due to Altiok [5, 7] and the three-moment approximation scheme is due to Osogami and Harchol-Balter [87].

For $Cv^2 < 1$, two-moment approximation scheme in [7] utilizes the generalized Erlang distribution shown in Figure 5.7. Given the first moment of the superposed process, m_1, and the squared coefficient of variation, Cv^2, the number of phases, k, is determined from $1/k \leq Cv^2 \leq 1/(k-1)$, and the parameters a and λ are given respectively by

$$1 - a = \frac{2kc + k - 2 - \sqrt{k^2 + 4 - 4kc}}{2(c+1)(k-1)},$$

and

$$\lambda = \frac{1 + (k-1)a}{m_1}.$$

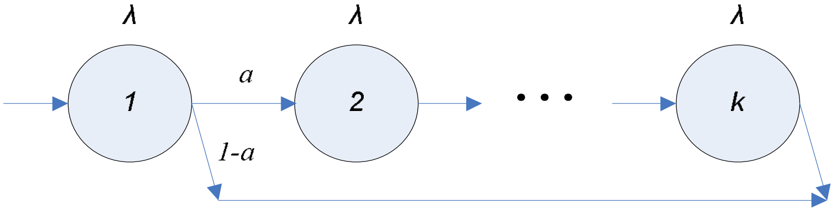


Figure 5.7: Phase diagram of generalized Erlang-k distribution

On the other hand, three-moment approximation scheme in [87] utilizes Erlang-Coxian

(EC) distributions and its variants shown in Figure 5.8. The EC distribution is simply an *MGE*-2 distribution appended to an Erlang distribution. It also allows positive probability to mass at point zero. EC distribution has six parameters to estimate and a closed-form solution is derived in [87]. Empirical studies suggest that using two-moments is sufficient for the domain $Cv^2 < 1$, though using three-moments captures the skewness of the distribution and brings more accuracy to the approximation.

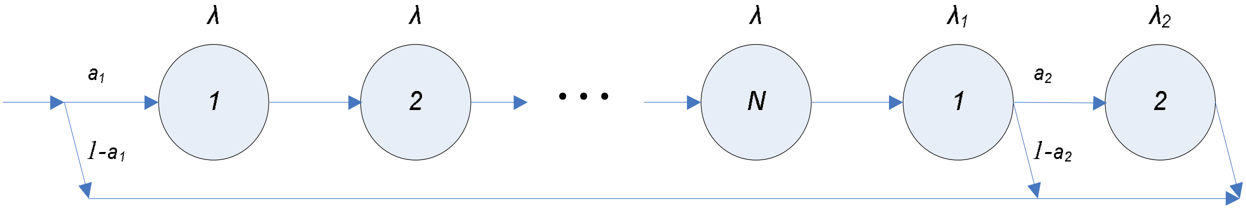


Figure 5.8: Phase diagram of Erlang-Coxian distribution

Let us illustrate the approximation concept in an example.

Example 5.3 *Consider a distribution inventory system with three identical retailers. The retailers face Poisson demand with rate one and follow a continuous review* $(R, Q) = (5, 10)$ *inventory control policy. Hence, the demand arrival process at the warehouse is a superposition of three Erlang-10 distributions with rate one. In order to characterize the demand arrival process at the warehouse, we superpose the first two Erlang-10 distributions, approximate it by using both two-moment and three-moment approximation schemes, and then superpose the resulting stream with the third Erlang-10 distribution.*

We superpose the first two arrival streams using the methodology given in the previous sections. The resulting process has first moment, $E[X] = 5$*, second moment,* $E[X^2] = 34.2334$*, third moment,* $E[X^3] = 273.5516$*, and the squared coefficient of variation,* $Cv^2 = 0.3693$*. We approximate it by using both two-moment and three-moment approximation schemes. Two-moment approximation scheme results in first moment,* $E[X] = 5$*, second moment,* $E[X^2] = 34.2334$*, third moment,* $E[X^3] = 294.5875$*, and the squared coefficient of variation,* $Cv^2 = 0.3693$*. On the other hand, three-moment approximation scheme results in first moment,* $E[X] = 5$*, second moment,* $E[X^2] = 34.2351$*, third moment,* $E[X^3] = 273.5927$*, and the squared coefficient of variation,* $Cv^2 = 0.3694$*.*

Then we superpose the approximate resulting stream with the third arrival process. The *final process, which used two-moment approximation scheme results in first moment,* $E[X] = 3.3314$*, second moment,* $E[X^2] = 16.4271$*, third moment,* $E[X^3] = 101.7050$*, and the squared coefficient of variation,* $Cv^2 = 0.4802$*. On the other hand, the final process, which used three-moment approximation scheme results in first moment,* $E[X] = 3.3334$*, second moment,* $E[X^2] = 16.8486$*, third moment,* $E[X^3] = 102.9705$*, and the squared coefficient of variation,* $Cv^2 = 0.5163$*.*

If we directly employ the methodology given in the previous sections to the three arrival streams, which requires significant computational effort, we get first moment of $E[X] = 3.3333$*, second moment of* $E[X^2] = 16.8363$*, third moment of* $E[X^3] = 103.8908$*, and the squared coefficient of variation,* $Cv^2 = 0.5153$*.*

We conclude that using three-moment approximation schemes results in more accuracy, especially when the number of superposed arrival streams increases.

5.1.4 Steady-State Analysis of the Subsystems

In this section, we calculate the steady-state probabilities of the underlying Markovian chains in the subsystems $\Omega(j)$, $j = W, 1, 2, \ldots, N$. Each of the subsystems, $\Omega(j)$, for $j = W, 1, 2, \ldots, N$, is a two-node subsystem with its own stock keeping policy, and phase-type procurement and demand inter-arrival times. The use of the phase-type random variables gives rise to a Markovian analysis, and matrix-recursive procedures based on [29, 64, 86] are used to obtain steady-state probabilities. For numerical convenience, we assume all transportation times follow a second order Erlang distribution (Erlang-2). Let β_j denote the phase rate of (Erlang-2) transportation time, $\beta_j^{'}$ denote the processing rate of $U_j^{'}$, and $\beta_j^{''}$ denote the processing rate of $U_j^{''}$, $j = W, 1, 2, \ldots, N$.

Analysis of Subsystem involving Warehouse

Let us start with the analysis of subsystem $\Omega(W)$, the subsystem involving the warehouse. Here, the effective procurement time has an Erlang-2 distribution (the phase rate of the (Erlang-2) is β_W). An important aspect of this subsystem is that the procurement orders and demand arrivals are both in batches. Still, we utilize the matrix recursive schemes to solve for the probabilities. Let $\{I_t, J_t, N_t, t \geq 0\}$ is a Markov chain where I_t represents the phase of $U_W^{'}$, J_t represents the phase of $U_W^{''}$, N_t denotes the number of inventories in the warehouse where $I_t = 1, 2, B$, $J_t = 1, 2$, and $N_t = Q_W + R_W, Q_W + R_W - q, Q_W + R_W - 2q \ldots$. The Markov chain has infinite number of states, and yet we truncate the state-space at a state with negligible holding probability. The state-space and the transitions of the Markov chain for one warehouse and two retailers (where $Q_W = 2q$, $Q_1 = 2q$, and $Q_2 = q$) are presented in Figure 5.9. Let the probabilities of the subsystem be:

$$\tilde{\mathbf{P}}(n)|_{n=1}^{2} = \begin{bmatrix} P(1, n, R_W - kq) \\ P(2, n, R_W - kq) \\ \vdots \\ P(1, n, R_W) \\ P(2, n, R_W) \\ P(B, n, R_W + q) \\ P(B, n, R_W + Q_W) \end{bmatrix}.$$

Here, k is a number that ensures a state with negligible holding probability. The flow-balance equations of the system in compact form are:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}(1) &= \mathbf{B}\tilde{\mathbf{P}}(3), \\ \mathbf{C}\tilde{\mathbf{P}}(2) &= a\mu\tilde{\mathbf{P}}(1), \\ \mathbf{C}\tilde{\mathbf{P}}(3) &= \mu\tilde{\mathbf{P}}(2). \end{aligned} \tag{5.2}$$

The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are given as:

$$\mathbf{A} = \begin{bmatrix} \mu_1 + \beta_W^{''} & 0 & -(1-a)\mu_1 p_1 & 0 & -(1-a)\mu_1 p_2 & 0 & \ldots \\ -\beta_W & \mu_1 + \beta_W^{''} & 0 & -(1-a)\mu_1 p_1 & 0 & -(1-a)\mu_1 p_2 & \ldots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & -\beta_W & \ldots & \mu_1 + \beta_W^{''} & 0 & -(1-a)\mu_1 p_1 & \ldots \\ 0 & 0 & \ldots & -\beta_W & \mu_1 + \beta_W^{''} & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & \mu_1 p_1 & 0 & \mu_1 p_2 & 0 & \ldots \\ 0 & 0 & 0 & \mu_1 p_1 & 0 & \mu_1 p_2 & \ldots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \end{bmatrix},$$

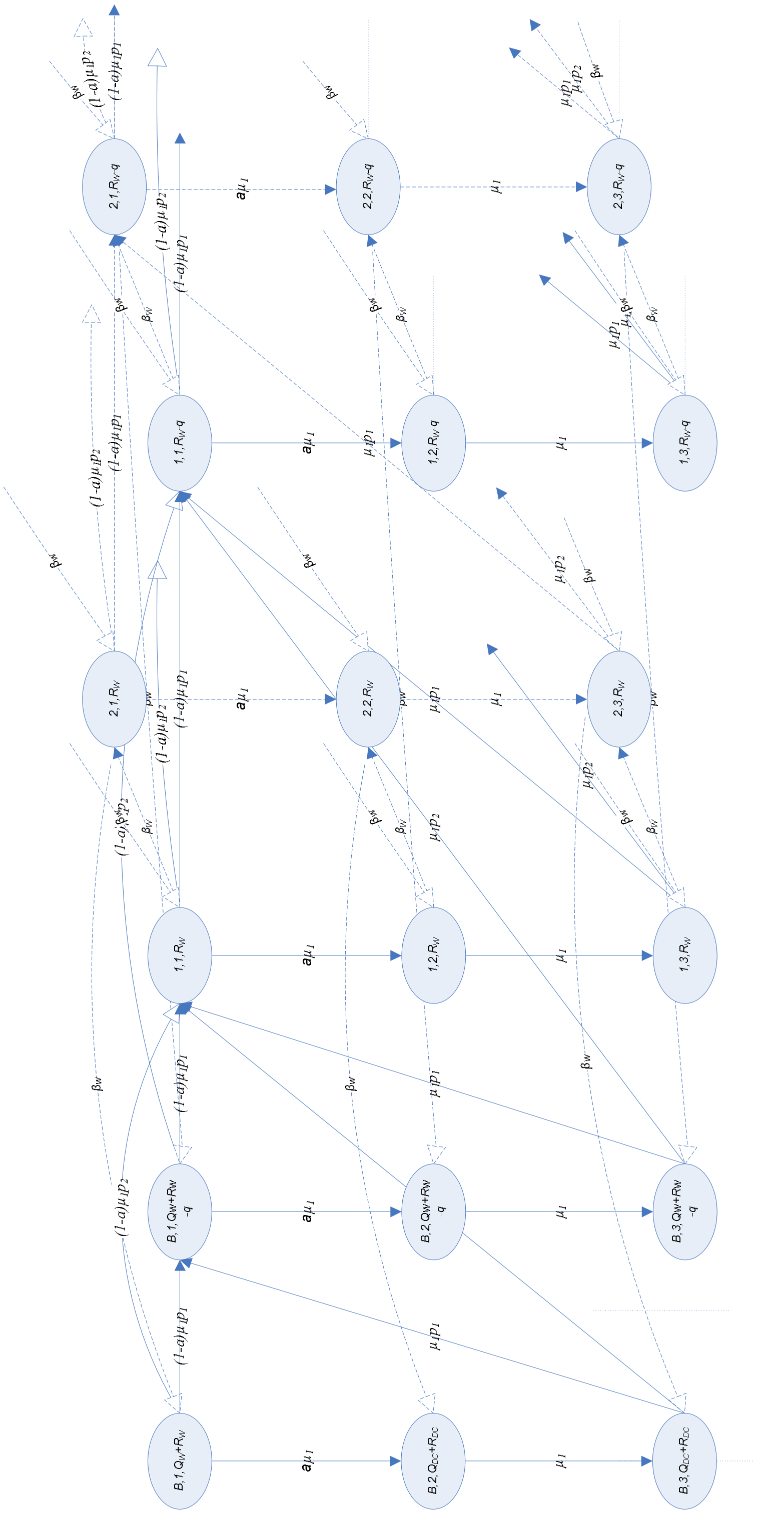


Figure 5.9: Transition diagram for subsystem $\Omega(W)$

$$\mathbf{C} = \begin{bmatrix} \mu_1 + \beta_W^{''} & 0 & 0 & 0 & 0 & 0 & \dots \\ -\beta_W & \mu_1 + \beta_W^{''} & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & -\beta_W & \dots & \mu_1 + \beta_W^{''} & 0 & 0 & \dots \\ 0 & 0 & \dots & -\beta_W & \mu_1 + \beta_W^{''} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix}.$$

After, representing $\tilde{\mathbf{P}}(2)$ and $\tilde{\mathbf{P}}(3)$ in terms of $\tilde{\mathbf{P}}(1)$, and utilizing Equation 5.2, we get $\mathbf{P} \times \tilde{\mathbf{P}}(1) = 0$ where

$$\mathbf{P} = \mathbf{A} - a\mu^2\mathbf{B}(\mathbf{C}^{-1})^2.$$

Finally, normalization is achieved by the equation

$$\mathbf{p} = \mathbf{e}_{(1\times t)} \times (\mathbf{I} + a\mu\mathbf{C}^{-1} + a\mu^2(\mathbf{C}^{-1})^2).$$

Replacing the first row of matrix $\mathbf{P}$ with row vector $\mathbf{p}$, we solve for

$$\mathbf{P} \times \tilde{\mathbf{P}}(1) = [1, 0, \dots, 0]_t^T.$$

Rest of the probabilities are given by:

$$\tilde{\mathbf{P}}(2) = a\mu\mathbf{C}^{-1}\tilde{\mathbf{P}}(1),$$
$$\tilde{\mathbf{P}}(3) = a\mu^2(\mathbf{C}^{-1})^2\tilde{\mathbf{P}}(1).$$

Analysis of Subsystems involving Retailers

The subsystems, $\Omega(i)$, $i = 1, 2, \dots, N$ model the behavior of the retailers where demand arrives in single units and according to a Poisson process, and the replenishment process takes place in batches. A queuing analogy of the above model is the system $M/PH^k/1$ where arrivals are from a Poisson process, and the service time distribution is of phase-type and in exact batches of k. Although general solution procedures for the above queuing system are given in [31], we will again use the matrix-recursive technique utilized in the previous subsystem. Typical approaches use the generating function of the steady-state distribution. Inverting this function to compute the probabilities may be problematic and may require more computational effort than our approach.

Let $\{I_t, N_t, t \geq 0\}$ be a Markov chain where I_t represents the phase of $U_i^{'}$, and N_t denotes the level of inventories at retailer i where $I_t = 1, 2, B$, and $N_t = Q_i + R_i, Q_i + R_i - 1, \dots$.

The effective procurement time has a complex phase structure. However, we use a three-moment *MGE-2* approximation (the parameters are γ_1, γ_2 and b). The Markov chain has infinite number of states, and we truncate the state-space for the sake of tractability. The state-space and the transitions of the Markov chain are presented in Figure 5.10. Let us denote the steady-state probabilities of the Markov chain by:

$$\tilde{\mathbf{P}}_w(n)|_{n=1}^{2} = \begin{bmatrix} P(n,R_i) \\ P(n,R_i-1) \\ \vdots \\ P(n,R_i-kQ_i) \end{bmatrix}_{(kQ_i+1)\times 1}, \quad \tilde{\mathbf{P}}(B) = \begin{bmatrix} P(B,R_i+Q_i) \\ \vdots \\ P(B,R_i+1) \end{bmatrix}_{Q_i\times 1}.$$

Then, the flow-balance equations are given below:

$$\begin{aligned} \mathbf{A}\tilde{\mathbf{P}}(B) &= \mathbf{B}\tilde{\mathbf{P}}_w(1) + \mathbf{C}\tilde{\mathbf{P}}_w(2), \\ \mathbf{D}\tilde{\mathbf{P}}_w(1) &= \mathbf{E}\tilde{\mathbf{P}}_w(2) + \mathbf{F}\tilde{\mathbf{P}}(B), \\ \mathbf{G}\tilde{\mathbf{P}}_w(2) &= \gamma_1 b\tilde{\mathbf{P}}_w(1), \end{aligned} \tag{5.3}$$

where

$$\mathbf{A} = \begin{bmatrix} \lambda & & & \\ -\lambda & \lambda & & \\ & & \ddots & \\ & & -\lambda & \lambda \end{bmatrix}_{Q_i\times Q_i}, \mathbf{B} = \begin{bmatrix} \gamma_1(1-b) & & & & \\ & \gamma_1(1-b) & & & \\ & & \ddots & & \dots \\ & & & \gamma_1(1-b) & \end{bmatrix}_{Q_i\times(kQ_i+1)},$$

$$\mathbf{C} = \begin{bmatrix} \gamma_2 & & & \\ & \gamma_2 & & \\ & & \ddots & \dots \\ & & \gamma_2 & \end{bmatrix}_{Q_i\times(kQ_i+1)}, \mathbf{D} = \begin{bmatrix} \lambda+\gamma_1 & \dots & -\gamma_1(1-b) & & & \\ -\lambda & \lambda+\gamma_1 & & & \ddots & \\ & & & \ddots & & \\ & & & & -\lambda & \lambda+\gamma_1 \\ & & & & & -\lambda & \gamma_1 \end{bmatrix},$$

$$\mathbf{E} = \begin{bmatrix} -\gamma_1(1-b) & \\ & \ddots \\ & \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \lambda+\gamma_2 & & & & \\ -\lambda & \lambda+\gamma_2 & & & \\ & & \ddots & & \\ & & -\lambda & \lambda+\gamma_2 & \\ & & & -\lambda & \gamma_2 \end{bmatrix},$$

and $\mathbf{D}$, $\mathbf{E}$, $\mathbf{G}$ are $(kQ_i+1)\times(kQ_i+1)$ matrices. Additionally,

$$\mathbf{F} = [f_{ij}] = \begin{cases} \lambda, & \text{if } i=1, j=Q_i \\ 0, & \text{otherwise,} \end{cases}$$

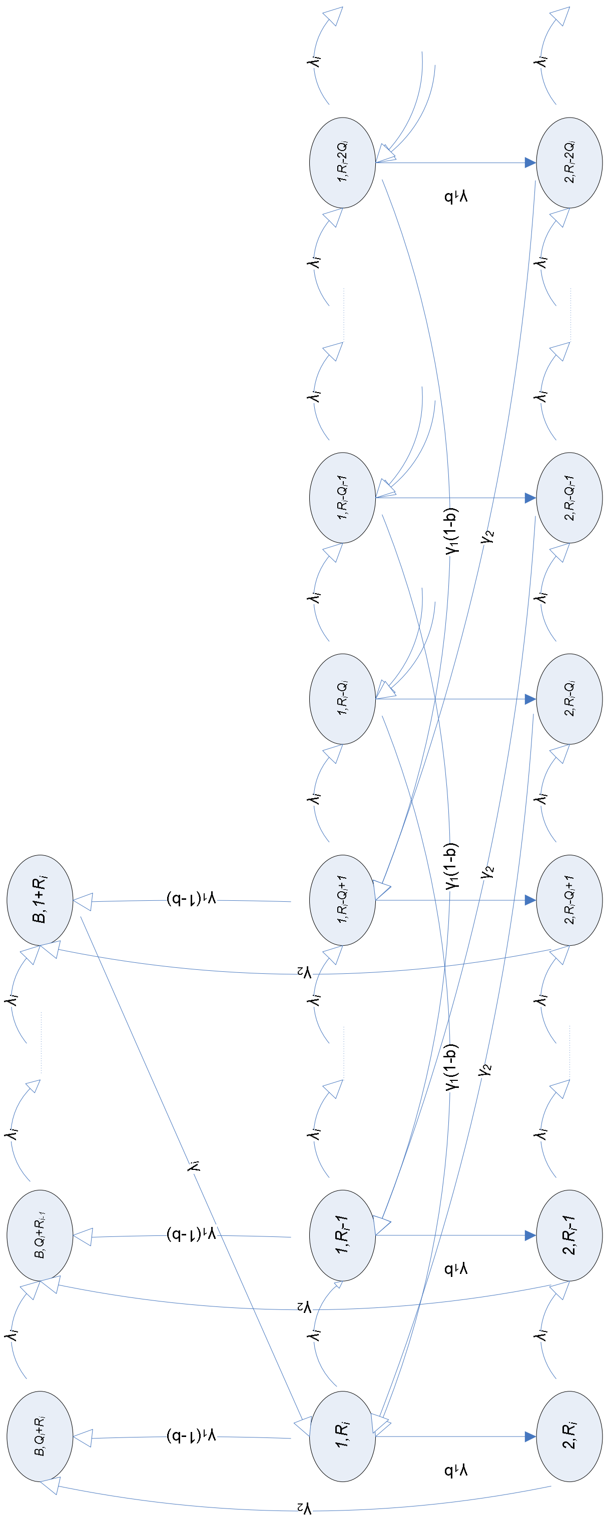


Figure 5.10: Transition diagram for subsystems $\Omega(i)$, $i = 1, 2, \ldots, N$

is a $(kQ_i + 1) \times (Q_i)$ matrix.

