Holdout Transshipment Policy

In Two-location Inventory Systems

Jiaqi Zhang

Thesis submitted for the degree of

Doctor of Philosophy

The University of Edinburgh

2008

To my wife Li Na

for her great support

Abstract

In two-location inventory systems, unidirectional transshipment policies are considered when an item is not routinely stocked at a location in the system. Unlike the past research in this area which has concentrated on the simple transshipment policies of complete pooling or no pooling, the research presented in this thesis endeavors to develop an understanding of a more general class of transshipment policy. The research considers two major approaches: a decomposition approach, in which the two-location system is decomposed into a system with independent locations, and Markov decision process approach.

For the decomposition approach, the transshipment policy is restricted to the class of holdout transshipment policy. The first attempt to develop a decomposition approach assumes that transshipment between the locations occurs at a constant rate in order to decompose the system into two independent locations with constant demand rates. The second attempt modifies the assumption of constant rate of transshipment to take account of local inventory levels to decompose the system into two independent locations with non-constant demand rates. In the final attempt, the assumption of constant rate of transshipment is further modified to model more closely the location providing transshipments. Again the system is decomposed into two independent locations with non-constant demand rates. For each attempt, standard techniques are applied to derive explicit expressions for the average cost rate, and an iterative solution method is developed to find an optimal holdout transshipment policy. Computational results show that these approaches can provide some insights into the performance of the original system.

A semi-Markov decision model of the system is developed under the assumption of exponential lead time rather than fixed lead time. This model is later extended to the case of phase-type distribution for lead time. The semi-Markov decision process allows more general transshipment policies, but is computationally more demanding. Implicit expressions for the average cost rate are derived from the optimality equation for dynamic programming models. Computational results illustrate insights into the management of the two-location system that can be gained from this approach.

Keywords: Inventory control management system, lateral transshipment policy, stochastic modeling, dynamic programming and simulation modeling

Prologue

I would like to take this opportunity to acknowledge my sincere gratitude to my supervisors Professor Glazebrook and Professor Archibald. Thanks for their help on the funding support to my PhD research. In particular, when Professor Archibald became my principal supervisor in the middle of my PhD study. I should be very grateful for his constructive guidance and effective supervisions guiding me to the final point. Thank him so much for giving me a good training which allows me to be able to continue my professional career after finishing my PhD at University of Edinburgh.

I also would like to say many thanks for all help which I received from colleagues and staffs in our department office and IT support team. Thanks for their kindness help and support during my study.

I must be very grateful for the strong support from my family, especially my wife Li Na. Thanks for her determination to support my career development by suspending her high education teaching career during my study in Edinburgh. I sincerely hope to start our family life from a new level as a reward for all she has given in support of my study in Edinburgh.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

Nowadays, most global enterprise operations are far more complicated than those of the traditional local family managed business. With significant influences from globalization and wide application of Information Communication Technology (ICT), logistics operations of global business giants exercise their controls beyond their geographic boundaries. Some multi-national business operations can be ranked in scale as a single trade country or entity in the world independently. For example, Wal-Mart, the world largest retail supermarket chain, reported in its annual report of year 2006 that it operates more than 2460 super centres around the world, with annual turnover over US Dollar 312,427 millions, and inventory worth at US Dollar 32,191 millions. In the manufacturing sector, some finished products are made up of components to a magnitude level. For instance, as revealed in Economist (2006), a Boeing 747 has more than 6 million parts, all of which have been ordered, tracked, assembled and carefully monitored throughout their service life.

According to the Economist (2006), cost savings can be achieved not only from purchasing power, but also from more efficient stock management. Hau Lee, a professor of operations, information and technology at Stanford University, calls one of the biggest sources of inefficiency in logistics, the “bull-whip effect”, after the way the amplitude of a whiplash increases down the length of the whip when it is cracked.

Some might argue that keeping high volume of inventory stock is far less important than two decades ago since the emerging and popular use of computer network and internet techniques. We agree with this argument from strong evidence that a new trend for most traditional businesses is to handle with the plummeting inventory. For instance, Dixons, one of the largest electronic retailers in UK, claims its long-term strategy is to close more high street outlets to improve competitiveness in responding to situations where shoppers move to online shopping systems. Top volume-car manufacturers like GM and Ford are fighting hard to survive by keeping stock at reasonable low levels in order to compete with their Japanese rivals. However, these only tell a part of the story.

In practice, the survey in the Economist (2006) suggests that business needs to offer tailored service levels for different products to meet different requirements. The strategy is never to be out of stock of high-velocity items, which tend to be the most profitable. On low-velocity items, there is slightly more room for errors, advised by Same Israelit, an expert in retail logistics for Bain & Company. The ultimate goal for the supply chain is to become leaner and more agile, says Chris Poole, P&G's director of outbound logistics in Western Europe.

On the one hand, for most business operations, enormous savings of tangible inventory goods become the effective decision policy for the senior manager to decrease the amount of current assets and to increase operation profits correspondingly. However, it does not mean that any cost saving strategy can be achieved successfully on compensations of customer services. In effect, for some businesses, e.g. Military supply, the requirement for a top quality service level is far more important than cost savings.

On the other hand, reductions on tangible goods lead to escalating demand for service business. Movements of redundant tangible goods turn out significantly to increase demand on products and services which are in intangible or virtual forms. Therefore, inventory control management strategies can be reused to such products or services, like online ordering system and call centre business. The precise and efficient control and management of inventory stocks are demanding good inventory control strategies. In Seifert (2006), they reported that Hewlett-Packard Company (HP) adopted integrated direct and indirect sales channels to cope with changing situations where more customers are keen to place order online. Precisely, they consider adding new online shopping platform as a new virtual sale channel and integrate it into the existing retail network. An unidirectional transshipment policy between the virtual sale channel and other retail sale channels is deployed to improve customer services and to save operation costs. Obviously, for some service businesses such as online ticket and *e*-procurement systems, conventional inventory control management techniques are still effective to provide the fundemental guidelines for better management performances.

Among a comprehensive collection of inventory control strategies, we put our research interests on the lateral transshipment policy to the multi-echelon and multi-location inventory control system because many business operations still use variants of lateral transshipments policy within their local retail network systems to improve the customer services and to achieve cost savings. In fact, Alliance Boots, the largest UK drugs and pharmacy company, uses a lateral transshipment policy to help customers whenever they cannot find the product from one particular shop. Royal Mail Holdings Plc., the largest UK postal service company, provides a transshipment service by charging a small amount of money for a redelivery service which delivers goods to the nominated local post office by a choice of the customer. In term of the lateral transshipment policy, it could be crucially important under some specific circumstances. For instance, Sergeant Steven Roberts, the British tank soldier who was killed in Iraq because of the lack of protective armour on the war site, could perhaps have been saved if an emergent lateral transshipment policy had been reviewed and executed from the nearest British military supply base to the supply point at the front line. In that sense, the considerate deployment of the lateral transshipment policy will make the warfare management better-off under some circumstances.

Nevertheless, there is no panacea in the world. Any logistics strategy has its context. For high-value and slow-moving inventory stock, the lateral transshipment policy proves to be effective and to be considered as a decision policy for most business. For example, as reported in Zhao *et al.* (2005), Caterpillar Inc. employs a lateral transshipment management policy to coordinate high-value inventory stock (some expensive tyres amount to ten of thousands dollar each) among the regional multi-location inventory control systems. Komatsu Ltd., the second largest heavy duty construction machinery and equipment manufacturer in the world, provides real-time Vehicle Health Monitoring System (VHMS) to their Repair & Maintenance (R&M) contractors or dealers, making them informed of the latest working conditions of their machinery to maintain the optimal repair and maintenance service at a top level of operating efficiency by avoiding unexpected down time and minimizing unexpected costs.

In maintenance business services, such as supplying expensive spare parts for airplane maintenance, supplying high-value spare parts to airspace manufacturing sites, and supplying the spare parts to nuclear power stations and other business environments which take a high degree of business performance as the top priority strategy, the lateral transshipment policy is widely used to keep a high level of customer service in case of huge loss on the customer's business performances.

In addition, even for business which handles with low-value and slow-moving items, for example, the transshipment service provided by Edinburgh City Library Network, a lateral transshipment policy is still attractive. For those who might have a busy schedule, time savings on the item collection could be a good option by paying a little extra money to re-deliver the item at the right place at right time.

1.1.1 Inventory type classifications

There are various classifications of inventory types according to different criteria. In respective of production planning and scheduling, conventionally, we can classify inventory types into six categories as:

1. **Cycle inventory:** the amount of inventory resulted as ordering or producing in batches
2. **Congestion stocks:** inventory items compete for limited capacity
3. **Safety stock:** the amount of inventory on hand to keep the uncertain demand and uncertain supply in the short run
4. **Anticipation inventory:** the amount of inventory holding for forecasting demand
5. **Pipeline inventory (work-in-progress):** the amount of inventory on the way
6. **Decoupling stock:** the amount of inventory is used when the separation of decision making at the different echelons is allowed in the multi-echelon system

According to further classifications in Silver (1985), we have a product-process matrix in Figure 1.1.

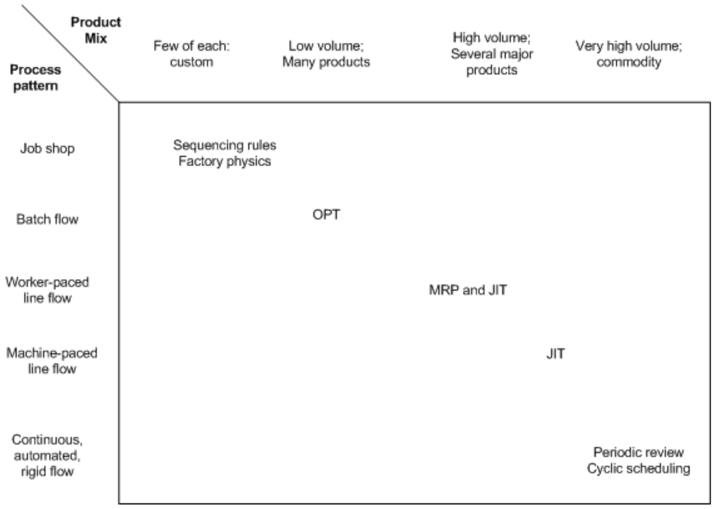

Figure 1.1. Production planning and scheduling and the Product-Process Matrix

For the senior inventory manager, decision methods of production planning and decision schedule are chosen by identifying the product type and process pattern in the matrix in Figure 1.1.

The stock unit identity (**SKU**) contains the unit's main characteristics i.e. specification, shape, size, colour, location. In terms of usage and value of the **SKU**, we have a so called $A - B - C$ classification for most inventory stocks by Silver (1985), in which class A item means the **SKU**s amounting up to the largest share of total inventory value and is given serious attention by any inventory manager; class B item represents moderate value of the **SKU**s for any inventory system; and class C item is the **SKU**s which make up a less important portion on the inventory value than those of class A and B items.

For some specific **SKU**s under the relevant context, there is a collection of decision policies for inventory manager to choose. However, it is very difficult to find one uniform scheduling and planning decision for general situations. All policies proposed in the sounding modelling are only effective within their assumptions and constraints.

In terms of our interests on the holdout lateral transshipment policy to the multi-location inventory system, to our knowledge from the related research, our policy is appropriate for inventory systems which deal with slow moving, high value **SKU**s and any loss of **SKU**s could result in a significant consequence on the customer service and business performance.

For the sake of any inventory control system, we need to define some terms to measure the system performances. Inventory total cost, the most important performance measure for any inventory system, consists of several components including ordering cost, holding cost, stockout cost, backorder cost and transshipment cost if necessary. Definitions of these cost components will be introduced in the later section. In addition, we use other performance measures such as direct fill rate, back order fill rate and transshipment fill rate to assess the system behaviours under the given policy.

Note 1.1. By default, in our research, the terms, location and depot are interchangeable.

1.1.2 Inventory management strategy review

Generally, the primary objective of any inventory system is to provide products or services with a reasonable cost at the right time and right place. On the whole, objectives for any inventory system operations are to achieve a good balance between a high-level of customer service, manageable inventory system and economical financial operations cost.

As far as large business organizations are concerned, management strategies are executed at hierarchy levels. In Silver (1998), four strategy levels are classified as follows.

1. **Enterprise strategy:** macroeconomic and microeconomic effectiveness of any strategy applied to the economy and society
2. **Corporate strategy:** operations identity and resource deployment
3. **Business strategy:** compete with competitors to maximize market share
4. **Functional area strategy:** maximization of resource productivity and the development of distinctive competencies

Clearly, the holdout lateral transshipment decision policy can be categorized as a functional area strategy, at the bottom level of these four strategy levels. The aim of our research is to give quantitative analysis support to the decision maker from the stochastic modelling and simulation implementations.

According to Silver (1998), direct or indirect consequences of inventory management strategy to the operations performance is in finance and marketing respects. A good cost-effective strategy brings a huge cost saving to whole enterprises and gain the competitive advantages by improving response times and satisfactions to customers. On the contrary, a poor inventory control management strategy results in catastrophes to some businesses. The lack of competitiveness for large US and European car manufacturers against Japanese and Korean competitors is a good example to show how important of good inventory control management strategy to business decision makers.

1.1.3 Dimensions of an inventory control system

In practice, an inventory control system consists of three basic dimensions: location structure, replenishment order policy and lateral transshipment policy.

Location structure of inventory control system: a location structure includes two types of structures: multi-echelon and multi-location. Within a multi-echelon inventory control system, at the bottom level, each local depot is supplied by one central regional distribution centre which is supplied by a distribution centre at the upper level. Among local depots, lateral transshipments are allowed in case of emergency **stockout** at requesting depot. A lateral transshipment between regional distribution centres is also allowed. The relationship is illustrated in Figure 1.2. Within a multi-location inventory control system, as showed in Figure 1.3, we only consider a local system in which each location is replenished by the same central distribution centre, however, a transshipment is permitted among them in case of stockouts at any location.

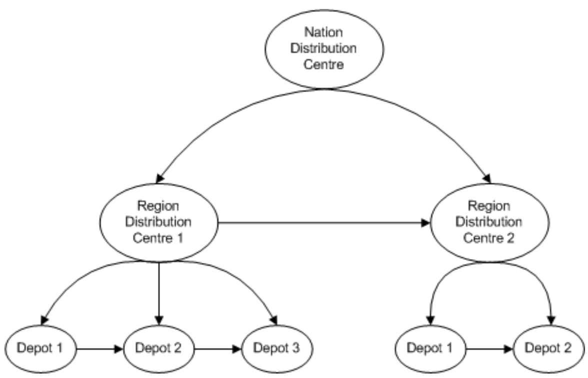

Figure 1.2. Three-echelon system

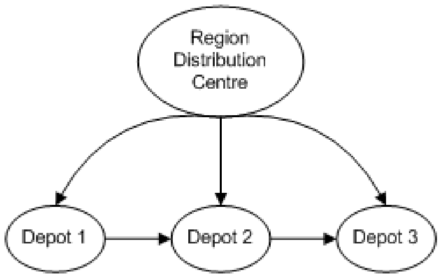

Figure 1.3. Three-location system with a region distribution centre

Replenishment order policy: for any inventory control system, we have to answer two questions: how often should the system be reviewed to place an order? How large should each order be? Answering the first question gives rise to the frequency of reviewing the system. The system is called a periodic review system whenever **inventory position** is reviewed at a fixed epoch time. On the contrary, we call the system the continuous review system whenever the inventory position is reviewed continuously and replenishment orders are placed if a pre-defined condition holds.

The answer to the second question leads to an assortment of replenishment order policies. The major replenishment order policy in our field comprises the following variants: (R, Q) and (s, S) polices given in Silver (1998).

Basically, a classification of replenishment order policies depends on a condition to trigger a replenishment order and an amplitude of order quantity once an order was placed. For a continuous review system, under (R, Q) replenishment order policy, an order of Q units of one type of item is placed whenever the inventory position reaches the reorder point R. For a periodic review system, (s, S) policy places a variable order quantity of one kind of item to restore the inventory position to S whenever the inventory position at a review epoch is less than or equal to s.

Lateral transshipment polices: Similar to the questions for the replenishment order policy, we need to answer three questions for lateral transshipment polices: when to make the transshipment; what type of transshipment needs to be carried out; what quantity for each transshipment delivery?

The conditions to trigger a transshipment delivery are dependent on the specific pooling policy. For instance, transshipment can be delivered when the inventory level or inventory position reaches one threshold value. There are two types of lateral transshipment policies: unidirectional and bidirectional. In addition, we also need to define the quantity for each transshipment delivery whenever a transshipment decision is made. The common practice is to deliver one item for each transshipment or to deliver a variable transshipment quantity depending on the specific pooling policy.

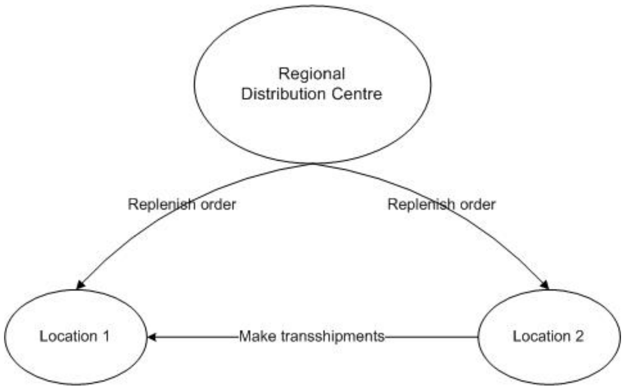

Figure 1.4. Two-location system with unidirectional transshipment policy

In our research, we pay attention to the single item transshipment in the continuous review two-echelon, two-location inventory system with (R, Q) replenishment policy and the unidirectional transshipment policy. The situation is illustrated in Figure 1.4.

1.1.4 Inventory control system performance measures

The assessment criterion for any inventory system is differentiated by comprehensive system performance measures. Principally, inventory cost consists of four classifications: order cost, holding cost, stockout cost and backorder cost.

1. **Order cost:** the procurement cost results from the order placed, including configuration cost for each order placed and personnel cost for each order placed

2. **Holding cost:** the cost is assumed to occur at a rate proportional to inventory level

    a. Space costs: costs charged for the spatial dimension of the **SKU** consumed

    b. Capital costs: inventory operations in the finance term

    c. Inventory service costs: costs occurred in term of insurance and taxes

    d. Inventory risk costs: costs arising from deterioration, damage or obsolescence of the **SKU**

3. **Stockout cost:** the cost is assumed to occur when demand cannot be met immediately due to a lack of inventory at the location facing the demand and a lack of a suitable location to transship from. It is assumed that the stockout cost is a fixed cost per item and that all stockouts are backordered.

4. **Backorder cost:** the costs incurred due to damaged goodwill as a result of the delay in meeting the demand when a stockout occurs. The cost is assumed to be proportional to the time taken to meet the demand.

Additionally, we define following terms to measure system performances:

1. **Fill rate:** the proportion of demand arising at the location that is met by local stock

2. **Backorder fill rate:** the proportion of demand arising at the location that is met by backorders.

3. **Transshipment fill rate:** the proportion of demand arising at the location that is met by transshipment

1.2 Research Objectives

In order to achieve a good understanding of our own research, it is vital to identify our research objectives and relevant methodologies. Normally, extensive reading and study of relevant papers are helpful to understand problem domains and to develop our own solutions.

From our study and observations on historical research of the lateral transshipment policy to the multi-echelon and multi-location inventory control system, we found that searching an optimal total cost to the multi-location inventory control system with general transshipment policy or with holdout transshipment policy still has more works to do due to its complexity. For example, there are more than 20 parameters in a two-location system, and the system performances are arbitrarily dependent and influenced by some individual parameter or combination of these parameters under some specific circumstances. Hence, there is no uniform optimal solution of general approximation models for most situations under the given policy. Most papers only explore models with complete pooling or no pooling policies. Some promising estimations for performance measures are only valid and feasible for some specific pooling policies with particular constraints.

In short, a lack of a general solution to the problem domain gives only limited helps to the inventory manager from the practical point of view. Meanwhile, more freedoms on assumptions to the inventory systems are harmful for a fundamental understanding of the holdout transshipment policy to multi-echelon or multi-location inventory control systems under given circumstances.

Put these issues together, we propose that our research at least contains four basic characteristics: **Simplification, Accuracy, Framework, Documentation.**

- **Simplification:** models under our study are simple and reflect fundamentals of the holdout transshipment policy.
- **Accuracy:** approximations on the optimal total cost and the relevant performance measure are verified with corresponding simulations.
- **Framework:** under the same assumptions, our approximations can be extended to an inventory system which has more than two locations.
- **Good documentation:** good documentation is helpful for the further research. The documents on the approximation models and simulation implementations need to be presented in a good manner conveniently for the further development.

Generally, we keep our research concentrations on the study of a single product in a single echelon two-location inventory control system, a continuous review system with (R, Q) replenishment order policy and one item for each transshipment delivery with unidirectional lateral transshipment policy. Conventional stochastic approximation models and Markov decision process approaches are considered in our study. Estimations of the optimal total system cost and performance measures are examined and verified with simulations.

Many reasons could lead us to draw a line between our research and others. However, the key reason we consider a single item unidirectional transshipment is due to the complexity of the system. There are more than 20 parameters which could give an impact to the system performance of a system which has intensive interactions due to the holdout transshipment policy. It is important to understand their impacts to the system under the holdout transshipment policy by identifying several sensitive parameters first. Therefore, we choose to concentrate our study on a single-item unidirectional transshipment model rather than a multiple-item bidirectional transshipment model.

1.3 Research methodologies

As illustrated in Figure 1.5, our research mainly considers conventional stochastic modelling and simulation modelling approaches. Under the given transshipment policy, stochastic approximations and simulations are formulated. On the vertical direction, there are mappings between theoretical and implementation layers for both simulation and Operations Research (**OR**) approach; On the horizontal direction, there are mappings between simulation and stochastic models at the theoretical layer and between simulations and numerical results at the implementation layer. Therefore, we use simulations to guide us to develop better stochastic approximation models.

To distinguish our research from others, we aim to establish the approximation models which are verified with the equivalent simulations. Our research approach can be classified as a combination study of stochastic modelling, computing simulation and business models, which heavily involves approximation modelling and simulation programming. We are determined to contribute a more practical approach providing more insights into the inventory management system under the given transshipment policy.

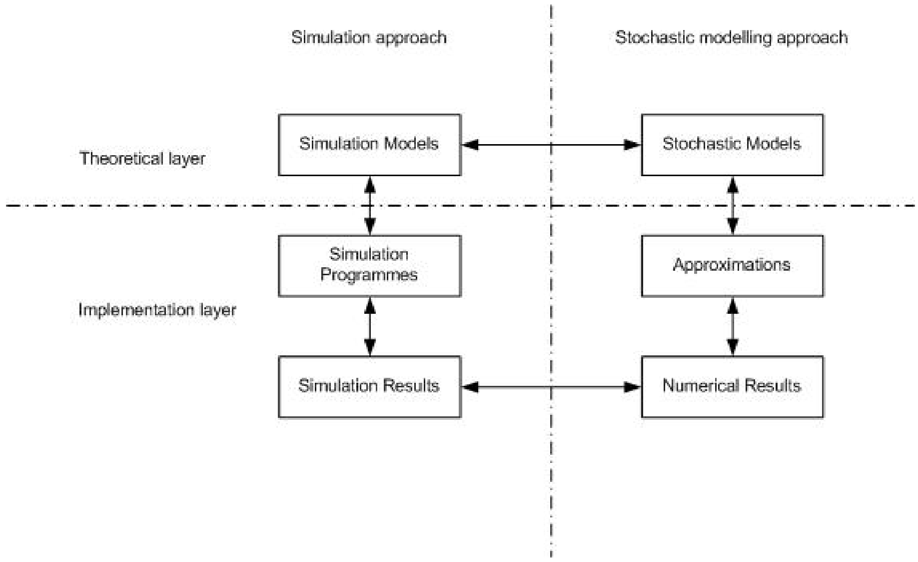

Figure 1.5. Four research quadrants

1.4 Outline of the thesis

Through our research, we define assumptions, establish stochastic approximation & semi-Markov process models under the given transshipment policy and provide analysis of these models from the numerical results. Generally, this thesis document is organized in the following manner:

In the introductory chapter, we give the fundamentals of the inventory management strategy and inventory control system. All relevant notations and terminologies are explained and discussed for the further research. Meanwhile, research objectives and methodologies are also identified and explained.

In the second chapter, theory foundations for the Markov decision process modelling, conceptions and classifications of major history research works are introduced. Most research works on the lateral transshipment or holdout transshipment policy are categorized. Among them, some key research contributions are discussed and those which are close to our research are examined in more details as well.

In the third chapter, before we start to discuss approximation models to the two-location system, we introduce theory and programming structures of simulation implementations. In fact, the simulation implementation in our research proves to be an effective tool to assist us to make a better approximation model.

In the fourth chapter, as the first step, we have a quick study of the single-depot system, which provides a foundation for other decomposition models. The model is formulated and verified with simulation respectively. The optimal average total cost is searched by using an exhaustive searching method. Then, we establish an approximation model, which decomposes a two-location system into two independent locations with constant demand rates. The transshipment agreement probability is introduced to approximate the unidirectional holdout transshipment policy. This approach is verified with the corresponding simulation.

In the fifth chapter, in order to achieve a high accurate degree of our approximation models, we decompose the two-location system into two independent systems with non-constant demand rates. In the first part of this chapter, we define a decomposition approximation model where the transshipment decision is still dependent on the transshipment agreement probability. In the second part of this chapter, we give a decomposition approximation model where the transshipment decision is dependent on the holdout threshold at location 2 explicitly. For each model, standard techniques are applied to derive explicit expressions for the average cost rate, and an iterative solution method is developed to find an optimal holdout transshipment policy. To make assure the correctness of our approximations, we check the accuracy of these approximations during a single cycle with corresponding simulations. The numerical results are provided to illustrate the effects when the partial pooling becomes the optimal transshipment policy.

In the sixth chapter, it is worth considering the Markov decision process technique to have more insights of the unidirectional holdout transshipment policy because of the nature of dynamic programming. Hence, we develop a semi-Markov decision processes (**SMDP**) model with exponential lead time to find a solution to inventory control system under the holdout transshipment policy. Furthermore, the **SMDP** model is extended to the case of phase-type distribution for the lead time in order to approximate the fixed lead time effectively. In addition, we investigate how optimal transshipment decisions depend on the state of the replenishment order process for the **SMDP** models with exponential lead time and the **SMDP** models with phase-type lead time via dynamic programming.

Finally, in the seventh chapter, we summarise all models we explored in our research, discuss the advantages and disadvantages of those modelling approaches and propose the directions for the further research.

For further documentary reference, the core classes and methods for the simulation and approximation models implemented by **JAVA** programming are provided in the appendix part.

Glossary

SKU: The stock unit identity contains the unit's main characteristics i.e. specification, shape, size, colour, location
stockout: The situation when the current stock level can not meet its local demand
inventory position: The addition of on hand stock and outstanding orders minus the outstanding backorders

Chapter 2
Relevant research

2.1 Introduction

Hitherto, comprehensive researches on the emergency transshipment policy to the multi-echelon and multi-location inventory control systems have been done since the **METRIC** model of Sherbrook (1968). A group of researchers have made their contributions in this field. Due to the complexity of multi-echelon and multi-location inventory control systems, we firstly need to understand the problem domains and methodologies in this field in order to develop our own research.

Basically, for any research in the field of multi-echelon and multi-location inventory control systems, we need to identify following relevant issues including the type of inventory control system, the type of inventory control system review, replenishment order policy, lateral transshipment policy and research methodology. In other words, any research in this field should at least handle one of these following issues:

- **Inventory control system type:** for most papers published in this area, only two echelons and single echelon inventory system are carefully examined due to the particular research interests of different researchers.

- **Inventory system review type:** basically, an inventory control system can be reviewed on the periodic or continuous basis. In practice, most supermarket retailers use periodic review inventory control system to monitor and replenish their food stock on a daily or weekly basis because it is easy to implement. However, for some high-quality service activities such like military and airplane maintenance, the inventory control system has to be reviewed continuously only because of a significant economic consequence for any unmet demand.

- **Replenishment order policy:** for any inventory control system, this is a fundamental question that needs to be answered. We should make it clear how the replenishment order is placed, under what conditions and what replenishment policy is imposed whenever the replenishment order is placed. In our research field, the conditions to trigger a replenishment order are possibly dependent on two factors: **inventory level** (stock on hand) or **inventory position** (stock on hand plus the outstanding orders minus back orders placed). Inventory system performance measures which are based either on the inventory level or on the inventory position result in different research interests. Once the condition for a replenishment order is met, a collection of possible replenishment order policies become available to choose from. The most popular replenishment order policies in our field are:

  For continuous review system

  - Order up to S: an order with variable quantity is placed to ensure that the inventory position reaches to the order-up-to level S when inventory position falls to the reorder point s
  - (R, Q): the system places an order with Q units of one kind of item when the inventory position is below R, this outstanding order will arrive after the lead time L.

  For periodic review system

  - Order-up-to-level (R, S): during R units time of each cycle, an order is placed to replenish inventory position up to the level of S
  - (R, s, S): during every R units time of each cycle, the inventory position is checked, if it is less than the reorder point s, an order is placed to raise it back to S. Otherwise, nothing needs to be done until the next review cycle.

- **Transshipment policy:** lateral transshipment policies are various in respect of transshipment directions, conditions on transshipment and quantity of transshipment. Furthermore, a transshipment decision is made either before all demand is realised, or after all demand is realised or in response to stockouts.

- Transshipment directions: the lateral transshipment direction among the inventory control system contains two types: unidirectional and bidirectional transshipment. Namely, for a multi-location depot system with k locations, $k = 1, ...,$ n, unidirectional transshipment allows a transshipment only from location $k+1$ to location $k, k = 1, ..., n-1$, vice versa not permitted. On the other hand, bidirectional transshipment allows transshipments between any two neighbouring depots.
- Conditions on the transshipment: complete pooling and no pooling policies are commonly used. Complete pooling policy means that when one stockout occurs at one location, a transshipment request is to be met if and only if the requested location has one stock; no pooling policy means that the transshipment request due to a stockout is never met by the transshipment from the requested location. In addition to these two pooling policies, we might define a partial pooling policy which means that a transshipment can be delivered when a certain condition is met, e.g. the **holdout pooling policy** is one of partial pooling policies in which a transshipment request will be met when the inventory level at the requested location is greater than or equal to one specific holdout threshold value. Otherwise, the transshipment request is rejected.
- Transshipment quantity: normally, we permit only one unit of a kind of item for each transshipment delivery. However some models also agree to a variable number for a kind of item for each transshipment delivery. The function of the transshipment quantity might depend on conditions of inventory levels between the transshipment requesting and requested locations under some circumstances.

- **Research methodology:** numerous methodologies in the lateral transshipment study for multi-echelon and multi-location inventory systems can be considered. Major research methodologies include simulation, analytic modelling, heuristic approximation, Markov decision process, and Nash Equilibrium techniques. Meanwhile, Approximations of system performance measures including the optimal average total cost, direct fill rate, transshipment fill rate and backorder fill rate for the specific model are discussed respectively.

2.2 Markov decision process theory foundation

In this section, we give an introduction of major definitions and theory foundations of Markov decision process which are relevant to our research. Due to the popularity of notations used by Tijms and Puterman from their series books on the stochastic modelling: Tijms (1986, 1994, 1995 and 2003) and Puterman (1994), we use their notations in our Markov decision process (**MDP**) models.

Principally, a **MDP** model is a controllable dynamic system in which the system evolves over a finite or infinite horizon by the possible actions made at each review epoch at the each specific state. At each review epoch, the system stays at one state and a decision of moving to next state is based on the possible actions at the current state. A **MDP** is a time homogeneous process, which means that the decision made is only based on the current state regardless of past historic state transitions. State transitions are reflected by the state transitions probability distributions.

The **MDP** modelling family comprises two classes, one is the discrete-Markov decision process (**DMDP**) and the other is the semi-Markov decision process (**SMDP**). The key difference between these two classes is the state review epoch method. For **DMDP** model, the system is reviewed at equidistant epoch time $t = 0, 1, 2, ..., N$, On the contrary, the system reviews the state epoch time at random pace in **SMDP** model. In that sense, **SMDP** is a continuous review Markov decision process.

At each review, the system is classified into one of a possible number of states and subsequently a decision has to be made. The set of possible states is denoted by I. For each state $i \in I$, a set $A(i)$ of decisions or actions is given. The state space I and the action sets $A(i)$ are assumed to be finite. The cost, an economic consequence of decisions taken at the review epoch is incurred after decision making. If at a decision epoch that action a is chosen in state i, then regardless of the past history of the system, the following terms hold:

a) $\boldsymbol{c_i(a)}$ = an immediate cost $c_i(a)$ is incurred

b) $\boldsymbol{R}$ = a stationary policy R is imposed on the whole horizon

c) $\boldsymbol{p_{ij}(a)}$ = at the next decision epoch the system will be in the state j with probability

$$p_{ij}(a), \text{ where } \sum_{j \in I} p_{ij}(a) = 1, i \in I$$

d) $\boldsymbol{\tau_i(a)}$ = the expected time until the next decision epoch when action a is chosen in state i

Note that the one-step immediate cost $c_i(a)$ and the one-step transition probability $p_{ij}(a)$ are assumed to be homogeneous. The immediate costs $c_i(a)$ often represent the expected cost incurred until the next decision epoch when action a is chosen at state i.

In order to achieve the unique equilibrium distribution $\{p_{ij}(R),\ j \in I\}$, we assume that the Markov chain is an ***unichain*** Markov Chain, which has no two disjoint closed sets.

In terms of computing implementation, normally, we apply a policy-iteration algorithm, linear programming algorithm or value-iteration algorithm to find the optimal cost. however the first two algorithms are not discussed due to the complexity and effectiveness of numerical computations. In our research, we only utilize the value-iteration algorithm to find the optimal average cost .

In order to apply the value-iteration algorithm to search the optimal average cost, for each average cost optimal stationary policy, we have two assumptions as such

- *The associated Markov chain $\{X_n\}$ has no two disjoint closed sets*
- *The associated Markov chain $\{X_n\}$ is aperiodic.*

Generally, the *value-iteration algorithm* computes recursively a sequence of value functions for searching the minimal average cost per time unit by dynamic programming.

Under the weak unichain assumption, for each average cost optimal stationary policy, the upper and lower bounds on the minimal average cost converge under the value-iteration algorithm.

The algorithm for computing the long-run average cost per time unit for **SMDP** is too complicated to implement. Instead of this, we extend the value-iteration algorithms from **DMDP** to the **SMDP**'s implementation, as each stationary policy the average cost per time unit in the **DMDP** is the same as in the **SMDP**.

Therefore, we will apply the data-transformation method followed to transform the **SMDP** problem to the **DMDP** problem.

Data-transformation method

First choose a number τ with

$$0 < \tau \leqslant \min_{i,a} \tau_i(a)$$

Consider now the discrete-time Markov decision process model in which basic elements are given by

$$\bar{I} = I, \bar{A}(i) = A(i), \ \ i \in \bar{I}$$

$$\bar{c}_i(a) = c_i(a)/\tau_i(a), \ \ a \in \bar{A}(i) \text{ and } i \in \bar{I}$$

$$\bar{p}_{ij}(a) = \begin{cases} (\tau/\tau_i(a))p_{ij}(a), & j \neq i, \ a \in \bar{A}(i) \text{ and } i \in \bar{I} \\ (\tau/\tau_i(a))p_{ij}(a) + [1 - (\tau/\tau_i(a))], & j = i, \ a \in \bar{A}(i) \text{ and } i \in \bar{I} \end{cases}$$

Value-iteration algorithm

For the **SMDP** model, the formulation of a value-iteration algorithm is not straightforward. However, by the data transformation method, the **SMDP** model is converted into a discrete-time **MDP** model and both models have the same average cost for each stationary policy. A value-iteration algorithm for the original **SMDP** is replaced by the value-iteration algorithm for the transformed discrete-time Markov decision model.

Hence, we have the formulation recursion method for the **SMDP** model as such

1. ***Initialization:*** Choose $V_0(i)$ such that $0 \leqslant V_0(i) \leqslant \min_a \{c_i(a)/\tau_i(a)\}$ for all i. Choose a number τ with $0 < \tau \leqslant \min_{i,a} \tau_i(a)$. Let $n := 1$

2. ***Value-iteration one step:*** Compute the function $V_n(i), i \in I$, from

$$V_n(i) = \min_{a \in A(i)} [\frac{c_i(a)}{\tau_i(a)} + \frac{\tau}{\tau_i(a)} \sum_{j \in I} p_{ij}(a) V_{n-1}(j) + (1 - \frac{\tau}{\tau_i(a)}) V_{n-1}(i)] \tag{2.1}$$

   Let $R(n)$ be a stationary policy whose actions minimize the right-hand side of equation (2.1).

3. ***Bounds on the minimal costs:*** Compute the bounds

$$m_n = \min_{j \in I} \{V_n(j) - V_{n-1}(j)\}, \quad M_n = \max_{j \in I} \{V_n(j) - V_{n-1}(j)\}$$

4. ***Stopping condition:*** when $0 \leqslant (M_n - m_n) \leqslant \epsilon m_n$, where ϵ is a specified accuracy number. Otherwise, go to step 3.

5. ***Next iteration:*** $n := n + 1$ and go to step 1.

In chapter 6, we will use the theory foundations for Markov decision process to derive our own **SMDP** models for the two-location inventory system.

From the following sections, we give an introduction to the relevant historical studies of the lateral transshipment policy and review some relevant research in the similar area. Giving a good reference to readers, we classify all relevant papers in Table 2.1. Afterwards, a collection of papers, which are either ranked as the most important contributions in this field or relevant to our research, are discussed in more details in the succeeding sections.

2.3 Historical transshipment research

2.3.1 Research classifications

In the preceding sections, we give an introduction to all relevant terms and theoretical foundations relating to the specific research approach. Clearly, these give rise to a research classification in this field. In Table 2.1, we present a chronological summary of all major published papers.

Paper	Model	Replenishment order policy	Transshipment	Research method	Dependence
Das (1975)	one period review	order-up-to S	lateral	analytic	N/A
Lee (1987)[1]	continuous review multi-echelon	one-for-one	lateral	analytic	N/A
Jonsson and Silver (1987)	period review two echelon	order-up-to S	no	approximation	N/A
Karmarkar (1987)	periodic review	unknown	no	dynamic programming	N/A
Axsäter (1990)[2]	period review two echelon	one-for-one	lateral	analytic	Lee(1987)[1]
Tagaras and Cohen (1992)[1]	period review two echelon	order-up-to S	lateral	analytic	N/A
Sherbrooke(1992)		one-for-one	lateral	simulation	N/A
Archibald *et al.* (1997)[2]	period review single echelon two depot	order-up-to S	lateral	Markov decision process	N/A
Alfredsson and Verrijdt (1999)	two echelon	one-for-one	lateral	analytic	Axsater(1990) Lee(1987) Sherbrooke(1992)
Evers(2001)	two location	fixed order Q	lateral	heuristics derivation	Lee(1987) Axsater(1990)
Tagaras (2001)	periodic review not multi location	order-up-to S	emergency	analytic simulation	N/A
Kukreja *et al.* (2001)	continuous review single echelon multi-location	one-for-one	complete pooling	analytic	Lee(1987) Axsater(1990)
Grahovac and Chakravarty (2001)	multi-echelon	$(S-1, S)$	Sharing and transshipment	analytic simulation	Lee(1987) Axsater(1990)
Rudi *et al.* (2001)	single period	(R, Q)	lateral	Nash equilibrium	Krishnan(1965) Robinson(1990)
Tagaras (2002)	periodic review	order-up-to S	lateral	simulation	N/A
Axsater(2003a)[2]	continuous review single echelon multi-location	order-up-to S (R, Q)	unidirectional transshipment	analytic simulation	Lee(1987) Axsater(1990)
Axsater(2003b)[2]	continuous review single echelon multi-location	(R, Q)	unidirectional transshipment	heuristic simulation	Lee(1987) Axsater(1990) Alfredsson and Verrijdt(1999)
Xu *et al.* (2003)	continuous review two location	(R, Q)	lateral	simulations analytic	Lee(1987) Axsater(1990) Sherbrooke(1992)
Dong (2004)	periodic newsvendor	Q	transshipment	Stackelberg equilibrium	N/A
Hu *et al.* (2005)	periodic review	(s, S)	emergency	dynamic programming simulation	Krishnan(1965) Robinson(1990)
Minner (2005)	continuous review	(R, Q)	transshipment	analytic	N/A
Zhao *et al.* (2005)	continuous review two location	(S, K)	sharing	game theory equilibrium	Krishnan(1965) Lee(1987) Rudi et al. (2001)
Wee and Dada (2005)	single period review	order-up-to S	lateral	mathematical programming	Krishnan(1965) Rudi et al. (2001) Robinson(1990)
Herer (2006)	periodic review	order-up-to S	sharing	LP/IPA	Robinson(1990)
Ozdemir (2006)	periodic review	order-up-to S	transshipment	LP/IPA	Herer *et al.* (2004)
Seifert *et al.* (2006a) Seifert *et al.* (2006b)			Unidirectional transshipment	mathematical model	Rudi *et al.* (2001)
Wong *et al.* (2006a)	continuous review	$(S-1, S)$	lateral	integer programming	Lee(1987) Sherbrooke(1992) Alfredsson and Verrijdt(1999) Grahovac and Chakravarty(2001)
Hu *et al.* (2007)	single period	(R, Q)	lateral	Nash equilibrium	Rudi *et al.* (2001)
Zhao *et al.* (2008)	continuous review	make-to-stock	lateral	Dynamic programming	Zhao *et al.* (2005)
Kutanoglu (2008)	continuous review	$(S-1, S)$	lateral	Markov analysis	Axsater (1990)

Table 2.1. Relevant research summary

Note 2.1. Paper marked with superscript[1] is categorized as the most important research in this field. Paper marked with superscript[2] uses the approach which is close to our research approach. The papers with N/A mark in research dependence means that no similar research was done prior to this one, or it is difficult to identify its dependence.

2.3.2 Introduction to the transshipment research

Firstly, we give a brief chronological review on the relevant research in the transshipment policy study to multi-echelon and multi-location inventory control systems in this section. Major research works which are very close to our research approaches are also explained in the subsequent sections.

In the paper from Jonsson and Silver (1987), they examined a two-echelon system with a periodic review system and compared the system service levels between the systems both with and without the redistribution policy.

The multi-location inventory problem considered in Karmarkar (1987) is a natural extension of the so-called "newsboy" problem to many locations and multiple time periods with proportional transfer costs and replenishment costs. A restricted Lagrangian decomposition of the problem that results in an easily computable lower bound for the problem and a dual relaxation that gives an upper bound are presented. In numerical example, two sets of tests were conducted on two-location two-period and three-period problems. The approach in Karmarkar (1987) is very different with our research approach.

Sherbrooke (1992) paid special interest to the simulation approximation of back orders within the multi-echelon system with the lateral transshipment policy from the VARI-METRIC model. It was assumed a Poisson demand process at each location and one-for-one replenishment order policy. The VARI-METRIC model assumes that resupply requests are filled on a first-come, first-served bases.

Simulation results show that when lateral transshipment times are less than the transshipment time from a depot to a base, any item completing depot repair should be sent to some base immediately.

In the paper of Sherbrooke (1992), it was reported that it is very common to achieve 30-50% backorder reductions from the introduction of lateral transshipment policy when the demand rates are low.

Alfredsson and Verrijdt (1999) developed a two-echelon model with one-for-one replenishment order policy and lateral transshipment policy. At each location, they assumed a Poisson demand process. In case of a demand at a location, the strategy for filling this demand are considered as the following order: fill the demand from stock on hand, the demand is satisfied by an emergency lateral transshipment from another randomly chosen location are situated in one area such that the transshipment is much shorter than the replenishment time from the central warehouse, the demand is satisfied by a direct delivery from the central warehouse if it has stock on hand, and the demand is satisfied by a direct delivery from the plant which has infinite supply.

They developed two models, the first model approximated the fraction of the total demand that is met by direct delivery from the central warehouse and the fraction of the total demand that is served by direct delivery from the plant if a customer arrives when there is no stock on hand, locally or at the central warehouse. The second model applies Markov analysis to derive the fraction of demand is met by transshipment and the fraction of demand from stock on hand.

Those system performances were validated with the numerical results of simulations, it was also revealed that a combination of lateral transshipment and direct deliveries resulted in a significant cost saving.

Evers (2001) presented two related heuristics in two important respects: the costs and benefits of transshipments are treated more explicitly and stock transfers of multiple units are considered. The first heuristic Economic Order Quantity (EOQ) model deals with the emergency transshipment of a single unit and the second heuristic Economic Order Quantity (EOQ) model addresses the general case of multiple units. The numerical examples showed that emergency transshipments may be quite appropriate on many occasions.

Rudi *et al.* (2001) paid their attention on the profit by the given pooling policy. They developed single-period models by investigating cases: the newsvendor problem, inventory decisions for each location are centrally coordinated to maximize aggregate profits, inventory decisions are made locally and the expected profit at each location now depends not only on order quantities, but also on transshipment prices. They considered the lateral transshipment problem by Nash Equilibrium approach. Through this approach, local decision makers optimize their own performance, rather than a single central decision maker optimizing overall performance. They found that if each location aims to maximize its own profits, their inventory choices will not, in general, maximize joint profits. To the extent that inventory choice varies with transshipment prices, transshipment prices that induce the locations to choose inventory levels are consistent with joint-profits maximization.

Kukreja *et al.* (2001) gave a study of a two-echelon system with a one-for-one replenishment policy. It was assumed that each plant faces independent and stationary Poisson demand, and complete pooling of stock is allowed among the plants. The sourcing rule was based on transshipment costs: from the locations that have stock on hand, transshipment from that location with the lowest transshipment cost to the location needing the unit.

The model is formulated for slow-moving and expensive parts in which complete pooling of stock is allowed between the locations. Experimentation was performed to validate the results from the model and to explore the benefits of pooling of stock in a pilot application. A queuing-based model was developed to assist in determining the optimal allocation of inventory throughout the collection of stocking parts.

The analytical model was tested on a variety of measures and reported to perform well. The experiments also showed that substantial savings were achieved by allowing for pooling of stock. The pilot study showed that the model bring about 68% cost reductions. The further research is needed for the non-complete pooling policy.

Tagaras and Vlachos (2001) considered a single-item periodic review inventory system that employs a base stock with order-up-to-S replenishment order policy. The review period is taken as given and determined by considerations such as the need for coordinating the regular replenishment with those of other items.

The numerical results proposed that the emergency replenishment with shorter lead times and higher acquisition costs could result in a big cost saving by their optimization algorithm.

Grahovac and Chakravarty (2001) considered a two-echelon system with one-for-one replenishment order policy. They assumed the Poisson demand process at all retailers and developed a solution methodology for analyzing a supply chain with expensive, low-demand items, and the particular pattern of transshipment in centralized and decentralized supply chain environments. The numerical results show that the ability to quickly move inventory within the lowest echelon can reduce the overall cost by up to approximately 20%. However, they concluded that this savings is not always accompanied by a reduction in the overall inventory in the supply chain. Furthermore, any reductions in inventory occur at the distribution centre, while retailers experience stable or even increasing inventory level. These contradictory trends can cause problems in decentralised supply chains in which some participants may need extra incentives and assurances to join the inventory-sharing and transshipment arrangement.

Tagaras and Vlachos (2002) considered a two-echelon two-location periodic review system. At each location, they assumed general random demand process. One of the main contributions of this paper is that it deals with non-negligible transshipment times. Furthermore, it places special emphasis on the investigation and sensitivity analysis with regard to the type (shape) and variability of the underlying demand distribution. It reviews the effect of the type and variability of the demand distributions on the pooling group performance. Specifically, Normal distribution stands for high demand with a low coefficient of variation and Poisson distribution stands for low demand items. From the numerical results, they concluded that the shape and variability of demand distribution affect all aspects of design and operation of pooling groups. No general statements can be made about the appropriateness of lateral transshipment policies without explicitly taking into account the characteristics of the demand distributions.

In the paper of Axs$\ddot{a}$ter (2003b), Axs$\ddot{a}$ter proposed a simple but effective approach to study lateral transshipment policy to the multi-location inventory control system. He consider a number of locations having independent compound Poisson demand process and assume an $(R,\ Q)$ replenishment order policy with the fixed lead time. The objective is to find a suitable replenishment policy which includes reorder points and batch quantities for normal replenishment as well as a decision rule for lateral transshipments.

The decision rule for the optimal transshipment policy was evaluated in a series of simulation studies for the situations include lower transshipment costs, fixed transshipment cost and (R, Q) from the no-pooling case. The numerical results indicates that it is a reasonable approximation to use the easily available reorder points and batch quantities that are optimal without lateral transshipments. This approach always gives expected savings compared with the no-pooling case.

Dong and Rudi (2004) paid more attention to the transshipment effect on both the retailers and manufacturer in terms of order quantities and profits. They considered a distribution system consisting of n retailers owned or operated by the same entity and one manufacturer that sells to these retailers in a single period. The lateral transshipment between any two locations can be allowed after observing demand before having to satisfy it.

They developed models to approximate the retailer's profit and manufacturer's profit. Furthermore, they considered the case of exogenous wholesale price, where the manufacturer is a price taker and can not affect the wholesale price and the case of endogenous whole sale price by modelling the interaction between the manufacturer and the retailers. Based on the Stackelberg equilibrium game between manufacturers and retailers, the numerical results demonstrated that the impact of transshipment on the manufacturer and the retailers is largely dependent on whether the manufacturer is a price taker or a price setter.

Hu $et\ al.$ (2005) considered a periodic-review system with centralized-ordering and emergency transshipments by using the dynamic programming approach. They developed the approximation models with the general emergency shipment and normal distributed demand approximation respectively. From their numerical results, they concluded that for a small number of stores and small transshipment costs, relative to the holding and stock-out costs, inventory policies may be obtained from a simplified model using zero transshipment costs but using transshipment as a means to solve emergency situations. If transshipment costs are equal or greater than the holding plus stock-out costs then a model without transshipment can be used. In general using transshipment could be a very cost effective way of reducing inventories for situations with a large number of stores.

Wee and Dada (2005) considered one warehouse and n-retailer inventory system. At each location, it was assumed an order-up-to S replenishment order policy before the selling period and random demand from an arbitrary multivariate distribution. The objective is to determine the optimal stocking levels so as to maximize the expected profit (net of transshipment and penalty cost). They advanced a series of studies on the system performances on the impact of lateral transshipment policy for five systems: retailer-only, retailer-first system, warehouse-first system, warehouse-only and no-pooling system. It reported that the optimal transshipment policy would exclude the partial transshipment policy from their periodic models.