After, representing $\tilde{\mathbf{P}}(B)$ and $\tilde{\mathbf{P}}_w(2)$ in terms of $\tilde{\mathbf{P}}_w(1)$, and by utilizing Equation 5.3, we obtain $\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = 0$ where

$$\mathbf{P} = \mathbf{D} - \gamma_1 b\mathbf{E}\mathbf{G}^{-1} - \mathbf{F}\mathbf{A}^{-1}\mathbf{B} - \gamma_1 b\mathbf{F}\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1}.$$

In addition, we have the normalization equation

$$\mathbf{p} = \mathbf{e}_{(1\times(kQ_R+1))} \times (\mathbf{I} + \gamma_1 b\mathbf{G}^{-1}) + \mathbf{e}_{(1\times(Q_R))} \times (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1}).$$

Replacing the first row of matrix $\mathbf{P}$ by the row vector $\mathbf{p}$, we solve for

$$\mathbf{P} \times \tilde{\mathbf{P}}_w(1) = [1, 0, \ldots, 0]^T_{(kQ_i+1)\times 1}.$$

The rest of the probabilities are obtained using

$$\begin{aligned} \tilde{\mathbf{P}}_w(2) &= \gamma_1 b\mathbf{G}^{-1}\tilde{\mathbf{P}}_w(1), \\ \tilde{\mathbf{P}}(B) &= (\mathbf{A}^{-1}\mathbf{B} + \gamma_1 b\mathbf{A}^{-1}\mathbf{C}\mathbf{G}^{-1})\tilde{\mathbf{P}}_w(1). \end{aligned}$$

5.1.5 An Aggregation Algorithm

The nature of the decomposition algorithm requires subsystems to supply information to each other. This is achieved by utilizing a fixed-point algorithm. The unknown parameters of the subsystems are μ_1, a, and $\omega_i(j),\ j = 0, 1, 2, \ldots$ for $i = 1, 2, \ldots, N$. As part of the algorithm, μ_1, a are used in the analysis of $\Omega(W)$. Similarly, $\omega_i(j),\ j = 0, 1, 2, \ldots$'s are used in the analysis of $\Omega(i)$ for $i = 1, 2, \ldots, N$. Yet, we have to assign values to these unknown probabilities.

Here, μ_1, a are obtained using the superposition approximation technique described in the previous section, and $\omega_i(j)$'s, $j = 0, 1, 2, \ldots$ for $i = 1, 2, \ldots, N$ are evaluated

$$\begin{aligned} \omega_i(0) &= Pr(N_W \geq Q_i \backslash N_i = R_i), \\ \omega_i(j) &= \sum_{k=(j-1)Q_W+1}^{jQ_W} Pr(N_W = Q_i - k \backslash N_i = R_i),\ j = 1, 2, \ldots. \end{aligned}$$

where N_W and N_i represent the inventory level at the warehouse and the retailers for $i = 1, 2, \ldots, N$, respectively. The $\omega_i(j)$'s are arrival-point probabilities. In this setting, it is difficult to compute the arrival rate probabilities. Instead, we use arbitrary time probabilities in the algorithm. The throughput of the subsystems are obtained using

$$\bar{\xi}_j = \frac{\text{Utilization of } M''_j}{E[U''_j]}, \; j = W, 1, 2, \ldots, N.$$

Due to backordering practice in the system, the throughput of the warehouse is known to be $\lambda = \sum_{i=1}^{N} \lambda_i$, and the throughput of the retailers are λ_i, for $i = 1, 2, \ldots, N$.

A summary of the algorithm is given in Table 5.1.

1.	Initialize: Obtain μ_1, a using the superposition approximation technique.
2.	Analyze $\Omega(W)$, obtain its steady-state probabilities, update $\omega_i(j)$, $j = 0, 1, 2, \ldots$, for $i = 0, 1, 2, \ldots, N$.
3.	Analyze $\Omega(i)$, obtain its steady-state probabilities, for $i = 0, 1, 2, \ldots, N$.
4.	Obtain customer service level at retailer i, for $i = 0, 1, 2, \ldots, N$.

Table 5.1: The approximation algorithm for multi-echelon distribution inventory system

5.2 Computational Accuracy

We test the accuracy of our approximation algorithm by comparing its results against simulation in a number of examples. The purpose of numerical examples is to see the ranges of the system parameters where the approximation is accurate and where it is not. The approximation procedure described above and the discrete-event simulation model runs are implemented on a Pentium IV PC operating at 2.80 GHz. The simulation model is developed using the Arena[1] simulation software. Each simulation run consists of 50,000,000 job departures to provide point estimates and 95% confidence intervals for key performance measures.

In this study, we focus on average inventory levels, average backorder levels, and customer service levels. Here, we define the customer service level as the probability of fully satisfying the demand of an arriving customer.

The approximation and the simulation results are given in Tables 5.2-5.9 for different

[1] Arena is a trademark of Rockwell Software.

systems and for different system parameters. Warehouse buffer capacity is chosen proportional to the retailer buffer capacities. In some settings, demand rate is varied while keeping other parameters constant.

We have three major experimental settings: a serial system (Table 5.2), a system with one warehouse and three retailers (Tables 5.3, 5.4), and a system with one warehouse and five retailers (Tables 5.5, 5.6, 5.7, 5.8, 5.9). We further differentiate the last two systems by assuming identical and non-identical retailers. In particular, Tables 5.3 and 5.5 refer to identical retailers whereas Tables 5.4, 5.6, 5.7, 5.8, and 5.9 refer to non-identical retailers.

	Warehouse	Retailer
	$R_W = 10$	$R_1 = 5$
Parameters:	$Q_W = 20$	$Q_1 = 10$
	$\beta_S = 1$	$\beta_W = 1$

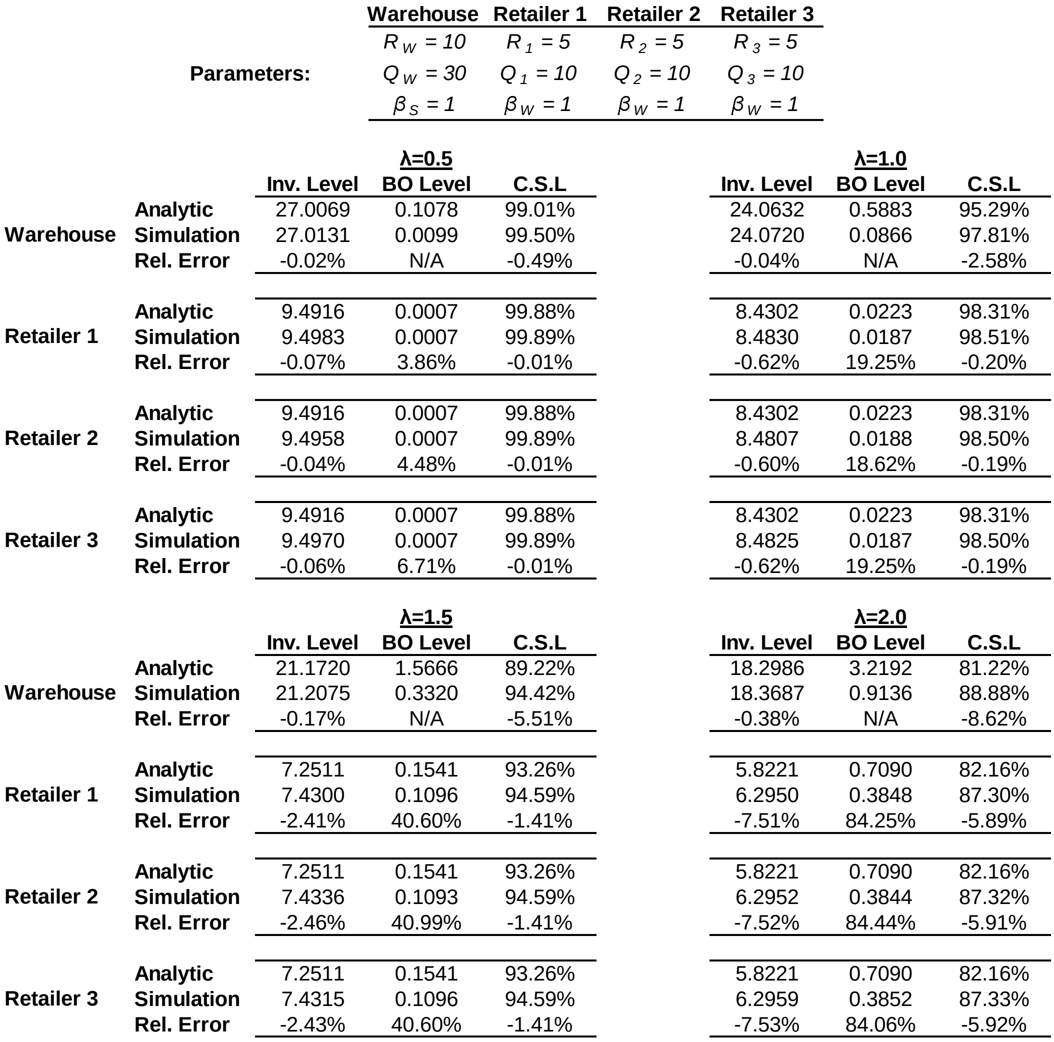
		λ=0.5			**λ=1.0**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Warehouse	**Analytic**	24.0000	0.0000	1.00%	23.0000	0.0000	100.00%
	Simulation	24.0012	0.0000	1.00%	23.0012	0.0000	100.00%
	Rel. Error	0.00%	0.00%	0.00%	-0.01%	0.00%	0.00%
Retailer	**Analytic**	9.5006	0.0007	99.89%	8.5104	0.0174	98.58%
	Simulation	9.4998	0.0006	99.89%	8.5094	0.0175	98.57%
	Rel. Error	0.01%	0.88%	0.00%	0.01%	-0.57%	0.01%
		λ=1.50			**λ=2.0**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Warehouse	**Analytic**	21.9997	0.0003	99.98%	20.9974	0.0027	99.88%
	Simulation	22.0009	0.0003	99.98%	20.9986	0.0027	99.88%
	Rel. Error	-0.01%	0.00%	0.00%	-0.01%	0.00%	0.00%
Retailer	**Analytic**	7.5332	0.0941	95.12%	6.5538	0.2960	89.32%
	Simulation	7.5320	0.0942	95.10%	6.5532	0.2961	89.29%
	Rel. Error	0.02%	-0.11%	0.02%	0.01%	-0.03%	0.03%
		λ=2.50			**λ=3.0**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Warehouse	**Analytic**	19.9886	0.0125	99.57%	18.9669	0.0388	98.91%
	Simulation	19.9898	0.0125	99.57%	18.9717	0.0382	98.92%
	Rel. Error	-0.01%	0.00%	0.00%	-0.03%	1.57%	-0.01%
Retailer	**Analytic**	5.5505	0.7368	81.01%	4.4988	1.6621	69.92%
	Simulation	5.5530	0.7350	81.00%	4.5150	1.6321	70.11%
	Rel. Error	-0.05%	0.24%	0.01%	-0.36%	1.84%	-0.27%
		λ=3.50			**λ=4.00**		
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
Warehouse	**Analytic**	17.9252	0.0947	97.79%	16.8566	0.1983	96.10%
	Simulation	17.9274	0.0937	97.80%	16.8562	0.1975	96.11%
	Rel. Error	-0.01%	1.07%	-0.01%	0.00%	0.41%	-0.01%
Retailer	**Analytic**	3.3705	3.7330	55.53%	2.1303	9.4627	37.07%
	Simulation	3.4313	3.5194	56.43%	2.2960	7.9471	39.87%
	Rel. Error	-1.77%	6.07%	-1.59%	-7.22%	19.07%	-7.02%

Table 5.2: Accuracy of the approximation algorithm for 1 Warehouse and 1 Retailer

In the serial system, the retailer follows a continuous review $(R_1, Q_1) = (5, 10)$ inventory control policy and the warehouse follows a continuous review $(R_W, Q_W) = (10, 20)$ inventory control policy. We assume, in particular, the transportation time from supplier to warehouse and from warehouse to retailer follow a 2'nd order Erlang (Erlang-2) distribution with rate 1. We vary the demand rate while keeping other parameters constant. The relative error of the performance estimates varies from -7.22% to 0.02% for average inventory levels, -0.57% to 19.07% for backorder levels, and -0.03% to 0.02% for customer service levels. This shows that our approximation algorithm is a strong alternative to the exact solution procedures, which require significant computational effort. In addition, it is clear from the results that the relative error gradually increases as the demand rate (system load) increases, which is expected.

In the one warehouse three identical retailer system, the relative error of the performance estimates varies from -7.53% to -0.02% for average inventory levels, 3.86% to 84.06% for backorder levels, and -8.62% to -0.01% for customer service levels. It is clear from the results that the percentage deviation gradually increases as the demand rate (system load) increases. Here, the accuracy in the backorder levels is somehow surprising. This is because backorder levels are low and approximating small probabilities does not seem to be quite successful. In addition, we use arbitrary time probabilities as surrogate of arrival rate probabilities and this results in less accurate results in retailers. Other tables can be interpreted accordingly.

As a final note, while the number of retailers increases, the accuracy of the results at warehouse also increases. This is because, the magnitude of autocorrelation of the demand arrival process decreases as there are more channels to send replenishment orders. In fact, we observe the highest negative autocorrelation at the superposed process when we consider two identical retailers. However, the negative lag-1 autocorrelation decreases as the number of superposed individual processes increases. This is, also, why our estimates are more accurate in a system with non-identical retailers than in a system with identical retailers. To give an idea of the magnitude of the lag-1 autocorrelation, we first consider a one warehouse and two identical retailers with Erlang-10 distributions with rate one. While the lag-1 autocorrelation is -0.5901 in a two retailer system, it is -0.3972 in a three retailer system. The lag-1 autocorrelation is expected to converge to zero as the number of superposed



	Warehouse	Retailer 1	Retailer 2	Retailer 3
	$R_W = 10$	$R_1 = 5$	$R_2 = 5$	$R_3 = 5$
Parameters:	$Q_W = 30$	$Q_1 = 10$	$Q_2 = 10$	$Q_3 = 10$
	$\beta_S = 1$	$\beta_W = 1$	$\beta_W = 1$	$\beta_W = 1$

		λ=0.5			λ=1.0		
		Inv. Level	BO Level	C.S.L	Inv. Level	BO Level	C.S.L
Warehouse	Analytic	27.0069	0.1078	99.01%	24.0632	0.5883	95.29%
	Simulation	27.0131	0.0099	99.50%	24.0720	0.0866	97.81%
	Rel. Error	-0.02%	N/A	-0.49%	-0.04%	N/A	-2.58%
Retailer 1	Analytic	9.4916	0.0007	99.88%	8.4302	0.0223	98.31%
	Simulation	9.4983	0.0007	99.89%	8.4830	0.0187	98.51%
	Rel. Error	-0.07%	3.86%	-0.01%	-0.62%	19.25%	-0.20%
Retailer 2	Analytic	9.4916	0.0007	99.88%	8.4302	0.0223	98.31%
	Simulation	9.4958	0.0007	99.89%	8.4807	0.0188	98.50%
	Rel. Error	-0.04%	4.48%	-0.01%	-0.60%	18.62%	-0.19%
Retailer 3	Analytic	9.4916	0.0007	99.88%	8.4302	0.0223	98.31%
	Simulation	9.4970	0.0007	99.89%	8.4825	0.0187	98.50%
	Rel. Error	-0.06%	6.71%	-0.01%	-0.62%	19.25%	-0.19%

		λ=1.5			λ=2.0		
		Inv. Level	BO Level	C.S.L	Inv. Level	BO Level	C.S.L
Warehouse	Analytic	21.1720	1.5666	89.22%	18.2986	3.2192	81.22%
	Simulation	21.2075	0.3320	94.42%	18.3687	0.9136	88.88%
	Rel. Error	-0.17%	N/A	-5.51%	-0.38%	N/A	-8.62%
Retailer 1	Analytic	7.2511	0.1541	93.26%	5.8221	0.7090	82.16%
	Simulation	7.4300	0.1096	94.59%	6.2950	0.3848	87.30%
	Rel. Error	-2.41%	40.60%	-1.41%	-7.51%	84.25%	-5.89%
Retailer 2	Analytic	7.2511	0.1541	93.26%	5.8221	0.7090	82.16%
	Simulation	7.4336	0.1093	94.59%	6.2952	0.3844	87.32%
	Rel. Error	-2.46%	40.99%	-1.41%	-7.52%	84.44%	-5.91%
Retailer 3	Analytic	7.2511	0.1541	93.26%	5.8221	0.7090	82.16%
	Simulation	7.4315	0.1096	94.59%	6.2959	0.3852	87.33%
	Rel. Error	-2.43%	40.60%	-1.41%	-7.53%	84.06%	-5.92%

Table 5.3: Accuracy of the approximation algorithm for 1 Warehouse and 3 identical Retailers

	Warehouse	Retailer 1	Retailer 2	Retailer 3
	$R_W = 10$	$R_1 = 5$	$R_2 = 5$	$R_3 = 5$
Parameters:	$Q_W = 30$	$Q_1 = 10$	$Q_2 = 5$	$Q_3 = 15$
	$\beta_S = 1$	$\beta_W = 1$	$\beta_W = 1$	$\beta_W = 1$

		Retailer 1	Retailer 2	Retailer 3	Retailer 1	Retailer 2	Retailer 3
		λ=1.0	λ=1.0	λ=1.0	λ=1.5	λ=1.0	λ=1.0
		Inv. Level	BO Level	C.S.L	Inv. Level	BO Level	C.S.L
	Analytic	21.6804	0.2778	89.99%	20.7490	0.4180	87.49%
Warehouse	Simulation	21.7106	0.2385	93.33%	20.7809	0.3347	92.04%
	Rel. Error	-0.14%	16.48%	-3.58%	-0.15%	24.89%	-4.94%
	Analytic	8.3406	0.0270	98.05%	7.2168	0.1532	93.23%
Retailer 1	Simulation	8.4239	0.0219	98.34%	7.3881	0.1164	94.36%
	Rel. Error	-0.99%	23.29%	-0.29%	-2.32%	31.62%	-1.20%
	Analytic	5.7345	0.0690	95.53%	5.6981	0.0748	95.27%
Retailer 2	Simulation	5.7850	0.0595	95.94%	5.7692	0.0611	95.85%
	Rel. Error	-0.87%	15.97%	-0.43%	-1.23%	22.42%	-0.61%
	Analytic	10.7152	0.0228	98.45%	10.6625	0.0251	98.34%
Retailer 3	Simulation	10.8145	0.0180	98.69%	10.7697	0.0198	98.60%
	Rel. Error	-0.92%	26.67%	-0.24%	-1.00%	26.77%	-0.26%

		Retailer 1	Retailer 2	Retailer 3	Retailer 1	Retailer 2	Retailer 3
		λ=1.0	λ=1.5	λ=1.0	λ=1.0	λ=1.0	λ=1.5
		Inv. Level	BO Level	C.S.L	Inv. Level	BO Level	C.S.L
	Analytic	20.7282	0.3655	89.15%	20.7938	0.4323	87.83%
Warehouse	Simulation	20.7675	0.3391	92.13%	20.8068	0.3556	91.03%
	Rel. Error	-0.19%	7.79%	-3.23%	-0.06%	21.57%	-3.52%
	Analytic	8.3113	0.0288	97.96%	8.2970	0.0297	97.91%
Retailer 1	Simulation	8.4038	0.0228	98.28%	8.3824	0.0237	98.23%
	Rel. Error	-1.10%	26.32%	-0.33%	-1.02%	25.32%	-0.33%
	Analytic	4.1298	0.6871	79.84%	5.6900	0.0761	95.22%
Retailer 2	Simulation	4.2922	0.5317	82.32%	5.7536	0.0631	95.76%
	Rel. Error	-3.78%	29.23%	-3.01%	-1.11%	20.60%	-0.56%
	Analytic	10.6645	0.0250	98.35%	9.5581	0.1172	94.98%
Retailer 3	Simulation	10.7701	0.0195	98.60%	9.7719	0.0872	95.89%
	Rel. Error	-0.98%	28.21%	-0.25%	-2.19%	34.40%	-0.95%

		Retailer 1	Retailer 2	Retailer 3	Retailer 1	Retailer 2	Retailer 3
		λ=1.5	λ=1.0	λ=1.5	λ=1.5	λ=1.5	λ=1.5
		Inv. Level	BO Level	C.S.L	Inv. Level	BO Level	C.S.L
	Analytic	19.9304	0.6065	85.48%	18.9155	0.8480	82.64%
Warehouse	Simulation	19.9031	0.4901	89.42%	18.9526	0.6629	87.82%
	Rel. Error	0.14%	23.75%	-4.41%	-0.20%	27.92%	-5.90%
	Analytic	7.1511	0.1675	92.82%	7.0825	0.1834	92.37%
Retailer 1	Simulation	7.3355	0.1250	94.06%	7.2973	0.1307	93.87%
	Rel. Error	-2.51%	34.00%	-1.32%	-2.94%	40.32%	-1.60%
	Analytic	5.6508	0.0828	94.93%	3.9118	0.8894	76.71%
Retailer 2	Simulation	5.7399	0.0641	95.70%	4.2160	0.5681	81.44%
	Rel. Error	-1.55%	29.17%	-0.80%	-7.22%	56.56%	-5.81%
	Analytic	9.4851	0.1275	94.68%	9.3963	0.1407	94.31%
Retailer 3	Simulation	9.7038	0.0943	95.65%	9.6320	0.1021	95.39%
	Rel. Error	-2.25%	35.21%	-1.01%	-2.45%	37.81%	-1.13%

Table 5.4: Accuracy of the approximation algorithm for 1 Warehouse and 3 Retailers

	Warehouse	Retailer 1	Retailer 2	Retailer 3	Retailer 4	Retailer 5
	$R_W = 10$	$R_1 = 5$	$R_2 = 5$	$R_3 = 5$	$R_4 = 5$	$R_5 = 5$
Parameters:	$Q_W = 30$	$Q_1 = 10$	$Q_2 = 10$	$Q_3 = 10$	$Q_4 = 10$	$Q_5 = 10$
	$\beta_S = 1$	$\beta_W = 1$	$\beta_W = 1$	$\beta_W = 1$	$\beta_W = 1$	$\beta_W = 1$