The paper written by Zhao *et al.* (2005) is another recent published work which models a continuous-review infinite-horizon dealer network consisting of two independent dealers. They assume each dealer uses a base-stock and rationing policy. They employed the Nash Equilibrium from Game theory to the decentralized supply chains system. Two system performances: dealer's inventory-sharing behaviour and the expected system back orders are examined.

They made their contributions from at least key respective. First, they review a network in which the dealers have the flexibility to share inventory to satisfy customer demand. Second, they consider inventory sharing as a multiple demand class problem, in which a dealer's own demand is considered as a higher priority class of demand, and sharing requests from other dealers, modelled endogenously are considered as a low-priority class of demand. Finally, they consider the interactions between the dealers in a decentralized setting.

From the numerical study, they concluded that increasing the manufacturer's incentives for sharing and reducing the cost of sharing are two different ways to affect dealers' inventory-sharing behaviour in a decentralized system. Incentives for sharing lead to achieve full sharing.

Wong $et\,al.$ (2005) reviewed a single-echelon system in which N locations coordinate by pooling their spare-part stock of a particular repairable item. In their work, they developed models with non-zero or zero lead time and delayed lateral transshipments. By considering the inventory situation in the group (aggregate) and in the individual level, the problem can be simplified as a one-dimension Markov chain. Numerical experiments show that proposed approximation method is quite accurate and computationally efficient and delayed lateral transshipment can reduce the expected number of backorders significantly. This could mean that the assumption of instantaneous lateral transshipment has a risk as it can lead to non-optimal stocking decisions.

The modelling approach in the paper of Minner and Silver (2005) is very straightforward. They assumed a distribution system with two identical locations. At each location, there is a compound Poisson demand process and employs an (R, Q) replenishment order policy. Comprehensive numerical results were conducted in terms of two extreme transshipment policies: no pooling and complete pooling policies. The numerical results show that the approximations are reasonably accurate, especially for the case of pure Poisson demand. For the instances with higher percentages demand and longer lead times, the error percentages increases, especially for the case of no transshipments. The accuracy becomes better under transshipment.

Herer *et al.* (2006) considered a single-echelon multi-location system on the periodic review basis. The demand distribution at each retailer in a period is assumed to be known and stationary over time. The system is reviewed periodically and replenishment orders are placed with the supplier.

They considered the multi-location dynamic transshipment problem via an integrated infinitesimal perturbation analysis (**IPA**)/linear programming (**LP**) algorithm. First, an arbitrary number of non-identical retailers was considered with possibly dependent stochastic demand. Second, they modelled the dynamic behaviour of the system in an arbitrary period as a network flow problem. Finally, a simulation-based method using **IPA** for optimization was used.

Ozdemir *et al.* (2006) considered a single-item two-echelon system with periodic review system. If the demand exceeds the current inventory level at one or more of the stocking locations, while the inventory level exceeds demand at one or more of the stocking locations, then before the demand is satisfied lateral transshipments take place to decrease the overall overage and underage in the system. At each location, two decisions should be made every period: transshipment and replenishment quantities.

They developed a **LP** multi-location transshipment model with N non-identical locations and with capacity transportation constraints by a sample-path-based optimization approach. Moreover, they also proposed a stochastic approximation using **Monte Carlo** simulation by formulating and validating **IPA** derivative estimators. Their approach provides the formulation and validation of **IPA**-based gradient estimators for a stochastic optimization algorithm to solve the multi-location transshipment problems with transportation capacities. The numerical experiments show that the impact of transshipment capacity on system behaviour heavily depends on the number of participating stocking locations as well as on the network structure.

Hu $et\ al$ (2007) considered a two-location production inventory model with lateral transshipment and uncertain capacity during the period. Their interest is in whether there exist coordinating transshipment price such that, if each location made locally optimal decisions, the two locations' total profits would be equal to the profits earned by a centralized system. Hence, the optimal decisions both for a central coordination and local depot are modelled

Numerical study shows that a firm that would like to coordinate two locations' production and transshipment is unlikely to achieve coordination in many instances by setting linear transshipment price schedule.

Kutanoglu (2008) considered a two-echelon system with N locations and one central warehouse. It was assumed that each location has one-for-one replenishment order policy and exponential lead time. He tried to evaluate total costs and time-based fill rates from two respective: response time service time and inventory sharing.

Numerical experiments show that a complete enumeration scheme can be useful when there are a small number of facilities each with relatively low demand. However, the models they chose is on the evaluation rather than optimization.

Zhao $et\ al$ (2008) is the latest paper in the transshipment policy. Rather than considering the conventional replenishment order policy, they considered the make-to-stock queuing system with transshipment policy. In the two-location, the system faces both production and demand filling decisions at each location, with each decision now involving a transshipment option.

It was assumed that a distribution network with two locations, each consisting of an inventory pool linked to a make-to-stock production facility. A dynamic programming formulation for the problem was presented and the structure of the optimal policy was proved.

Their study showed that: the optimal policy will use transshipment not only for emergency demand filling, however also to provide capacity flexibility through the sharing of production capacity; a dual transshipment policy, in which transshipment is allowed both before and after demand arrival, significantly outperforms more restrictive policies, which allow only one type of transshipment; a common restriction on transshipment used in practice, i.e., that transshipment will be used if and only if there is an emergency, is optimal under certain reasonable assumptions on the problem parameters; as determined by the optimal policy, production, demand filling, and transshipment decisions are state dependent.

2.3.3 Key contributions to transshipment research

In the previous discussions on the relevant research of the transshipment policy to the multi-echelon and multi-location inventory control system, several pioneering works are regarded as the key research contributions in this field. Therefore it is worth to have a concise review of these works before we introduce our own approach. For convenience of reference, we present these momentous papers in a chronological order.

Das (1975) considered a two-location system with periodic review. It was assumed that replenishment may be ordered only at times of periodic review but transshipments is allowed between the locations at predetermined times within the replenishment cycle.

The model for a single product system of two locations with independent demands is formulated. The system is reviewed periodically for purposes of replenishment and redistribution of stock and assumed that the transshipment and ordering are not carried out concurrently. The system cost comprises of three operating costs: holding cost, ordering cost and transshipment cost.

Das defined the optimal expected total cost function for each period. In order to derive the optimal transfer and ordering rules, Das gave very detailed analysis and proof using convex function features and its component functions. The optimal transfer rule for the single period model and optimal ordering rule under the assumption of complete backlogging of excess demand are discussed.

Das's approach provides us with a study of a single period for one single product two-location inventory system with stock transfer policy. With the strict assumptions on the convex component function of expected cost, the optimal transfer and ordering rules are defined in a very general manner. However, Das (1975) does not mention a method of finding optimal transfer and ordering rule for a specific situation. It is very difficult to judge the validity of his approach without simulation.

The paper of Lee (1987) is one of most important papers in the 1980's on the transshipment policy because of its popular reference rate. Lee considered a multi-echelon inventory system with continuous review system. A one-for-one replenishment policy is used and emergency lateral transshipments between some identical bases are allowed. In this paper, Lee called the transshipment of stock goods from one base to another an emergency lateral transshipment when the stock-out takes place at one base. In the model, a two-echelon multi-location inventory system where a group of base stock is supplied by a replenishment depot and bases are sub-categories by different pooling groups where the emergency lateral transshipment is permitted among all bases within the same pooling group.

Three transshipment decision policies are examined to choose the source base in such orders

- Rule 1: the source base is chosen randomly from any base within the pooling group with stock on hand
- Rule 2: the base with maximum stock on hand is chosen as the source base, if no such base exists, go to random rule 1
- Rule 3: first choose the base with the maximum stock on hand, if no such base, search the base with the smallest number of outstanding orders. If no such base exists again, go the random rule 1

Lee (1987) proposed two measures to assess the system performance under the given policy: performance measures and optimal costs.

Performance measures:

Lee (1987) identifies the number of back-orders and the quantity of emergency lateral transshipment as the two key performance measures: direct fill rate and transshipment fill rate.

Optimal total cost:

Based on the definitions and derivations on the performance measures, Lee deduced the objective function of optimal minimum total cost composed of holding cost, backorder cost and emergency lateral transshipment cost subject to some service level constraints.

In the end, the numerical results on the performance measures and optimal total cost are compared and analyzed with those from the simulations.

It must be highlighted here that it is the first time this type of methodology is used, which is believed to be a more reasonable and sound framework than those before. Since this paper, many researchers followed this manner in their research approach. The reason making them consider this approach is because the ultimate objective of our study on the inventory system is to find the optimal cost policy. To get this goal, we need to define the terms for the system performance measures and approximate these terms to have more insights of the given pooling policy. Meanwhile, the approximation model should be verified by the corresponding simulation. In this sense, we think the approach proposed by Lee (1987) delivers a sound methodology guideline for our research.

2.3.4 Related transshipment research

In the preceding section, several key papers are examined. However, some of them only give solutions to some general problems. For our research interests of the holdout transshipment policy to the multi-location inventory system, we need to review the papers which specially handle the specific problems such like unidirectional transshipment, holdout transshipment, simulation and approximations.

Sven Axsäter is one of the key researchers who concentrates his research interest on the lateral transshipment issue. Therefore, it is worth paying more attention to his paper Axsäter (1990). In the work of Axsäter (1990), he extends the model from Lee (1987). Rather than focusing on the approximation of the number of outstanding orders, Axsäter makes more efforts on the modeling of demand.

Under assumptions of one-for-one replenishment policy, several demands for depot and group are approximated for the two-echelon multi-location inventory system. Then approximations of the direct fill rate, transshipment fill rate and backorder fill rate were examined in respect of identical base and non-identical base scenarios.

In the end, numerical algorithms for searching the correct values of backorder fill rate and transshipment fill rate were discussed with a comparison of results between Lee (1987)'s simulation and Axsäter's approximation. As pointed in the concluding part of Axsäter (1990), Axsäter's approximations advance Lee (1987)'s approximations to the non-identical situation. The numerical results with simulations show that, in case of identical bases, approximations have good improvement compared to the method suggested by Lee (1987) when the proportion of emergency transshipment is relatively large. These approximation techniques can also be extended to non-identical bases and performs equally well in such cases.

Tagaras and Cohen (1992) presented a conventional approach to the lateral transshipment inventory system. Basically, a periodic single-echelon two-location system is under review. Pooling policy is considered either by the inventory level or inventory position at each location, that is to say, the location which is being asked to provide a transshipment wishes to maintain a safety stock before agreeing to deliver a transshipment. They define four pooling policies. Among them, policy 1 and 2 are dependent on the inventory level, meanwhile, policy 3 and 4 are dependent on the inventory position. The definition of the threshold safety stock value is similar to the meaning of the holdout concept at the location where the transshipment is requested. When two thresholds are equal to zero for two locations, policy 1 and policy 2 become the complete pooling policy.

The expected cost per period is examined and inequalities with respect to the cost difference are discussed. When the inequality condition is met, the optimal cost gives rise to one directional transshipment. A series of optimal cost simulations based on the four pooling policies demonstrates pooling policies 1 and 2 are equally determining on the optimal cost, however pooling policy 3 becomes dominating pooling policy in most cases to achieve optimal cost.

To improve optimal cost searching in the simulation, a heuristic algorithm of complete pooling, which is based on the insensitivity of the optimal safety factor to the replenishment lead time in non-pooling situations, provides near-optimal solutions. Additionally, the approximate estimates on the fill rate and average transshipment are reviewed in terms of complete pooling policy.

The paper of Tagaras and Cohen (1992) might be the first paper introducing the so called **holdout** concept for the multi-location inventory system whenever a stockout occurs at one location. It is very interesting to see its effects over the system under different pooling policies. However, further particular research on the holdout transshipment policy needs to be carried out.

The method which Archibald applies in the work of Archibald *et al.* 1997 is different from most research in this subject. Rather than choosing conventional approximations of the performance measures and optimal total cost, the system is modelled as a continuous time Markov decision process.

Firstly, a Markov decision process (**MDP**) model of a single product two-location inventory system is examined. Inventory levels at the two locations are defined as the finite state space. A transshipment action is made for situations when inventory level at one location is zero and the other location has stock on hand. The minimum expected total cost of satisfying the demand within a period, given the stock level at location 1 and location 2 at the beginning of the period, is formulated as the sum of the expected cost of the first unmet demand occurring at location 1 and location 2, and the expected cost of no unmet demand occurring before the next review epoch.

The practical method of getting the optimal total cost is to discretize time and calculate, working backward in time. Then various modifications of the model are discussed by the evaluation of the optimal transfer policy. In particular, the likelihood of transfer from location 1 to location 2 is dependent on various relations between the increase in the demand at location 1 and location 2. Furthermore, an extended model for a two depot multiple item inventory system is discussed and an example of numerical computation on the two depot two item inventory system is analyzed.

Archibald *et al.* (1997) shows an innovative approach that reviews the lateral transshipment policy for inventory control. By taking advantage of the Markov decision process and dynamic programming, we will advance Archibald *et al.* (1997)'s approach to the two-location inventory system with non-zero replenishment lead time and holdout transshipment policy.

The methodology used in Axs*ä*ter 2003a concentrates on simulation techniques. The lateral transshipment policy is simple in the paper, that is the unidirectional lateral transshipment among the two-echelon multi-location inventory system. In other words, the transshipment is only considered from the posterior location to the earlier location on the same supply chain multi-location system. To simplify the solution, only two and three location inventory system are examined with order-up-to S or (R, S) replenishment order policy under the unidirectional lateral transshipment policy. In general, he gave the approximations on the key performance measures: direct fill rate, fill rate (including lateral transshipment) and indirect fill rate (only lateral transshipment). In addition to these approximations, the total cost (which consists of holding order cost, back order cost and transshipment cost) is formulated.

Under several identical and non-identical circumstances, computation results on the fill rate, stock on hand rate and back orders from the approximations and simulations are provided in respective of order-up-to S and $(R,\ S)$ policy for two-location and three-location inventory system. The numerical results of optimal total cost and simulation for two-location inventory system with and without transshipment policy are compared. It is reported in Axs*ä*ter (2003a) that some discrepancies are found in the performance measures and total cost approximations.

Pros:

The advantage of choosing a simulation approach in Axs*ä*ter (2003a) is because it is very straightforward and proves to be a quick method to learn system performances by the given policy. Another highlight point is the unidirectional transshipment applied. Due to the complexity of multi-echelon and multi-location inventory control systems, a choice of simple transshipment policy is helpful to have a good understanding of the key parameter which contributes to the optimal transshipment policy.

Cons:

The simulation approach is clear-cut to understand impact on a system under the given lateral transshipment policy. However, there are many ways to gain an improved approximation on the performance measures and optimal transshipment policy. In addition, no holdout concept is discussed in Axs*ä*ter (2003a).

The paper of Xu *et al.* (2003) is a recent study especially concentrating on the **holdout** transshipment policy under continuous review for the two-location inventory system with (R, Q) replenishment policy and is the work which might is close to our research problem domain when we prepare this literature review.

Generally, Xu *et al.* (2003) derives the assumption of continuous review models from the work by Lee and Nahmias (1993), and transshipment policy is similar to these in Krishnan and Rao. (1965) and Tagaras and Cohen (1992). More specifically, the transshipment quantity that location i receives from location j is dependent on the inventory level of location j and the related control policy. Three pooling policies are discussed as below:

A) Complete pooling policy: location j makes the transshipment without any condition when a transshipment request is received from location i

B) Reorder-point pooling policy: location j keeps enough inventory on hand to cover the reorder-point inventory position before agreeing to deliver a transshipment to location i

C) Holdout pooling policy: location j keeps a threshold value of inventory on hand, but unrelated to the reorder-point inventory position, before agreeing to deliver a transshipment to location i

Under the so called (R_i, Q_i, H_i) emergency transshipment policy, the following performance measures of the inventory system are examined:

1. no stock-out probability at location i before pooling
2. no stock-out probability at location i after pooling
3. fill rate at location i before pooling
4. fill rate at location i after pooling

These performance measure estimations are validated by checking corresponding results from simulations both for systems with non-identical locations and identical locations. Big discrepancies between analytic and simulation models for the non-identical situation are reported. It is suggested that a low Q_i relative to average demand during lead time might explain such big discrepancy. However a small value of R_i leads to large inbound transshipment and a good level of agreement between estimation and simulation. On the contrary, the disagreement between the analytic models with simulation for identical location situations becomes less than that for non-identical location situation.

Pros:

This is the first paper in which the so called (R, Q, H) replenishment/transshipment policy is formally used. The performance measures of the two-location continuous review system are carefully examined and analyzed in respect of three different pooling policies, though results from these estimations do not agree with these of simulations.

Cons:

It is assumed that each location can determine its actual demand during a lead time before the end of the lead time. This is similar to the assumption that transshipments occur after demand is realised but before it must be satisfied. Meanwhile, further study on the optimal total cost under such holdout pooling policy is needed for a considerate review.

Seifert (2006a) and (2006b) are recent research works which show the impact of unidirectional transshipment policy.

Instead of allowing transshipments between retailers or transshipments from the virtual store to a retail store, they consider that an unidirectional transshipment is allowed from the existing retail network (indirect sale channel) to the virtual on-line store (direct sale channel). In the dedicated supply chain, the retail stores serve all in-store customers and the virtual store serves all online customers because transshipments are not allowed. In the integrated supply chain, the excess inventory at the retail stores can be used to fill those online demands that the virtual store cannot meet from stock. Hence, the problem is modelled as a discounted stationary multiple-period newsvendor problem with lost sales at the retail stores and lost sales at the virtual store.

The dedicated supply chain and integrated supply chain models are developed in terms of optimal base-stock and optimal cost. The examples of cost savings are used to show how the benefits of integrating the virtual store depend on the following characteristics of the supply chain: relative channel volume, product characteristics and replenishment lead time. These results show that integration of the direct and indirect channels improves supply chain performance by substantial channel-stock reductions and reductions in lost sales if the number of retail stores is large, transshipment penalty cost is relatively low and lost-sales cost, financial inventory holding cost, demand variability and lead time are high. Their work also demonstrates that relative channel volume affects the magnitude of the system performance improvements.

This is the one of few works that directly considers the unidirectional transshipment for two-location inventory systems. Their approaches provide a good framework to review the impact of the different characteristics of the system. However, it is assumed that transshipment occurs after the demand is realised and these models need to be validated by the counterpart simulations. In addition, they only handle the complete pooling policy for transshipment, not partial pooling (holdout transshipment) and derive the total cost rather than each cost component.

2.4 Conclusions

From our relevant research study, we learn that most models allowing the transshipment policy can been classified as three categories: periodic review systems that allow transshipments at a single point during a period before the demand for the period is fully known; periodic review systems that allow transshipments after the demand for the period is known, however before it has to be satisfied; continuous review systems that allow transshipments in response to stockouts and use a one-for-one replenishment policy.

However, we are interested to examine the models which can be classified as continuous review systems that allow transshipments (not only one transshipment) in response to stockouts and use a (R,Q) replenishment policy. Rather than considering the models which only allow the no pooling or the complete pooling policy, we consider the models with more general pooling policies which include no pooling, partial pooling and complete pooling policy.

Most papers choose either analytic approximation model or simulation model. Not many works are conducted by validating the analytic approximation model with the counterpart simulation model, even for those which apply the two approaches together, such as Xu *et al.* (2003). Most report some discrepancy between the analytic approximation and simulation models. Therefore, we aim to develop an accurate analytic approximation model that is consistent with the counterpart simulation model.

Rather than estimations of a range of performance measures, our research aims to find the optimal total cost by approximating all cost components at each location. In order to develop a solid understanding of transshipment policy for inventory systems, we decide to consider a basic two-location system with unidirectional holdout transshipment policy.

Furthermore, we also develop the **SMDP** models to take advantage of dynamic programming. Using the **SMDP** approach, we develop the models with general unidirectional transshipment policy for two-location systems. We compare the optimal average total cost rate under general unidirectional transshipment policy with that under unidirectional holdout transshipment policy to evaluate the performance of the holdout policies. This approach is also used to approximate the optimal holdout threshold at location 2 and to develop a new dynamic holdout transshipment policy for two-location inventory systems. By applying approximation, simulation and dynamic programming techniques, we aim to get a good understanding of the unidirectional transshipment policy for two-location inventory systems.

Glossary

Unichain: A Markov chain is said to be ***unichain*** if it has no two disjoint closed sets

Aperiodic: If the greatest common divisor of all n for which $P^n\{X_n = i|X_{n-1} = i\} > 0$ is equal to 1, then state i is said to be ***aperiodic***

Chapter 3
Problems descriptions & Simulations

3.1 Introduction

In the preceding chapter, we give the theoretical foundations of Markov decision processes and a summary of the research on multi-echelon and multi-location inventory systems. For the remainder of the thesis, we will establish a series of approximation models to help us understand the impact of the general unidirectional transshipment and holdout unidirectional transshipment policies on two-location inventory systems. In order to guide us in the validation and evaluation of such approximation models, we use implementations of simulation models of a variety of inventory systems.

3.2 Problem domain descriptions

The problem domain to which we pay special attention is the impact of the unidirectional transshipment policy to the multi-location inventory system. As the problem is more complicated than expected due to the transshipment interactions among the locations, we decide to concentrate our study on the two-location inventory system. We define the following terms:

Definition 3.1. Inventory Level: *number of items in stock at a location*

Definition 3.2. Backorder level: *number of outstanding backorders at a location*

Definition 3.3. Inventory Position: *inventory level minus the backorder level plus the number of items ordered, but not yet delivered*

Note that the two terms inventory level and stock level are used interchangeably.

We concentrate our study on the unidirectional transshipment policy, that is, a lateral transshipment is only allowed to deliver from location 2 to satisfy a stockout at location 1. We consider the general transshipment and holdout transshipment policies respectively. With the general transshipment policy, a transshipment decision is made when the inventory level at location 2 is greater than zero and location 1 has a stockout. With the holdout transshipment policy, a transshipment occurs if and only if the inventory level at location 2 is greater than a specific holdout threshold when there is a stockout at location 1. The holdout transshipment policy can be regarded as a partial pooling policy compared to the two extremes of the no pooling and complete pooling policies.

We are keen to find the optimal holdout transshipment policy by searching on the average total cost under the unidirectional transshipment policy. Doing so, we identify the sensitive parameters which might contribute to the optimal holdout transshipment policy for the two-location inventory system.

In addition, it is worth introducing several terms as system performance measures to evaluate out approximation models. At each location, we define the fill rate, which consists of the direct fill rate and indirect fill rate (also known as the transshipment fill rate), and the backorder fill rate. Hence, we define these fill rates for the two-location system as such

1. **Direct fill rate:** the proportion of demand arising at location k, (k=1,2) that is met from local stock.

2. **Indirect fill rate:** the proportion of demand arising at location 1 that is met by transshipment from location 2.

3. **Backorder fill rate:** the proportion of demand arising at the location k, (k=1,2) that is met by backorders and this is equal to 1 minus the sum of direct and indirect fill rates.

3.3 Simulation implementations

In the **OR** field, it is intuitive to use simulation techniques due to their straightforward and accurate reflection of the original problem domain. In practice, a good simulation, which reflects more system features under the specific policy, can provide a good guide to the development of effective mathematical modelling. From the highly developed simulator for NASA's space projects to the simple financial mortgage calculator, we encounter simulations in diversified forms in our daily life.

However, as mentioned in Winston (2004), like most research techniques, any simulation approach has twofold features. On the one hand, simulation gives modellers not only a more flexible but also more straightforward method to implement than approximation modeling approach. Compared to the approximation model approach, simulations do not imply as many restrictive assumptions which must be prerequisite for most approximation models, on the contrary, simulations can be extended directly to other complicated systems under more flexible constraint environments. Hence, the simulation approach gives us a quick look at the consequences of a given policy. However, simulations do not provide the optimal solutions in most cases. In the search of the optimal operation cost, we have to rely on the approximation models. Overall, we believe that simulation helps us have more accurate details of the system's behaviour which can be used as a reference for our approximation models.

In the following sections, we will introduce our simulation implementations. First we give the assumptions and notations of our simulations, then flow charts of programming control are presented. The core programming codes are provided for further references in the appendix of the thesis.

3.3.1 Preliminary results

Simulations for multi-location inventory control models are slightly more complicated than simulation of the single-depot system. Unlike a Poisson process in a single-depot system, more Poisson processes are involved in multi-location inventory control models. Therefore, the following features of merged Poisson processes are used.

Let X_1, X_2, $...X_n$ be a sequence of non-negative, independent random variables having a common probability distribution function. Letting S_0=0, $S_n=\sum\limits_{i=1}^{n} X_i$, $n = 1, 2, ...$ Define for each $t \geqslant 0$, $N(t)$ equals to the largest integer $n \geqslant 0$ for which $S_n \leqslant t$, the random variable $N(t)$ represents the number of events up to time t.

Suppose that $\{N_1(t),\, t \geqslant 0\}$ and $\{N_2(t),\, t \geqslant 0\}$ are two independent Poisson processes with respective rates λ_1 and λ_2. Then it can be proved that the merged process $\{N(t) = N_1(t) + N_2(t)\}$ is another Poisson process with rate $\lambda_1 + \lambda_2$. The probability that an arrival in the merged process is from process 1 rather than process 2 is equal to $\frac{\lambda_1}{\lambda_1 + \lambda_2}$.

With these results, we can use the merged Poisson process in our simulations to model the occurence of the demand in the system and determine the origin of the demand from the merged process.

3.3.2 Assumptions

Before we discuss simulations for single-depot and two-location inventory control systems, we define assumptions and notations as follows.