			λ=0.75			**λ=1.00**	
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	22.6320	1.1996	91.59%	20.2293	2.3328	85.69%
Warehouse	**Simulation**	22.6508	0.2848	94.39%	20.2449	0.6643	90.30%
	Rel. Error	-0.08%	N/A	-2.97%	-0.08%	N/A	-5.11%
	Analytic	8.8915	0.0077	99.25%	8.2437	0.0372	97.58%
Retailer 1	**Simulation**	8.9475	0.0058	99.39%	8.3795	0.0236	98.22%
	Rel. Error	-0.63%	32.76%	-0.14%	-1.62%	57.63%	-0.65%
	Analytic	8.8915	0.0077	99.25%	8.2437	0.0372	97.58%
Retailer 2	**Simulation**	8.9454	0.0059	99.38%	8.3826	0.0238	98.22%
	Rel. Error	-0.60%	30.51%	-0.13%	-1.66%	56.30%	-0.65%
	Analytic	8.8915	0.0077	99.25%	8.2437	0.0372	97.58%
Retailer 3	**Simulation**	8.9487	0.0058	99.39%	8.3822	0.0238	98.23%
	Rel. Error	-0.64%	32.76%	-0.14%	-1.65%	56.30%	-0.66%
	Analytic	8.8915	0.0077	99.25%	8.2437	0.0372	97.58%
Retailer 4	**Simulation**	8.9455	0.0059	99.38%	8.3815	0.0237	98.22%
	Rel. Error	-0.60%	30.51%	-0.13%	-1.64%	56.96%	-0.65%
	Analytic	8.8915	0.0077	99.25%	8.2437	0.0372	97.58%
Retailer 5	**Simulation**	8.9469	0.0059	99.38%	8.3795	0.0238	98.22%
	Rel. Error	-0.62%	30.51%	-0.13%	-1.62%	56.30%	-0.65%

			λ=1.25			**λ=1.50**	
		Inv. Level	**BO Level**	**C.S.L**	**Inv. Level**	**BO Level**	**C.S.L**
	Analytic	17.8252	4.0308	78.57%	15.4045	6.5446	70.35%
Warehouse	**Simulation**	17.8469	1.3426	84.94%	15.4622	2.4930	78.18%
	Rel. Error	-0.12%	N/A	-7.50%	-0.37%	N/A	-10.02%
	Analytic	7.4758	0.1360	94.00%	6.4981	0.4532	86.97%
Retailer 1	**Simulation**	7.7664	0.0704	96.08%	7.0796	0.1727	92.60%
	Rel. Error	-3.74%	93.18%	-2.16%	-8.21%	162.42%	-6.08%
	Analytic	7.4758	0.1360	94.00%	6.4981	0.4532	86.97%
Retailer 2	**Simulation**	7.7672	0.0696	96.09%	7.0880	0.1712	92.64%
	Rel. Error	-3.75%	95.40%	-2.18%	-8.32%	164.72%	-6.12%
	Analytic	7.4758	0.1360	94.00%	6.4981	0.4532	86.97%
Retailer 3	**Simulation**	7.7648	0.0698	96.08%	7.0839	0.1729	92.62%
	Rel. Error	-3.72%	94.84%	-2.16%	-8.27%	162.12%	-6.10%
	Analytic	7.4758	0.1360	94.00%	6.4981	0.4532	86.97%
Retailer 4	**Simulation**	7.7682	0.0694	96.10%	7.0835	0.1727	92.63%
	Rel. Error	-3.76%	95.97%	-2.19%	-8.26%	162.42%	-6.11%
	Analytic	7.4758	0.1360	94.00%	6.4981	0.4532	86.97%
Retailer 5	**Simulation**	7.7703	0.0695	96.10%	7.0816	0.1740	92.58%
	Rel. Error	-3.79%	95.68%	-2.19%	-8.24%	160.46%	-6.06%

Table 5.5: Accuracy of the approximation algorithm for 1 Warehouse and 5 identical Retailers

		λ=1.0				λ=1.0			
		Parameters	Inv. Level	BO Level	C.S.L	Parameters	Inv. Level	BO Level	C.S.L
	Analytic	$R_W = 10$	17.9942	1.1042	84.16%	$R_W = 10$	18.0339	1.2652	79.88%
Warehouse	Simulation	$Q_W = 30$	17.9915	0.8923	86.76%	$Q_W = 30$	18.0596	1.0533	83.27%
	Rel. Error	$\beta_S = 1$	0.02%	23.75%	-3.00%	$\beta_S = 1$	-0.14%	20.12%	-4.07%
	Analytic	$R_1 = 5$	8.1669	0.0392	97.45%	$R_1 = 5$	8.1406	0.0414	97.35%
Retailer 1	Simulation	$Q_1 = 10$	8.2947	0.0285	97.96%	$Q_1 = 10$	8.2548	0.0306	97.85%
	Rel. Error	$\beta_W = 1$	-1.54%	37.54%	-0.52%	$\beta_W = 1$	-1.38%	35.29%	-0.51%
	Analytic	$R_2 = 5$	5.5528	0.1015	94.18%	$R_2 = 5$	5.5290	0.1067	93.98%
Retailer 2	Simulation	$Q_2 = 5$	5.6843	0.0720	95.34%	$Q_2 = 5$	5.6718	0.0738	95.25%
	Rel. Error	$\beta_W = 1$	-2.31%	40.97%	-1.22%	$\beta_W = 1$	-2.52%	44.58%	-1.33%
	Analytic	$R_3 = 5$	10.4910	0.0340	97.93%	$R_3 = 5$	10.4836	0.0347	97.90%
Retailer 3	Simulation	$Q_3 = 15$	10.6199	0.0249	98.32%	$Q_3 = 15$	10.6146	0.0251	98.31%
	Rel. Error	$\beta_W = 1$	-1.21%	36.55%	-0.40%	$\beta_W = 1$	-1.23%	38.25%	-0.42%
	Analytic	$R_4 = 5$	5.5528	0.1015	94.18%	$R_4 = 5$	8.1406	0.0414	97.35%
Retailer 4	Simulation	$Q_4 = 5$	5.6854	0.0708	95.36%	$Q_4 = 10$	8.2575	0.0309	97.84%
	Rel. Error	$\beta_W = 1$	-2.33%	43.36%	-1.24%	$\beta_W = 1$	-1.42%	33.98%	-0.50%
	Analytic	$R_5 = 5$	5.5528	0.1015	94.18%	$R_5 = 5$	8.1406	0.0414	97.35%
Retailer 5	Simulation	$Q_5 = 5$	5.6866	0.0711	95.37%	$Q_5 = 10$	8.2582	0.0306	97.85%
	Rel. Error	$\beta_W = 1$	-2.35%	42.76%	-1.25%	$\beta_W = 1$	-1.42%	35.29%	-0.51%

		λ=1.0				λ=1.0			
			Inv. Level	BO Level	C.S.L		Inv. Level	BO Level	C.S.L
	Analytic	$R_W = 10$	18.0730	1.4697	77.03%	$R_W = 10$	18.0141	1.1860	82.29%
Warehouse	Simulation	$Q_W = 30$	18.1161	1.2933	79.96%	$Q_W = 30$	18.0250	0.9691	85.17%
	Rel. Error	$\beta_S = 1$	-0.24%	13.64%	-3.66%	$\beta_S = 1$	-0.06%	22.38%	-3.38%
	Analytic	$R_1 = 5$	8.1234	0.0433	97.27%	$R_1 = 5$	8.1548	0.0403	97.40%
Retailer 1	Simulation	$Q_1 = 10$	8.2384	0.0322	97.78%	$Q_1 = 10$	8.2775	0.0293	97.91%
	Rel. Error	$\beta_W = 1$	-1.40%	34.47%	-0.52%	$\beta_W = 1$	-1.48%	37.54%	-0.52%
	Analytic	$R_2 = 5$	5.4878	0.1158	93.64%	$R_2 = 5$	5.5393	0.1044	94.07%
Retailer 2	Simulation	$Q_2 = 5$	5.6277	0.0794	94.96%	$Q_2 = 5$	5.6724	0.0729	95.27%
	Rel. Error	$\beta_W = 1$	-2.49%	45.84%	-1.39%	$\beta_W = 1$	-2.35%	43.21%	-1.26%
	Analytic	$R_3 = 5$	10.4735	0.0358	97.86%	$R_3 = 5$	10.4872	0.0344	97.91%
Retailer 3	Simulation	$Q_3 = 15$	10.6032	0.0259	98.27%	$Q_3 = 15$	10.6178	0.0247	98.32%
	Rel. Error	$\beta_W = 1$	-1.22%	38.22%	-0.42%	$\beta_W = 1$	-1.23%	39.27%	-0.42%
	Analytic	$R_4 = 5$	10.4735	0.0358	97.86%	$R_4 = 5$	5.5393	0.1044	94.07%
Retailer 4	Simulation	$Q_4 = 15$	10.6070	0.0256	98.28%	$Q_4 = 5$	5.6757	0.0724	95.30%
	Rel. Error	$\beta_W = 1$	-1.26%	39.84%	-0.43%	$\beta_W = 1$	-2.40%	44.20%	-1.29%
	Analytic	$R_5 = 5$	10.4735	0.0358	97.86%	$R_5 = 5$	8.1548	0.0403	97.40%
Retailer 5	Simulation	$Q_5 = 15$	10.6068	0.0258	98.28%	$Q_5 = 10$	8.2744	0.0295	97.91%
	Rel. Error	$\beta_W = 1$	-1.26%	38.76%	-0.43%	$\beta_W = 1$	-1.45%	36.61%	-0.52%

Table 5.6: Accuracy of the approximation algorithm for 1 Warehouse and 5 Retailers

		λ=1.0				λ=1.0			
		Parameters	**Inv. Level**	**BO Level**	**C.S.L**	**Parameters**	**Inv. Level**	**BO Level**	**C.S.L**
Warehouse	**Analytic**	$R_W = 10$	18.0354	1.3051	80.95%	$R_W = 10$	18.0354	1.3051	80.95%
	Simulation	$Q_W = 30$	18.0569	1.1000	83.55%	$Q_W = 30$	18.0894	1.1736	81.65%
	Rel. Error	$\beta_S = 1$	-0.12%	18.65%	-3.11%	$\beta_S = 1$	-0.30%	11.20%	-0.86%
Retailer 1	**Analytic**	$R_1 = 5$	8.1465	0.0412	97.36%	$R_1 = 5$	8.1465	0.0412	97.36%
	Simulation	$Q_1 = 10$	8.2652	0.0301	97.88%	$Q_1 = 10$	8.2468	0.0317	97.80%
	Rel. Error	$\beta_W = 1$	-1.44%	36.88%	-0.53%	$\beta_W = 1$	-1.22%	29.97%	-0.45%
Retailer 2	**Analytic**	$R_2 = 5$	5.5192	0.1088	93.90%	$R_2 = 5$	5.5192	0.1088	93.90%
	Simulation	$Q_2 = 5$	5.6537	0.0756	95.15%	$Q_2 = 5$	5.6520	0.0765	95.13%
	Rel. Error	$\beta_W = 1$	-2.38%	43.92%	-1.31%	$\beta_W = 1$	-2.35%	42.22%	-1.29%
Retailer 3	**Analytic**	$R_3 = 5$	10.4813	0.0350	97.89%	$R_3 = 5$	10.4813	0.0350	97.89%
	Simulation	$Q_3 = 15$	10.6109	0.0253	98.30%	$Q_3 = 15$	10.6114	0.0256	98.28%
	Rel. Error	$\beta_W = 1$	-1.22%	38.34%	-0.42%	$\beta_W = 1$	-1.23%	36.72%	-0.40%
Retailer 4	**Analytic**	$R_4 = 5$	5.5192	0.1088	93.90%	$R_4 = 5$	8.1465	0.0412	97.36%
	Simulation	$Q_4 = 5$	5.6528	0.0759	95.14%	$Q_4 = 10$	8.2493	0.0313	97.82%
	Rel. Error	$\beta_W = 1$	-2.36%	43.35%	-1.30%	$\beta_W = 1$	-1.25%	31.63%	-0.47%
Retailer 5	**Analytic**	$R_5 = 5$	10.4813	0.0350	97.89%	$R_5 = 5$	10.4813	0.0350	97.89%
	Simulation	$Q_5 = 15$	10.6108	0.0253	98.29%	$Q_5 = 15$	10.6104	0.0255	98.29%
	Rel. Error	$\beta_W = 1$	-1.22%	38.34%	-0.41%	$\beta_W = 1$	-1.22%	37.25%	-0.41%

Table 5.7: Accuracy of the approximation algorithm for 1 Warehouse and 5 Retailers

processes increases. We conclude that even using a three-moment approximation scheme does not guarantee a good approximation of the inherited autocorrelation in the superposed processes.

λ_1=0.5 λ_2=1.0 λ_3=1.5 λ_4=1.0 λ_5=0.5

		Parameters	Inv. Level	BO Level	C.S.L	Parameters	Inv. Level	BO Level	C.S.L
	Analytic	$R_W = 10$	18.9204	0.8910	85.51%	$R_W = 10$	18.9449	0.9681	82.17%
Warehouse	Simulation	$Q_W = 30$	18.9479	0.6861	88.05%	$Q_W = 30$	18.9778	0.8036	85.32%
	Rel. Error	$\beta_S = 1$	-0.15%	29.86%	-2.88%	$\beta_S = 1$	-0.17%	20.47%	-3.69%
	Analytic	$R_1 = 5$	9.3490	0.0016	99.78%	$R_1 = 5$	9.3387	0.0016	99.78%
Retailer 1	Simulation	$Q_1 = 10$	9.3859	0.0012	99.82%	$Q_1 = 10$	9.3725	0.0013	99.82%
	Rel. Error	$\beta_W = 1$	-0.39%	33.33%	-0.04%	$\beta_W = 1$	-0.36%	23.08%	-0.04%
	Analytic	$R_2 = 5$	5.5973	0.0928	94.52%	$R_2 = 5$	5.5825	0.0957	94.41%
Retailer 2	Simulation	$Q_2 = 5$	5.7062	0.0687	95.48%	$Q_2 = 5$	5.6946	0.0706	95.39%
	Rel. Error	$\beta_W = 1$	-1.91%	35.08%	-1.01%	$\beta_W = 1$	-1.97%	35.55%	-1.03%
	Analytic	$R_3 = 5$	9.3931	0.1416	94.29%	$R_3 = 5$	9.3886	0.1429	94.26%
Retailer 3	Simulation	$Q_3 = 15$	9.6374	0.1022	95.40%	$Q_3 = 15$	9.6306	0.1018	95.39%
	Rel. Error	$\beta_W = 1$	-2.53%	38.55%	-1.16%	$\beta_W = 1$	-2.51%	40.37%	-1.18%
	Analytic	$R_4 = 5$	5.5973	0.0928	94.52%	$R_4 = 5$	8.1927	0.0373	97.54%
Retailer 4	Simulation	$Q_4 = 5$	5.7055	0.0682	95.48%	$Q_4 = 10$	8.2991	0.0285	97.97%
	Rel. Error	$\beta_W = 1$	-1.90%	36.07%	-1.01%	$\beta_W = 1$	-1.28%	30.88%	-0.44%
	Analytic	$R_5 = 5$	6.8876	0.0026	99.63%	$R_5 = 5$	9.3387	0.0016	99.78%
Retailer 5	Simulation	$Q_5 = 5$	6.9149	0.0021	99.68%	$Q_5 = 10$	9.3731	0.0013	99.81%
	Rel. Error	$\beta_W = 1$	-0.39%	23.81%	-0.05%	$\beta_W = 1$	-0.37%	23.08%	-0.03%

			Inv. Level	BO Level	C.S.L		Inv. Level	BO Level	C.S.L
	Analytic	$R_W = 10$	18.9761	1.0874	79.83%	$R_W = 10$	18.9268	0.9080	84.60%
Warehouse	Simulation	$Q_W = 30$	18.9946	0.9710	82.64%	$Q_W = 30$	18.9399	0.7374	87.06%
	Rel. Error	$\beta_S = 1$	-0.10%	11.99%	-3.40%	$\beta_S = 1$	-0.07%	23.14%	-2.83%
	Analytic	$R_1 = 5$	9.3328	0.0017	99.77%	$R_1 = 5$	9.3464	0.0016	99.78%
Retailer 1	Simulation	$Q_1 = 10$	9.3610	0.0014	99.80%	$Q_1 = 10$	9.3789	0.0012	99.82%
	Rel. Error	$\beta_W = 1$	-0.30%	21.43%	-0.03%	$\beta_W = 1$	-0.35%	33.33%	-0.04%
	Analytic	$R_2 = 5$	5.5516	0.1019	94.16%	$R_2 = 5$	5.5927	0.0937	94.49%
Retailer 2	Simulation	$Q_2 = 5$	5.6594	0.0751	95.17%	$Q_2 = 5$	5.6983	0.0696	95.43%
	Rel. Error	$\beta_W = 1$	-1.90%	35.69%	-1.06%	$\beta_W = 1$	-1.85%	34.63%	-0.99%
	Analytic	$R_3 = 5$	9.3812	0.1451	94.21%	$R_3 = 5$	9.3922	0.1419	94.28%
Retailer 3	Simulation	$Q_3 = 15$	9.6234	0.1035	95.35%	$Q_3 = 15$	9.6234	0.1026	95.37%
	Rel. Error	$\beta_W = 1$	-2.52%	40.19%	-1.20%	$\beta_W = 1$	-2.40%	38.30%	-1.14%
	Analytic	$R_4 = 5$	10.5391	0.0319	98.03%	$R_4 = 5$	5.5927	0.0937	94.49%
Retailer 4	Simulation	$Q_4 = 15$	10.6566	0.0237	98.39%	$Q_4 = 5$	5.6972	0.0704	95.41%
	Rel. Error	$\beta_W = 1$	-1.10%	34.60%	-0.37%	$\beta_W = 1$	-1.83%	33.10%	-0.96%
	Analytic	$R_5 = 5$	11.7561	0.0015	99.80%	$R_5 = 5$	9.3464	0.0016	99.78%
Retailer 5	Simulation	$Q_5 = 15$	11.7941	0.0011	99.84%	$Q_5 = 10$	9.3828	0.0012	99.82%
	Rel. Error	$\beta_W = 1$	-0.32%	36.36%	-0.04%	$\beta_W = 1$	-0.39%	33.33%	-0.04%

Table 5.8: Accuracy of the approximation algorithm for 1 Warehouse and 5 non-identical Retailers

λ_1=0.5 λ_2=1.0 λ_3=1.5 λ_4=1.0 λ_5=0.5

		Parameters	Inv. Level	BO Level	C.S.L	Parameters	Inv. Level	BO Level	C.S.L
	Analytic	$R_W = 10$	18.9374	0.9567	83.87%	$R_W = 10$	18.9547	1.0067	81.43%
Warehouse	**Simulation**	$Q_W = 30$	18.9498	0.8045	86.17%	$Q_W = 30$	18.9922	0.8663	84.45%
	Rel. Error	$\beta_S = 1$	-0.07%	18.92%	-2.67%	$\beta_S = 1$	-0.20%	16.21%	-3.58%
	Analytic	$R_1 = 5$	9.3446	0.0016	99.78%	$R_1 = 5$	9.3368	0.0017	99.77%
Retailer 1	**Simulation**	$Q_1 = 10$	9.3797	0.0013	99.81%	$Q_1 = 10$	9.3702	0.0013	99.81%
	Rel. Error	$\beta_W = 1$	-0.37%	23.08%	-0.03%	$\beta_W = 1$	-0.36%	30.77%	-0.04%
	Analytic	$R_2 = 5$	5.5829	0.0956	94.41%	$R_2 = 5$	5.5731	0.0976	94.33%
Retailer 2	**Simulation**	$Q_2 = 5$	5.6860	0.0711	95.36%	$Q_2 = 5$	5.6853	0.0716	95.33%
	Rel. Error	$\beta_W = 1$	-1.81%	34.46%	-1.00%	$\beta_W = 1$	-1.97%	36.31%	-1.05%
	Analytic	$R_3 = 5$	9.3890	0.1428	94.26%	$R_3 = 5$	9.3862	0.1436	94.24%
Retailer 3	**Simulation**	$Q_3 = 15$	9.6254	0.1029	95.37%	$Q_3 = 15$	9.6306	0.1028	95.37%
	Rel. Error	$\beta_W = 1$	-2.46%	38.78%	-1.16%	$\beta_W = 1$	-2.54%	39.69%	-1.18%
	Analytic	$R_4 = 5$	5.5829	0.0956	94.41%	$R_4 = 5$	8.1891	0.0376	97.53%
Retailer 4	**Simulation**	$Q_4 = 5$	5.6872	0.0711	95.37%	$Q_4 = 10$	8.2907	0.0289	97.95%
	Rel. Error	$\beta_W = 1$	-1.83%	34.46%	-1.01%	$\beta_W = 1$	-1.23%	30.10%	-0.43%
	Analytic	$R_5 = 5$	11.7592	0.0015	99.81%	$R_5 = 5$	11.7581	0.0015	99.81%
Retailer 5	**Simulation**	$Q_5 = 15$	11.7916	0.0011	99.85%	$Q_5 = 15$	11.7924	0.0011	99.84%
	Rel. Error	$\beta_W = 1$	-0.27%	36.36%	-0.04%	$\beta_W = 1$	-0.29%	36.36%	-0.03%

Table 5.9: Accuracy of the approximation algorithm for 1 Warehouse and 5 non-identical Retailers

Chapter 6
Forecasting Using TES Processes

Improving decision making practices in a supply chain is a major source of competitive advantage in today's uncertain business environments. Resolving uncertainty in early phases of the decision making process will result in better planning and accuracy of supply chain activities, and improved customer service levels, lesser inventories and lower costs. Forecasting is one of the key ingredients necessary to handle uncertainties in the early stages of planning. It is a crucial driver for procurement, manufacturing and distribution activities in a supply chain.

Improving the quality of forecasts has been a challenging problem. Failure to account for large autocorrelations, trend, and seasonality in data sets is key ingredient contributing to lack of accuracy in forecasting. Time series models such as Winters exponential smoothing, Box-Jenkins auto regressive integrated moving average (ARIMA), and multiple regression have been widely used to account for these type of patterns. Likewise, TES (Transform-Expand-Sample) models were utilized to generate forecasts for correlated data sets [73]. Melamed [81] introduced TES processes to model autocorrelated time series in Monte Carlo simulation.