In general, we consider a two-location system, in which the demand occurs according to a Poisson process at rate of λ_k at location k, $k = 1, 2$. Meanwhile, an $(R_k,\, Q_k)$ replenishment order policy is used to manage inventory at location k. That is to say, an order of Q_k units of one kind of product is placed when the inventory position falls down to R_k and this outstanding order will arrive after a fixed lead time L_k at location k. The system allows more than one outstanding order, however, we only allow unidirectional transshipment between the two locations in the system. More precisely, the transshipment is only permitted from location 2 to location 1, but not vice verse. Transshipment is assumed to be instantaneous and involves a transshipment cost.

The simulation models are used to estimate performance measures including the total cost rate, order cost rate, backorder cost rate, stockout cost rate, transshipment cost rate, direct fill rate and indirect fill rate based on observations of the system for a fixed time period after a reasonable warm up time period. It is possible to estimate these performance measures based on the inventory level and position at each location and the number of transshipments during the simulation run.

Let $\mathbf{SL}_{\boldsymbol{k}}$ and $\mathbf{SP}_{\boldsymbol{k}}$ denote the current inventory level and the current inventory position at location k. Let **NT** denote the number of transshipment deliveries.

We note that the simulation models of the general transshipment policy and the holdout transshipment policy are designed in a way that could easily be extended to systems with more than two locations.

3.3.3 Simulation for single-depot system

The simulation of a single-depot system is straightforward by using a discrete event simulation model. Events are defined to be instances of demand and deliveries of replenishment orders. When the next event is a demand, the inventory level and inventory position are updated accordingly and the inventory position is checked to determine whether a replenishment order needs to be placed. When the next event is a delivery, the inventory level is updated accordingly. After a long-run, the system attains at equilibrium mode.

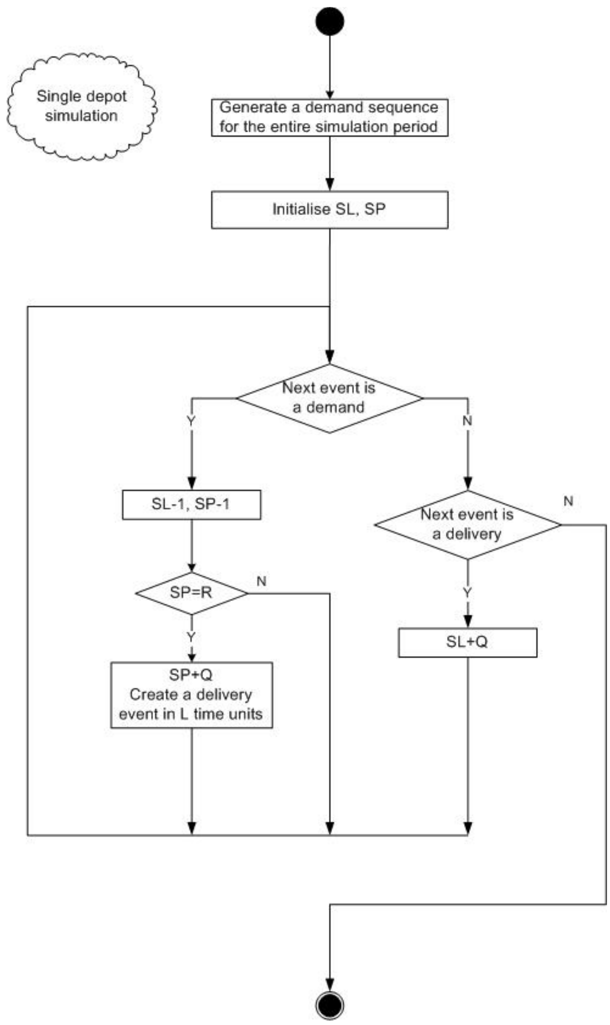

Figure 3.1. Simulation of single-depot model

The flow chart of the single-depot system in Figure 3.1 is intuitive. Initially, we generate a demand sequence according to a Poisson demand process from zero to the end time of the simulation. If next event is a demand, we update the inventory level and inventory position due to an occurence of a demand. More specifically, both the inventory level and inventory position are decreased by 1. When the inventory position reaches R then, due to the (R, Q) replenishment order policy, a replenishment order for Q items is placed and this order will arrive after lead time L. Accordingly, the inventory position is increased by Q items and, provided the time until the end of the simulation is at least L, a new delivery event is inserted into the demand sequence.

If the next event is a delivery of an outstanding order, the inventory level is updated by adding Q items. When there are no more events, the simulation finishes and all relevant system performances including the direct and backorder fill rates are worked out.

3.3.4 Simulation for two-location system

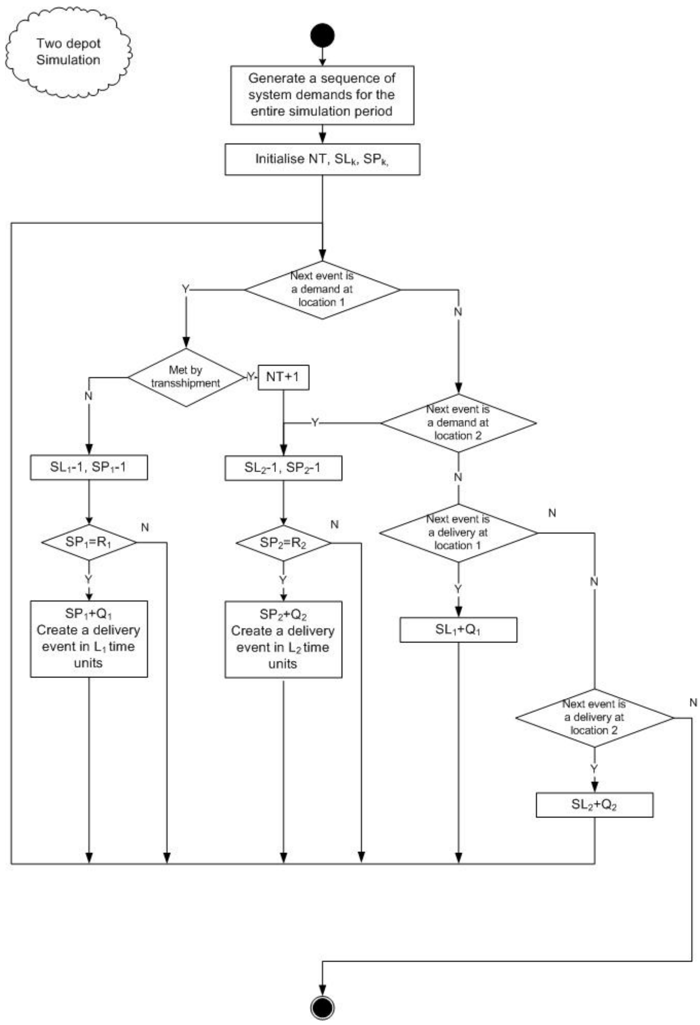

Figure 3.2. Simulation of two-location system with the original transshipment policy

The simulation model of the two-location system uses a similar approach as the simulation model of the single-depot system. Again it is a discrete event simulation in which the basic events are instances of demand and deliveries of replenishment orders. However, there are now five distinct events: demand at location 1 that is met from local stock, demand at location 1 that is met by transshipment, demand at location 2, delivery at location 1 and delivery at location 2.

Figure 3.2 shows a flow chart of the simulation model. Initially, we generate a sequence of system demands according to the merged Poisson process with rate $\lambda_1+\lambda_2$ from zero to the end time of the simulation. If the next event is a demand at location 1, we first consider whether it is met by transshipment or not. This depends on the inventory levels of the two locations and the transshipment policy being considered. For example, if the inventory level at location 1 is greater than zero, transshipment will never be used. However, if the inventory level at location 1 is less than or equal to zero, the inventory level at location 2 is greater than zero and the policy is complete pooling, transshipment would be used. If the demand is to be met from local stock, we update the inventory level and inventory position and check if a replenishment order should be placed for location 1.

If the demand is to be met by transshipment, we increase the counter of transshipment by 1, update the inventory level and inventory position at location 2, and check if a replenishment order should be placed at location 2. If the next event is a demand at location 2, we update the inventory level and position at location 2 and check if a replenishment order should be placed at location 2.

If the next event is a delivery of an outstanding order at location k, the inventory level is updated by adding Q_k items. When there are no more events within the simulation period, the simulation finishes and all relevant system performances including the direct fill rate, backorder fill rate and indirect fill rate are worked out.

In this thesis we use the simulation model from Figure 3.2 to simulate complete pooling, no pooling and holdout transshipment policies by varying the condition in the "Met by transshipment" box.

Complete pooling: for complete pooling, transshipment is used if and only if the inventory level at location 1 is less than one and the inventory level at location 2 is greater than zero.

No pooling: for no pooling, demand at location 1 is never met by transshipment.

Partial pooling: a policy in which location 2 may refuse a transshipment request from location 1 when location 2 has on-hand inventory

Holdout transshipment: for a holdout transshipment policy with threshold I_2, transshipment is used if and only if the inventory level at location 1 is less than 1 and the inventory level at location 2 is greater than I_2.

To help the verification of our approximation modelling, some simulation results are provided in the relevant chapters. Meanwhile, the core simulation implementation codes are provided in an appendix of the thesis.

Glossary

Inventory Level: number of items in stock at a location
Backorder level: number of outstanding backorders at a location
Inventory Position: inventory level minus the backorder level plus the number of items ordered, but not yet delivered
Direct fill rate: the proportion of demand arising at location k, (k=1,2) that is met from local stock
Indirect fill rate: the proportion of demand arising at location 1 that is met by transshipment from location 2
Backorder fill rate: the proportion of demand arising at the location k, (k=1,2) that is met by backorders and this is equal to 1 minus the sum of direct and indirect fill rates
SL$_k$: denote the current inventory level at location k
SP$_k$: denote the current inventory position at location k
NT: denote the number of transshipment deliveries
Complete pooling: transshipment is used if and only if the inventory level at location 1 is less than one and the inventory level at location 2 is greater than zero
No pooling: demand at location 1 is never met by transshipment
Partial pooling: a policy in which location 2 may refuse a transshipment request from location 1 when location 2 has on-hand inventory
Holdout transshipment partial pooling: for a holdout transshipment policy with threshold I_2, transshipment is used if and only if the inventory level at location 1 is less than 1 and the inventory level at location 2 is greater than I_2

Chapter 4

Decomposition approach based on independent locations with constant demand rates

4.1 Introduction

In this chapter, we seek a simple approximation model that accurately reflects the unidirectional holdout transshipment policy within a two-location inventory control system. We endeavour to use this model to gain an insight on optimal transshipment policies with little computation. We use a decomposition approach in which the two-location system is decomposed into two independent single-depot systems with constant demand rates. To achieve the decomposition, we employ an approximation to the system in which location 2 randomly decides whether or not to satisfy transshipment requests from location 1. The demand rates in the single-depot systems are based on the original demand rates modified according to the average rate at which transshipment occurs during a period. Under the approximation, it is hoped that the decomposition approach will capture important features of the holdout transshipment policy.

The motivation for the decomposition approach is that the single-depot system is well-understood and relatively straightforward to analyse. Therefore, before discussing the decomposition approach in detail, we first present in Section 4.2 an analysis of the single-depot system and derive exact expressions for a range of long-run average performance measures including average total cost, direct fill rate and backorder fill rate. The **JAVA** program written to approximate these performance measures is verified by comparison with a simulation model of the single-depot system. We also present a search algorithm to find the optimal average total cost and a numerical experiment to illustrate the algorithm.

Section 4.3 provides details of the decomposition approach for the two-location system. Following a brief introduction, the assumptions of the two-location system are given in Section 4.3.2 Section 4.3.3 explains how the two-location system is decomposed into two independent single-depot systems and Section 4.3.4 derives approximate expressions for a range of performance measures including the long-run average total cost and its individual average cost components, direct fill rate, backorder fill rate and transshipment fill rate. One search algorithm on the optimal average total cost under the so called financial budget constraint is demonstrated with a numerical experiment.

In the end, we compare numerical results which are based on the decomposition approach and financial budget search algorithm with the original holdout transshipment simulation model. For five snapshots of system characteristics, we investigate the possible reasons for the differences in the results of the decomposition approximation and simulation models.

4.2 The single depot model

4.2.1 Assumptions

Customer demand is modelled as a Poisson process with constant rate λ. The inventory at the depot is managed using an $(R,\, Q)$ replenishment policy and each order placed is delivered after a fixed lead time L. Let c denote the fixed order cost and h denote the holding cost per item per time unit. Any demand that cannot be met from inventory will be backordered. There is a one-off cost of $\hat{b}$ when a backorder is placed and a further cost of b per time unit until this backorder is satisfied. For convenience, we refer to these costs as the stockout cost and backorder cost respectively.

4.2.2 Steady-state distribution of inventory level

It is well-known, for example from the book by Hadley and Whitin (1963), that under the above assumptions, the steady-state distribution of the inventory position is uniform between $R+1$ and $R+Q$. Hence, when the system is in steady-state, the probability that the inventory position is i is equal to $\frac{1}{Q}$ for $R+1 \leqslant i \leqslant R+Q$. If the inventory position at time $t-L$ is i, then the inventory level at time t will be x if and only if $i-x$ demands occur in the interval $(t-L, t)$. This is because during the interval $(t-L, t)$, all outstanding orders at time $t-L$ will arrive and all orders placed will be outstanding at time t. Since demand must be non-negative, the inventory level at time t must be less than or equal to the inventory position at time $t-L$. Hence, if the inventory level at time t is x, the inventory position at time $t-L$ must have been between $\max(R+1, x)$ and $R+Q$. Let D to be a random variable representing the demand during the lead time. Conditioning on the inventory position L time units earlier, the probability that the inventory level is x any time when the system is in steady-state is given by

$$p(x) = \frac{1}{Q} \sum_{i=\max(R+1,x)}^{R+Q} Pr(D = i - x) \text{ for } x \leqslant R+Q \tag{4.1}$$

Note that in equation (4.1), negative inventory corresponds to outstanding backorders.

4.2.3 Performance measures

We now are in a position to derive exact expressions for the performance measures of interest. It follows immediately from the definition of $p(x)$ in equation (4.1) that: the long-run average inventory level $= \sum_{x=1}^{R+Q} x\,p(x)$ and the long-run average backorder level $= \sum_{x=1}^{\infty} x\,p(-x)$.

The direct fill rate α is calculated as the steady-state probability that the depot has at least one item in inventory which is equal to $\sum_{x=1}^{R+Q} p(x)$. Since all demands that cannot be satisfied from inventory are backordered, therefore, the backorder fill rate β is equal to $1-\alpha$.

The total demand between successive orders is Q. Hence, the average time between orders is $\frac{Q}{\lambda}$ and the average fixed order cost per time unit is $c\frac{\lambda}{Q}$. On average, an instance of demand leads to a backorder being placed with probability $1-\alpha$. Hence, stockouts occur at a rate of $\lambda(1-\alpha)$ and the average stockout cost per time unit is $\hat{b}\lambda(1-\alpha)$.

The other cost terms follow immediately from the expressions for the average inventory level and average backorder level above. Hence, the long-run average total cost rate for the single-depot system is given by

$$C(R,Q) = c\frac{\lambda}{Q} + \hat{b}\lambda(1-\alpha) + b\sum_{x=1}^{\infty} xp(-x) + h\sum_{x=1}^{R+Q} xp(x) \tag{4.2}$$

4.2.4 Model verification

We have written a **JAVA** program to calculate the performance measures from Section 4.2.3. Because the number of backorders is unbounded, the expressions for some performance measures involve infinite summations that need to be approximated in the **JAVA** program. The purpose of this section is, therefore, to establish that these approximations are sufficiently accurate and the **JAVA** program is correct. We do this by comparing the results of the program with the results of a simulation (also written in **JAVA**) of a single-depot system under assumptions of Section 4.2.1. A flow chart representation of the structure of the simulation model is given in Figure 3.1. Another purpose of this comparison is to confirm that the system performance measures can be estimated satisfactorily from the steady-state distribution of the inventory level.

In the verification we increase Q from 1 to 20 while holding all other parameters constant. We also choose $R = \lambda L$. The relatively low values for Q means that there will often be many outstanding orders at the one time which will test the implementation of the simulation model. The relatively low values for Q and the choice of R ensure that the stockout and backorder costs are never negligible which is important for testing the calculations. We define one time unit to be equal to the lead time (i.e. $L = 1$). The other parameters are given values as follows: $\lambda = 20, c = 10, \hat{b} = 10, b = 8$ and $h = 1$.

For the simulation implementation, in order to make a good balance between computational time and accuracy, we performed 500 independent simulation runs of 50,000 time units with a warm-up period of 500 time units, for each value of Q. For simplicity, we arbitrarily choose the initial inventory level and inventory position between $R+1$ and $R+Q$. Hence, there are no outstanding orders initially. Table 4.1 compares estimates of the total average cost, average inventory level, average backorder level and direct fill rate from the simulation model and the analytic expressions based on steady-state probability of the inventory level. Note that indirect fill rate β can be calculated by $1-\alpha$ directly. Columns **sim**, **se** and **an** for each cost component represent simulation result, standard error of the simulation result and analytic result respectively.

	$C(R,Q)$			Avg. Inventory level			Avg. backorder level			α		
Q	sim	se	an	sim	se	an	sim	se	an	sim	se	an
1	301.3	0.18	301.2	2.34	0.00	2.34	1.34	0.01	1.34	0.56	0.00	0.56
2	191.7	0.15	191.6	2.66	0.00	2.66	1.16	0.01	1.16	0.60	0.00	0.60
3	149.6	0.14	149.5	3.01	0.00	3.01	1.01	0.01	1.01	0.64	0.00	0.64
4	124.9	0.12	124.8	3.38	0.00	3.38	0.88	0.01	0.88	0.68	0.00	0.68
5	107.8	0.11	107.7	3.77	0.00	3.77	0.77	0.00	0.77	0.71	0.00	0.71
6	94.9	0.11	94.9	4.18	0.00	4.18	0.68	0.00	0.68	0.74	0.00	0.74
7	84.8	0.10	84.7	4.60	0.00	4.60	0.60	0.00	0.60	0.77	0.00	0.77
8	76.6	0.09	76.5	5.03	0.00	5.04	0.54	0.00	0.54	0.79	0.00	0.79
9	69.9	0.08	69.8	5.48	0.00	5.48	0.48	0.00	0.48	0.81	0.00	0.81
10	64.4	0.08	64.3	5.94	0.00	5.94	0.44	0.00	0.44	0.83	0.00	0.83
11	59.8	0.07	59.7	6.40	0.00	6.40	0.40	0.00	0.40	0.84	0.00	0.85
12	55.9	0.07	55.9	6.84	0.00	6.87	0.37	0.00	0.37	0.85	0.00	0.86
13	52.7	0.06	52.7	7.34	0.01	7.34	0.34	0.00	0.34	0.86	0.00	0.86
14	50.0	0.06	50.0	7.82	0.01	7.81	0.32	0.00	0.32	0.87	0.00	0.87
15	47.7	0.06	47.7	8.30	0.01	8.29	0.29	0.00	0.29	0.88	0.00	0.88
16	45.7	0.05	45.7	8.77	0.01	8.78	0.28	0.00	0.28	0.89	0.00	0.89
17	44.1	0.05	44.0	9.26	0.01	9.26	0.26	0.00	0.26	0.90	0.00	0.90
18	42.6	0.04	42.6	9.75	0.01	9.74	0.25	0.00	0.25	0.90	0.00	0.90
19	41.4	0.04	41.3	10.23	0.01	10.23	0.23	0.00	0.23	0.91	0.00	0.91
20	40.3	0.04	40.3	10.72	0.01	10.72	0.22	0.00	0.23	0.91	0.00	0.91

Table 4.1. Performance measures for single depot model

	C_o			C_h			$C_{\hat{b}}$			C_b		
Q	sim	se	an	sim	se	an	sim	se	an	sim	se	an
1	200.0	0.05	200.0	2.3	0.00	2.3	88.2	0.12	88.2	10.7	0.02	10.7
2	100.0	0.03	100.0	2.7	0.00	2.7	79.8	0.11	79.7	9.3	0.02	9.3
3	66.7	0.02	66.7	3.0	0.00	3.0	71.8	0.11	71.8	8.0	0.02	8.0
4	50.0	0.01	50.0	3.4	0.00	3.4	64.5	0.10	64.5	7.0	0.02	7.0
5	40.0	0.01	40.0	3.8	0.00	3.8	57.9	0.10	57.8	6.1	0.01	6.1
6	33.3	0.01	33.3	4.2	0.00	4.2	52.0	0.09	51.9	5.4	0.01	5.4
7	28.6	0.01	28.6	4.6	0.00	4.6	46.8	0.08	46.7	4.8	0.01	4.8
8	25.0	0.01	25.0	5.0	0.00	5.0	42.3	0.08	42.2	4.3	0.01	4.3
9	22.2	0.01	22.2	5.5	0.00	5.5	38.3	0.07	38.3	3.9	0.01	3.9
10	20.0	0.01	20.0	5.9	0.00	5.9	35.0	0.07	34.9	3.5	0.01	3.5
11	18.2	0.00	18.2	6.4	0.00	6.4	32.0	0.06	32.0	3.2	0.01	3.2
12	16.7	0.00	16.7	6.9	0.00	6.9	29.5	0.06	29.4	2.9	0.01	2.9
13	15.4	0.00	15.4	7.3	0.01	7.3	27.3	0.06	27.2	2.7	0.01	2.7
14	14.3	0.00	14.3	7.8	0.01	7.8	25.4	0.05	25.3	2.5	0.01	2.5
15	13.3	0.00	13.3	8.3	0.01	8.30	23.7	0.05	23.7	2.4	0.01	2.4
16	12.5	0.00	12.5	8.8	0.01	8.9	22.2	0.05	22.2	2.2	0.01	2.2
17	11.8	0.00	11.8	9.3	0.01	9.3	21.0	0.05	20.9	2.0	0.01	2.1
18	11.1	0.00	11.1	9.8	0.01	9.8	19.7	0.04	19.7	2.0	0.01	2.0
19	10.5	0.00	10.5	10.2	0.01	10.2	18.7	0.04	18.7	1.9	0.01	1.9
20	10.0	0.00	10.0	10.7	0.01	10.7	17.8	0.04	17.8	1.8	0.01	1.9

Table 4.2. Component of total average cost per time unit

The verification results in Table 4.1 demonstrate that the minimum and maximum differences in the analytic and simulation based estimates of average total cost, average inventory level, backorder level and fill rate are from 0.12% to 0.03%, from 0.00% to 0.20%, from 0.00%

to 4.84% and from 0.002% to 0.054% respectively. Table 4.2 provides further breakdown details of the cost terms including order cost (C_o), holding cost (C_h), stockout cost ($C_{\hat{b}}$) and backorder cost (C_b). Again the results reveal a high degree of consistency between the analytic and simulation based estimates. The minimum and maximum differences between the simulation and analytic approximation results for average order cost, average holding cost, average stockout cost and average backorder cost range from 0.002% to 0.003%, 0.00% to 0.020%, 0.026% to 0.245% and 0.000% to 4.839% respectively. Thus, we can trust our method of calculating the performance measures for the single-depot system and reuse it in future research.

In addition, the verification results demonstrate interesting trends in system performance along with increases on the value of Q. While Q increases, the direct fill rate improves and the stockout and backorder costs fall. When the system is in steady-state, stockouts will occur if the demand in the next L time units exceeds the current inventory position. As Q increases, the average inventory position increases and the risk of stockouts falls.

4.2.5 Optimal average cost algorithm

In equation (4.2), there are two decision variables R and Q, therefore, we aim to find the optimal total cost by searching on R and Q. From the properties of the Poisson demand process, we know that the lead time demand has mean λL and standard deviation $\sqrt{\lambda L}$. Using the normal distribution to approximate the Poisson distribution, we conclude that there is approximately a 2.5% chance of the lead time demand exceeding $\lambda L + 2\sqrt{\lambda L}$. We argue that the reorder position should be at least as great as the mean lead time demand and that 2.5% is an acceptable level of risk. Hence, we assume that the reorder position R satisfies the inequality equation (4.3).

$$\lambda L \leqslant R \leqslant \lambda L + 2\sqrt{\lambda L} \tag{4.3}$$

We assume the supplier imposes a limit $\bar{Q}$ on the size of an order, so that the order quantity Q satisfies $1 \leqslant Q \leqslant \bar{Q}$. With these constraints, it is possible to perform an exhaustive search on R and Q to find the optimal average total cost.

4.2.6 Numerical experiments

We provide one example using the exhaustive search method, in which the optimal average total cost is found when $R = 26$ and $Q = 22$. The problem parameters are as follows: $\lambda = 20$,

$c = 10, \hat{b} = 10, b = 8, h = 1, L = 1$ and $\bar{Q} = 50$.

Table 4.3 shows a series of optimal Q values for the values of R given by the constraint on R in inequality (4.3). The values marked by stars show the (R, Q) replenishment policy which minimizes the average total cost. The cost saving compared to the average total cost when $R = 20, Q = 35$ is 21.85%.

R	20	21	22	23	24	25	26	27	28	29
Optimal Q	35	31	29	27	25	23	22*	21	21	21
Av. Total Cost	35.00	32.96	31.30	30.06	29.23	28.80	28.72*	28.96	29.42	30.07

Table 4.3. Optimal search on R and Q

From the result in Table 4.4 and Figure 4.1, we observe that function $C(R, Q)$ is a convex function on Q in this case. We observed this property in all examples that we considered.

Q	Cost	Q	Cost	Q	Cost	Q	Cost
1	301.27	14	49.99	27	36.05	40	35.40
2	191.71	15	47.66	28	35.77	41	35.50
3	149.55	16	45.72	29	35.56	42	35.66
4	124.92	17	44.05	30	35.35	43	35.85
5	107.82	18	42.56	31	35.23	44	36.03
6	94.91	19	41.36	32	35.13	45	36.23
7	84.77	20	40.28	33	35.03	46	36.45
8	76.61	21	39.37	34	35.03	47	36.67
9	69.89	22	38.55	35*	35.01*	48	36.92
10	64.38	23	37.90	36	35.05	49	37.14
11	59.76	24	37.32	37	35.10	50	37.41
12	55.94	25	36.82	38	35.19		
13	52.72	26	36.41	39	35.28		

Table 4.4. Exhaustive search on Q when $R = 20$

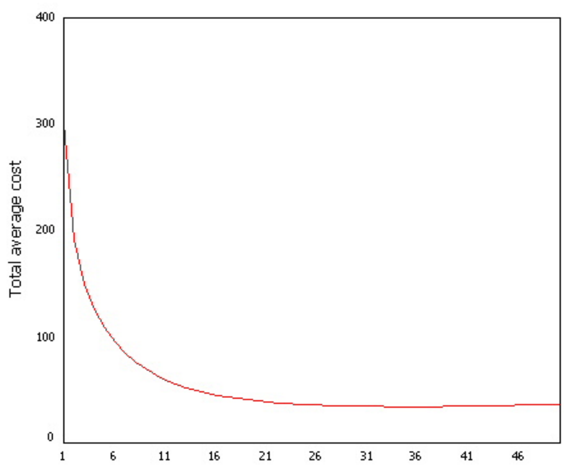

Figure 4.1. Exhaustive search on Q when $R = 20$

4.2.7 Conclusion

We have developed an efficient and reliable method to estimate important performance measures for a single-depot system. This method will be used in the development of a model of a two-location system via the decomposition approach. In all examples of the single-depot system we considered, $C(R, Q)$ is a convex function of Q. However, we offer no proof of this result in general. When $C(R, Q)$ is convex in Q, the optimal value of Q for a fixed R can be found more efficiently using, for example, a bisection search method. In the remainder of this chapter, we will assume $C(R, Q)$ has this property to speed up the search for an optimal Q for the fixed R.

4.3 Decomposition approach based on independent locations with constant demand rates

4.3.1 Introduction

Now we consider the two-location inventory system with unidirectional holdout transshipment policy. Rather than dealing with the explicit holdout transshipment policy directly, we consider an alternative modelling to reflect the holdout transshipment policy. Essentially, the transshipment from location 2 to location 1 has the effect of decreasing the average demand rate at location 1 and increasing the average demand rate at location 2. Hence, one might expect to establish a modelling to reflect the effect of such unidirectional transshipment policy by modifying demand rates at two locations to decompose the whole two-location system into two independent single-depot systems. By this decomposition approach, the level of holdout at location 2 is controlled by a parameter that models the proportion of transshipment requests that location 2 agrees to meet. We refer to this parameter as the transshipment agreement probability.

4.3.2 Assumptions

Principally, at location k, $k = 1, 2$, an independent Poisson demand process is used to model local demand and an $(R_k,\ Q_k)$ replenishment order policy is used to manage inventory. In addition, when a stockout occurs at location 1, transshipment from location 2 to location 1 is allowed to satisfy the unmet demand. Transshipment from location 1 to location 2 is never considered.

At location $k, k = 1, 2$, let c_k be the fixed order cost per order and h_k be the holding cost rate per item. Unmet demand at location 1 must be met either by backorder or by transshipment from location 2. Transshipment is assumed to be instantaneous and involves a cost of t per item. When a backorder is placed at location k, there is one-off stockout cost $\hat{b_k}$ and a further backorder cost of b_k per time unit until the backorder is satisfied.

4.3.3 Approximation of the two-location system

First consider the situation in which location 1 operates independently of location 2. At location 1, from the analysis in Section 4.2.3, the steady-state distribution of inventory level $p_1(x)$ is given by equation (4.1) and hence the fill rate $\alpha_1^0 = \sum_{x=1}^{R_1+Q_1} p_1(x)$.

Whenever a stockout occurs at location 1, a transshipment request will be sent to location 2 and demand rates at locations will only be affected if the request is met. If location 2 agrees to meet the transshipment request, the average demand rate at location 1 will be reduced and the average demand rate at location 2 will be increased. To model this, we introduce the transshipment agreement probability z. Precisely, z is the probability that location 2 agrees to meet a transshipment request. Assume that location 2 will randomly choose whether or not to meet a transshipment request from location 1 and define $\epsilon = (1 - \alpha_1^0)z$. Under our assumption ϵ has the interpretation of the steady probability that a demand at location 1 results in a transshipment and hence is an approximation to the transshipment fill rate. With this level of transshipment, on average the demand rate at location 1 is deflated by $\lambda_1\epsilon$, the demand rate at location 2 is inflated by $\lambda_1\epsilon$, and transshipment occur at a rate of $\lambda_1\epsilon$. Hence, we decompose the two-location system into a system of two independent locations, one with average demand rate $\lambda_1(1-\epsilon)$ and the other with average demand rate $\lambda_2+\lambda_1\epsilon$. Again, using equation (4.1) of Section 4.2.3, we deduce the steady-state distribution of inventory level $p_k^\epsilon(x)$ with the modified demand rate and the fill rate $\alpha_k^\epsilon = \sum_{x=1}^{R_k+Q_k} p_k^\epsilon(x)$ at location $k\,(k=1,2)$.

We hope to use the transshipment agreement probability z to approximate a holdout transshipment policy. That is to say, the transshipment agreement probability z corresponds to a holdout threshold value I_2 at location 2 where the probability that the inventory level at location 2 is great than I_2 is approximately equal to z, i.e. $\sum_{x=I_2}^{R_2+Q_2} p_2^\epsilon(x) \simeq z$.

4.3.4 Performance measures

Firstly by the arguments above the average rate at which transshipments occur is approximated as $\lambda_1\epsilon$, so we can easily derive the average transshipment cost rate $C_t^\epsilon(R_1,Q_1,z)=t\lambda_1\epsilon$ in equation (4.6). At location 1, we know that the sum of direct fill rate, backorder fill rate and transshipment fill rate should be equal to 1. The direct fill rate at location 1 can approximated directly from our assumptions as α_1^0. The indirect fill rate, or the transshipment fill rate at location 1 is approximated as ϵ. Hence the backorder fill rate at location 1 is $1-\alpha_1^0-\epsilon$ and the average stockout cost rate is $\hat{b}_1\lambda_1(1-\alpha_1^0-\epsilon)$.

Since we decompose the two-location system into two independent single-depot systems, we can derive all average cost components at each location with the modified demand rate using the results in the Section 4.2.3. From equation (4.2) with demand rate $\lambda_1(1-\epsilon)$ at location 1, the long-run average order cost rate, average holding cost rate and average backorder cost rate are $c_1\frac{\lambda_1(1-\epsilon)}{Q_1}$, $h_1\sum_{x=1}^{R_1+Q_1} xp_1^\epsilon(x)$ and $b_1\sum_{x=1}^{\infty} xp_1^\epsilon(-x)$ respectively. Hence, the long-run average cost rate $C_1(R_1,Q_1,z)$ at location 1 is given by equation (4.4).

Note that while we obtain an approximation to the stockout cost rate at location directly from the transshipment agreement probability, we could also approximate this cost rate using equation (4.2). Using that method the long-run average stockout cost rate would be $\hat{b}_1\lambda_1(1-\alpha_1^\epsilon)$. If modifying the demand rates accurately captures the effect of transshipment, we would expect the two approximations to give similar estimates. Note also that, if the approach of modifying the demand rates is effective, one would expect $\alpha_1^0+\lambda_1=\alpha_1^\epsilon$.

Similarly, from equation (4.2) with demand rate $\lambda_2 + \lambda_1\epsilon$ at location 2, the long-run average order cost rate, average holding cost rate, average stockout cost rate and average backorder cost rate are $c_2\frac{\lambda_2 + \lambda_1\epsilon}{Q_2}$, $h_2 \sum_{x=1}^{R_2+Q_2} xp_2^{\epsilon}(x)$, $\hat{b_2}\lambda_2(1 - \alpha_2^{\epsilon})$, and $b_2 \sum_{x=1}^{\infty} xp_2^{\epsilon}(-x)$ respectively. Therefore, the long-run average cost rate $C_2^{\epsilon}(R_2, Q_2)$ at location 2 is given by equation (4.5). The long-run average total cost rate $C(R_1, Q_1, R_2, Q_2, z)$ in equation (4.7) includes all costs at location 1 and location 2. From Section 4.2.3, the direct fill rate and backorder fill rate at location 2 are approximated as α_2^{ϵ} and $1-\alpha_2^{\epsilon}$ respectively.

$$C_1(R_1, Q_1, z) = c_1\frac{\lambda_1(1-\epsilon)}{Q_1} + h_1\sum_{x=1}^{R_1+Q_1} xp_1^{\epsilon}(x) + \hat{b_1}\lambda_1(1-\alpha_1^0 - \epsilon) + b_1\sum_{x=1}^{\infty} xp_1^{\epsilon}(-x) \tag{4.4}$$

$$C_2^{\epsilon}(R_2, Q_2) = c_2\frac{\lambda_2 + \lambda_1\epsilon}{Q_2} + h_2\sum_{x=1}^{R_2+Q_2} xp_2^{\epsilon}(x) + \hat{b_2}\lambda_2(1-\alpha_2^{\epsilon}) + b_2\sum_{x=1}^{\infty} xp_2^{\epsilon}(-x) \tag{4.5}$$

$$C_t^{\epsilon}(R_1, Q_1, z) = t\lambda_1\epsilon \tag{4.6}$$

$$C(R_1, Q_1, R_2, Q_2, z) = C_1(R_1, Q_1, z) + C_2^{\epsilon}(R_2, Q_2) + C_t^{\epsilon}(R_1, Q_1, z) \tag{4.7}$$

4.3.5 Optimal financial budget algorithm

Practically, in any multi-location inventory control system, we will face a constraint, such as financial budget, volume or weight restriction, on the maximum inventory level at each location. Let F_k denote the limit on inventory level due to the financial budget constraint at location $k\,(k = 1, 2)$ and note that the maximum inventory level at location k is $R_k + Q_k$. Since $R_k \geqslant 0$ and $Q_k > 1$, R_k and Q_k must satisfy $1 \leqslant R_k + Q_k \leqslant F_k$. To allow for an exhaustive search on z, we further assume that z can only take one of a finite number of possible values between 0 and 1. For each possible combination of R_1, Q_1 and z, we calculate ϵ and decompose the two-location system into a two independent single-depot system. This allows us to calculate the average cost rate at location 1. Finally, we search for the optimal average cost rate at location 2 subject to the constrains $0 \leqslant R_2 \leqslant F_2 - 1$ and $1 \leqslant Q_2 \leqslant F_2 - R_2$. Following the comments of Section 4.2.7, we perform an exhaustive search on R_2, but calculate the optimal Q_2 for each value of R_2 assuming the function $C_2^{\epsilon}(R_2, Q_2)$ is convex in Q_2. Eventually, we have the optimal total cost rate $C(R_1, Q_1, R_2, Q_2, z)$ and the corresponding cost rates at each location.

4.3.6 Numerical experiments

In order to investigate the performance of our approximation approach, we design five snapshots of problem parameters in which b_1 is gradually increased to an extremely large value to provide incentive for transshipment. At the same time b_2 is increased (keeping the ratio of b_1 to b_2 equal 5 to 3) to encourage the use of a partial pooling policy. All other parameters are held constant. In Table 4.5, we provide detailed values of b_1 and b_2 for each snapshot. Meanwhile the other parameters are given values as follows: $\lambda_1 = 20$, $\lambda_2 = 20$, $F_1 = 25$, $F_2 = 25$, $c_1 = 10$, $h_1 = 1$, $\hat{b_1} = 60$, $c_2 = 10$, $h_2 = 1$, $\hat{b_2} = 20$, $t = 5$, $L = 1$.

Snapshot S/N	b_1	b_2	b_1: b_2
1	600	360	5:3
2	700	420	5:3
3	800	480	5:3
4	900	540	5:3
5	1000	600	5:3

Table 4.5. Snapshot summary on b_1 and b_2

z	C^*	C_1^*	R_1^*	Q_1^*	$C_{o_1}^*$	$C_{h_1}^*$	$C_{b_1}^*$	C_2^*	R_2^*	Q_2^*	$C_{o_2}^*$	$C_{h_2}^*$	$C_{b_2}^*$	C_t^*
0	898.1	572.0	23	2	100.0	4.9	467.1	326.1	23	2	100.0	4.9	221.2	0.0
0.1	874.8	509.9	23	2	98.2	5.2	406.5	363.1	23	2	101.9	4.6	256.6	1.9
0.2	859.9	452.4	23	2	96.3	5.5	350.6	403.8	23	2	103.7	4.3	295.8	3.7
0.3*	853.2*	399.3	23	2	94.5	5.9	299.0	448.3	23	2	105.5	4.1	338.8	5.5
0.4	854.4	350.2	23	2	92.6	6.2	251.4	496.9	23	2	107.4	3.8	385.7	7.4
0.5	863.4	304.7	23	2	90.8	6.5	207.4	549.5	23	2	109.2	3.5	436.7	9.2
0.6	879.8	262.5	23	2	88.9	6.9	166.8	606.2	23	2	111.1	3.3	491.8	11.1
0.7	903.2	223.3	23	2	87.1	7.2	129.0	667.0	23	2	112.9	3.0	551.0	12.9
0.8	933.3	186.6	23	2	85.2	7.5	93.9	731.9	23	2	114.8	2.8	614.3	14.8
0.9	961.2	239.0	24	1	171.8	7.9	59.3	708.1	23	2	114.1	2.9	591.1	14.1
1.0	990.5	209.6	24	1	168.7	8.2	32.7	765.2	23	2	115.7	2.7	646.8	15.7

Table 4.6. Snapshot 1 of approximation model with constant demand rates

z	C^*	C_1^*	R_1^*	Q_1^*	$C_{o_1}^*$	$C_{h_1}^*$	$C_{b_1}^*$	C_2^*	R_2^*	Q_2^*	$C_{o_2}^*$	$C_{h_2}^*$	$C_{b_2}^*$	C_t^*
0	1029.0	653.9	23	2	100.0	4.9	548.9	375.2	23	2	100.0	4.9	270.3	0.0
0.1	1001.6	578.9	23	2	98.2	5.2	475.5	420.9	23	2	101.9	4.6	314.4	1.9
0.2	985.2	510.2	23	2	96.3	5.5	408.3	471.3	23	2	103.7	4.3	363.3	3.7
0.3*	979.6*	447.2	23	2	94.5	5.9	346.9	526.8	23	2	105.5	4.1	417.2	5.5
0.4	984.4	389.7	23	2	92.6	6.2	290.9	587.4	23	2	107.4	3.8	476.2	7.4
0.5	999.4	336.9	23	2	90.8	6.5	239.6	653.2	23	2	109.2	3.5	540.5	9.2
0.6	1024.1	288.6	23	2	88.9	6.9	192.8	724.5	23	2	111.1	3.3	610.1	11.1
0.7	1053.7	322.4	24	1	178.1	7.3	137.1	720.3	23	2	111.0	3.3	606.0	11.0
0.8	1083.6	286.4	24	1	174.9	7.6	103.9	784.7	23	2	112.5	3.1	669.1	12.5
0.9	1119.7	252.5	24	1	171.8	7.9	72.8	853.1	23	2	114.1	2.9	736.1	14.1
1.0	1161.5	220.5	24	1	168.7	8.2	43.7	925.4	23	2	115.7	2.7	807.0	15.7

Table 4.7. Snapshot 3 of approximation model with constant demand rates

z	C^*	C_1^*	R_1^*	Q_1^*	$C_{o_1}^*$	$C_{h_1}^*$	$C_{b_1}^*$	C_2^*	R_2^*	Q_2^*	$C_{o_2}^*$	$C_{h_2}^*$	$C_{b_2}^*$	C_t^*
0	1148.6	724.3	24	1	200.0	5.3	519.0	424.3	23	2	100.0	4.9	319.4	0.0
0.1	1127.8	656.1	24	1	196.9	5.6	453.7	470.1	23	2	101.6	4.7	363.9	1.6
0.2	1110.5	567.9	23	2	96.3	5.5	466.7	538.9	23	2	103.7	4.3	430.9	3.7
0.3*	1106.0*	495.2	23	2	94.5	5.9	394.9	605.2	23	2	105.5	4.1	495.6	5.5
0.4	1114.4	429.1	23	2	92.6	6.2	330.3	677.9	23	2	107.4	3.8	566.7	7.4
0.5	1135.4	369.1	23	2	90.8	6.5	271.9	757.0	23	2	109.2	3.5	644.3	9.2
0.6	1159.5	385.3	24	1	181.2	7.0	197.1	764.8	23	2	109.4	3.5	651.9	9.4
0.7	1191.3	342.6	24	1	178.1	7.3	157.2	837.7	23	2	111.0	3.3	723.5	11.0
0.8	1230.3	302.9	24	1	174.9	7.6	120.4	914.8	24	1	225.1	3.4	686.3	12.5
0.9	1271.4	266.0	24	1	171.8	7.9	86.3	991.3	24	1	228.2	3.2	759.9	14.1
1.0	1319.7	231.4	24	1	168.7	8.2	54.6	1072.6	24	1	231.4	3.0	838.3	15.7

Table 4.8. Snapshot 5 of approximation model with constant demand rates

Tables 4.6 to 4.8 provide details of the optimal replenishment policy and the corresponding cost rate for different values of z and three of the snapshots including the optimal R_k^* and Q_k^* at location $k\,(k=1,2)$ and optimal average transshipment cost (C_t^*). In addition, we provide the average cost (C_k^*), order cost ($C_{o_k}^*$), holding cost ($C_{h_k}^*$), the sum of stockout and backorder cost ($C_{b_k}^*$) at location $k\,(k=1,2)$ for each z. Note that optimal order quantity is rather small because of the tight financial budget constraint in these snapshots. The results from Tables 4.6 to 4.8 demonstrate how the system performance changes as backorder cost rate increases. In fact, we find consistency in the optimal transshipment policy for the five snapshots when the ratio between b_1 and b_2 is 5:3.

Figure 4.2 shows the optimal average total cost rates for the five snapshots. Note that, $z=0$ corresponds to a no pooling policy; $z=1$ corresponds to a complete pooling policy; and other values of z correspond to a holdout transshipment policy. In each snapshot, the optimal average total cost rate occurs when $z=0.3$ indicating that, based on the assumption of the decomposition model, a partial pooling policy is optimal. In snapshots 1, 3 and 5, the predicted cost savings from the optimal holdout policy are 5.27%, 5.05% and 3.85% respectively compared to no pooling ($z=0$) and 16.09%, 18.58% and 19.33% respectively compared to complete pooling ($z=1$).

We can derive the optimal holdout threshold value at location 2 suggested by the model by searching for the value of I_2 for which the optimal transshipment agreement probability $z \simeq \sum_{x=I_2+1}^{R_2+Q_2} p_2^{\epsilon}(x)$. In Table 4.9, we provide estimates of $z \cong \sum_{x=I_2+1}^{R_2+Q_2} p_2^{\epsilon}(x)$ for values of I_2 in the range from 1 to 10. We deduce that holdout threshold values of I_2=6 or I_2=5 most closely correspond to a transshipment agreement probability of 0.3 for these five snapshots. These results illustrate how a partial pooling policy consisting of a holdout value at location 2 can be determined for a given transshipment agreement probability. We conclude that the approximation method developed in the sections above can be used to determine replenishment and transshipment policies for a two-location inventory system with unidirectional transshipment policy.

	I_2									
Snapshot	1	2	3	4	5*	6*	7	8	9	10
1	0.67	0.59	0.50	0.42	0.33	0.26	0.19	0.13	0.09	0.05
2	0.67	0.59	0.50	0.42	0.33	0.26	0.19	0.13	0.09	0.05
3	0.67	0.59	0.50	0.42	0.33	0.26	0.19	0.13	0.09	0.05
4	0.67	0.59	0.50	0.42	0.33	0.26	0.19	0.13	0.09	0.05
5	0.67	0.59	0.50	0.42	0.33	0.26	0.19	0.13	0.09	0.05

Table 4.9. The transshipment agreement probability implied by different holdout thresholds

We now check the accuracy of the approximation model's predictions for these five snapshots using the simulation model of the original two-location system with an unidirectional holdout transshipment policy. For all five snapshots, the approximation model suggests that the optimal values for R and Q are 23 and 2 respectively at both locations for most values of z and in particular for the values of z that minimize the total average cost rate. We apply this replenishment policy at both locations and simulate different holdout transshipment policies. The results, plotted in Figure 4.3 show that for this replenishment policy, the optimal transshipment policy for each snapshot is complete pooling. Furthermore, the holdout policy predicted by the approximation model leads to total average cost rates that are 57.79%, 62.76% and 65.05% higher than the complete pooling for snapshots 1, 3 and 5 respectively. Although the holdout transshipment policy predicted by the approximation model does lead to average total cost rates that are 0.10%, 0.11% and 0.87% lower than the no pooling for snapshots 1, 3 and 5 respectively, we conclude that the approximation model does not predict the optimal transshipment policy for the original system with sufficient accuracy.

z	C^*	C_1^*	$C_{o_1}^*$	$C_{h_1}^*$	$C_{b_1}^*$	C_2^*	$C_{o_2}^*$	$C_{h_2}^*$	$C_{b_2}^*$	C_t^*	α_1	γ_1	α_2
0	898.1	572.0	100.0	4.9	467.1	326.1	100.0	4.9	221.2	0.0	0.82	0.00	0.82
0.1	874.7	509.9	98.2	5.2	406.5	363.1	101.8	4.6	256.6	1.8	0.82	0.02	0.79
0.2	859.7	452.4	96.3	5.5	350.6	403.8	103.7	4.3	295.7	3.6	0.82	0.04	0.77
0.3	852.9	399.3	94.5	5.8	299.0	448.3	105.5	4.0	338.8	5.2	0.82	0.06	0.74
0.4	853.9	350.2	92.6	6.2	251.4	496.9	107.4	3.8	385.7	6.8	0.82	0.07	0.71
0.5	862.2	304.7	90.8	6.5	207.4	549.5	109.2	3.5	436.7	8.4	0.82	0.09	0.69
0.6	878.5	262.5	88.9	6.8	166.8	606.2	111.1	3.3	491.8	9.9	0.82	0.11	0.66
0.7	901.5	223.3	87.1	7.2	129.0	666.9	112.9	3.0	551.0	11.3	0.82	0.13	0.63
0.8	931.1	186.6	85.2	7.5	93.9	731.9	114.8	2.8	614.3	12.6	0.82	0.15	0.60
0.9	967.0	152.3	83.4	7.9	59.3	800.9	116.6	2.6	591.1	13.9	0.82	0.17	0.61
1.0	1008.9	119.9	81.5	8.2	32.7	873.9	118.5	2.4	646.8	15.1	0.82	0.18	0.58

Table 4.10. Snapshot 1 of the approximation model with constant demand rates

z	C	C_1	C_{o_1}	C_{h_1}	C_{b_1}	C_2	C_{o_2}	C_{h_2}	C_{b_2}	C_t	α_1	γ_1	α_2
0	900.2	574.2	100.0	4.9	469.2	326.0	100.0	4.9	221.1	0	0.82	0.00	0.82
0.1	801.7	437.0	98.4	5.1	333.4	363.1	101.6	4.7	256.8	1.6	0.84	0.02	0.79
0.2	741.5	343.1	97.3	5.2	240.5	395.7	102.8	4.5	288.4	2.7	0.87	0.03	0.78
0.3	703.3	276.8	96.5	5.3	175.0	422.9	103.6	4.4	314.9	3.5	0.88	0.04	0.77
0.4	678.2	228.8	95.8	5.4	127.5	445.3	104.2	4.3	336.8	4.2	0.90	0.04	0.76
0.5	662.5	192.9	95.4	5.5	92.0	465.0	104.7	4.2	356.0	4.7	0.91	0.05	0.75
0.6	649.2	164.9	95.0	5.5	64.4	479.3	105.1	4.2	370.0	5.1	0.92	0.05	0.74
0.7	641.7	143.3	94.6	5.6	43.1	493.0	105.4	4.2	383.4	5.4	0.92	0.05	0.74
0.8	635.7	125.9	94.3	5.6	26.0	504.1	105.7	4.1	394.2	5.7	0.93	0.06	0.73
0.9	631.0	111.6	94.1	5.7	11.9	513.4	106.0	4.1	403.3	5.9	0.93	0.06	0.73
1.0	628.9	99.6	93.9	5.7	0.0	523.1	106.2	4.1	412.9	6.2	0.94	0.06	0.73

Table 4.11. Snapshot 1 of the simulation model with transshipment agreement probability

To further investigate the reasons for the inaccuracies in the approximation model, we consider a simulation model of the original two-location system in which location 2 agrees to transshipment requests from location 1 at random according to the transshipment agreement probability. We apply the replenishment policy predicted by the approximation model $R_k = 23$ and $Q_k = 2$ at each location and simulate different transshipment agreement probabilities. The results, shown in Figure 4.4, suggest that complete pooling (i.e. $z = 1$) is optimal for all five snapshots. This is, of course, consistent with the simulation model of the original two-location system with a holdout transshipment policy. That gives us a hint that it might still be possible to capture the effect of a holdout transshipment policy by introducing the such transshipment agreement probability.