The primary objective of time series modeling (TSM) is to draw inferences from past data. It relies on the argument that data points taken through time may have an underlying structure (such as autocorrelation, trend or seasonal variation) and this structure may persist over time. The approach consists of establishing a mathematical model to represent a given data set. Then, the model is employed to describe and analyze the sample data, and make forecasts for the future. The main advantage of time series models is that they can handle any persistent patterns in data [25, 28].

TES is a methodology [71, 72] to model empirical time series from a stationary probability law. Its merit is to capture both the empirical distribution and autocorrelation function, simultaneously. It can model a wide variety of autocorrelation functions (e.g. monotone, oscillating, alternating etc.) and is suitable for Monte Carlo simulation of autocorrelated time series. The analytical formulas of TES processes provide calculation of autocorrelations as well as its transition structure. Forecasts for the future can be calculated by utilizing the known transition structure of TES processes [73].

This chapter reports on an experimental study that compares TES process forecasting to traditional Box-Jenkins ARIMA models. Similar comparative studies exist in the literature. Among the recent ones, Alon *et al.* [4] presents a study that compares artificial neural networks to time series forecasting methods in predicting US aggregate retail sales. Thomakos and Guerard [99] compare naive, ARIMA, nonparametric and transfer function models on several data sets. Zou and Yang [111] suggest combining several time series models to get forecasts that are more accurate and compare them to individual methods. Our study differs from others since it exploits TES forecasting procedure. Jagerman and Melamed [73] also implement the TES forecasting methodology based on the use of mixture of uniform random variables as the innovation density. This chapter contains an extensive computational study of TES forecasting, and exploits phase-type random variables as the innovation density.

The remainder of the chapter is organized as follows. The next section gives an overview of TSM methodology. The second section explains TES processes and its empirical modeling. The third section illustrates the numerical implementation. The fourth section contains a comparison study of TES forecasting to general ARIMA models and final comments.

6.1 Time Series Models

Time series models are used to draw inferences from past data. In these models, data is analyzed in order to identify patterns recurring over time. Then, forecasts for future periods are developed based on such underlying patterns. The applications of time series models include forecasting future values of the series, testing hypothesis, monitoring and simulation, among others.

A *discrete time series* $\{X_t\}$, $t = 0, 1, 2, \ldots$ is a sequence of observations recorded at time t, correspondingly, a *continuous time series* is the one where observations recorded continuously. The autocorrelation function of a stationary time series, $\{X_t\}$, with common mean $\mu_X < \infty$ and variance $\sigma_X^2 < \infty$ is defined at lag τ as

$$\rho_x(\tau) = \frac{E[(X_{t+\tau} - \mu_X)(X_t - \mu_X)]}{\sigma_X^2}, \quad \tau = 1, 2, 3, \dots . \tag{6.1}$$

Box and Jenkins [25] provides a methodology for fitting a model to an empirical data set. The systematic approach identifies a class of models appropriate for empirical data sequence at hand and estimates its parameters. A general class of Box and Jenkins models includes ARIMA models that can model a large class of autocorrelation functions [25, 28]. The model is a combination of auto regressive (AR) and moving average (MA) models for differenced data. An AR model is simply a regression of the current observation to the previous ones. On the other hand, an MA model is a regression of the current value against the previous white noise.

6.2 TES Processes

TES is a modeling methodology [71, 72, 81, 82] of empirical time series that captures a very strong statistical signature such as the marginal distribution as well as the autocorrelation function. In addition, its analytical background makes it a viable tool to forecast future values of empirical time series data [73].

The TES modeling procedure satisfies three important requirements of fitting a model to an empirical data set. The first one is to match the marginal distribution of the model to the marginal distribution of the time series, which is a first-order characteristic of the data. The second one is to approximate the autocorrelation function of the data, a second-order statistics. Finally, the third requirement is that the sample paths generated by the TES model should resemble their empirical counterparts.

A TES process utilizes background and foreground schemes in the sequence generation procedure. That is, an auxiliary sequence is generated from a stationary process by a recursive relationship. Then, the target foreground sequence is obtained by making a transformation using the background sequence.

There are two types of TES processes, namely TES$^+$ and TES$^-$. The former can generate positive lag-1 autocorrelations while the latter can generate negative lag-1 autocorrelations, respectively. In this paper, we are mainly interested in TES$^+$ processes. We will append proper superscripts (plus or minus) wherever it is necessary to distinguish between TES$^+$ and TES$^-$. A TES$^+$ process is generated as follows. First, a background variate, U_n^+, is generated by utilizing the following recursive relationship:

$$U_n^+ = \langle U_{n-1}^+ + V_n \rangle, \quad n > 0 \tag{6.2}$$

where U_0 is a uniform random number in $(0,1)$, V_n is an i.i.d. random sequence (called the *innovation sequence* since they bring added randomness at each step) with a common density function, f_v, independent of $U_0^+, \ldots U_{n-1}^+$, and $\langle\rangle$ is the modulo-1 arithmetic operator, i.e., $\langle x \rangle = x - \max\{\text{integer } n : n \le x\}$, resulting in the fractional part of n. U_n^+ turns out to have Uniform $(0,1)$ marginal distribution. Then, the foreground sequence X_n^+ is obtained using a transformation (called distortion) from U_n^+, i.e.,

$$X_n^+ = D(U_n^+), \quad n > 0. \tag{6.3}$$

In order to smooth the sample paths generated by TES models, an intermediary *stitching* transformation is applied to the background sequence. It is a piecewise linear transformation and it preserves the uniformity of the original sequence. The transformation is given by

$$S_\xi(y) = \begin{cases} \frac{y}{\xi}, & 0 \le y < \xi \\ \\ \frac{1-y}{1-\xi}, & \xi \le y < 1 \end{cases} \tag{6.4}$$

where $\xi \in [0,1)$.

6.2.1 The Autocorrelation Function of TES Processes

The autocorrelation function of a foreground TES$^+$ sequence X_n^+ is given by

$$\rho_x^+(\tau) = \frac{2}{\sigma_X^2} \sum_{\nu=1}^{\infty} Re[\tilde{f_v^\tau}(i2\pi\nu)] |\tilde{D}(i2\pi\nu)|^2 \tag{6.5}$$

where $Re[]$ denotes the real part of a complex number, and $\tilde{f_v}$ and $\tilde{D}$ are the Laplace transforms of the innovation density and the distortion, respectively. Details of 6.5 can be found in [71].

6.2.2 The Empirical TES Modeling Methodology

Given an empirical time series $\{Y_n\}_{n=0}^{N}$, TES modeling methodology aims to fit TES models whose marginal distribution matches the marginal distribution of the time series, and whose autocorrelation function, $\rho_X(\tau)$, approximates its empirical counterpart, $\hat{\rho}_Y(\tau)$. The methodology consists of selecting the TES model, TES$^+$ or TES$^-$, a transformation (distortion), D, a stitching parameter, ξ, and an innovation density, f_v.

Initially, TES model is selected by investigating the empirical autocorrelations. Then, the general practice is to express data sequence as an empirical density (histogram) since a mixture of uniform distributions can approximate any general density function [8]. Let $\hat{H}_Y$ denote the associated cumulative distribution function of the empirical density. In particular, the construction of the distortion of the form $D = \hat{H}_Y^{-1}(S_\xi)$ makes sure that the random sequence $\{X_n\}$ has the same marginal distribution as the empirical histogram, (due to the inversion transformation method [8, 27]). A formulation of empirical density function is given in Appendix C. It remains to select an appropriate stitching parameter and an innovation density. This selection requires an extensive search procedure. In fact, the choice of (f_v, ξ) determines the model's autocorrelation structure.

Successful applications of the TES models consist of machine failures, financial time series models, MPEG-compressed VBR video, texture synthesis, and H.261-Compressed video [65, 66]. An algorithmic empirical TES model fitting methodology using mixture of uniform innovation sequences is described in [8, 74].

6.3 Forecasting Using TES Processes

In order to use TES models in forecasting, one first needs to model the data set using a TES process. As part of the empirical TES modeling methodology, choosing a proper innovation density requires extensive computational effort. However, it is possible to limit the search to a subset of innovation densities. Among the candidate densities are mixture of uniform innovations as well as phase-type distributions because of their generality and versatility. Earlier work on TES modeling used mixture of uniform random variables as the innovation variables. In this study, we propose using phase-type random variables [7, 86]. Using phase-type random variables as the innovation density substantially reduces the search space since they are more likely to have fewer parameters than mixtures of uniform random variables. Below, we show how to calculate the autocorrelation function 6.5 of TES processes using different innovation densities.

6.3.1 The Innovation Variables

In the current implementation of TES modeling [82], mixtures of uniform distributions have been used as innovation variables, having the density function

$$f_v(x) = \sum_{k=1}^{K} \frac{P_k}{R_k - L_k} \mathbf{1}_{[L_k, R_k)}(x) \tag{6.6}$$

where K is the number of Uniform (L_k, R_k) variates with mixing probabilities P_k.

In this study, we have implemented a subset of phase-type distributions as innovation variables consisting of mixtures of generalized Erlang (*MGE*) distributions that have been widely used in the analysis of manufacturing and communication systems [7, 86]. A special case is the *MGE-2* distribution consisting mixtures of two exponential phases with respective rates μ_1 and μ_2 and a density function

$$f_v(x) = c_1\mu_1 e^{-\mu_1 x} + c_2\mu_2 e^{-\mu_2 x} \tag{6.7}$$

with mixing probabilities $c_1 = (\mu_1(1-a_1) - \mu_2)/(\mu_1 - \mu_2)$ and $c_2 = 1 - c_1$, where $\mu_1 \neq \mu_2$, and a_1 being the conditional probability of going to phase 2 given that phase 1 is completed. Additionally, its Laplace transform is given by

$$\tilde{f}_v(s) = \frac{s\mu_1(1-a_1) + \mu_1\mu_2}{s^2 + s(\mu_1+\mu_2) + \mu_1\mu_2}. \tag{6.8}$$

6.3.2 Computation of $\tilde{f}_v(i2\pi\nu)$

For the innovation density 6.6, $\tilde{f}_v(i2\pi\nu)$ is given by

$$\tilde{f}_v(i2\pi\nu) = \sum_{k=1}^{K} P_k \frac{e^{-i2\pi\nu\alpha_k\phi_k}\sin(\pi\nu\alpha_k)}{\pi\nu\alpha_k} \tag{6.9}$$

where $\alpha_k = R_k - L_k$ and $\phi_k = (R_k + L_k)/\alpha_k$, and for the innovation density 6.7, it is given by

$$\begin{aligned} &\tilde{f}_v(i2\pi\nu) \\ &= \frac{\sqrt{\mu_1^2\mu_2^2+4\pi^2\nu^2\mu_1^2(1-a_1)^2}}{\sqrt{(\mu_1\mu_2-4\pi^2\nu^2)^2+(2\pi\nu(\mu_1+\mu_2))^2}} e^{i(\tan^{-1}(\frac{2\pi\nu\mu_1(1-a_1)}{\mu_1\mu_2})-\tan^{-1}(\frac{2\pi\nu(\mu_1+\mu_2)}{\mu_1\mu_2-4\pi^2\nu^2}))} \end{aligned} \tag{6.10}$$

Computation of $|\tilde{D}(i2\pi\nu)|^2$ is given in [71, 72] and summarized in Appendix C.

6.3.3 TES Model Fitting Methodology

TES modeling guarantees fitting of the empirical distribution function by utilizing distortions of the form $D = \hat{H}_Y^{-1}(S_\xi)$. However, fitting the empirical autocorrelation function requires an extensive search procedure over the candidate pairs of (f_v, ξ). As a result, the problem is to find TES models whose autocorrelation function, $\rho_{f_v,\xi}$, determined by the pair (f_v, ξ) approximates its empirical counterpart, $\hat{\rho}_Y(\tau)$. Formally, for a fixed histogram inverse distribution, $\hat{H}_Y^{-1}$, the problem is to find an optimal innovation density and stitching parameter, (f_v^*, ξ^*), such that

$$(f_v^*, \xi^*) = \arg\min_{(f_v,\xi)} \{\sum_{t=1}^{T} [\rho_{f_v,\xi} - \hat{\rho}_Y(\tau)]^2\} \tag{6.11}$$

where T is the maximal autocorrelation lag to be approximated. The problem is similar to one described in [74]. Recall that, we use a subset of phase-type distributions as innovation variables consisting of mixtures of generalized Erlang (*MGE*) distributions, and ξ is in $[0, 1)$.

6.3.4 Outline of the Fitting Methodology

A brief outline of the empirical TES model fitting methodology is:

- Select the TES model, TES$^+$ or TES$^-$.
- Construct the empirical distribution function, $\hat{H}_Y$, from which $\hat{H}_Y^{-1}$ is easily constructed.
- Discretize the parameter space of ξ into a number of equidistant values (ξ is in $[0, 1)$).

- Start with an initial value of ξ, solve the optimization problem 6.11 with using *MGE* distributions.

Now, a TES model is fitted to the empirical data. If $\rho_X(\tau)$ sufficiently approximates $\hat{\rho}_Y(\tau)$, and simulated sample paths of the TES model resembles its empirical counterpart, the model is accepted. Otherwise, the search continues with different ξ values until a satisfactory model is found.

6.3.5 The TES Forecasting Methodology

TES-based forecasting is described in detail in [73]. However, because of its importance, we summarize it here. The forecast for τ periods ahead ($\tau = 0, 1, 2, \ldots$) given the current value of the background sequence is given by the following formula

$$F_X^+(u,\tau) = E[X^+_{n+\tau}|U^+_n = u] = \sum_{\nu=-\infty}^{\infty} \bar{\tilde{f}}_v(i2\pi\nu)^{\tau} s_v(\xi) e^{i2\pi\nu u} \tag{6.12}$$

where $\bar{\tilde{f}}_v$ is the complex conjugate of $\tilde{f}_v$, and

$$q_\nu(\xi) = \xi \int_0^1 e^{-i2\pi\nu\xi v} D(v)dv, \tag{6.13}$$

$$r_\nu(\xi) = (1-\xi) \int_0^1 e^{i2\pi\nu(1-\xi)v} D(v)dv, \tag{6.14}$$

$$s_\nu(\xi) = q_\nu(\xi) + r_\nu(\xi) \tag{6.15}$$

are the Fourier coefficients, with

$$s_0(\xi) = \int_0^1 D(v)dv = E[X_n^+]. \tag{6.16}$$

Here, the computation of conditional expectations, $E[X^+_{n+\tau}|U^+_n = u]$, used to forecast future values, is based on the current background event $U^+_n = u$. The problem is how to obtain u_n from the foreground sequence, x_n. Since a stitching transformation is usually 2-to-1, it follows that the mapping from x_n to u_n is 1-to-2, and one has to select between two possible values, as explained in [73]. This is done by retrograde forecasting, taking advantage of the fact that both a background TES sequence and its time-reversed version are both Markovian with known transition densities. One chooses that u_n whose use provides better retrograde forecasts.

6.3.6 Testing methodology

After we fit a TES model to an empirical data set using phase-type innovation variables, all the autocorrelations and transition densities of the model can be calculated using accurate analytical formulas. Utilizing the transition structure of the TES model, forecasts for future periods can be obtained as conditional expectations of the process given a current value. TES point estimation can be found in [73] and is summarized above.

We have implemented the TES forecasting procedure using the data from Dow Jones Utilities Index (DJUI), recorded between August 28 and December 18, 1972 [28, 80] A TES model was constructed by matching the empirical distribution and autocorrelation function, simultaneously. We have used mixtures of generalized Erlang (MGE) random variables in the construction of the TES model (as the innovation variables).

Here, it is important to comment on the innovation variables. In phase-type modeling of the innovations, it is desirable to have small K (number of phases) values, since smaller K values reduces computational burden (computation of $\tilde{f}_v(i2\pi\nu)$). In the procedure, we have started with small K values and incremented it successively. The case of $K = 1$ (exponential distribution) has yielded unsatisfactory TES models. However, we have achieved satisfactory models with $K = 2$. Larger values of K did not yield much better models whereas they increased the computational burden considerably. Consequently, we decided $K = 2$ is the minimal K value that yields satisfactory models.

Matlab Optimization Toolbox was utilized to solve problem 6.11. Although the toolbox uses standard algorithms for nonlinear optimization problems, there is no guarantee that the global minimum is achieved. These algorithms have been chosen for their robustness and efficiency.

Here, the search requires an additional parameter, N_ξ, where N_ξ is the number of equidistant values which ξ can take. Increasing N_ξ increases the computational requirements. On the other hand, smaller N_ξ values yield unsatisfactory TES models. It is critical to decide on the value of N_ξ where satisfactory TES models can be identified. In general, with $N_\xi = 10$, we were able to identify satisfactory TES models. Nevertheless, it is easy to increase N_ξ whenever it is necessary. The procedure described above is highly heuristic and the effort needed depends on the data set as well as the modeler.

Below, we have included the autocorrelation functions of the empirical data set and the candidate TES model. We can see the almost exact match between the autocorrelation functions of the empirical observations and the TES model from Figure 6.1.

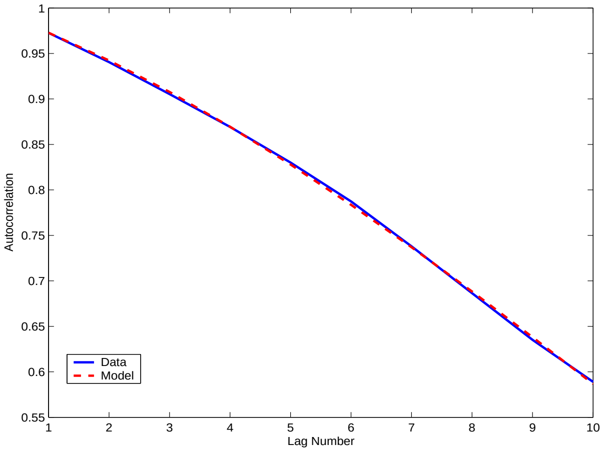


Figure 6.1: Autocorrelation functions of the DJUI data and the TES model

The DJUI data set has 78 points. We split the data into fit and test periods in order to check the accuracy of the forecasting method rigorously. The TES model was identified by using the first 68 points. Using the model, we estimated values of these 68 data points, which we refer to as *validation*. Then, we used the model for forecasting the values of the remaining 10 data points called *test-period*. Recall that, the computation of conditional expectations, $E[X^+_{n+\tau}|U^+_n = u]$, used to forecast future values, is based on the current background event $U^+_n = u$. Since the mapping from x_n to u_n is 1-to-2, we have to select between two possible values. In choosing the u_n, we forecast past ten values of the data series depending on both possible values. We choose that u_n whose use provides better retrograde forecasts.

Forecasts generated by the TES model using DJUI data are presented in Figure 6.2. The forecasts are calculated using expectations conditioned on the current value of the time series. Therefore, for a given data point, both one-period-ahead and multiple-period-ahead predictions can be computed. In order to show the goodness of fit of the TES model to the validation period data, we have started conditioning on the first data point and computed an estimate for the second period, conditioned on the second data point and computed an estimate for the third period, and so on. In Figure 6.2, the actual data and the one-period-ahead forecasts at every point are presented.

In addition, we have illustrated the accuracy of the one-period-ahead forecasts in the

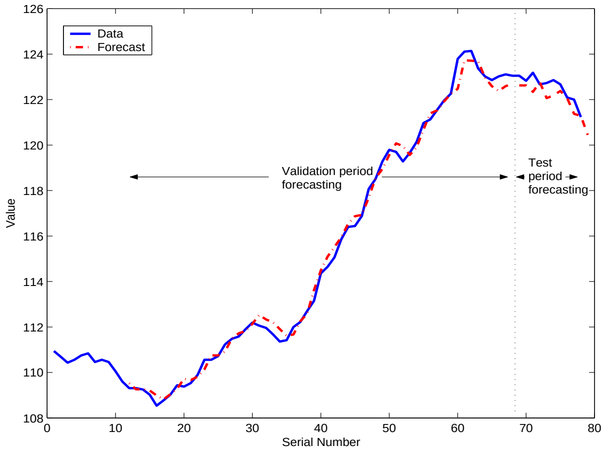


Figure 6.2: In-sample and out-of-sample forecasting for DJUI

test-period. In the last 10 data points in Figure 6.2, we have forecasted only at the out-of-sample period using the model developed based on the in-sample period. For forecasting performance, we have used the last 10 points of the data, and calculated the error measures such as the root mean squared error (RMSE) and the mean absolute percentage error (MAPE) given below. The forecasting error is obtained by $e_t = Y_t - F_t$ where Y_t is the time series data, and F_t is its forecast at time t as described above. RMSE is given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}}, \tag{6.17}$$

and MAPE is given by

$$MAPE = \frac{\sum_{i=1}^{n} |Y_i - F_i|/Y_i}{n} 100. \tag{6.18}$$

The resulting RMSE and MAPE for out-of-sample data are 0.4820 and 0.32%, respectively for DJUI data. A comparison of this result to traditional ARIMA models is given in next section.

6.4 Comparison of TES Forecasting to ARIMA Models

In this section, we compare the accuracy of TES forecasting methodology with the traditional ARIMA models using several empirical data sets. We have first utilized the TES modeling methodology to fit a model to an empirical data set and then generated forecasts using the fitted model. Then, we have developed ARIMA models using the same data set, generated forecasts, and finally, compared with TES forecasting using the error measures above.

Most of the data sets are borrowed from [28] and for all of them; we have specified part of the data except the last ten points as the validation period and the remaining as the test period. After fitting the models, forecasts were computed and the error measures were calculated using the test points.

The "Lake Huron" data shows the water level at Lake Huron in feet (reduced by 570), between 1875 and 1972 [28, 80]. Data set "Sales" is sales data from [25]. "Appc" represents private housing units started in the U.S.A. (monthly, from the Makridakis competition, series 922). Data set "Petroleum" is from *Monthly Energy Review* database and represents monthly total domestic field petroleum production from January 1984 to December 2003 (URL: http://www.eia.doe.gov/emeu/mer/petro.html). The "Sbl" data is the number of car drivers killed or seriously injured monthly in Great Britain for ten years beginning in January 1975. The "Deaths" data is monthly accidental deaths in the U.S.A. between 1973 and 1978 (National Safety Council). All the computations were conducted by using Matlab Release 12.1. Matlab Optimization Toolbox was utilized to fit the TES models. Forecasts generated by the TES model using above data sets are presented in Figure 6.3.