To allow more detailed comparison between these simulation results and the results of the approximation model, we provide a cost breakdown of snapshot 1 for both the simulation and approximation models in Tables 4.10 to 4.11. Note that we apply the same notation as we used in Tables 4.6 to 4.8. In addition, we provide a breakdown of direct fill rate (α_k), backorder fill (β_k) rate and transshipment fill rate γ_1 at the location $k, (k = 1, 2)$ for comparison.

These comparison results show that the two models are equivalent when $z = 0$ as no transshipment occurs. As the value of z increases, the results demonstrate that the cost components estimated by the simulation and approximation models for each location display the same trends. To be specific, at location 1, the order cost, backorder and stock out cost decrease due to the transshipment from location 2 to location 1. Meanwhile, the holding cost at location 1 increases due to the higher stock level there. At location 2, we see the changes in the opposite directions because of the transshipment from location 2 to location 1. Order cost, backorder and stock out cost increase, while, at the same time, the holding cost decrease. However, comparison with simulation shows that such trends from approximation models are exaggerated due to the lack of accuracy.

It is clear from Tables 4.10 and 4.11 that transshipment has a significant affect on the direct fill rate at location 1 which is not well captured in the approximation model. In fact, the approximation model exaggerates the extent to which transshipment is used and the affect of transshipment on the direct fill rate at location 2. However the trends of these two measures are correct because γ_1 increases with z and α_2 decreases with z.

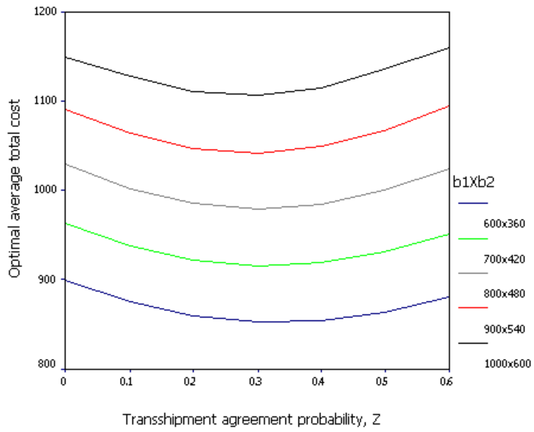

Figure 4.2. Average total cost for approximation model with constant demand rates

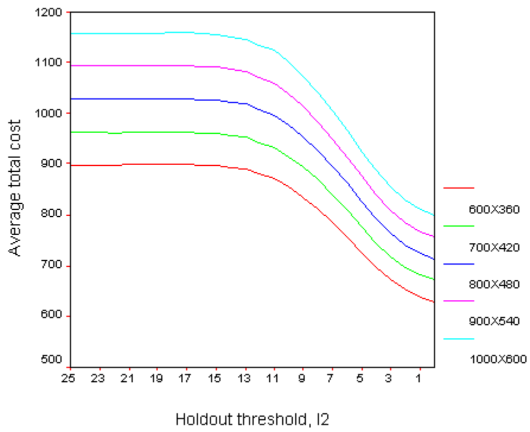

Figure 4.3. Average total cost rate for the original holdout simulation model

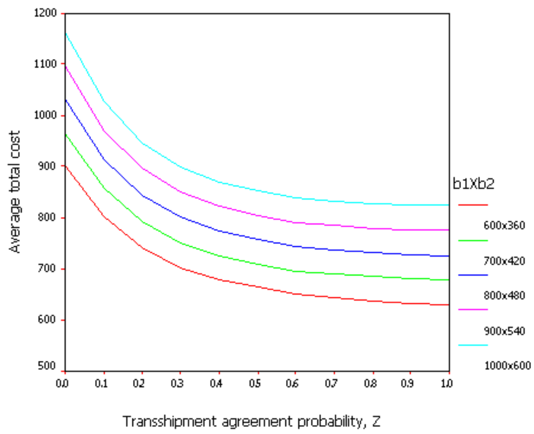

Figure 4.4. Average total cost rate for simulation model with transshipment agreement probability

4.4 Conclusions

In Section 4.3, we formulated an approximation model for the two-location system by decomposing the system into two independent single-depot systems with constant demand rates. This approach provides a simple way to model the unidirectional transshipment between location 2 and location 1. Extensive numerical experiments have shown that the form of the transshipment policy predicted by the approximation model is particularly sensitive to the stockout and backorder costs at the two locations. Detailed results were presented for five snapshots to illustrate cases where the approximation model predicts that partial pooling offers significant savings over the naive heuristics of no pooling and complete pooling. However, a simulation of the original two-location system with unidirectional transshipment shows that these predictions are not accurate and that complete pooling is optimal for the cases considered. It appears that the approximation model does not estimate the impact of transshipment accurately.

From the discussion in Section 4.3.6, we learn that the introduction of the so called transshipment agreement probability works to some extent as we have seen evidence of consistency between the simulation model with transshipment agreement probability and the simulation model of a holdout transshipment policy. For the five snapshots considered in detail, we observe agreement between the pooling policies suggested by the simulation model of the two-location system with the transshipment agreement probability and the simulation model of the two-location system with a holdout transshipment policy. This suggests that the concept of a transshipment agreement probability might be a useful tool for approximating the impact of transshipment in the original two-location system. The discrepancies between the predictions of the approximation model and the behaviour of the original approximation model maybe due to the use of constant demand rates in the sub-problems. Therefore, we need to establish a new modelling approach to reflect dynamic changes in the demands rates at the two locations in the system.

Glossary

R: reorder point of inventory position for a replenishment order
Q: replenishment order quantity
(R,Q): replenish an order with Q units of one kind of item when the inventory position reaches R
L: lead time

Chapter 5

Decomposition approach based on independent locations with variable demand rates

5.1 Introduction

In the preceding chapter, we examined an approximation model of a two-location system with unidirectional transshipment via a decomposition approach. By this approach, we decomposed the system into two independent single-depot systems with constant demand rates by assuming the unidirectional transshipment decision is made at random, independently of the current inventory level at location 2, according to the transshipment agreement probability. The study of the decomposition approach model showed that the approach does not always accurately approximate the behaviour of the original system. However, the results also gave us a hint that the transshipment agreement probability may still be used to approximate the original system to some degree. It is possible that the interactions between the two locations in the original system cannot be captured using independent locations with constant demand rates.

In this chapter, we develop two approximation models of the two-location inventory system by decomposing the system into two independent single-depot systems with non-constant demand rates. The first approximation model recognises that location 1 will only benefit from transshipment when it is out of stock and that location 2 only meets transshipment requests when it has stock. Hence, the effective demand rate at location 1 is only reduced when location 2 has no stock and the effective demand rate at location 2 is only increased when it has stock. However, the transshipment decision is still made randomly according to the transshipment agreement probability and we refer to this model as the **TAP** model to reflect this. The second approximation model attempts to model a holdout transshipment policy explicitly and we refer to this as the **HOT** model. The **HOT** model also recognises that location 1 only benefits from transshipment when it has no stock. However, in the **HOT** model, location 2 only meets transshipment requests when its inventory level is above the holdout threshold and, therefore, the effective demand rate at location 2 is only increased when its inventory level is above the holdout threshold.

Once we have decomposed the system into two independent subsystems, we analyse the period between the arrivals of two successive orders at each location. We define this period as a single cycle for a location. From the renewal theory, we learn that the long-run average cost rate for a location can be worked out from the average cost rate during a single cycle divided by the average length of a cycle. During a single cycle, we model the customer demand at a location as a Poisson process with a variable rate which depends dynamically on the condition of the current inventory level at that location.

We present the transshipment agreement probability **(TAP)** model in Section 5.4 and the explicit holdout transshipment **(HOT)** model in Section 5.5. As the two models share a similar framework, we provide general assumptions in Section 5.2 for both models. In order to decrease repetitions in derivations of the approximations, we derive a number of commonly used expressions for the later cost approximations in Section 5.3.

5.2 Assumptions

We consider a two-location inventory control system with unidirectional transshipment from location 2 to location 1. At location k, $k = 1, 2$, a Poisson demand process with rates λ_k is used to model the customer demand. An (R, Q) replenishment order policy is employed at each location and the lead time for replenishment at each location is assumed to be constant. We denote the reorder position, order quantity and lead time for replenishment by R_k, Q_k, L_k respectively at location k, $k = 1, 2$. Further, we assume that there never can be more than one outstanding replenishment order at a location. In practical terms, this means that the order quantity has to be much larger than the average lead time demand (i.e. $Q_k \gg \lambda_k L_k$), so that it is very unlikely that the reorder position will be reached again before the end of the lead time of an order.

At location $k\,(k=1,2)$, the fixed order cost is c_k and the cost of holding inventory is h_k per item per time unit. Demand that can not be met from on-hand inventory may be backordered. When a backorder is placed at location $k\,(k=1,2)$, there is one-off stockout cost $\hat{b}_k$ and a further backorder cost of b_k per time unit until the backorder is satisfied. Alternatively unmet demands at location 1 may be met by transshipment from location 2. Transshipment is assumed to be instantaneous and involves a cost of t per item.

The two approaches considered in the chapter decompose the system into two independent locations each facing a Poisson demand process with non-constant demand rate. Due to our assumptions about the replenishment process, the time between successive orders at a location (which we refer to as a cycle) consists of an interval when the inventory level is above the reorder position followed by the constant lead time for orders. Hence, the inventory level at the end of a cycle is equal to the reorder position minus the lead time demand. We define the random variable $X_k(R_k)$ to represent the inventory level at the end of a cycle at location k. A cycle at location k ends with the delivery of Q_k items. Hence, the inventory level at the beginning of a cycle is equal to $Q_k + X_k(R_k)$. We derive the probability distributions of the inventory level at the end of a cycle at each location and, hence, deduce approximations to a number of performance measures at each location under the assumptions of the decomposition approach.

5.3 Preliminary results

Before we consider the decomposition approaches in detail, we analyze three situations which frequently arise in the expected cost approximations. Consider a location facing a Poisson demand process at a rate which may depend on the inventory level at the location. It is convenient to define the function $F(x,\lambda)$, the probability that a Poisson random variable with mean λ does not exceed x and its complement

$$F(x,\lambda) = \begin{cases} \sum_{j=0}^{x} \frac{\lambda^j}{j!} e^{-\lambda} & \text{when } x \geqslant 0 \\ 0 & \text{when } x < 0 \end{cases} \quad \text{and} \quad \overline{F}(x,\lambda) = \begin{cases} \sum_{j=x+1}^{\infty} \frac{\lambda^j}{j!} e^{-\lambda} & \text{when } x \geqslant 0 \\ 1 & \text{when } x < 0 \end{cases}$$

Assuming no replenishment orders arrive, we model the inventory level over an interval of time. When the inventory level is greater than zero, holding costs are incurred at a rate of h per item per time unit. When the inventory level is less than zero, backorder costs are incurred at a rate of b per item per time unit.

Lemma 5.1 derives an expression for the expected holding cost for the situation where the inventory level falls from X to Y $(X > Y \geqslant 0)$ according to a Poisson demand process with constant rate. **Lemma** 5.2 derives expressions for the expected holding cost and backorder cost for the situation when inventory level starts from $X \geqslant 0$ and falls according to a Poisson demand process with constant rate during an interval of length L. **Lemma** 5.3 derives expressions for the expected holding cost and backorder cost for the situation where inventory level falls from X, according to a Poisson demand process whose rate changes when inventory level reaches $Y\,(X > Y \geqslant 0)$ during an interval of length L.

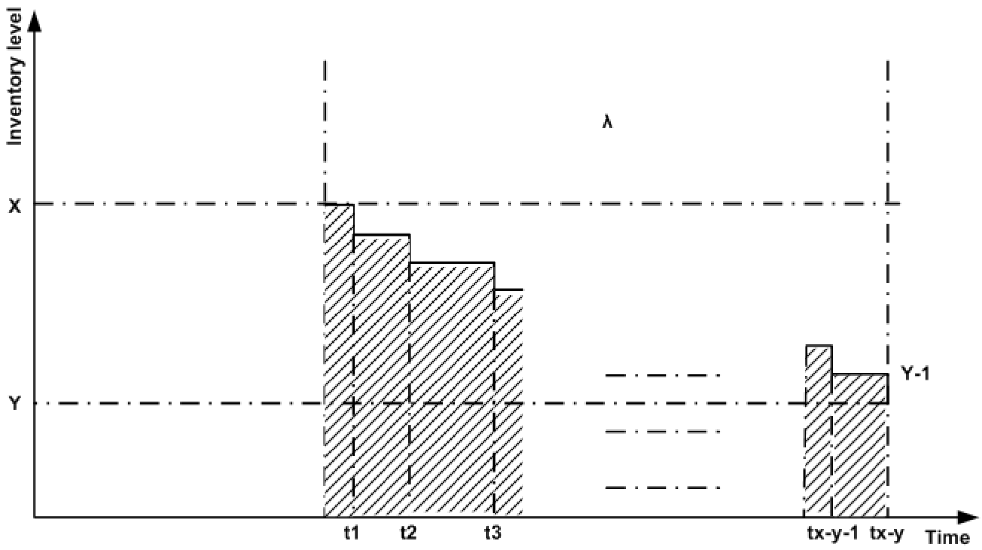

Figure 5.1. Inventory level falls from X to Y due to a Poisson demand process with constant rate λ

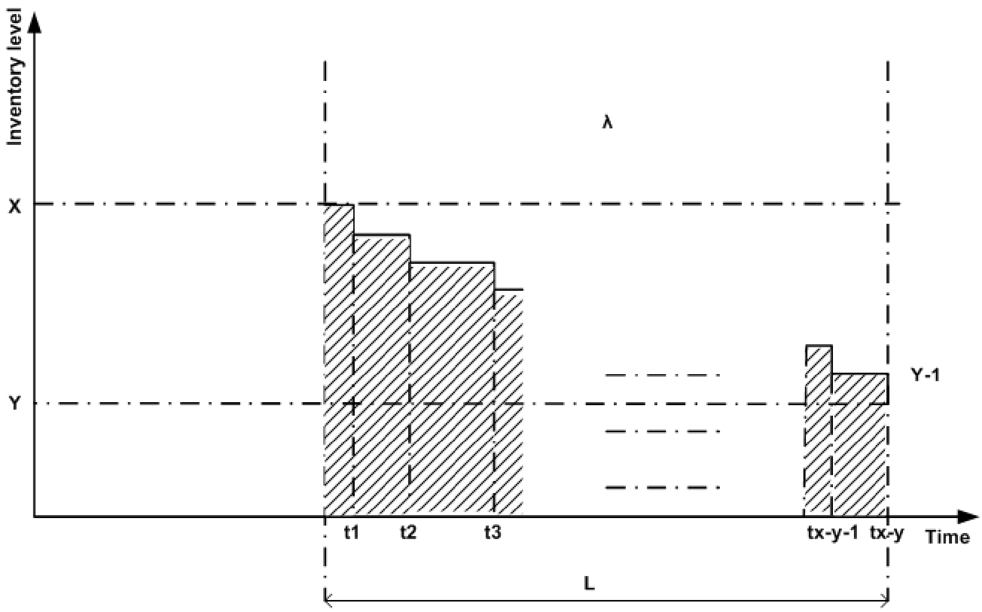

Figure 5.2. Inventory level falls from X to Y in L time units due to a Poisson demand process with constant rate λ

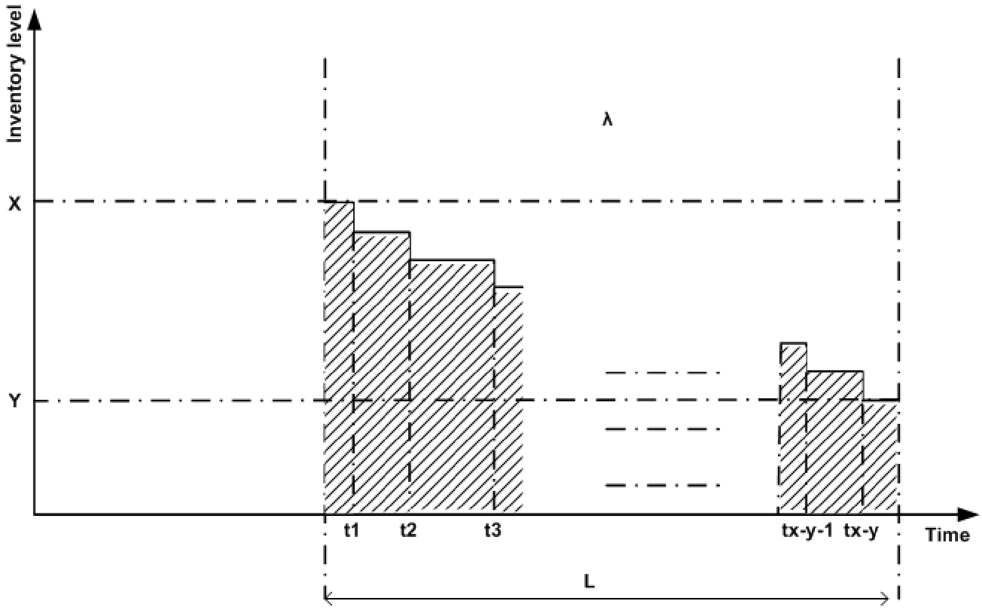

Figure 5.3. Inventory level falls from X to Y during an interval of length L due to a Poisson demand process with constant rate λ

Lemma 5.1. *Consider an interval in which the inventory level falls from* X *to* Y *(*$X > Y \geqslant 0$*) as a result of demands which occur according to a Poisson process with constant rate* λ.

a) *Assume the interval ends when the inventory level first reaches Y, the expected holding cost during the interval is given by*

$$\frac{h}{2\lambda}(X-Y)(X+Y+1) \tag{5.1}$$

b) *Assume the interval ends when the inventory level first reaches Y and the interval is of length L. The expected holding cost during the interval is given by*

$$\frac{hL}{2}(X+Y+1) \tag{5.2}$$

c) *Assume the interval is of length L and demand during the interval is* $X - Y$*. The expected holding cost during the interval is given by*

$$\frac{hL}{2}(X+Y) \tag{5.3}$$

Proof. Let t_i be the time (measured from the start of interval) at which the ith demand occurs, where $1 \leqslant i \leqslant X-Y$.

Three cases are illustrated in Figures 5.1, 5.2 and 5.3. Since demand occurs according to a Poisson process at a constant rate within the interval, we can assume that, for the purpose of calculating expected holding cost, the instances of demand are uniformly distributed within the interval.

a) By considering the area under the graph in Figure 5.1, during the interval we see that the expected holding cost is $h\,E[\sum_{j=0}^{X-Y-1}(X-j)(t_{j+1}-t_j)]$ where $t_0=0$. By the properties of a Poisson process with constant rate λ, we can assume that, in expectation, $t_{j+1}-t_j=1/\lambda$ for $0\leqslant j\leqslant X-Y-1$. Hence, the expected holding cost incurred during the interval is given by

$$\frac{h}{\lambda}\sum_{j=0}^{X-Y-1}(X-j)=\frac{h}{\lambda}[(X-Y)X-\tfrac{1}{2}(X-Y)(X-Y-1)]$$

$$=\frac{h(X-Y)}{2\lambda}(2X-X+Y+1)=\frac{h}{2\lambda}(X-Y)(X+Y+1)$$

b) Consider the situation when the inventory level falls from X and reaches Y at the end of an interval of length L. By considering the area under the graph in Figure 5.2, we see that the expected holding cost during the interval is $h\,E[\sum_{j=0}^{X-Y-1}(X-j)(t_{j+1}-t_j)]$ where $t_0=0$. Since $X-Y$ demands occur just within an interval of length L, we can assume that, in expectation, $t_{j+1}-t_j=\frac{L}{X-Y}$ for $0\leqslant j\leqslant X-Y-1$. Hence, the expected holding cost incurred during the interval is given by

$$\frac{hL}{X-Y}\sum_{j=0}^{X-Y-1}(X-j)=\frac{hL}{X-Y}[X(X-Y)-\tfrac{1}{2}(X-Y)(X-Y-1)]$$

$$=\frac{hL}{2}(2X-X+Y+1)=\frac{hL}{2}(X+Y+1)$$

c) By considering the area under the graph in Figure 5.3, we see that the expected holding cost during the interval is $h\,E[\sum_{j=0}^{X-Y}(X-j)(t_{j+1}-t_j)]$ where $t_0=0$ and $t_{X-Y+1}=L$. Since $X-Y$ demands occur within an interval of length L, we can assume that, in expectation, $t_{j+1}-t_j=\frac{L}{X-Y+1}$ for $0\leqslant j\leqslant X-Y$. Hence, the expected holding cost incurred during the interval is given by

$$\frac{hL}{X-Y+1}\sum_{j=0}^{X-Y}(X-j)=\frac{hL}{X-Y+1}[(X-Y+1)X-\tfrac{1}{2}(X-Y+1)(X-Y)]$$

$$=\frac{hL}{2}(2X-X+Y)=\frac{hL}{2}(X+Y)$$

□

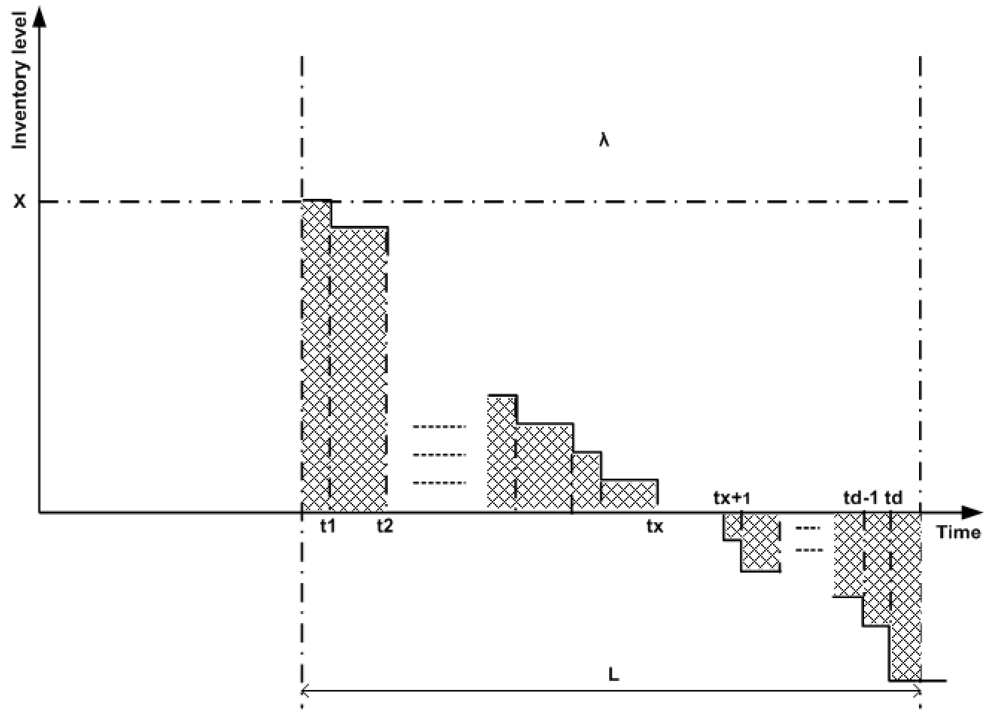

Figure 5.4. Inventory level falls from X for an interval of length L due to a Poisson demand process with constant rate λ

Lemma 5.2. *Consider an interval of length L in which the inventory level starts from $X \geqslant 0$ and demand occurs according to a Poisson process with constant rate λ.*

a) The expected holding cost during the interval is given by

$$hLXF(X-1,\lambda L) - \frac{h\lambda L^2}{2}F(X-2,\lambda L) + \frac{hX(X+1)}{2\lambda}\overline{F}(X,\lambda L) \tag{5.4}$$

b) The expected backorder cost during the interval is given by

$$b\frac{\lambda L^2}{2}\overline{F}(X-1,\lambda L) - bLX\overline{F}(X,\lambda L) + \frac{bX(X+1)}{2\lambda}\overline{F}(X+1,\lambda L) \tag{5.5}$$

Proof. Let D be the random variable representing demand during the interval. Assume $D = d$. Let t_i be the time (measured from the start of the interval) at which the ith demand occurs. Define $t_0 = 0$ and $t_{d+1} = L$. Since demand occurs according to a Poisson process with a constant rate within the interval, we can assume that demands are uniformly distributed within this interval. Hence, in expectation, $t_{j+1} - t_j = \frac{L}{d+1}$ for $0 \leqslant j \leqslant d$. The inventory level as a function of time is illustrated in Figure 5.4. Considering the area under the graph above the horizontal axis in the figure, we see that the expected holding cost during the interval when $X > 0$ is

$$hE[\sum_{j=0}^{\min(X-1,d)}(X-j)(t_{j+1}-t_j)]$$

$$=\frac{hL}{d+1}\sum_{j=0}^{\min(X-1,d)}(X-j)=\frac{hL}{d+1}[X\min(X,d+1)-\frac{1}{2}\min(X-1,d)\min(X,d+1)]$$

$$=\frac{hL}{d+1}\min(X,d+1)[X-\frac{1}{2}\min(X-1,d)]$$

Conditioning on the demand during the interval, we infer that when $X>0$ the expected holding cost during the interval is given by

$$\sum_{d=0}^{\infty}Pr[D=d]\frac{hL}{d+1}\min(X,d+1)[X-\frac{1}{2}\min(X-1,d)]$$

$$=\sum_{d=0}^{X-1}Pr[D=d]\frac{hL}{d+1}(d+1)(X-\frac{1}{2}d)+\sum_{d=X}^{\infty}Pr[D=d]\frac{hL}{d+1}X[X-\frac{1}{2}(X-1)]$$

$$=hLX\sum_{d=0}^{X-1}Pr[D=d]-\frac{hL}{2}\sum_{d=0}^{X-1}dPr[D=d]+\frac{hLX(X+1)}{2}\sum_{d=X}^{\infty}\frac{Pr[D=d]}{d+1}$$

Since D is a Poisson random variable with mean λL, $Pr[D=d]=\frac{\lambda L}{d}Pr[D=d-1]$ for $d\geqslant 1$. Hence, we can simplify the expression for the expected holding cost during the interval as follows.

$$hLX\sum_{d=0}^{X-1}Pr[D=d]-\frac{hL}{2}\sum_{d=1}^{X-1}\lambda LPr[D=d-1]+\frac{hLX(X+1)}{2}\sum_{d=X}^{\infty}\frac{Pr[D=d+1]}{\lambda L}$$

$$=hLXF(X-1,\lambda L)-\frac{h\lambda L^2}{2}F(X-2,\lambda L)+\frac{hX(X+1)}{2\lambda}\overline{F}(X,\lambda L)$$

Note that this expression is equal to 0 when $X=0$, so the expression is valid for all $X\geqslant 0$.

Consider the area under the graph below the horizontal axis in Figure 5.4. When $d>X$, the expected backorder cost during the interval is

$$bE[\sum_{j=X+1}^{d}(t_{j+1}-t_j)(j-X)]=\frac{bL}{d+1}\sum_{j=X+1}^{d}(j-X)=\frac{bL}{d+1}\sum_{j=1}^{d-X}j=\frac{bL}{2(d+1)}(d-X)(d-X+1)$$

Conditioning on the demand during the interval, we infer that the expected backorder cost during the interval is given by

$$\sum_{d=X+1}^{\infty}Pr[D=d]\frac{bL}{2(d+1)}(d-X)(d-X+1)=\frac{bL}{2}\sum_{d=X+1}^{\infty}Pr[D=d]\frac{(d-X)[(d+1)-X]}{d+1}$$

$$=\frac{bL}{2}\sum_{d=X+1}^{\infty}Pr[D=d]\frac{d(d+1)-dX-(d+1)X+X^2}{d+1}$$

$$= \frac{bL}{2} \sum_{d=X+1}^{\infty} Pr[D=d] \frac{d(d+1) - 2(d+1)X + X^2 + X}{d+1}$$

$$= \frac{bL}{2} \sum_{d=X+1}^{\infty} Pr[D=d]d - bLX \sum_{d=X+1}^{\infty} Pr[D=d] + \frac{bL(X^2+X)}{2} \sum_{d=X+1}^{\infty} \frac{Pr[D=d]}{d+1}$$

Again using the fact that $Pr[D=d] = \frac{\lambda L}{d} Pr[D=d-1]$ for $d \geqslant 1$, we can simply the above equation as follows

$$= \frac{b\lambda L^2}{2} \sum_{d=X}^{\infty} Pr[D=d-1] - bLX \sum_{d=X+1}^{\infty} Pr[D=d] + \frac{bL(X^2+X)}{2} \sum_{d=X+2}^{\infty} \frac{Pr[D=d+1]}{\lambda L}$$

$$= b\frac{\lambda L^2}{2} \overline{F}(X-1, \lambda L) - bLX\overline{F}(X, \lambda L) + \frac{bX(X+1)}{2\lambda} \overline{F}(X+1, \lambda L)$$

□

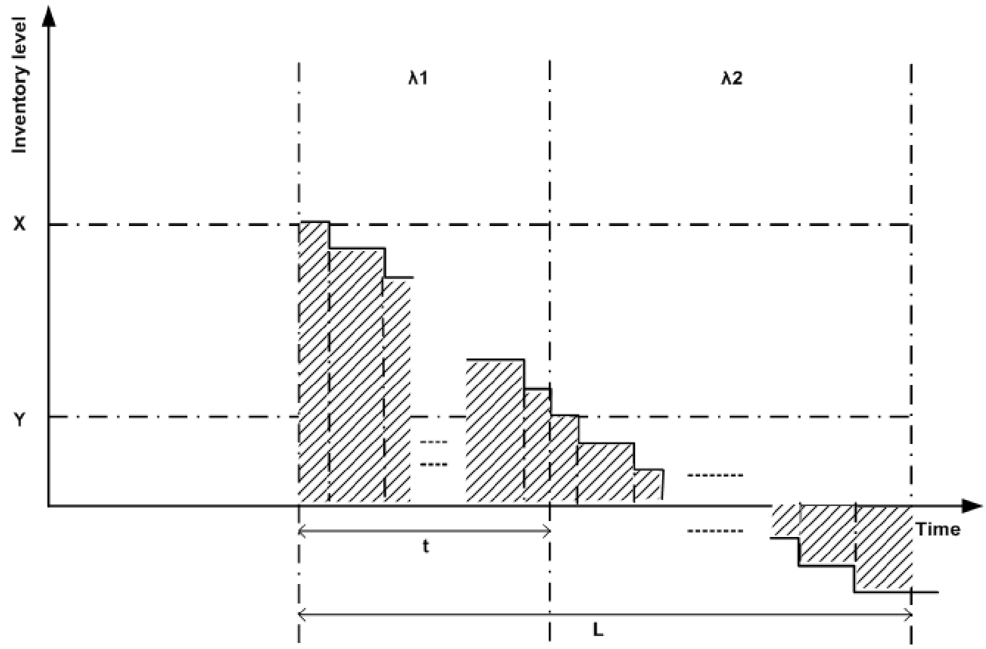

Figure 5.5. Inventory level falls from X for an interval of length L due to a Poisson demand process with a rate that changes when inventory level reaches Y

Lemma 5.3. *Consider an interval of length L in which the inventory level starts from X and demand occurs according to a Poisson process with rate λ_1, when inventory level is greater than Y and with rate λ_2 when inventory level is less than or equal to Y ($X > Y \geqslant 0$).*

a) The expected holding cost during the interval before the change of demand rate is given by

$$hLXF(X-Y-1, \lambda_1 L) - h\frac{\lambda_1 L^2}{2} F(X-Y-2, \lambda_1 L) +$$

$$h\frac{(X+Y+1)(X-Y)}{2\lambda_1}\overline{F}(X-Y,\lambda_1 L) \tag{5.6}$$

b) The expected holding cost during the interval after the change of demand rate is given by

$$h\int_0^L \frac{\lambda_1^{X-Y}t^{X-Y-1}}{(X-Y-1)!}e^{-\lambda_1 t}\{(L-t)YF(Y-1,\lambda_2(L-t))-$$
$$\frac{\lambda_2(L-t)^2}{2}F(Y-2,\lambda_2(L-t))+\frac{Y(Y+1)}{2\lambda_2}\overline{F}(Y,\lambda_2(L-t))\}dt \tag{5.7}$$

c) The expected backorder cost during the interval is given by

$$b\int_0^L \frac{\lambda_1^{X-Y}t^{X-Y-1}}{(X-Y-1)!}e^{-\lambda_1 t}\{\frac{\lambda_2(L-t)^2}{2}\overline{F}(Y-1,\lambda_2(L-t))-(L-t)Y\overline{F}(Y,\lambda_2(L-t))+$$
$$\frac{Y(Y+1)}{2\lambda_2}\overline{F}(Y+1,\lambda_2(L-t))\}dt \tag{5.8}$$

Proof. We consider the expected holding cost during the interval when the demand rate is λ_1 and the expected holding cost during the interval when demand rate is λ_2 separately.

For the expected holding cost during the interval when the demand rate is λ_1, we consider two situations: one in which the inventory level does not reach Y, and the other in which the inventory level falls from X to Y within a period of length t.

For the first situation, let D be the random variable representing the demand during the interval. When demand rate is λ_1, during the interval of length L, inventory level falls from X to $X-D>Y\geqslant 0$ and occurs according to a Poisson process with constant rate λ_1. By Lemma 5.1c, the expected holding cost is given by $E[\frac{hL}{2}(X+X-D)]$

For the second situation, note that the time until the inventory level first falls to Y has an **Erlang** distribution with shape parameter $X-Y$ and scale parameter $\lambda_1 L$. Assume that inventory level first falls to Y after time t. During this interval, demand occurs according to a Poisson process with constant rate λ_1. By Lemma 5.1b, the expected holding cost is $\frac{ht}{2}(X+Y+1)$. Hence, the expected holding cost during the interval when the demand rate is λ_1 is given by

$$E[\tfrac{hL}{2}(2X-D)] + h\int_0^L \tfrac{\lambda_1^{X-Y}t^{X-Y-1}}{(X-Y-1)!}e^{-\lambda_1 t}\tfrac{t}{2}(X+Y+1)dt$$
$$= \sum_{j=0}^{X-Y-1} Pr[D=j]\tfrac{hL}{2}(2X-j) + h\tfrac{(X+Y+1)(X-Y)}{2\lambda_1}\int_0^L \tfrac{\lambda_1^{X-Y+1}t^{X-Y}}{(X-Y)!}e^{-\lambda_1 t}dt$$
$$= hLX\sum_{j=0}^{X-Y-1} Pr[D=j] - \tfrac{hL}{2}\sum_{j=1}^{X-Y-1} jPr[D=j] + h\tfrac{(X+Y+1)(X-Y)}{2\lambda_1}\int_0^L \tfrac{\lambda_1^{X-Y+1}t^{X-Y}}{(X-Y)!}e^{-\lambda_1 t}dt$$

Recall D is a Poisson random variable with mean $\lambda_1 L$, so $jPr[D=j] = \lambda_1 L Pr[D=j-1]$. Note also that the integral in the expression above equals the probability that the $(X-Y+1)$th instance of demand from a Poisson process with mean $\lambda_1 L$ occurs during the interval of length L and so is simply the probability that D is greater than $X-Y$. Hence, we can simplify this expression further as follows.

$$= hLX\sum_{j=0}^{X-Y-1} Pr[D=j] - \tfrac{hL}{2}\sum_{j=1}^{X-Y-1} \lambda_1 L Pr[D=j-1] + h\tfrac{(X+Y+1)(X-Y)}{2\lambda_1}\sum_{j=X-Y+1}^{\infty} Pr[D=j]$$
$$= hLXF(X-Y-1,\lambda_1 L) - h\tfrac{\lambda_1 L^2}{2}F(X-Y-2,\lambda_1 L) + h\tfrac{(X+Y+1)(X-Y)}{2\lambda_1}\overline{F}(X-Y,\lambda_1 L)$$

Expressions for the expected holding cost and the expected backorder cost when the demand rate is λ_2 in Figure 6.5, can be obtained by applying Lemma 5.2. We see that, during an interval of length $L-t$, inventory level falls from Y due to a Poisson demand process with constant rate λ_2. By Lemma 5.2, the expected holding cost during this interval is given by

$$h(L-t)YF(Y-1,\lambda_2(L-t)) - \frac{h\lambda_2(L-t)^2}{2}F(Y-2,\lambda_2(L-t)) + \frac{hY(Y+1)}{2\lambda_2}\overline{F}(Y,\lambda_2(L-t))$$

and the expected backorder cost during this interval is given by

$$\frac{b\lambda_2(L-t)^2}{2}\overline{F}(Y-1,\lambda_2(L-t)) - b(L-t)Y\overline{F}(Y,\lambda_2(L-t)) + \frac{bY(Y+1)}{2\lambda_2}\overline{F}(Y+1,\lambda_2(L-t))$$

Recall that t has an **Erlang** distribution with shape parameter $X-Y$ and scale parameter $\lambda_1 L$. Hence, the expected holding cost during the interval when the demand rate is λ_2 is given by

$$h\int_0^L \tfrac{\lambda_1^{X-Y}t^{X-Y-1}e^{-\lambda_1 t}}{(X-Y-1)!}\{(L-t)YF(Y-1,\lambda_2(L-t)) -$$
$$\tfrac{\lambda_2(L-t)^2}{2}F(Y-2,\lambda_2(L-t)) + \tfrac{Y(Y+1)}{2\lambda_2}\overline{F}(Y,\lambda_2(L-t))\}dt$$

Finally, the expected backorder cost during the interval of length L is given by

$$\int_0^L \tfrac{\lambda_1^{X-Y}t^{X-Y-1}}{(X-Y-1)!}e^{-\lambda_1 t}\{\tfrac{b\lambda_2(L-t)^2}{2}\overline{F}(Y-1,\lambda_2(L-t)) - b(L-t)Y\overline{F}(Y,\lambda_2(L-t)) +$$
$$\tfrac{bY(Y+1)}{2\lambda_2}\overline{F}(Y+1,\lambda_2(L-t))\}dt \qquad \square$$

5.4 Transshipment agreement probability (TAP) model to approximate holdout policy

Similar to the decomposition approach we examined in Chapter 4, we reuse the transshipment agreement probability to model the unidirectional transshipment from location 2 to location 1. However, we decompose the two-location system into two independent single-depot systems with non-constant demand rates and call this the **TAP** model.

The particular assumptions for **TAP** model are introduced in Section 5.4.1. The distribution of the inventory level at the end of a cycle and mean cycle time at location 1 and 2 are derived in Section 5.4.2 and Section 5.4.3. Subsequently, we provide approximations to a range of performance measures including average cost rate and direct fill rate at location 1 and 2 in Section 5.4.4. In Section 5.4.5, we present an algorithm to estimate two factors θ and ϕ and explain how this algorithm can be used to approximate the optimal transshipment policy.

In Section 5.4.6, we verify the correctness of our approximations to the performance measures at two locations using simulation of a single location with non-constant demand rate. Finally, a series of numerical experiments designed to evaluate the **TAP** model are presented in Section 5.4.7.

5.4.1 Assumptions

In addition to the general assumptions of Section 5.2, we define following assumptions for the **TAP** model.

We approximate the transshipment policy by assuming that, when location 2 has a positive inventory level, there is a probability of location 2 satisfying a transshipment request from location 1. We call this probability the 'transshipment agreement probability' and denote it by z. More specifically, if $z = 0$, the transshipment policy is equivalent to the no pooling policy; if $z = 1$, the transshipment policy approximates to the complete pooling policy; if $0 < z < 1$, the transshipment policy approximates to a holdout transshipment policy. We hope the approximations are sufficiently accurate to inform the choice of transshipment policy.

Define X_k be the inventory level at location $k, (k = 1, 2)$. We introduce $\phi = P[X_2 \geqslant 1]$ representing the probability that location 2 can consider a transshipment request and $\theta = P[X_1 \leqslant 0]$ representing the probability that location 1 needs to make a transshipment request. We use these probabilities to decompose the two-location system in the following manner.

When the inventory level at location 1 is greater than zero, location 1 is able to meet all of its demands from its local stock. Hence, the inventory level at location 1 falls due to a Poisson demand process with rate λ_1. When the inventory level of location l reaches R_1, an order for Q_1 items is placed which will arrive after a fixed time L_1. When the inventory level at location 1 is less than or equal to zero, location 1 makes transshipment requests at a rate of λ_1 per time unit. If location 2 is able to consider transshipment requests, a transshipment request will be satisfied with probability z. The probability that location 2 is able to consider transshipment requests is ϕ. Hence, the probability that a transshipment request at location 1 results in a backorder is $1 - \phi z$. Therefore, when the inventory level at location 1 is less than or equal to zero, inventory level at location 1 falls due to a Poisson demand process with rate $\lambda_1(1 - \phi z)$. The situation at location 1 is illustrated in Figure 5.6.

Similarly, at location 2, when the inventory level is greater than zero, location 2 is able to meet all of its demands from its local stock and consider all transshipment requests from location 1. Since location 1 only makes transshipment requests when it is out of stock, transshipment requests occur at an average rate of $\lambda_1\theta$ per time unit. Hence, when the inventory level at location 2 is greater than zero, the inventory level at location 2 falls due to a Poisson demand process with rate $\lambda_2 + \lambda_1 z\theta$. When the inventory level of location 2 reaches R_2, an order for Q_2 items is placed which will arrive after a fixed time L_2. When the inventory level at location 2 is less than or equal to zero, location 2 backorders all of its demands and is unable to consider any other transshipment requests from location 1. Hence, the inventory level at location 2 falls due to a Poisson demand process with rate λ_2. The situation at location 2 is illustrated in Figure 5.7.

Hence, we decompose the two-location system into two independent single-depot systems with non-constant demand rates. Because our approach focuses on each location over a cycle, we need to derive the mean cycle time and distribution of inventory level at the end of a cycle at each location. Compared to the decomposition approach we considered in Chapter 4, demand process at each location under this decomposition approach depends on the local inventory levels, which hopefully can give us a good approximation to the interactions between the locations in the two-location system with unidirectional transshipment.

To apply this approach to the original two-location system with unidirectional transshipment, we need to be able to associate a value of z with a holdout transshipment policy. Let I_2 denote the holdout threshold value at location 2. The transshipment agreement probability is the probability that location 2 meets a transshipment request given that it is able to consider the request. Hence, this is the probability that the inventory level at location 2 is greater than I_2 given that it is least 1. If we can find the value of z which minimises the average total cost for the whole system, z^* say, we can derive a corresponding holdout policy by finding I_2 satisfying $P[X_2 > I_2 | X_2 \geqslant 1] \simeq z^*$.

5.4.2 Distribution of inventory level & mean cycle time for location 1

Under the assumption that there is never more than one outstanding order at each location, whenever an order arrives at location 1, the inventory level is restored to a level greater than R_1. Figure 5.6 depicts the inventory level as a function of time during a single cycle for location 1.

Demand rate distributions

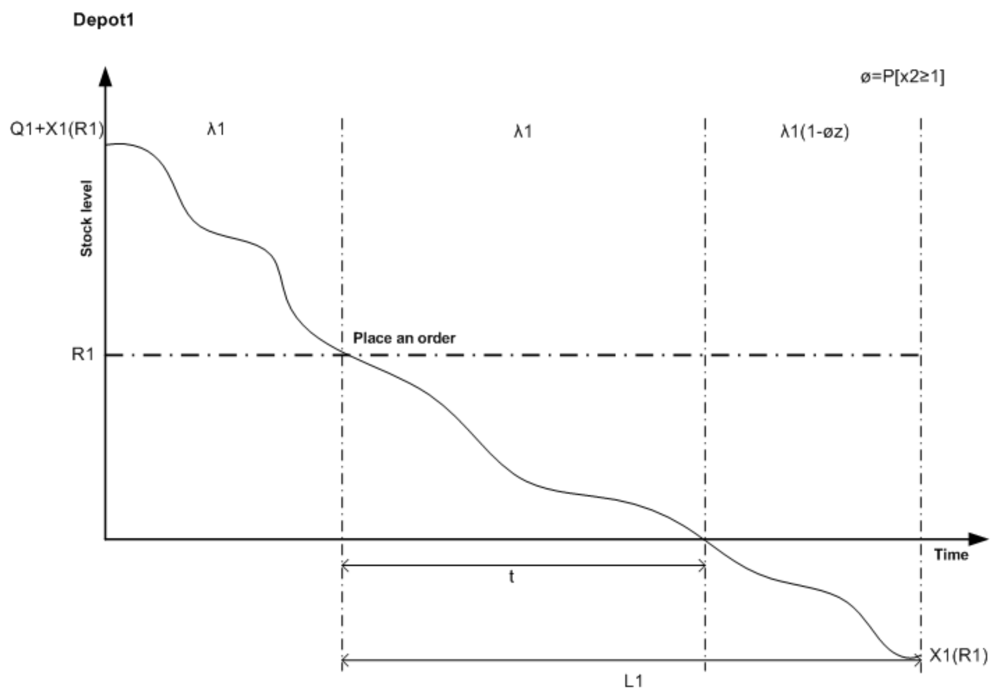

Figure 5.6. Inventory level at location 1 as a function of time during a single cycle

We first derive the probability distribution of $X_1(R_1)$, the inventory level at the end of a cycle at location 1.

Distribution of $X_1(R_1)$

Define D_1 to be a random variable representing the lead time demand. We consider two cases for the demand during the lead time. For the case when the lead time demand is less then R_1, then the rate of the Poisson demand process is equal to λ_1 throughout the cycle. Hence, for $0 \leqslant d < R_1$, the probability that the lead time demand is equal to d is given by a simple Poisson probability as follows:

$$P[D_1 = d] = \frac{(\lambda_1 L_1)^d}{d!} e^{-\lambda_1 L_1} \tag{5.9}$$

For the case when the lead time demand is greater than R_1, the time at which the inventory level reaches zero has an **Erlang** distribution with shape parameter R_1 and scale parameter $\lambda_1 L_1$. If t is the time at which the rate changes, the demand during the remainder of the lead time is a Poisson random variable with mean $\lambda_1(1-\phi z)(L_1 - t)$. Conditioning on the time at which the inventory level reaches zero, the probability that the lead time demand equals d, where $d \geqslant R_1$ is as follows

$$\begin{aligned} P[D_1 = d] &= \int_0^{L_1} \frac{\lambda_1^{R_1} t^{R_1-1} e^{-\lambda_1 t}}{(R_1-1)!} \{ \frac{[\lambda_1(1-\phi z)(L_1-t)]^{d-R_1}}{(d-R_1)!} e^{-\lambda_1(1-\phi z)(L_1-t)} \} dt \\ &= \frac{\lambda_1^{R_1}[\lambda_1(1-\phi z)]^{d-R_1}}{(R_1-1)!(d-R_1)!} e^{-\lambda_1(1-\phi z)L_1} \int_0^{L_1} t^{R_1-1}(L_1-t)^{d-R_1} e^{-\lambda_1 \phi z t} dt \end{aligned}$$

Note that $X_1(R_1) = R_1 - D_1$ and $P[X_1(R_1) = j] = P[D_1 = R_1 - d]$. Since $X_1(R_1) = R_1 - D_1$ and $D_1 \geqslant 0$, $X_1(R_1) \leqslant R_1$. Hence, we can infer the probability distribution of $X_1(R_1)$ as follows

For $1 \leqslant j \leqslant R_1$

$$P[X_1(R_1) = j] = \frac{(\lambda_1 L_1)^{R_1 - j}}{(R_1 - j)!} e^{-\lambda_1 L_1} \tag{5.10}$$

While for $j \leqslant 0$

$$P[X_1(R_1) = j] = \frac{\lambda_1^{R_1}[\lambda_1(1-\phi z)]^{-j}}{(R_1-1)!(-j)!} e^{-\lambda_1(1-\phi z)L_1} \int_0^{L_1} t^{R_1-1}(L_1-t)^{-j} e^{-\lambda_1 \phi z t} dt \tag{5.11}$$

We then have

$$E[X_1(R_1)] = \sum_{j=-\infty}^{R_1} j P[X_1(R_1) = j] \tag{5.12}$$

$$E[X_1(R_1)^2] = \sum_{j=-\infty}^{R_1} j^2 P[X_1(R_1) = j] \tag{5.13}$$

Having established the probability distribution of the inventory level at the end of a cycle, we are now able to derive the expected cycle time.

Cycle time

As illustrated in Figure 5.6, the mean cycle time consists of two parts, one part before the lead time starts and the other part is the lead time. During the first part, the inventory level falls from $Q_1 + X_1(R_1)$ to R_1 according to a Poisson process with constant rate λ_1. Thus the mean cycle time is given by

$$\frac{E[Q_1 + X_1(R_1) - R_1]}{\lambda_1} + L_1 = \frac{(Q_1 - R_1)}{\lambda_1} + \frac{E[X_1(R_1)]}{\lambda_1} + L_1 \tag{5.14}$$

5.4.3 Distribution of inventory level & mean cycle time for location 2

We now derive the mean cycle time and the probability distribution of $X_2(R_2)$, the inventory level at the end of a single cycle at location 2. Under the assumption that there is never more than one outstanding order at each location, whenever an order arrives at location 2, the inventory level is restored to a level greater than R_2. Figure 5.7 shows the inventory level as a function of time during a single cycle for location 2.