Table 6.1 shows the computational results for both forecasting methods. In the first row, we have used the DJUI data to compare the two forecasting procedures. TES forecasting procedure yields RMSE = 0.4820 and MAPE = 0.32%. In the meantime, ARIMA model yields RMSE = 0.4366 and MAPE = 0.29%. Other rows are interpreted accordingly. The table also shows the detail in the fitted ARIMA model. For DJUI data, the identified ARIMA model is an AR model of order 1 to the transformed data (differenced at lag 1).

DATA SET	**TES**		**ARIMA**		
	RMSE	MAPE	RMSE	MAPE	MODEL
DJUI	0.4820	%0.32	0.4366	%0.29	(1,1,0)
Lake Huron	0.7441	%7.65	0.7466	%7.6	(1,0,1)
Sales	1.4561	%0.45	0.9861	%0.31	(1,1,1)
Appc	149.27	%6.81	169.89	%7.42	(1,1,3)
Petroleum	125.31	%1.49	106.03	%0.95	(3,1,3)
Sbl	155.11	%9.50	142.82	%7.76	(5,1,4)
Deaths	452.21	%3.57	296.10	%2.43	(1,1,1)

Table 6.1: Square root of MSE and MAPE for TES and ARIMA models

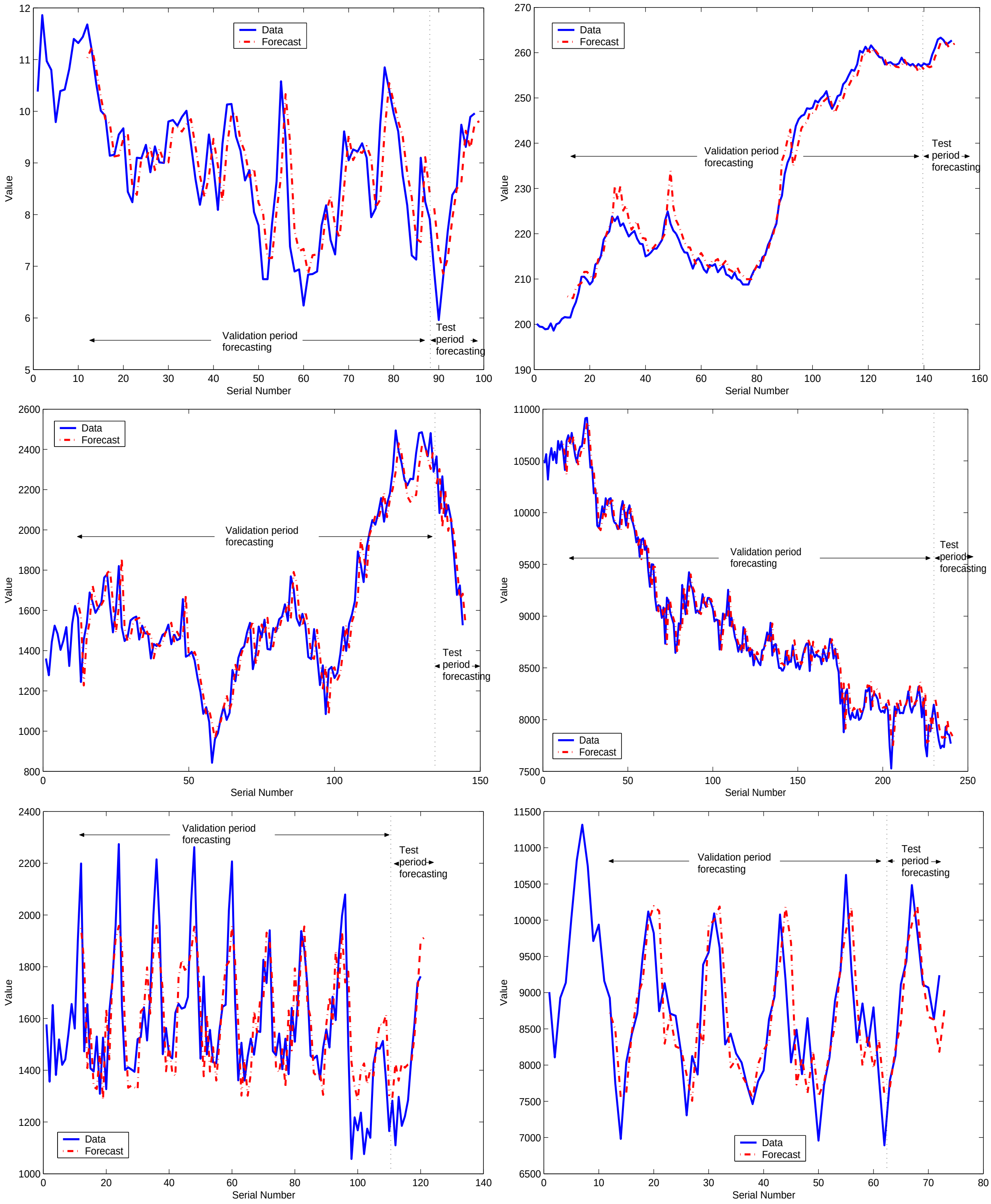


Figure 6.3: In-sample and out-of-sample forecasting for Lake Huron, Sales, Appc, Petroleum, Sbl, Deaths data sets

As can be seen from the computational results, TES forecasting methodology yields forecasts as accurate as ARIMA models. This makes TES forecasting procedure an attractive complement to time series models, especially when data exhibits high autocorrelations.

However, as the values of the autocorrelations decrease, the accuracy of the forecasts generated by the TES model decreases. The underlying reason is that identifying a satisfactory TES model for a given data set becomes more difficult and time consuming (but yet doable) as the value of the empirical autocorrelations decreases.

The data sets included were highly autocorrelated, which is appropriate for TES modeling. Our analysis suggests that in addition to its analytical modeling of autocorrelated time series and Monte Carlo simulation, use of TES models as a forecasting tool yields forecasts as accurate as other time series models. Furthermore, using phase-type random variables as the innovation density considerably decreases the search effort for model fitting, which in turn makes it possible to frequently update the fitted model as new data arrive. In addition, TES processes are extremely useful in modeling empirical data series, especially in capturing autocorrelations.

Chapter 7
Conclusion and Future Research

In this thesis, we have studied a typical supply chain consisting of a supplier, a plant, a DC, and a retailer. We have used batch ordering inventory policies to control material flow between stages. The supply chain is capacitated in the sense that it has a finite production rate, and transit times between echelons are random variables. We have presented an efficient decomposition technique to obtain long-run performance measures of the system: time averages of inventory and backorder levels as well as the customer service levels. The model was validated against simulation, yielding good agreement in robust performance metrics.

The metrics were then used within an optimization framework to help design the supply chain. The objective of optimization in our problem is to determine appropriate production and inventory policy parameters. We have employed a cost-minimizing objective function that assigns penalties for holding inventory and shortages to solve the optimization problem. In addition, a penalty per set-up or ordering is charged to avoid excessive set-ups or replenishment orders, respectively. The outcome of the optimization framework specifies not only how much and where to hold inventory but also how to move inventory across the supply chain, i.e., reorder levels and replenishment batch sizes.

The proposed model takes into account the interactions between the echelons, especially the demand process that propagates backward to the upstream stages and the lead time process that propagates forward to the downstream stages. Moreover, it requires limited computational requirements, which in turn helps update the performance measures and optimal system parameters frequently so as to be more responsive to short-term changes in demand or supply. In addition, it can be used as a decision support system for effective decision making as opposed to using simplistic inventory models, which results in significantly higher operating costs. Also, the model accommodates both backordering and partial lost sales assumptions. Modeling difference between backorder and lost sales cases is that analysis of the effective demand inter-arrival and procurement times is different. The procurement times are simpler; however, the demand inter-arrival times are more involved. We assume that the actual demand is lost only at the retailer.

In a similar vein, we have considered a distribution inventory system with one warehouse and several retailers. The challenge in this system is to describe the demand arrival process at the warehouse. We have proposed a procedure to characterize the demand arrival process at the warehouse as a superposition of several independent Erlang processes. This characterization can also be applied to a queue to which the arrival process is the superposition of separate arrival streams, each of whose inter-arrival times is of Erlang distribution. We have presented a methodology to characterize such arrival streams as Markovian processes which have been extended to phase-type arrival streams as well. Our methodology exactly describes the superposed process, however the state-space of the proposed Markovian process increases considerably. We, in addition, have developed a three-moment approximation scheme to efficiently use the methodology in practice.

Finally, we have reported an experimental study that compares TES process forecasting to traditional Box-Jenkins ARIMA models. TES is a methodology [71, 72] to model empirical time series from a stationary probability law. Its merit is to capture both the empirical distribution and autocorrelation function, simultaneously. Our analysis suggests that in addition to its analytical modeling of autocorrelated time series and Monte Carlo simulation, use of TES models as a forecasting tool yields forecasts as accurate as other time series models. Furthermore, using phase-type random variables as the innovation density considerably decreases the search effort for model fitting, which in turn makes it possible to frequently update the fitted model as new data arrive. In addition, TES processes are extremely useful in modeling empirical data series, especially in capturing autocorrelations.

Several extensions to this study are as follows. An important issue is to investigate the stability of the supply chain. Most studies assume an unlimited capacity for the plants. Here, we have assumed a finite production rate. Thus, an ineffective policy may lead to high backorder levels. An exception was [59] in which they have investigated the stability of a multi-echelon system under a base-stock policy. We will look for conditions where the inventory and backorder levels are stable.

For both the backorder and the lost sales case, the Poisson demand assumption does not model the real world problem accurately. A compound Poisson process is a more general assumption. Therefore, we will investigate this problem as well. In addition, the optimization procedure can very well be applied to inventory distribution system as well.

Appendix A
Phase-Type Distributions

Consider a Markov chain with state space $1, 2, \ldots, k, k+1$ where $k+1$ being the absorbing state with infinitesimal generator

$$\mathbf{Q} = \begin{bmatrix} \mathbf{T} & \underline{T}^0 \\ \mathbf{0} & 0 \end{bmatrix}$$

where the $m \times m$ matrix $\mathbf{T}$ satisfies $T_{ii} < 0$, for $1 \leq i \leq m$, $T_{ij} \geq 0$ for $i \neq j$, and $\mathbf{T}\underline{e}+\underline{T}^0 = \underline{0}$, where $\underline{e}$ is the unit column vector. States $1, 2, \ldots, k$ are transient, so that absorption into state $k+1$, from any initial state, is certain. The initial probability vector of the Markov chain is given by $(\underline{\alpha}, \alpha_{k+1})$, with $\underline{\alpha}\ \underline{e} + \alpha_{k+1} = 1$. Then, the distribution of the random variable representing the time until absorption in the above Markov chain is said to be of phase-type with an $(\underline{\alpha}, \mathbf{T})$ representation. It is a probability distribution on $[0, \infty)$.

Let X be a phase-type random variable with an $(\underline{\alpha}, \mathbf{T})$ representation. The moments of X are all finite, and are given by

$$E[X^n] = (-1)^n n! (\underline{\alpha}\mathbf{T}^{-n}\underline{e})$$

for $n \geq 1$. In addition, the density function of X is

$$f_X(x) = \underline{\alpha} e^{(\mathbf{T}x)} \underline{T}^0$$

for $x \geq 0$.

Phase-type distributions are dense on $[0, \infty)$. They are closed under some operations. The convolution of phase-type distributions is also a phase-type distribution. In addition, a finite mixture of phase-type distributions is also a phase-type distribution

Appendix B

Impact of Cost on System Parameters

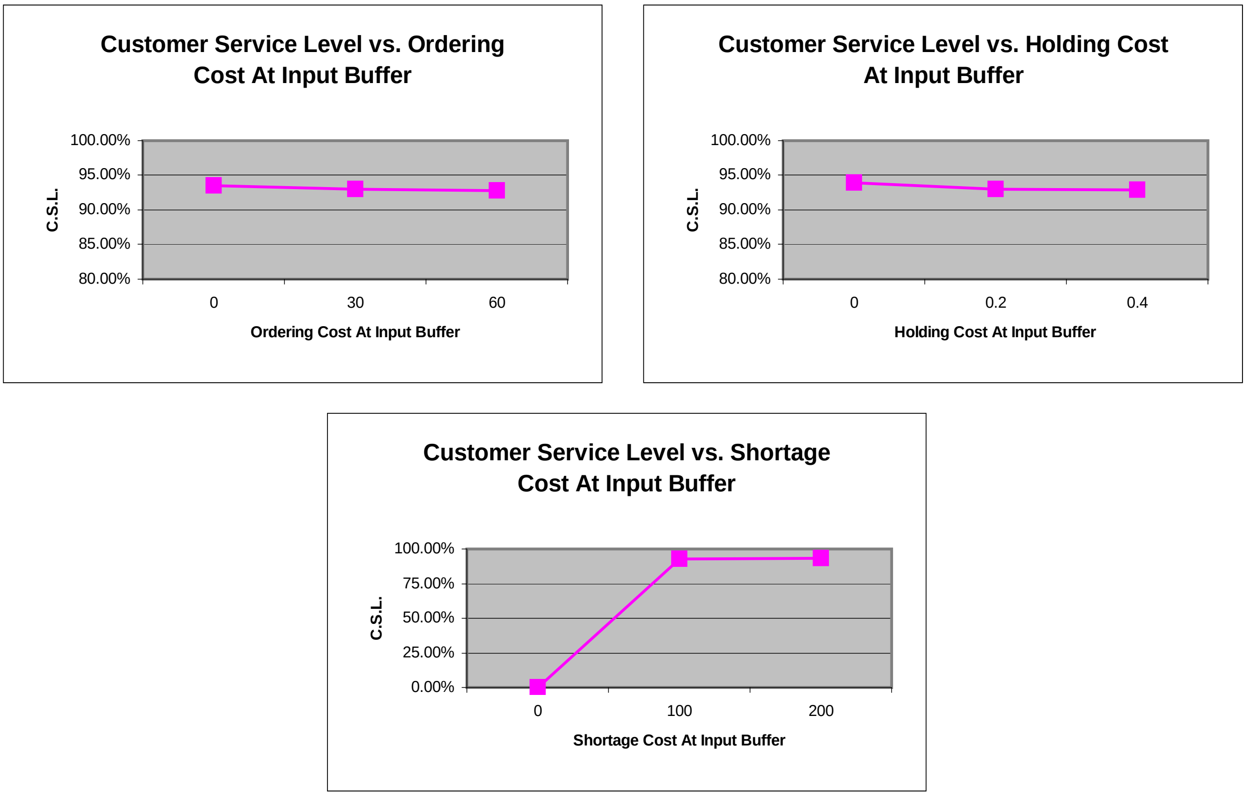


Figure B.1: Input buffer attributes that drive the performance of the supply chain

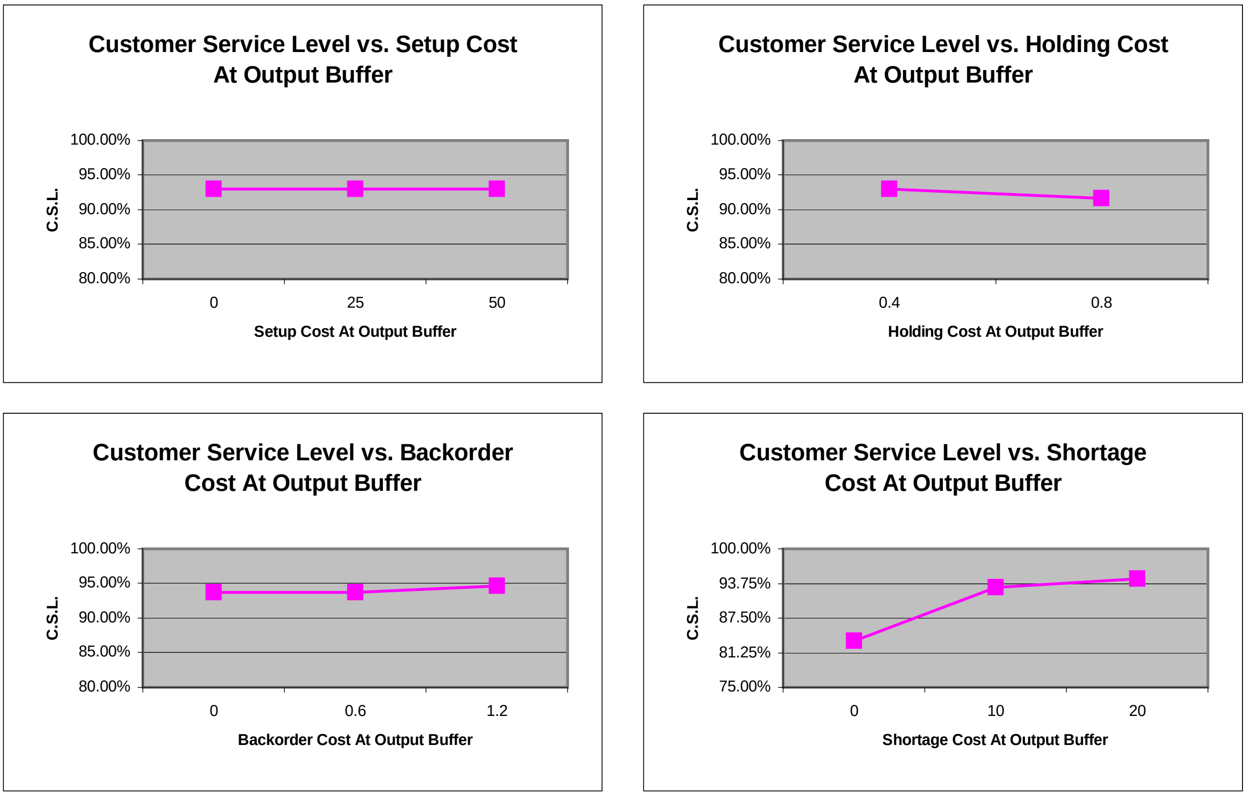


Figure B.2: Output buffer attributes that drive the overall performance of the supply chain

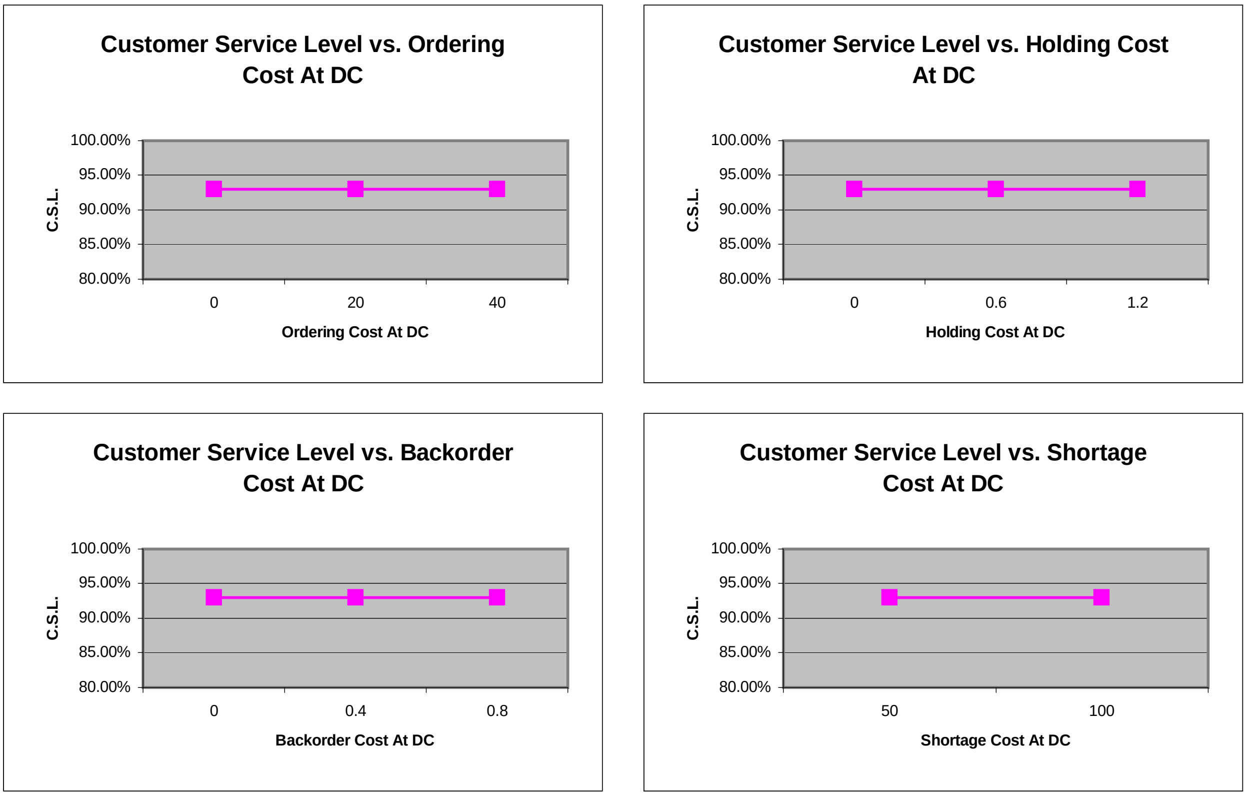


Figure B.3: Distribution center attributes that drive the overall performance of the supply chain

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 2$	$\beta_S = 1$	***K***	30	25	20	15
λ=1.4	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	$a = 0.1$	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	6.2957	10.8764	14.5133	9.6983	41.3837	95.99%
14	9	29	9	19	9	11	4	6.0535	10.8199	12.578	9.8177	39.2691	
		21	1	13	10				6.5424	12.2547		34.6683	
15	8	21	1	12	10	11	4	6.1106	6.5426	11.8733	9.7805	34.3071	
		19	1	12	10				6.1977	11.8733		33.9621	
16	7	19	1	12	10	11	4	5.9572	6.2113	11.7797	9.7687	33.7169	
		19	1	12	10				6.2113	11.7797		33.7169	
17	7	19	1	12	10	11	4	5.8713	6.2094	11.7781	9.7685	33.6273	
		19	1	12	10				6.2094	11.7781		33.6273	
18	7	19	1	12	10	11	4	5.8063	6.2076	11.7767	9.7684	33.559	
		19	1	12	10				6.2076	11.7767		33.559	
19	7	19	1	12	10	11	4	5.7587	6.2061	11.7754	9.7682	33.5084	
		19	1	12	10				6.2061	11.7754		33.5084	
20	7	19	1	12	10	11	4	5.7259	6.2047	11.7743	9.7681	33.4729	
		19	1	12	10				6.2047	11.7743		33.4729	
21	7	19	1	12	10	11	4	5.7057	6.2034	11.7733	9.7679	33.4504	
		19	1	12	10				6.2034	11.7733		33.4504	
22	6	19	1	12	10	11	4	5.6899	6.2143	11.7821	9.769	33.4553	
		19	1	12	10				6.2143	11.7821		33.4553	
23	6	19	1	12	10	11	4	5.6817	6.2127	11.7808	9.7689	33.4441	
		19	1	12	10				6.2127	11.7808		33.4441	
23	6	19	1	12	10	11	4	5.6817	6.2127	11.7808	9.7689	33.4441	
		19	1	12	10				6.2127	11.7808		33.4441	
													C.S.L
23	6	19	1	12	10	11	4	5.6817	6.2127	11.7808	9.7689	33.4441	94.12%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.1: Convergence path of the optimization procedure with single steps (medium production rate, $\lambda = 1.4$)

				I.B.	O.B.	DC	Retailer
	μ_1 = 2	β_S = 1	***K***	30	25	20	15
λ=1.45	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		**O.B.**		**DC**		**Retailer**				**Cost**			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	**I.B.**	**O.B.**	**DC**	**Retailer**	**TC**	**C.S.L**
13	***10***	***30***	***10***	***20***	***10***	***10***	***5***	***6.4032***	***11.5507***	***14.3726***	***9.905***	**42.2315**	**95.56%**
14	***9***	***30***	***10***	***19***	***9***	***10***	***5***	***6.1571***	***11.5969***	***13.0914***	***10.1169***	**40.9624**	
		30	***10***	***20***	***1***				***11.5969***	***9.0222***		**36.8931**	
15	***8***	***30***	***10***	***20***	***1***	***10***	***5***	***5.979***	***11.6641***	***9.0102***	***9.9056***	**36.5589**	
		30	***10***	***20***	***1***				***11.6641***	***9.0102***		**36.5589**	
16	***8***	***30***	***10***	***20***	***1***	***10***	***5***	***5.8687***	***11.6491***	***9.0128***	***9.9055***	**36.4362**	
		30	***10***	***20***	***1***				***11.6491***	***9.0128***		**36.4362**	
17	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.7799***	***11.7418***	***8.9967***	***9.906***	**36.4244**	
		30	***10***	***20***	***1***				***11.7418***	***8.9967***		**36.4244**	
18	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.7041***	***11.7234***	***8.9998***	***9.9059***	**36.3333**	
		30	***10***	***20***	***1***				***11.7234***	***8.9998***		**36.3333**	
19	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.6478***	***11.7068***	***9.0027***	***9.9058***	**36.2632**	
		30	***10***	***20***	***1***				***11.7068***	***9.0027***		**36.2632**	
20	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.6072***	***11.692***	***9.0053***	***9.9057***	**36.2102**	
		30	***10***	***20***	***1***				***11.692***	***9.0053***		**36.2102**	
21	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.58***	***11.6787***	***9.0076***	***9.9057***	**36.172**	
		30	***10***	***20***	***1***				***11.6787***	***9.0076***		**36.172**	
22	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.5644***	***11.6667***	***9.0097***	***9.9056***	**36.1464**	
		30	***10***	***20***	***1***				***11.6667***	***9.0097***		**36.1464**	
23	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.5588***	***11.6559***	***9.0116***	***9.9056***	**36.1319**	
		30	***10***	***20***	***1***				***11.6559***	***9.0116***		**36.1319**	
23	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.5589***	***11.6559***	***9.0117***	***9.9056***	**36.132**	
		30	***10***	***20***	***1***				***11.6559***	***9.0117***		**36.132**	
													C.S.L
23	***7***	***30***	***10***	***20***	***1***	***10***	***5***	***5.5589***	***11.6559***	***9.0117***	***9.9056***	**36.132**	**95.55%**

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.2: Convergence path of the optimization procedure with single steps (medium production rate, $\lambda = 1.45$)

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 2$	$\beta_S = 1$	***K***	30	25	20	15
λ=1.55	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	$a = 0.1$	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	6.6193	15.7386	29.6206	29.1049	81.0834	46.82%
14	9	31	11	20	11	11	6	6.3657	15.7494	23.9241	26.0196	72.0589	
		40	18	22	12				15.1269	21.937		69.4493	
15	8	41	19	22	12	12	7	6.1891	15.3813	19.8488	14.4812	55.9004	
		42	18	24	13				15.5231	19.8269		56.0203	
16	8	42	18	24	13	13	6	6.0641	15.4692	19.895	13.5162	54.9445	
		43	17	26	14				15.6432	20.2277		55.4513	
17	8	43	17	26	14	14	5	5.9666	15.5951	20.2546	12.9668	54.7831	
		45	17	28	15				15.7874	20.8429		55.5638	
18	8	45	17	28	15	14	5	5.891	15.7514	20.8025	12.4623	54.9072	
		45	17	28	15				15.7514	20.8025		54.9072	
19	7	46	18	28	15	15	5	5.8237	16.042	21.1513	12.7809	55.7979	
		47	17	30	16				16.2396	21.8799		56.7241	
20	7	47	17	30	16	14	5	5.7736	16.1861	21.8485	12.5054	56.3136	
		45	17	28	15				15.9922	21.1358		55.4069	
21	7	45	17	28	15	15	5	5.7323	15.9432	21.0633	12.7912	55.53	
		47	17	30	16				16.1333	21.7917		56.4485	
22	7	47	17	30	16	14	5	5.7124	16.1347	21.7616	12.3844	55.9931	
		45	17	28	15				15.897	21.0082		55.002	
23	7	45	17	28	15	15	5	5.6943	15.86	20.9476	12.6494	55.1513	
		47	17	30	16				16.0663	21.7176		56.1276	
24	7	47	17	30	16	14	5	5.6948	16.0373	21.6973	12.2683	55.6978	
		45	17	28	15				15.8221	20.9028		54.688	
24	7	45	17	28	15	14	5	5.6903	15.8255	20.9006	12.5921	55.0085	
		45	17	28	15				15.8255	20.9006		55.0085	
24	7	45	17	28	15	14	5	5.6923	15.8255	20.8992	12.5905	55.0075	
		45	17	28	15				15.8255	20.8992		55.0075	
													C.S.L
24	7	45	17	28	15	14	5	5.6923	15.8132	20.8847	12.5716	54.9617	90.38%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.3: Convergence path of the optimization procedure with single steps (medium production rate, $\lambda = 1.55$)

				I.B.	O.B.	DC	Retailer
	μ_1 = 3	β_S = 1	***K***	30	25	20	15
λ=1.9	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		**O.B.**		**DC**		**Retailer**				**Cost**			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	**I.B.**	**O.B.**	**DC**	**Retailer**	**TC**	**C.S.L**
13	*10*	*30*	*10*	*20*	*10*	*10*	*5*	*7.8487*	*12.6389*	*15.1333*	*12.9898*	**48.6108**	**90.61%**
14	*11*	*30*	*10*	*20*	*9*	*11*	*6*	*7.5749*	*12.4962*	*14.4187*	*12.1765*	**46.6663**	
		32	*10*	*22*	*7*				*12.6874*	*13.6487*		**46.0875**	
15	*10*	*32*	*10*	*22*	*6*	*12*	*6*	*7.3696*	*12.7608*	*13.1107*	*12.0222*	**45.2633**	
		34	*10*	*24*	*4*				*12.9893*	*12.4533*		**44.8344**	
16	*10*	*34*	*10*	*24*	*3*	*13*	*6*	*7.1913*	*12.9704*	*11.859*	*11.9749*	**43.9955**	
		35	*9*	*26*	*1*				*13.2233*	*11.2979*		**43.6874**	
17	*10*	*35*	*9*	*26*	*1*	*13*	*6*	*7.0464*	*13.2042*	*11.2901*	*11.9642*	**43.5048**	
		35	*9*	*26*	*1*				*13.2042*	*11.2901*		**43.5048**	
18	*10*	*35*	*9*	*26*	*1*	*13*	*6*	*6.9186*	*13.1874*	*11.2867*	*11.9629*	**43.3556**	
		35	*9*	*26*	*1*				*13.1874*	*11.2867*		**43.3556**	
19	*10*	*35*	*9*	*26*	*1*	*13*	*6*	*6.8207*	*13.1704*	*11.2836*	*11.9617*	**43.2364**	
		35	*9*	*26*	*1*				*13.1704*	*11.2836*		**43.2364**	
20	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.7355*	*13.2533*	*11.3011*	*11.968*	**43.2578**	
		35	*9*	*26*	*1*				*13.2533*	*11.3011*		**43.2578**	
21	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.6638*	*13.2353*	*11.2967*	*11.9665*	**43.1623**	
		35	*9*	*26*	*1*				*13.2353*	*11.2967*		**43.1623**	
22	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.6093*	*13.2173*	*11.2929*	*11.9652*	**43.0846**	
		35	*9*	*26*	*1*				*13.2173*	*11.2929*		**43.0846**	
23	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.5682*	*13.2031*	*11.2896*	*11.964*	**43.025**	
		35	*9*	*26*	*1*				*13.2031*	*11.2896*		**43.025**	
24	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.539*	*13.1867*	*11.2868*	*11.9629*	**42.9754**	
		35	*9*	*26*	*1*				*13.1867*	*11.2868*		**42.9754**	
25	*9*	*35*	*9*	*26*	*1*	*13*	*6*	*6.5201*	*13.1758*	*11.2844*	*11.962*	**42.9422**	
		35	*9*	*26*	*1*				*13.1758*	*11.2844*		**42.9422**	
26	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.5094*	*13.2623*	*11.3035*	*11.9688*	**43.0439**	
		35	*9*	*26*	*1*				*13.2623*	*11.3035*		**43.0439**	
27	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.4989*	*13.2479*	*11.2998*	*11.9676*	**43.0143**	
		35	*9*	*26*	*1*				*13.2479*	*11.2998*		**43.0143**	
28	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.4981*	*13.2336*	*11.2965*	*11.9664*	**42.9946**	
		35	*9*	*26*	*1*				*13.2336*	*11.2965*		**42.9946**	
28	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.4983*	*13.2331*	*11.2965*	*11.9664*	**42.9943**	
		35	*9*	*26*	*1*				*13.2331*	*11.2965*		**42.9943**	
													C.S.L
28	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.4983*	*13.2331*	*11.2965*	*11.9664*	**42.9943**	**95.05%**

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.4: Convergence path of the optimization procedure with single steps (high production rate)

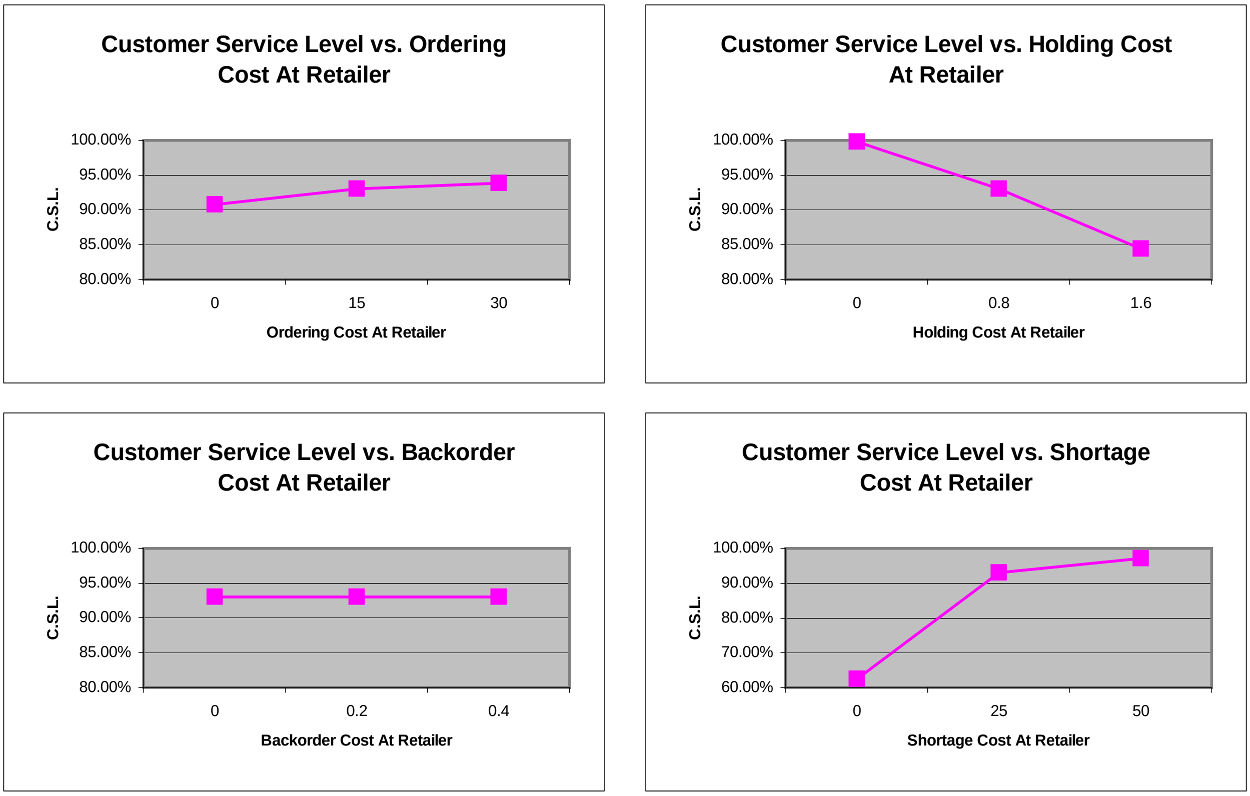


Figure B.4: Retailer attributes that drive the overall performance of the supply chain

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 3$	$\beta_S = 1$	***K***	30	25	20	15
λ=1.9	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	a = 0.1	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		**O.B.**		**DC**		**Retailer**				**Cost**			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	**I.B.**	**O.B.**	**DC**	**Retailer**	**TC**	**C.S.L**
13	*10*	*30*	*10*	*20*	*10*	*10*	*5*	*7.8487*	*12.6389*	*15.1333*	*12.9898*	**48.6108**	**90.61%**
28	*8*	*30*	*10*	*20*	*11*	*11*	*6*	*6.4843*	*12.5576*	*14.7393*	*12.1938*	**45.975**	
		32	*10*	*22*	*9*				*12.745*	*14.8382*		**46.2613**	
28	*8*	*32*	*10*	*22*	*1*	*12*	*6*	*6.4981*	*12.7443*	*10.216*	*12.0193*	**41.4777**	
		34	*10*	*24*	*1*				*12.9737*	*10.6983*		**42.1894**	
28	*8*	*34*	*10*	*24*	*1*	*13*	*6*	*6.5018*	*12.9761*	*10.695*	*11.9755*	**42.1484**	
		35	*9*	*26*	*1*				*13.2301*	*11.2989*		**43.0064**	
28	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.5049*	*13.231*	*11.2961*	*11.9663*	**42.9983**	
		35	*9*	*26*	*1*				*13.231*	*11.2961*		**42.9983**	
													C.S.L
28	*8*	*35*	*9*	*26*	*1*	*13*	*6*	*6.4983*	*13.2331*	*11.2965*	*11.9664*	**42.9943**	**95.05%**

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

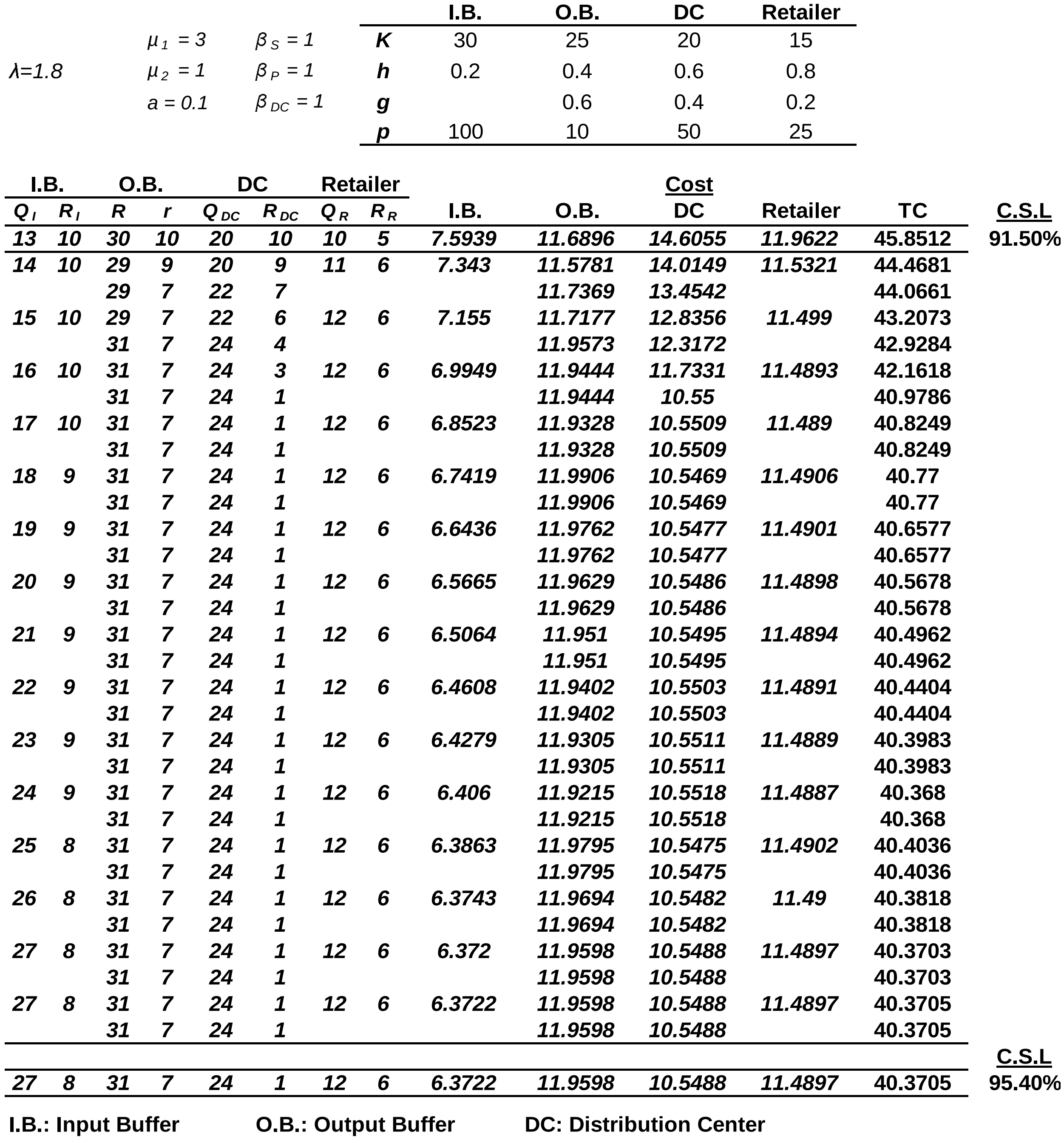
Table B.5: Convergence path of the optimization procedure with optimized steps (high production rate)

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 3$	$\beta_S = 1$	**K**	30	25	20	15
λ=1.9	$\mu_2 = 1$	$\beta_P = 1$	**h**	0.2	0.4	0.6	0.8
	a = 0.1	$\beta_{DC} = 1$	**g**		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
22	20	15	12	10	15	5	20	8.1495	25.0978	18.504	142.7724	194.5237	93.44%
27	8	22	12	10	16	6	19	6.438	12.9765	15.4515	20.8737	55.7396	
		23	11	12	14				12.5533	14.5769		54.4418	
27	8	23	11	12	13	7	18	6.4586	12.557	14.0476	18.0478	51.1111	
		25	11	14	11				12.3685	13.7692		50.6441	
27	8	25	11	14	8	8	17	6.4714	12.3781	12.0205	17.3436	48.2135	
		27	11	16	9				12.3451	12.6054		48.7654	
27	8	27	11	16	9	8	16	6.4808	12.3518	12.602	16.6948	48.1294	
		27	11	16	9				12.3518	12.602		48.1294	
28	8	27	11	16	9	8	15	6.4731	12.343	12.598	16.0145	47.4286	
		27	11	16	9				12.343	12.598		47.4286	
28	8	27	11	16	9	9	14	6.4732	12.3429	12.598	15.362	46.7761	
		29	11	18	7				12.4157	13.1776		47.4284	
28	8	29	11	18	1	9	13	6.488	12.4156	9.7943	14.7095	43.4074	
		29	11	18	1				12.4156	9.7943		43.4074	
28	8	29	11	18	1	9	12	6.4804	12.4193	9.8011	14.1007	42.8014	
		29	11	18	1				12.4193	9.8011		42.8014	
28	8	29	11	18	1	10	11	6.4803	12.4193	9.8011	13.5479	42.2486	
		30	10	20	1				12.5537	9.8513		42.4332	
28	8	30	10	20	1	10	10	6.4936	12.5537	9.9366	13.0214	42.0053	
		30	10	20	1				12.5537	9.9366		42.0053	
28	8	30	10	20	1	11	9	6.4862	12.5568	9.9406	12.6092	41.5928	
		32	10	22	1				12.7443	10.2287		42.0683	
28	8	32	10	22	1	11	8	6.4981	12.7443	10.216	12.2417	41.7001	
		32	10	22	1				12.7443	10.216		41.7001	
28	8	32	10	22	1	12	7	6.4909	12.7467	10.2179	12.0441	41.4997	
		34	10	24	1				12.976	10.6994		42.2105	
28	8	34	10	24	1	13	6	6.5018	12.9762	10.695	11.9755	42.1484	
		35	9	26	1				13.2301	11.2989		43.0063	
28	8	35	9	26	1	13	6	6.5049	13.2313	11.2961	11.9663	42.9986	
		35	9	26	1				13.2313	11.2961		42.9986	
													C.S.L
28	8	35	9	26	1	13	6	6.4983	13.2331	11.2965	11.9664	42.9943	95.05%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.6: Convergence path of the optimization procedure with optimized steps (high production rate)

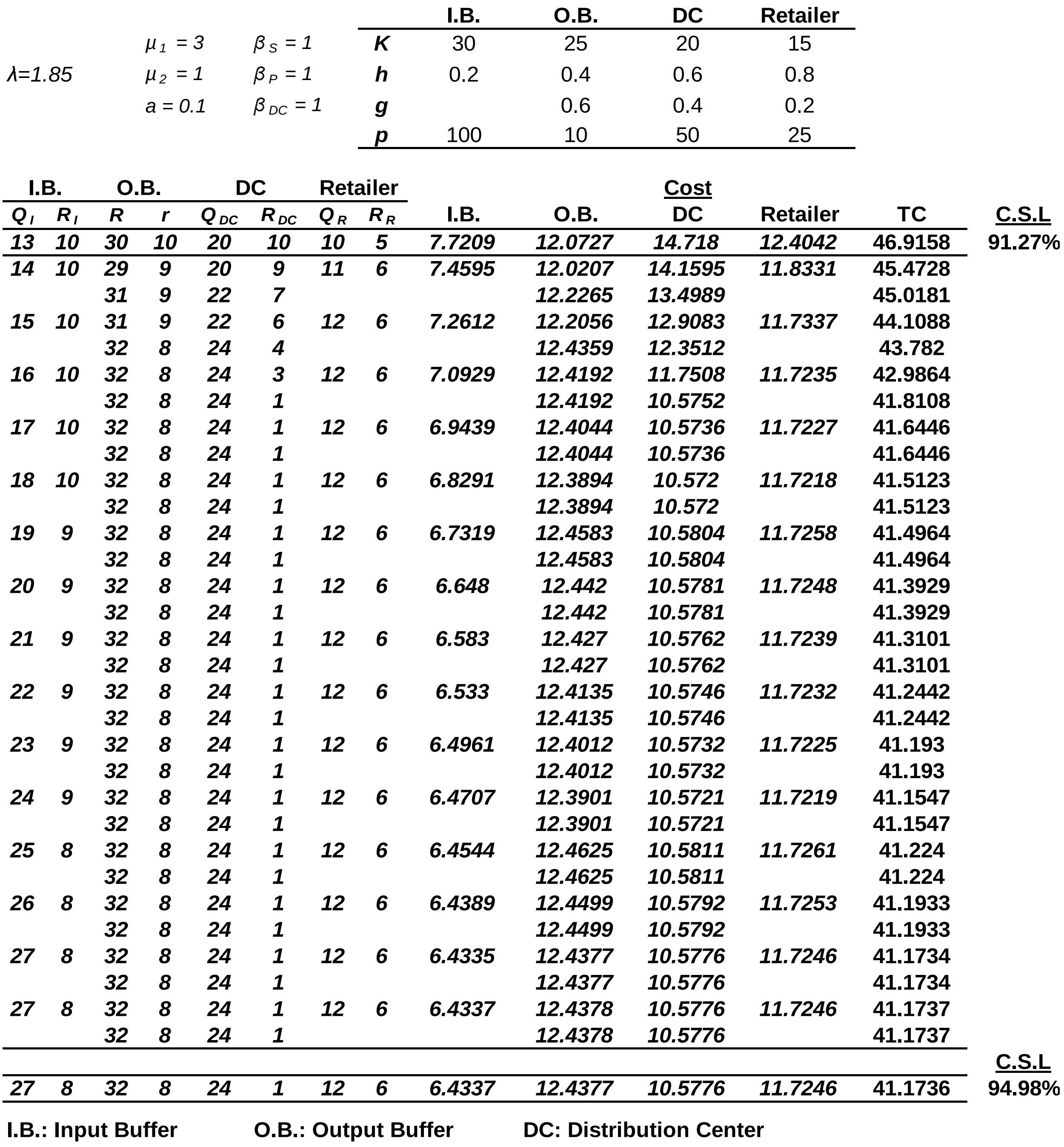


				I.B.	O.B.	DC	Retailer
	μ_1 = 3	β_S = 1	**K**	30	25	20	15
λ=1.8	μ_2 = 1	β_P = 1	**h**	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	**g**		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	7.5939	11.6896	14.6055	11.9622	45.8512	91.50%
14	10	29	9	20	9	11	6	7.343	11.5781	14.0149	11.5321	44.4681	
		29	7	22	7				11.7369	13.4542		44.0661	
15	10	29	7	22	6	12	6	7.155	11.7177	12.8356	11.499	43.2073	
		31	7	24	4				11.9573	12.3172		42.9284	
16	10	31	7	24	3	12	6	6.9949	11.9444	11.7331	11.4893	42.1618	
		31	7	24	1				11.9444	10.55		40.9786	
17	10	31	7	24	1	12	6	6.8523	11.9328	10.5509	11.489	40.8249	
		31	7	24	1				11.9328	10.5509		40.8249	
18	9	31	7	24	1	12	6	6.7419	11.9906	10.5469	11.4906	40.77	
		31	7	24	1				11.9906	10.5469		40.77	
19	9	31	7	24	1	12	6	6.6436	11.9762	10.5477	11.4901	40.6577	
		31	7	24	1				11.9762	10.5477		40.6577	
20	9	31	7	24	1	12	6	6.5665	11.9629	10.5486	11.4898	40.5678	
		31	7	24	1				11.9629	10.5486		40.5678	
21	9	31	7	24	1	12	6	6.5064	11.951	10.5495	11.4894	40.4962	
		31	7	24	1				11.951	10.5495		40.4962	
22	9	31	7	24	1	12	6	6.4608	11.9402	10.5503	11.4891	40.4404	
		31	7	24	1				11.9402	10.5503		40.4404	
23	9	31	7	24	1	12	6	6.4279	11.9305	10.5511	11.4889	40.3983	
		31	7	24	1				11.9305	10.5511		40.3983	
24	9	31	7	24	1	12	6	6.406	11.9215	10.5518	11.4887	40.368	
		31	7	24	1				11.9215	10.5518		40.368	
25	8	31	7	24	1	12	6	6.3863	11.9795	10.5475	11.4902	40.4036	
		31	7	24	1				11.9795	10.5475		40.4036	
26	8	31	7	24	1	12	6	6.3743	11.9694	10.5482	11.49	40.3818	
		31	7	24	1				11.9694	10.5482		40.3818	
27	8	31	7	24	1	12	6	6.372	11.9598	10.5488	11.4897	40.3703	
		31	7	24	1				11.9598	10.5488		40.3703	
27	8	31	7	24	1	12	6	6.3722	11.9598	10.5488	11.4897	40.3705	
		31	7	24	1				11.9598	10.5488		40.3705	
													C.S.L
27	8	31	7	24	1	12	6	6.3722	11.9598	10.5488	11.4897	40.3705	95.40%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.7: Convergence path of the optimization procedure with single steps (high production rate, $\lambda = 1.8$))



				I.B.	O.B.	DC	Retailer
	μ_1 = 3	β_S = 1	**K**	30	25	20	15
λ=1.85	μ_2 = 1	β_P = 1	**h**	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	**g**		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	7.7209	12.0727	14.718	12.4042	46.9158	91.27%
14	10	29	9	20	9	11	6	7.4595	12.0207	14.1595	11.8331	45.4728	
		31	9	22	7				12.2265	13.4989		45.0181	
15	10	31	9	22	6	12	6	7.2612	12.2056	12.9083	11.7337	44.1088	
		32	8	24	4				12.4359	12.3512		43.782	
16	10	32	8	24	3	12	6	7.0929	12.4192	11.7508	11.7235	42.9864	
		32	8	24	1				12.4192	10.5752		41.8108	
17	10	32	8	24	1	12	6	6.9439	12.4044	10.5736	11.7227	41.6446	
		32	8	24	1				12.4044	10.5736		41.6446	
18	10	32	8	24	1	12	6	6.8291	12.3894	10.572	11.7218	41.5123	
		32	8	24	1				12.3894	10.572		41.5123	
19	9	32	8	24	1	12	6	6.7319	12.4583	10.5804	11.7258	41.4964	
		32	8	24	1				12.4583	10.5804		41.4964	
20	9	32	8	24	1	12	6	6.648	12.442	10.5781	11.7248	41.3929	
		32	8	24	1				12.442	10.5781		41.3929	
21	9	32	8	24	1	12	6	6.583	12.427	10.5762	11.7239	41.3101	
		32	8	24	1				12.427	10.5762		41.3101	
22	9	32	8	24	1	12	6	6.533	12.4135	10.5746	11.7232	41.2442	
		32	8	24	1				12.4135	10.5746		41.2442	
23	9	32	8	24	1	12	6	6.4961	12.4012	10.5732	11.7225	41.193	
		32	8	24	1				12.4012	10.5732		41.193	
24	9	32	8	24	1	12	6	6.4707	12.3901	10.5721	11.7219	41.1547	
		32	8	24	1				12.3901	10.5721		41.1547	
25	8	32	8	24	1	12	6	6.4544	12.4625	10.5811	11.7261	41.224	
		32	8	24	1				12.4625	10.5811		41.224	
26	8	32	8	24	1	12	6	6.4389	12.4499	10.5792	11.7253	41.1933	
		32	8	24	1				12.4499	10.5792		41.1933	
27	8	32	8	24	1	12	6	6.4335	12.4377	10.5776	11.7246	41.1734	
		32	8	24	1				12.4377	10.5776		41.1734	
27	8	32	8	24	1	12	6	6.4337	12.4378	10.5776	11.7246	41.1737	
		32	8	24	1				12.4378	10.5776		41.1737	
													C.S.L
27	8	32	8	24	1	12	6	6.4337	12.4377	10.5776	11.7246	41.1736	94.98%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.8: Convergence path of the optimization procedure with single steps (high production rate, $\lambda = 1.85$))

				I.B.	O.B.	DC	Retailer
	μ_1 = 3	β_S = 1	***K***	30	25	20	15
λ=1.95	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer		Cost					
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	7.9771	13.4709	16.1915	14.1409	51.7804	90.00%
14	11	31	11	20	11	11	6	7.6853	13.1901	14.7511	12.7002	48.3267	
		33	11	22	9				13.3534	15.2801		49.019	
15	10	34	12	22	8	12	7	7.4769	13.4475	14.7028	12.3549	47.9821	
		35	11	24	6				13.6585	13.9105		47.4008	
16	10	35	11	24	5	13	6	7.2901	13.6311	13.3589	12.3191	46.5993	
		37	11	26	3				13.8696	12.6724		46.1512	
17	10	37	11	26	2	13	6	7.1378	13.8459	12.082	12.2369	45.3027	
		37	11	26	1				13.8459	11.5025		44.7232	
18	10	37	11	26	1	13	6	7.0046	13.8244	11.4881	12.2333	44.5503	
		37	11	26	1				13.8244	11.4881		44.5503	
19	10	37	11	26	1	13	6	6.9011	13.8051	11.4747	12.2299	44.4109	
		37	11	26	1				13.8051	11.4747		44.4109	
20	9	37	11	26	1	13	6	6.8175	13.9062	11.5235	12.2768	44.5239	
		37	11	26	1				13.9062	11.5235		44.5239	
21	9	37	11	26	1	13	6	6.7409	13.8823	11.5146	12.2702	44.408	
		37	11	26	1				13.8823	11.5146		44.408	
22	9	37	11	26	1	13	6	6.682	13.8611	11.5087	12.2502	44.302	
		37	11	26	1				13.8611	11.5087		44.302	
23	9	37	11	26	1	13	6	6.637	13.8423	11.4998	12.2363	44.2153	
		37	11	26	1				13.8423	11.4998		44.2153	
24	9	37	11	26	1	13	6	6.604	13.8252	11.4882	12.2333	44.1507	
		37	11	26	1				13.8252	11.4882		44.1507	
25	9	37	11	26	1	13	6	6.5818	13.81	11.478	12.2307	44.1004	
		37	11	26	1				13.81	11.478		44.1004	
26	9	37	11	26	1	13	6	6.5689	13.7946	11.4687	12.2283	44.0605	
		37	11	26	1				13.7946	11.4687		44.0605	
27	8	37	11	26	1	13	6	6.5628	13.8971	11.5202	12.2748	44.2548	
		37	11	26	1				13.8971	11.5202		44.2548	
28	8	37	11	26	1	13	6	6.5572	13.8792	11.514	12.2695	44.2199	
		37	11	26	1				13.8792	11.514		44.2199	
28	8	37	11	26	1	13	6	6.5575	13.8817	11.5141	12.2696	44.2229	
		37	11	26	1				13.8817	11.5141		44.2229	
													C.S.L
28	8	37	11	26	1	13	6	6.5575	13.8807	11.5141	12.2696	44.2218	94.50%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.9: Convergence path of the optimization procedure with single steps (high production rate, $\lambda = 1.95$))

				I.B.	O.B.	DC	Retailer
	$\mu_1 = 1$	$\beta_S = 1$	***K***	30	25	20	15
λ=0.85	$\mu_2 = 1$	$\beta_P = 1$	***h***	0.2	0.4	0.6	0.8
	a = 0.1	$\beta_{DC} = 1$	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		**O.B.**		**DC**		**Retailer**				**Cost**			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	**I.B.**	**O.B.**	**DC**	**Retailer**	**TC**	**C.S.L**
13	*10*	*30*	*10*	*20*	*10*	*10*	*5*	*5.0343*	*11.2897*	*18.1491*	*10.5687*	**45.0419**	**99.16%**
14	*9*	*31*	*11*	*20*	*11*	*11*	*4*	*4.8046*	*11.2667*	*15.8899*	*9.8906*	**41.8518**	
		33	*11*	*22*	*12*				*11.433*	*16.3394*		**42.4676**	
15	*8*	*33*	*11*	*22*	*12*	*11*	*3*	*4.6042*	*11.4477*	*16.383*	*9.1468*	**41.5818**	
		33	*11*	*22*	*12*				*11.4477*	*16.383*		**41.5818**	
16	*7*	*33*	*11*	*22*	*12*	*12*	*2*	*4.4295*	*11.4751*	*16.4192*	*9.1271*	**41.4509**	
		35	*11*	*24*	*13*				*11.6683*	*17.1445*		**42.3694**	
17	*6*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2975*	*11.7154*	*17.2022*	*8.7957*	**42.0108**	
		33	*11*	*22*	*12*				*11.5257*	*16.5016*		**41.1205**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2126*	*11.6318*	*16.6003*	*9.2323*	**41.677**	
		35	*11*	*24*	*13*				*11.8166*	*17.2893*		**42.5507**	
17	*5*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2155*	*11.8163*	*17.291*	*9.0075*	**42.3303**	
		34	*12*	*22*	*12*				*11.6316*	*16.6644*		**41.519**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2122*	*11.6319*	*16.6004*	*9.2325*	**41.677**	
		35	*11*	*24*	*13*				*11.8167*	*17.2893*		**42.5507**	
17	*5*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2155*	*11.8163*	*17.291*	*9.0075*	**42.3303**	
		34	*12*	*22*	*12*				*11.6316*	*16.6644*		**41.519**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2122*	*11.6319*	*16.6004*	*9.2325*	**41.677**	
		35	*11*	*24*	*13*				*11.8167*	*17.2893*		**42.5507**	
17	*5*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2155*	*11.8163*	*17.291*	*9.0075*	**42.3303**	
		34	*12*	*22*	*12*				*11.6316*	*16.6644*		**41.519**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2122*	*11.6319*	*16.6004*	*9.2325*	**41.677**	
		35	*11*	*24*	*13*				*11.8167*	*17.2893*		**42.5507**	
17	*5*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2155*	*11.8163*	*17.291*	*9.0075*	**42.3303**	
		34	*12*	*22*	*12*				*11.6316*	*16.6644*		**41.519**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2122*	*11.6319*	*16.6004*	*9.2325*	**41.677**	
		35	*11*	*24*	*13*				*11.8167*	*17.2893*		**42.5507**	
17	*5*	*35*	*11*	*24*	*13*	*11*	*2*	*4.2155*	*11.8163*	*17.291*	*9.0075*	**42.3303**	
		34	*12*	*22*	*12*				*11.6316*	*16.6644*		**41.519**	
17	*5*	*34*	*12*	*22*	*12*	*12*	*2*	*4.2122*	*11.6319*	*16.6004*	*9.2325*	**41.677**	
		35	*11*	*24*	*13*				*11.8167*	*17.2893*		**42.5507**	
													C.S.L
17	*5*	*34*	*12*	*22*	*12*	*11*	*2*	*4.2136*	*11.6316*	*16.6*	*9.2596*	**41.7047**	**83.96%**
17	*5*	*35*	*11*	*24*	*13*	*12*	*2*	*4.214*	*11.8167*	*17.2913*	*9.0125*	**42.3344**	**86.70%**

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.10: Convergence path of the optimization procedure with single steps (low production rate)

				I.B.	O.B.	DC	Retailer
	μ_1 = 1	β_S = 1	***K***	30	25	20	15
λ=0.85	μ_2 = 1	β_P = 1	***h***	0.2	0.4	0.6	0.8
	a = 0.1	β_{DC} = 1	***g***		0.6	0.4	0.2
			p	100	10	50	25

I.B.		O.B.		DC		Retailer				Cost			
Q_I	R_I	R	r	Q_{DC}	R_{DC}	Q_R	R_R	I.B.	O.B.	DC	Retailer	TC	C.S.L
13	10	30	10	20	10	10	5	5.0343	11.2897	18.1491	10.5687	45.0419	99.16%
17	5	32	12	20	11	11	4	4.2139	11.4644	16.2015	10.1472	42.0271	
		34	12	22	12				11.631	16.5693		42.5614	
17	5	34	12	22	12	11	3	4.2153	11.6312	16.5994	9.3174	41.7633	
		34	12	22	12				11.6312	16.5994		41.7633	
17	5	34	12	22	12	12	2	4.2136	11.6316	16.6	9.2318	41.6769	
		35	11	24	13				11.8163	17.2891		42.5508	
17	5	35	11	24	13	11	2	4.2155	11.8163	17.291	9.0075	42.3303	
		34	12	22	12				11.6316	16.6644		41.519	
17	5	34	12	22	12	12	2	4.2122	11.6319	16.6004	9.2325	41.677	
		35	11	24	13				11.8167	17.2893		42.5507	
17	5	35	11	24	13	11	2	4.2155	11.8163	17.291	9.0075	42.3303	
		34	12	22	12				11.6316	16.6644		41.519	
17	5	34	12	22	12	12	2	4.2122	11.6319	16.6004	9.2325	41.677	
		35	11	24	13				11.8167	17.2893		42.5507	
17	5	35	11	24	13	11	2	4.2155	11.8163	17.291	9.0075	42.3303	
		34	12	22	12				11.6316	16.6644		41.519	
17	5	34	12	22	12	12	2	4.2122	11.6319	16.6004	9.2325	41.677	
		35	11	24	13				11.8167	17.2893		42.5507	
17	5	35	11	24	13	11	2	4.2155	11.8163	17.291	9.0075	42.3303	
		34	12	22	12				11.6316	16.6644		41.519	
													C.S.L
17	5	34	12	22	12	11	2	4.2136	11.6316	16.6	9.2596	41.7047	83.96%
17	5	35	11	24	13	12	2	4.214	11.8167	17.2913	9.0125	42.3344	86.70%

I.B.: Input Buffer **O.B.: Output Buffer** **DC: Distribution Center**

Table B.11: Convergence path of the optimization procedure with optimized steps (low production rate)

Appendix C
TES Model Fitting Formulas

C.0.1 The empirical density function

A *histogram* is a mixture of uniform random variables. Formally, let X be a mixture of N Uniform (l_n, r_n) variates with mixing probabilities p_n and let C_n be the cumulative distribution function of p_n, i.e., $C_n = \sum_{j=1}^{n} p_j$, with $C_0 = 0$. Also, let us denote $w_n = r_n - l_n$ and $\mathbf{1}_A(x)$ is the indicator function. Then, the histogram has the step function density

$$f_X(x) = \sum_{n=1}^{N} \mathbf{1}_{[l_n, r_n)}(x) \frac{p_n}{w_n} \tag{C.1}$$

and the corresponding cumulative distribution function

$$F_X(x) = \sum_{n=1}^{N} \mathbf{1}_{[l_n, r_n)}(x)[C_{n-1} + (x - l_n)\frac{p_n}{w_n}]. \tag{C.2}$$

To be able to generate random variates for X, the distortion function is defined as the inverse of $F_X(x)$ given by

$$D(x) = \sum_{n=1}^{N} \mathbf{1}_{[C_{n-1}, C_n)}(x)[l_n + (x - C_{n-1})\frac{w_n}{p_n}]. \tag{C.3}$$

C.0.2 Computation of $|\tilde{D}(i2\pi\nu)|^2$

In this part, only the stitched distortions are considered. Let $D_\xi(x) = D(S_\xi(x))$. Then, $|\tilde{D}_\xi(i2\pi\nu)|^2 = a^2_{\xi,\nu} + b^2_{\xi,\nu}$ where $a^2_{\xi,\nu}$ and $b^2_{\xi,\nu}$ are given by

$$\begin{aligned} a^2_{\xi,\nu} &= \textstyle\sum_{n=1}^{N} \frac{r_n[\sin(2\pi\nu\xi C_n)+\sin(2\pi\nu(1-\xi)C_n)]}{2\pi\nu} \\ &- \textstyle\sum_{n=1}^{N} \frac{l_n[\sin(2\pi\nu\xi C_{n-1})+\sin(2\pi\nu(1-\xi)C_{n-1})]}{2\pi\nu} \\ &+ \textstyle\sum_{n=1}^{N} \frac{w_n}{p_n} \left\{ \frac{\cos(2\pi\nu\xi C_n)-\cos(2\pi\nu\xi C_{n-1})}{\xi(2\pi\nu)^2} \right. \\ &\qquad\qquad \left. + \frac{[\cos(2\pi\nu(1-\xi)C_n)-\cos(2\pi\nu(1-\xi)C_{n-1})]}{(1-\xi)(2\pi\nu)^2} \right\} \end{aligned} \tag{C.4}$$

$$
\begin{aligned}
b^2_{\xi,\nu} \quad &= \quad \textstyle\sum_{n=1}^{N} \frac{r_n[\cos(2\pi\nu\xi C_n) - \cos(2\pi\nu(1-\xi)C_n)]}{2\pi\nu} \\
&- \quad \textstyle\sum_{n=1}^{N} \frac{l_n[\cos(2\pi\nu\xi C_{n-1}) - \cos(2\pi\nu(1-\xi)C_{n-1})]}{2\pi\nu} \\
&- \quad \textstyle\sum_{n=1}^{N} \frac{w_n}{p_n} \left\{ \frac{\sin(2\pi\nu\xi C_n) - \sin(2\pi\nu\xi C_{n-1})}{\xi(2\pi\nu)^2} \right. \\
&\qquad \left. - \frac{[\sin(2\pi\nu(1-\xi)C_n) - \sin(2\pi\nu(1-\xi)C_{n-1})]}{(1-\xi)(2\pi\nu)^2} \right\}.
\end{aligned}
\tag{C.5}
$$

where ξ is known as the stitching parameter with typical values in $(0, 1)$.

References

[1] Adida, E., and Perakis, G., "A robust optimization approach to dynamic pricing and inventory control with no backorders", *Mathematical Programming*, Vol. 107, 97–129, 2006.

[2] Agrawal, V., and Seshadri, S., "Distribution free bounds for service constrained (Q, r) inventory systems", *Naval Research Logistics*, Vol. 47, 635–656, 2000.

[3] Albin, S. L., "Approximating a point process by a renewal process, II: superposition arrival processes to queues", *Operations Research*, Vol. 32, No. 5, 1133–1162, 1984.

[4] Alon, I., Qi, M., and Sadowski, R. J., "Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods", *Journal of Retailing and Consumer Services*, Vol. 8, 147–156, 2001.

[5] Altıok, T., "On the phase-type approximations of general distributions", *IIE Transactions*, Vol. 17, 110–116, 1985.

[6] Altıok, T., "(R, r) production/inventory systems", *Operations Research*, Vol. 37, No. 2, 266–276, 1989.

[7] Altıok, T., *Performance Analysis of Manufacturing Systems*, Springer, 1997.

[8] Altıok, T., and Melamed, B., *Simulation Modeling and Analysis with Arena*, Cyber Research, Inc. 2001.

[9] Altıok, T., and Ranjan, R., "Multi-stage pull-type production/inventory systems", *IIE Transactions*, Vol. 27, 190–200, 1995.

[10] Axsäter, S., "Simple solution procedures for a class of two-echelon inventory problems", *Operations Research*, Vol. 38, No. 1, 64–69, 1990.

[11] Axsäter, S., "Exact and approximate evaluation of batch-ordering policies for two-level inventory systems", *Operations Research*, Vol. 41, No. 4, 777–785, 1993.

[12] Axsäter, S., "Exact analysis of continuous-review (R, Q) policies in two-echelon inventory systems with compound Poisson demand", *Operations Research*, Vol. 48, No. 5, 686–696, 2000.

[13] Axsäter, S., "Scaling down multi-echelon inventory problems", *International Journal of Production Economics*, Vol. 71, 255–261, 2001.

[14] Axsäter, S., "Approximate optimization of a two-level distribution inventory system", *International Journal of Production Economics*, Vol. 81-82, 545–553, 2003.

[15] Axsäter, S., "Note: optimal policies for serial inventory systems under fill rate constraints", *Management Science*, Vol. 49, No. 2, 247–253, 2003.

[16] Axsäter, S., "A simple procedure for determining order quantities under a fill rate constraint and normally distributed lead-time demand", *European Journal of Operational Research*, Vol. 174, 480–491, 2006.

[17] Badinelli, R. D., "A model for continuous-review pull policies in serial inventory systems", *Operations Research*, Vol. 40, No. 1, 142–156, 1992.

[18] Balcioğlu, B., Jagerman, D. L., and Altıok, T., "Approximate mean waiting time in a $GI/D/1$ queue", *forthcoming in IIE transactions.*

[19] Bashyam, S., and Fu, M. C., "Optimization of (s, S) inventory systems with random lead times and a service level constraint", *Management Science*, Vol. 44, No. 12, S243–S256, 1998.

[20] Beamon, B. M., "Supply chain design and analysis: models and methods", *International Journal of Production Economics*, Vol. 55, 281–294, 1998.

[21] Bernstein, F., and DeCroix, G. A., "Inventory policies in a decentralized assembly system", *Operations Research*, Vol. 54, No. 2, 324–336, 2006.

[22] Bitran, G. R., and Dasu, S., "Approximating nonrenewal processes by Markov chains: use of Super-Erlang (SE) chains", *Operations Research*, Vol. 41, No. 5, 903–923, 1993.

[23] Bitran, G. R., and Dasu, S., "Analysis of the $\Sigma Ph_i/Ph/1$ queue", *Operations Research*, Vol. 42, No. 1, 158–174, 1994.

[24] Boute, R. N., Lambrecht, M. R., and Houdt, B., "Performance evaluation of a production/inventory system with periodic review and endogenous lead times", *Naval Research Logistics*, Vol. 54, 462–473, 2007.

[25] Box, G. E. P., and Jenkins, G. M., *Time Series Analysis: Forecasting and Control*, Holden-Day, 1976.

[26] Boyaci, T., and Gallego, G., "Serial production/distribution systems under service level constraints", *Manufacturing & Service Operations Management*, Vol. 3, No. 1, 43–50, 2001.

[27] Bratley, P., Fox, B. L., and Schrage, L. E., *A Guide to Simulation*, Springer-Verlag, 1987.

[28] Brockwell, P. J., and Davis, R. A., *Introduction to Time Series and Forecasting*, Springer, 2002.

[29] Buzacott, J. A., and Kostelski, D., "Matrix-geometric and recursive algorithm solution of a two-stage unreliable flow line", *IIE Transactions*, Vol. 19, No.4, 429–438, 1987.

[30] Cachon, G. P., "Exact evaluation of batch-ordering inventory policies in two-echelon supply chains with periodic review", *Operations Research*, Vol. 49, No. 1, 79–98, 2001.

[31] Chaudhry, M. L., and Templeton, J. G. C., *A First Course in Bulk Queues*, Wiley & Sons, 1983.

[32] Chen, F., "Optimal policies for multi-echelon inventory problems with batch ordering", *Operations Research*, Vol. 48, No. 3, 376–389, 2000.

[33] Chen, Y., Ray, S., and Song, Y., "Optimal pricing and inventory control policy in periodic-review systems with fixed ordering cost and lost sales", *Naval Research Logistics*, Vol. 53, 117–136, 2006.

[34] Chen, X., and Simchi-Levi, D., "Coordinating inventory control and pricing strategies: the continuous review model", *Operations Research Letters*, Vol. 34, 323–332, 2006.

[35] Chen, F., and Song, J. S., "Optimal policies for multi-echelon inventory problems with Markov-modulated demand", *Operations Research*, Vol. 49, No. 2, 226–234, 2001.

[36] Chen, F., and Zheng, Y. S., "One-warehouse multiretailer systems with centralized stock information", *Operations Research*, Vol. 45, No. 2, 275–287, 1997.

[37] Chen, F., and Zheng, Y. S., "Sensitivity analysis of an (s, S) inventory model", *Operations Research Letters*, Vol. 21, 19–23, 1997.

[38] Chen, F., and Zheng, Y. S., "Near-optimal echelon-stock (R, nQ) policies in multistage serial systems", *Operations Research*, Vol. 46, No. 4, 592–602, 1998.

[39] Chen, F., Federgruen, A., and Zheng, Y. S., "Coordination mechanisms for a distribution system with one supplier and multiple retailers", *Management Science*, Vol. 47, No. 5, 693–708, 2001.

[40] Chew, E. P., and Tang, L. C., "Warehouse-retailer system with stochastic demands - non-identical retailer case", *European Journal of Operational Research*, Vol. 82, 98–110, 1995.

[41] Chiang, W. K., and Monahan, G. E., "Managing inventories in a two-echelon dual-channel supply chain", *European Journal of Operational Research*, Vol. 162, 325–341, 2005.

[42] Chiang, C., "Optimal ordering policies for periodic-review systems with replenishment cycles", *European Journal of Operational Research*, Vol. 170, 44–56, 2006.

[43] Clark, A. J., and Scarf, H., "Optimal policies for a multi-echelon inventory problem", *Management Science*, Vol. 6, No. 4, 475–490, 1960.

[44] Cohen, M. A., and Lee, H. L., "Strategic analysis of integrated production-distribution systems: models and methods", *Operations Research*, Vol. 36, No. 2, 216–228, 1988.

[45] Davis, T., "Effective supply chain management", *Sloan Management Review*, Vol. 34, No. 4, 35–46, 1993.

[46] De Bodt, M. A., and Graves, S. C., "Continuous-review policies for a multi-echelon inventory problem with stochastic demand", *Management Science*, Vol. 31, No. 10, 1286–1299, 1985.

[47] Deuermeyer, B. L., and Schwarz, L. B., "A model for the analysis of system service level in warehouse-retailer distribution systems: The identical retailer case", Chapter 8 in *Multi-level Production/Inventory Control Systems: Theory and Practice*, North-Holland, Amsterdam, 1981.

[48] Diks, E. B., de Kok, A. G., and Lagodimos, A. G., "Multi-echelon systems: a service measure perspective", *European Journal of Operational Research*, Vol. 95, 241–263, 1996.

[49] Duri, C., Frein, Y., and Mascolo, M. Di., "Performance evaluation and design of base stock systems", *European Journal of Operational Research*, Vol. 127, 172–188, 2000.

[50] Erenguc, S. S., Simpson, N. C., and Vakharia, A. J., "Integrated production/distribution planning in supply chains: An invited review", *European Journal of Operational Research*, Vol. 115, 219–236, 1999.

[51] Erkip, N., Hausman, W. H., and Nahmias, S., "Optimal centralized ordering policies in multi-echelon inventory systems with correlated demands", *Management Science*, Vol. 36, No. 3, 381–392, 1990.

[52] Ettl, M., Feigin, G., Lin, G., and Yao, D. D., "A supply network model with base-stock control and service requirements", *Operations Research*, Vol. 48, No. 2, 216–232, 2000.

[53] Federgruen, A., and Zheng, Y. S., "An efficient algorithm for computing an optimal (r, Q) policy in continuous review stochastic inventory problems", *Operations Research*, Vol. 40, No. 4, 808–813, 1992.

[54] Federgruen, A., and Zipkin, P., "An efficient algorithm for computing optimal (s, S) policies", *Operations Research*, Vol. 32, No. 6, 1268–1285, 1984.

[55] Federgruen, A., and Zipkin, P., "Computational issues in an infinite-horizon, multiechelon inventory model", *Operations Research*, Vol. 32, No. 4, 818–836, 1984.

[56] Forsberg, R., "Exact evaluation of (R, Q)-policies for two-level inventory systems with Poisson demand", *European Journal of Operational Research*, Vol. 96, 130–138, 1996.

[57] Gallego, G., and Paul, Z., "Stock positioning and performance estimation in serial production-transportation systems", *Manufacturing & Service Operations Management*, Vol. 1, No. 1, 77–88, 1999.

[58] Girish, M. K., and Hu, J. Q., "Higher order approximations for the single server queue with splitting, merging, and feedback", *European Journal of Operational Research*, Vol. 124, 447–467, 2000.

[59] Glasserman, P., and Tayur, S., "The stability of a capacitated, multi-echelon production-inventory system under a base-stock policy", *Operations Research*, Vol. 42, No. 5, 913–925, 1994.

[60] Glasserman, P., and Wang, Y., "Fill-rate bottlenecks in production-inventory networks", *Manufacturing & Service Operations Management*, Vol. 1, No. 1, 62–76, 1999.

[61] Gurgur, C. Z., and Altıok, T., "Approximate analysis of decentralized, multi-stage, pull-type production/inventory systems", *Annals of Operations Research*, Vol. 125, 95–116, 2004.

[62] Hayya, J. C., Kim, J. G., Disney, S. M., Harrison, T. P., and Chatfield, D., "Estimation in supply chain inventory management", *International Journal of Production Research*, Vol. 44, No. 7, 1313–1330, 2006.

[63] He, Q. M., Jewkes, E. M., and Buzacott, J., "Optimal and near-optimal inventory control policies for a make-to-order inventory-production system", *European Journal of Operational Research*, Vol. 141, 113–132, 2002.

[64] Herzog, U., Woo, L., and Chandy, K. M., "Solution of queuing problems by a recursive technique", *IBM Journal of Research and Development*, Vol. 19, 295–300, 1975.

[65] Hill, J. R., and Melamed, B., "A survey of TES modeling applications", *Simulation*, Vol. 64, No. 6, 353–370, 1995.

[66] Hill, J. R., and Melamed, B., "TEStool: a visual interactive environment for modeling autocorrelated time series", *Performance Evaluation*, Vol. 24, No. 1&2, 3–22, 1995.

[67] Hill, R. M., and Johansen, S. G., "Optimal and near-optimal policies for lost sales inventory models with at most one replenishment order outstanding", *European Journal of Operational Research*, Vol. 169, 111–132, 2006.

[68] Hooke, R., and Jeeves, T. A., "Direct search of numerical and statistical problems", *Journal of ACM*, Vol. 8, 212–229, 1966.

[69] Houtum, G. J., Inderfurth, K., and Zijm, W. H. M., "Materials coordination in stochastic multi-echelon systems", *European Journal of Operational Research*, Vol. 95, 1–23, 1996.

[70] Ishii, K., Takahashi, K., and Muramatsu, R., "Integrated production, inventory and distribution systems", *International Journal of Production Research*, Vol. 26, No. 3, 473–482, 1988.

[71] Jagerman, D. L., and Melamed, B., "The transition and autocorrelation structure of TES processes part I: general theory", *Stochastic Models* 8(2), 193–219, 1992.

[72] Jagerman, D. L., and Melamed, B., "The transition and autocorrelation structure of TES processes part II: special cases", *Stochastic Models* 8(3), 499–527, 1992.

[73] Jagerman, D. L., and Melamed, B., "Bidirectional estimation and confidence regions for TES processes", *Proceedings of MASCOTS '95*, Durham, North Carolina, 94–98, January 1995.

[74] Jelenkovic, P., and Melamed, B., "Algorithmic modeling of TES processes", *IEEE Trans. on Automatic Control*, Vol. 40, No. 7, 1305–1312, 1995. A short version appeared as "Automated TES Modeling of Compressed Video", *Proceedings of IEEE INFOCOM '95*, Boston, Massachusetts, 746–752, April 1995.

[75] Jemai, Z., and Karaesmen, F., "Decentralized inventory control in a two-stage capacitated supply chain", *IIE Transactions*, Vol. 39, 501–512, 2007.

[76] Kim, E., "Optimal inventory replenishment policy for a queuing system with finite waiting room capacity", *European Journal of Operational Research*, Vol. 161, 256–274, 2005.

[77] Lee, H. L., and Billington, C., "Managing supply chain inventory: pitfalls and opportunities", *Sloan Management Review*, Vol. 33, No. 3, 65–73, 1992.

[78] Lee, H. L., and Billington, C., "Material management in decentralized supply chains", *Operations Research*, Vol. 41, No. 5, 835–847, 1993.

[79] Liu, L., Liu, X., and Yao, D. D., "Analysis and optimization of a multi-stage inventory-queue system", *Management Science*, Vol. 50, No. 3, 365–380, 2004.

[80] Makridakis, S., and Wheelwright, S. C., *Forecasting Methods for Management*, John Wiley & Sons, 1989.

[81] Melamed, B., "TES: a class of methods for generating autocorrelated uniform variates", *ORSA J. on Computing* 3, 317–329, 1991.

[82] Melamed, B., "The empirical TES methodology: modeling empirical time series", *J. of Applied Mathematics and Stochastic Analysis*, Vol. 10, No. 4, 333–353, 1997.

[83] Moinzadeh, K., and Lee, H. L., "Batch size and stocking levels in multi-echelon repairable systems", *Management Science*, Vol. 32, No. 12, 1567–1581, 1986.

[84] Naddor, E., "Sensitivity to distributions in inventory systems", *Management Science*, Vol. 24, No. 16, 1769–1772, 1978.

[85] Nahmias, S., and Smith, S. A., "Optimizing inventory levels in a two-echelon retailer system with partial lost sales", *Management Science*, Vol. 40, No. 5, 582–596, 1994.

[86] Neuts, M., *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Dover, 1994.

[87] Osogami, T., and Harchol-Balter, M., "A closed-form solution for mapping general distributions to minimal PH distributions", in *Computer Performance Modeling Techniques and Tools* (Kemper, P. and Sanders, W. H., Eds.), 200–217, *Lecture Notes in Computer Science*, Springer-Verlag, 2003.

[88] Platt, D. E., Robinson, L. W., and Freund, R. B., "Tractable (Q, R) heuristic models for constrained service levels", *Management Science*, Vol. 43, No. 7, 951–965, 1997.

[89] Pyke, D. F., and Cohen, M. A., "Performance characteristics of stochastic integrated production-distribution systems", *European Journal of Operational Research*, Vol. 68, 23–48, 1993.

[90] Reklaitis, G. V., Ravindran, A., and Ragsdell, K. M., *Engineering Optimization: Methods and Applications*, John Wiley & Sons, 1983.

[91] Schwarz, L. B., Deuermeyer, B. L., and Badinelli, R. D., "Fill-rate optimization in a one-warehouse N-identical retailer distribution system", *Management Science*, Vol. 31, No. 4, 488–498, 1985.

[92] Sherbrooke, C. C., "Metric: a multi-echelon technique for recoverable item control", *Operations Research*, Vol. 16, No. 1, 122–141, 1968.

[93] Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E., *Designing & Managing the Supply Chain: Concepts, Strategies & Case Studies*, McGraw-Hill, 2003.

[94] So, K. C., and Pinault, S. C., "Allocating buffer storages in a pull system", *International Journal of Production Research*, Vol. 26, No. 12, 1959–1980, 1988.

[95] Song, J. S., and Yao, D. D., "Performance analysis and optimization of assemble-to-order systems with random lead times", *Operations Research*, Vol. 50, No. 5, 889–903, 2002.

[96] Svonoros, A., and Zipkin, P., "Estimating the performance of multi-level inventory systems", *Operations Research*, Vol. 36, No. 1, 57–72, 1988.

[97] Svonoros, A., and Zipkin, P., "Evaluation of one-for-one replenishment policies for multi-echelon inventory systems", *Management Science*, Vol. 37, No. 1, 68–83, 1991.

[98] Swaminathan, J. M., and Tayur, S. R., "Tactical planning models for supply chain management", in *Supply Chain Management: Design, Coordination and Operation, Handbooks in Operations Research and Management Science*, (de Kok, A. G., and Graves, S. C., Eds.), Vol. 11, 525–559, Elsevier Amsterdam, The Netherlands, 2003.

[99] Thomakos, D. D., and Guerard, J. B., "Naive, ARIMA, nonparametric, transfer function and VAR models: a comparison of forecasting performance", *International Journal of Forecasting*, Vol. 20, 53–67, 2004.

[100] Thomas, D. J., and Griffin, P. M., "Coordinated supply chain management", *European Journal of Operational Research*, Vol. 94, 1–15, 1996.

[101] Toktay, L. B., and Wein, L. M., "Analysis of a forecasting-production-inventory system with stationary demand", *Management Science*, Vol. 47, No. 9, 1268–1281, 2001.

[102] Veatch, M. H, and Wein, L. M., "Optimal control of a two-station tandem production/inventory system", *Operations Research*, Vol. 42, No. 2, 337–350, 1994.

[103] Vericourt, F., Karaesmen, F., and Dallery, Y., "Optimal stock allocation for a capacitated supply system", *Management Science*, Vol. 48, No. 11, 1486–1501, 2002.

[104] Vuuren, M., and Adan, I. J., "Approximating multiple arrival streams by using aggregation", *Stochastic Models*, Vol. 22, 423–440, 2006.

[105] Whitt, W., "Approximating a point process by a renewal process, I: two basic methods", *Operations Research*, Vol. 30, No. 1, 125–147, 1982.

[106] Whitt, W., "Poisson arrivals see time averages", *Operations Research*, Vol. 30, No. 2, 223–231, 1982.

[107] Zheng, Y. S., and Federgruen, A., "Finding optimal (s, S) policies is about as simple as evaluating a single policy", *Operations Research*, Vol. 39, No. 4, 654–665, 1991.

[108] Zipkin, P., "Models for design and control of stochastic, multi-item batch production systems", *Operations Research*, Vol. 34, No. 1, 91–104, 1986.

[109] Zipkin, P., "Stochastic lead-times in continuous time inventory models", *Naval Research Logistics Quarterly*, Vol. 33, 763–774, 1986.

[110] Zipkin, P., "The use of phase-type distributions in inventory control models", *Naval Research Logistics Quarterly*, Vol. 35, No. 2, 247–257, 1988.

[111] Zou, H., and Yang, Y., "Combining time series for forecasting", *International Journal of Forecasting*, Vol. 20, 69–84, 2004.

Vita

ABDULLAH SIDDIK KARAMAN

EDUCATION

2001-2007 **Rutgers University**, New Brunswick, NJ
Ph.D. in Industrial and Systems Engineering

1999-2001 **Bilkent University**, Ankara, Turkey
M.S. in Industrial Engineering

1994-1999 **Koç University**, Istanbul, Turkey
B.S. in Mathematics

TEACHING EXPERIENCE

2001-2005 **Rutgers University**, New Brunswick, NJ
Teaching Assistant, Industrial and Systems Engineering

RESEARCH EXPERIENCE

2005-2006 **Rutgers University**, New Brunswick, NJ
Research Assistant, Industrial and Systems Engineering

WORK EXPERIENCE

2006-Present **marketRx**, Bridgewater, NJ
Analyst, Professional Services