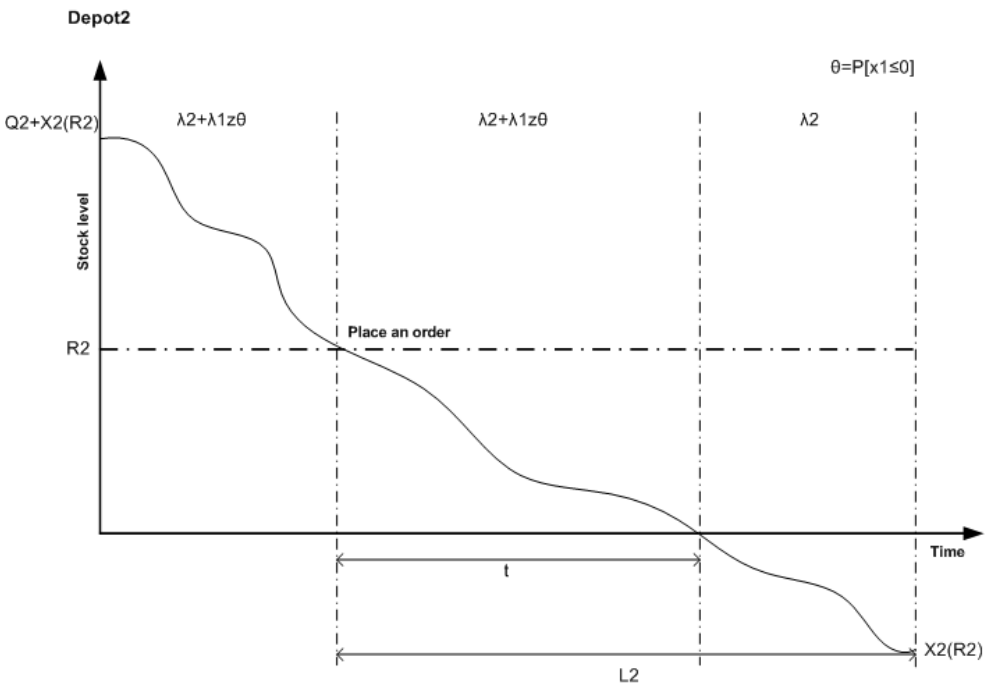

Figure 5.7. Inventory level at location 2 as a function of time during a single cycle

Distribution of $X_2(R_2)$

Define D_2 to be a random variable representing the lead time demand. We consider two cases for the demand during the lead time. For the case when the lead time demand is less than R_2, then the rate of the Poisson demand process is equal to $\lambda_2 + \lambda_1 z\theta$ throughout the cycle. Hence, for $0 \leqslant d < R_2$, the probability that the lead time demand is equal to d is given by a simple Poisson probability as follows:

$$P[D_2 = d] = \frac{[(\lambda_2 + \lambda_1 z\theta)L_2]^d}{d!} e^{-(\lambda_2 + \lambda_1 z\theta)L_2} \tag{5.15}$$

For the case when the lead time demand is greater than R_2, the time at which the inventory level reaches zero and, hence the rate of the Poisson demand process changes, has an **Erlang** distribution with shape parameter R_2 and scale parameter $(\lambda_2 + \lambda_1 z\theta)L_2$. If t is the time at which the rate changes, the demand during the remainder of the lead time is a Poisson random variable with mean $\lambda_2(L_2 - t)$. Conditioning on the time at which the inventory level reaches zero, the probability that the lead time demand equals d, where $d \geqslant R_2$ is as follows

$$\begin{aligned} P[D_2 = d] &= \int_0^{L_2} \frac{(\lambda_2 + \lambda_1 z\theta)^{R_2} t^{R_2 - 1} e^{-(\lambda_2 + \lambda_1 z\theta)t}}{(R_2 - 1)!} \left[\frac{[\lambda_2(L_2 - t)]^{d - R_2}}{(d - R_2)!} e^{-\lambda_2(L_2 - t)}\right] dt \\ &= \frac{[(\lambda_2 + \lambda_1 z\theta)]^{R_2}}{(R_2 - 1)!(d - R_2)!} e^{-\lambda_2 L_2} \lambda_2^{d - R_2} \int_0^{L_2} t^{R_2 - 1} e^{-\lambda_1 z\theta t} (L_2 - t)^{d - R_2} dt \end{aligned} \tag{5.16}$$

Since $X_2(R_2) = R_2 - D_2$ and $D_2 \geqslant 0$, we can infer the probability of $X_2(R_2)$ as follows

For $1 \leqslant j \leqslant R_2$

$$P[X_2(R_2) = j] = \frac{[(\lambda_2 + \lambda_1 z\theta)L_2]^{R_2 - j}}{(R_2 - j)!} e^{-(\lambda_2 + \lambda_1 z\theta)L_2} \tag{5.17}$$

While for $j \leqslant 0$

$$P[X_2(R_2) = j] = \frac{(\lambda_2 + \lambda_1 z\theta)^{R_2} \lambda_2^{-j}}{(R_2 - 1)!(-j)!} e^{-\lambda_2 L_2} \int_0^{L_2} t^{R_2 - 1} (L_2 - t)^{-j} e^{-\lambda_1 z\theta t} dt \tag{5.18}$$

Then we infer the expected mean of $X_2(R_2)$ and $X_2(R_2)^2$

$$E[X_2(R_2)] = \sum_{j=-\infty}^{R_2} j P[X_2(R_2) = j] \tag{5.19}$$

$$E[X_2(R_2)^2] = \sum_{j=-\infty}^{R_2} j^2 P[X_2(R_2) = j] \tag{5.20}$$

Having established the probability distribution of the inventory level at the end of a cycle, we are now able to derive the expected cycle time.

Cycle time

As illustrated in Figure 5.7, the mean cycle time consists of two parts, one part before the lead time starts and the other part is the lead time. During the first part, the inventory level falls from $Q_2 + X_2(R_2)$ to R_2 according to a Poisson process with rate $\lambda_2 + \lambda_1 z\theta$. Thus the mean cycle time is given by

$$\frac{E[Q_2 + X_2(R_2) - R_2]}{\lambda_2 + \lambda_1 z\theta} + L_2 = \frac{(Q_2 - R_2) + E[X_2(R_2)]}{\lambda_2 + \lambda_1 z\theta} + L_2 \tag{5.21}$$

5.4.4 System performance measure approximations

From the renewal theory, we obtain the result that

$$\text{Average cost rate at location } k = \frac{\text{mean cost over } a \text{ single cycle at location } k}{\text{mean time of } a \text{ single cycle at location } k} \tag{5.22}$$

Because we have already computed the mean cycle time for each location, in order to estimate the average cost rate at a location, we need to find an expression for the mean cost over a cycle at the location. We consider four cost components at each location : **ORDER COST, HOLDING COST, BACKORDER COST AND STOCKOUT COST.** The average transshipment cost per time unit is estimated using a similar approach and added to obtain the average cost rate for the system as a whole.

Order cost over a cycle

During a cycle at location k, $k = 1, 2$, exactly one replenishment order is placed. Hence the order cost over a cycle at location k is given by c_k.

Holding cost over a cycle at location 1

During the period preceding the point at which the order is placed, inventory level falls from $Q_1 + X_1(R_1)$ to R_1 due to a Poisson demand process with rate λ_1. By Lemma 5.1a, the holding cost during this period is given by

$$\frac{h_1}{2\lambda_1}[Q_1 + X_1(R_1) - R_1][Q_1 + X_1(R_1) + R_1 + 1]$$

Hence the expected holding cost during this period is given by

$$E\{\tfrac{h_1}{2\lambda_1}[Q_1+X_1(R_1)-R_1][Q_1+X_1(R_1)+R_1+1]\}$$
$$=\tfrac{h_1}{2\lambda_1}\{Q_1^2+Q_1-R_1^2-R_1+(2Q_1+1)E[X_1(R_1)]+E[X_1(R_1)^2]\} \tag{5.23}$$

During the lead time of length L_1, the inventory level starts from R_1 and falls due to a Poisson process with rate λ_1 until it reaches zero. After this point, no further holding cost is incurred. Hence by Lemma 5.2, the expected holding cost during this period is given by

$$h_1L_1R_1F(R_1-1,\lambda_1L_1)-\frac{h_1\lambda_1L_1^2}{2}F(R_1-2,\lambda_1L_1)+\frac{h_1R_1(R_1+1)}{2\lambda_1}\overline{F}(R_1,\lambda_1L_1) \tag{5.24}$$

Hence, the expected holding cost over a single cycle at location 1 is equal to the sum of equations (5.23) and (5.24).

Holding cost over a cycle at location 2

Between the start of a cycle and the point at which an order is placed, the inventory level falls from $Q_2+X_2(R_2)$ to R_2 due to a Poisson demand process with constant rate $\lambda_2+\lambda_1 z\theta$. By Lemma 5.1a the expected holding cost during this period is given by

$$E\{\tfrac{h_2}{2(\lambda_2+\lambda_1 z\theta)}[Q_2+X_2(R_2)-R_2][Q_2+X_2(R_2)+R_2+1]\}$$
$$=\tfrac{h_2}{2(\lambda_2+\lambda_1 z\theta)}\{Q_2^2+Q_2-R_2^2-R_2+(2Q_2+1)E[X_1(R_1)]+E[X_2(R_2)^2]\} \tag{5.25}$$

During the lead time of length L_2, the inventory level starts from R_2 and falls due to a Poisson process with rate $\lambda_2+\lambda_1 z\theta$ until it reaches zero. After this point, no further holding cost is incurred. Hence by Lemma 5.2,

$$h_2L_2R_2F(R_2-1,(\lambda_2+\lambda_1 z\theta)L_2)-\tfrac{h_2(\lambda_2+\lambda_1 z\theta)L_2^2}{2}F(R_2-2,(\lambda_2+\lambda_1 z\theta)L_2)+$$
$$\tfrac{h_2R_2(R_2+1)}{2(\lambda_2+\lambda_1 z\theta)}\overline{F}(R_2,(\lambda_2+\lambda_1 z\theta)L_2) \tag{5.26}$$

Hence, the expected holding cost over a single cycle at location 1 is equal to the sum of equations (5.25) and (5.26).

Stockout cost over a cycle

The inventory level at a location is only ever replenished at the beginning of a cycle. Therefore, any backorder placed at a location during a cycle will still be outstanding at the end of that cycle. It follows that the number of backorders placed at location k, $k = 1, 2$ during a cycle is equal to $max(-X_k(R_k), 0)$. Hence, the expected stockout cost during a cycle at location k is given by

$$\sum_{j=1}^{\infty} j\hat{b}_k P[X_k(R_k) = -j] \tag{5.27}$$

Backorder cost over a cycle at location 1

If the backorder cost is incurred during a cycle, it is incurred during an interval at the end of the lead time. During the lead time, the inventory level falls from R_1 to zero according to a Poisson process with constant rate λ_1. After this point, the inventory level continues to fall according to a Poisson process with constant rate $\lambda_1(1 - \phi z)$. By Lemma 5.3c, the expected backorder cost during a cycle is given by

$$\begin{aligned} & b_1 \int_0^{L_1} \frac{\lambda_1^{R_1} t^{R_1 - 1} e^{-\lambda_1 t}}{(R_1 - 1)!} \frac{\lambda_1(1-\phi z)(L_1 - t)^2}{2} \overline{F}(-1, \lambda_1(1-\phi z)(L_1 - t))dt \\ &= b_1 \frac{\lambda_1^{R_1+1}(1-\phi z)}{2(R_1 - 1)!} \int_0^{L_1} t^{R_1 - 1} e^{-\lambda_1 t}(L_1 - t)^2 dt \end{aligned} \tag{5.28}$$

Backorder cost over a cycle at location 2

If the backorder cost is incurred during a cycle, it is incurred during an interval at the end of the lead time. During the lead time, the inventory level falls from R_2 to zero according to a Poisson process with constant rate $\lambda_2 + \lambda_1 z\theta$. After this point, the inventory level continues to fall according to a Poisson process with constant rate λ_2. By Lemma 5.3c, then expected backorder cost during a cycle is given by

$$\begin{aligned} & b_2 \int_0^{L_2} \frac{(\lambda_2 + \lambda_1 z\theta)^{R_2} t^{R_2 - 1}}{(R_2 - 1)!} e^{-(\lambda_2 + \lambda_1 z\theta)t} \frac{\lambda_2(L_2 - t)^2}{2} \overline{F}(-1, \lambda_2(L_2 - t))dt \\ &= b_2 \frac{(\lambda_2 + \lambda_1 z\theta)^{R_2}\lambda_2}{2(R_2 - 1)!} \int_0^{L_2} t^{R_2 - 1} e^{-(\lambda_2 + \lambda_1 z\theta)t}(L_2 - t)^2 dt \end{aligned} \tag{5.29}$$

Transshipment cost over a cycle

Recall that we assume transshipment is unidirectional from location 2 to location 1 in this model. Under the assumption of the **TAP** model, a demand at location 1 results in a transshipment with probability z when location 1 has no stock and location 2 has some stocks. Hence, the indirect fill rate, or transshipment fill rate, at location 1 is approximated as $z\theta\phi$ and transshipment occurs at a rate of $\lambda_1 z\theta\phi$. Hence the average transshipment cost per time unit is given by

$$zt\lambda_1\phi\theta \tag{5.30}$$

Average cost rates

The expected total cost at location 1 over a single cycle is equal to the sum of c_1 and equations (5.23), (5.24), (5.27) and (5.28). The average cost rate at location 1 is obtained by dividing the result by equation (5.14).

Similarly, we derive the expected total cost at location 2 over a single cycle is equal to the sum of c_2 and equations (5.25), (5.26), (5.27) and (5.29). The average cost rate at location 2 is obtained by dividing the result by equation (5.21).

As a whole, the expected total cost rate for the two location system is the sum of the expected total cost rate at location 1, the expected total cost rate at location 2 and the expected transshipment cost rate from equation (5.30).

Direct fill rates

By definitions of θ and ϕ, we derive the direct fill rate α_1 at location 1 as $1-\theta$ and the direct fill rate α_2 at location 2 as ϕ. (Note that a method of estimating ϕ and θ is derived in Section 5.4.5).

Backorder fill rates

Unmet demand at location 1 will be met by a backorder if location 2 refuses the transshipment request. Therefore, the backorder fill rate β_1 at location 1 is derived as $\theta(1-\phi z)$. At location 2, all unmet demand will be backordered. Hence, the backorder fill rate β_2 at location 2 is derived as $1-\phi$.

Indirect fill rate at location 1

As explained above in the section on transshipment cost, the indirect fill rate, or transshipment fill rate, at location 1 is derived as $\phi\theta z$.

5.4.5 Solution algorithm

The decomposition approach and approximations to the system performance measures depend on θ and ϕ. Therefore before we use the expressions derived in the sections above, we need a method of estimating θ and ϕ. In this section we propose an iterative scheme that starts with estimates based on the assumption that the locations are independent. While we offer no proof of convergence, the scheme is found to converge after relatively few iterations in all examples we considered. We first derive expressions for ϕ and θ.

Factor ϕ

We address the question of estimating ϕ, the probability that location 2 has stock, using renewal type arguments as follows.

$$\phi = P[x_2 \geqslant 1] = 1 - P[x_2 \leqslant 0] = 1 - \frac{\text{mean time spent with } x_2 \leqslant 0 \text{ in } a \text{ cycle}}{\text{mean cycle time in equation (5.21)}}$$

As shown in Figure 5.7, when the lead time demand is greater than R_2, the time at which the inventory level reaches zero has an **Erlang** distribution with shape parameter R_2 and scale parameter $(\lambda_2 + \lambda_1 z\theta)L_2$. Since orders can never be delivered during a cycle, the inventory level will at or below zero for the remainder of the cycle. If t is the time at which the inventory level reaches zero, the time for which the inventory level is less than or equal to zero during the cycle is $L_2 - t$. Conditioning on the time at which the inventory level reaches zero, the mean time for which inventory level is less than or equal to zero during a single cycle is estimated as follows

$$\begin{aligned}
&\int_0^{L_2} \frac{(\lambda_2+\lambda_1 z\theta)^{R_2} t^{R_2-1} e^{-(\lambda_2+\lambda_1 z\theta)t}}{(R_2-1)!}(L_2-t)dt \\
=&L_2\int_0^{L_2} \frac{(\lambda_2+\lambda_1 z\theta)^{R_2} t^{R_2-1} e^{-(\lambda_2+\lambda_1 z\theta)t}}{(R_2-1)!}dt - \int_0^{L_2} \frac{(\lambda_2+\lambda_1 z\theta)^{R_2} t^{R_2} e^{-(\lambda_2+\lambda_1 z\theta)t}}{(R_2-1)!}dt \\
=&L_2\{\sum_{j=R_2}^{\infty} \frac{[L_2(\lambda_2+\lambda_1 z\theta)]^j}{j!} e^{-(\lambda_2+\lambda_1 z\theta)L_2}\} - \frac{R_2}{(\lambda_2+\lambda_1 z\theta)}\{\sum_{j=R_2+1}^{\infty} \frac{[L_2(\lambda_2+\lambda_1 z\theta]^j}{j!} e^{-(\lambda_2+\lambda_1 z\theta)L_2}\} \\
=&L_2\overline{F}(R_2-1, L_2(\lambda_2+\lambda_1 z\theta)) - \frac{R_2}{\lambda_2+\lambda_1 z\theta}\overline{F}(R_2, L_2(\lambda_2+\lambda_1 z\theta))
\end{aligned}$$

Thus we can further derive ϕ as

$$=1 - \frac{L_2\overline{F}(R_2-1, L_2(\lambda_2+\lambda_1 z\theta)) - \frac{R_2}{\lambda_2+\lambda_1 z\theta}\overline{F}(R_2, L_2(\lambda_2+\lambda_1 z\theta))}{\frac{(Q_2-R_2)}{(\lambda_2+\lambda_1 z\theta)} + \frac{E[X_2(R_2)]}{(\lambda_2+\lambda_1 z\theta)} + L_2}$$

Factor θ

Now we address the question of estimating the factor θ, the probability that location 1 has no stock. Again, by renewal-type arguments we have

$$\theta = P[x_1 \leqslant 0] = \frac{\text{mean time spent with } x_1 \leqslant 0 \text{ in } a \text{ cycle}}{\text{mean cycle time from equation (5.14)}}$$

As illustrated in Figure 5.6, when the lead time demand is greater than R_1, the time at which the inventory level reaches zero has an **Erlang** distribution with shape parameter R_1 and scale parameter $\lambda_1 L_1$. Orders are never delivered during a cycle, so the inventory level is non-increasing during a cycle. If t is the time at which the inventory level reaches zero, the time for which the inventory level is less than or equal to zero during the cycle is $L_1 - t$. Conditioning on the time at which the inventory level reaches zero, the mean time for which inventory level is less than or equal to zero during a single cycle is as follows

$$\int_0^{L_1} (L_1 - t)\frac{\lambda_1^{R_1} t^{R_1-1} e^{-\lambda_1 t}}{(R_1-1)!}dt = L_1 \int_0^{L_1} \frac{\lambda_1^{R_1} t^{R_1-1} e^{-\lambda_1 t}}{(R_1-1)!}dt - \frac{R_1}{\lambda_1}\int_0^{L_1} \frac{\lambda_1^{R_1+1} t^{R_1} e^{-\lambda_1 t}}{R_1!}dt$$
$$= L_1[\sum_{j=R_1}^{\infty} \frac{(\lambda_1 L_1)^j}{j!} e^{-\lambda_1 L_1}] - \frac{R_1}{\lambda_1}[\sum_{j=R_1+1}^{\infty} \frac{(\lambda_1 L_1)^j}{j!} e^{-\lambda_1 L_1}] = L_1 \overline{F}(R_1 - 1, \lambda_1 L_1) - \frac{R_1}{\lambda_1}\overline{F}(R_1, \lambda_1 L_1)$$

Hence, we derive

$$\theta = \frac{L_1 \overline{F}(R_1 - 1, \lambda_1 L_1) - \frac{R_1}{\lambda_1}\overline{F}(R_1, \lambda_1 L_1)}{\frac{(Q_1 - R_1)}{\lambda_1} + \frac{E[X_1(R_1)]}{\lambda_1} + L_1} \tag{5.31}$$

Factor ω

In order to get the predicted value of I_2 corresponding to the optimal value of z, we need to derive the probability that the inventory level at location 2 is greater than the threshold value I_2. Define:

$$\omega = P[X_2 > I_2] = 1 - P[X_2 \leqslant I_2] = 1 - \frac{\text{mean time spent with } x_2 \leqslant I_2 \text{ in } a \text{ cycle}}{\text{mean cycle time in equation (5.21)}}$$

We need to consider three cases: R_2 greater than I_2, R_2 equal to I_2 and R_2 less than I_2 respectively.

For the case when $R_2 > I_2$, when the lead time demand is greater than $R_2 - I_2$, the time at which the inventory level reaches I_2 has an **Erlang** distribution with shape parameter $R_2 - I_2$ and scale parameter $(\lambda_2 + \lambda_1 z\theta)L_2$. Since inventory level is decreasing during a cycle, if t is the time at which the inventory level reaches I_2, the time for which the inventory level is less than or equal to I_2 during the cycle is $L_2 - t$. Conditioning on the time at which the inventory level reaches I_2, the mean time for which the inventory level is less than or equal to I_2 during a single cycle is estimated as

$$\int_0^{L_2} \frac{(\lambda_2+\lambda_1 z\theta)^{R_2-I_2} t^{R_2-I_2-1} e^{-(\lambda_2+\lambda_1 z\theta)t}}{(R_2-I_2-1)!}(L_2-t)dt$$

$$= L_2\overline{F}\left(R_2-I_2-1,(\lambda_2+\lambda_1 z\theta)L_2\right) - \frac{R_2-I_2}{\lambda_2+\lambda_1 z\theta}\overline{F}\left(R_2-I_2,(\lambda_2+\lambda_1 z\theta)L_2\right)$$

Thus we can further derive ω as

$$1-\frac{L_2\overline{F}\left(R_2-I_2-1,(\lambda_2+\lambda_1 z\theta)L_2\right) - \frac{R_2-I_2}{\lambda_2+\lambda_1 z\theta}\overline{F}\left(R_2-I_2,(\lambda_2+\lambda_1 z\theta)L_2\right)}{\text{mean cycle time from equation (5.21)}} \tag{5.32}$$

For the case when $R_2 = I_2$, as the inventory level starts with R_2 at the beginning of the lead time, the mean time for the inventory level is less than or equal to I_2 is equal to the length of the lead time L_2. Hence, ω is derived as follows

$$1-\frac{L_2}{\text{mean cycle time from equation (5.21)}} \tag{5.33}$$

For the case when $R_2 < I_2$, the inventory level at the start of a cycle, $Q_2 + X_2(R_2)$, maybe above, below or equal to I_2. The time at which the inventory level is less than or equal to I_2 has two parts, one part before the lead time starts and the other part is the lead time. During the first part, the inventory level falls either from I_2 to R_2 or from $Q_2 + X_2(R_2)$ to R_2 depending on whether the inventory level at the beginning of the cycle is greater than I_2 or less than or equal to I_2 respectively. The fall occurs according to a Poisson process with rate λ_2. Thus ω is given by

$$1-\frac{\frac{E\{min(I_2,Q_2+X_2(R_2))\}-R_2}{\lambda_2}+L_2}{\text{mean cycle time from equation (5.21)}} \tag{5.34}$$

Computation algorithm

From the expressions derived for θ and ϕ, we learn that ϕ is dependent on $E[X_2(R_2)]$ and θ, and θ is dependent on $E[X_1(R_1)]$. Meanwhile, $E[X_1(R_1)]$ is dependent on ϕ and $E[X_2(R_2)]$ is dependent on θ. We devise an iterative approach on two pairs $\{\phi,\ \theta\}$ and $\{E[X_1(R_1)],\ \ E[X_2(R_2)]\}$. Specifically, at iteration n we first calculate the estimate $E[X_1(R_1)]^n$ and $E[X_2(R_2)]^n$ based on known values ϕ^n and θ^n. We then calculate the estimate ϕ^{n+1} from $E[X_2(R_2)]^n$ and the estimate θ^{n+1} from $E[X_1(R_1)]^n$. To initialise the iteration we set $\phi^{(0)}$ equal to the fill rate at location 2 and $\theta^{(0)}$ equal to the backorder fill rate at location 1 under the assumption that the locations are independent. We continue the calculation until the values ϕ^n and θ^n converge in the sense that $\max(|\,\phi^n - \phi^{n-1}|, |\theta^n - \theta^{n-1}|) < \epsilon$ where ϵ is pre-defined tolerance.

$$\begin{Bmatrix}\phi^{(0)}\\ \\ \theta^{(0)}\end{Bmatrix}\rightarrow\begin{Bmatrix}E[X_1(R_1)]^{(0)}\\ \\ E[X_2(R_2)]^{(0)}\end{Bmatrix}\rightarrow\begin{Bmatrix}\phi^{(1)}\\ \\ \theta^{(1)}\end{Bmatrix}\rightarrow\begin{Bmatrix}E[X_1(R_1)]^{(1)}\\ \\ E[X_2(R_2)]^{(1)}\end{Bmatrix}\cdots\cdots\rightarrow\text{continue until converage}\begin{Bmatrix}\phi^{*}\\ \theta^{*}\end{Bmatrix}$$

Using the converged estimates ϕ^* and θ^*, we calculate the system performance measures derived in Section 5.4.4.

Optimal total cost algorithm

Given the fixed values of (R_k, Q_k) at location $k, k = 1, 2,$ we perform an exhaustive search to find the value of transshipment agreement probability z, which minimises the approximation to the average total cost rate. Specifically, for each value of z between 0 and 1, we first estimate ϕ and θ using the iterative algorithm described above and then calculate the average total cost rate for the system.

5.4.6 Model verifications

Before we perform a numerical investigation to evaluate the solution algorithm, we need to verify the correctness of our **JAVA** implementation of the approximations to the system performance during a cycle. There are two major reasons to do the verifications. Firstly, the approximations are the result of a detailed mathematical analysis of the situation at each location and we want to verify that we have performed this analysis correctly. Secondly, the approximations involve terms, such as infinite summations and integrals, which are not straightforward to compute and we want to verify that we deal with these terms appropriately in our **JAVA** implementation.

Therefore, we compare approximations to the performance measures at each location from our **JAVA** implementation with the results of simulation of a single location facing a Poisson demand process with a non-constant demand rate. The algorithm used to simulate a single cycle and calculate holding cost, stockout cost, backorder cost and end of cycle inventory level is described in Figure 5.8. We assume that the location uses a $(R,\ Q)$ replenishment policy and there can be at most one outstanding order. Hence the inventory level at the start of a cycle must exceed R. We further assume that the rate of demand is equal to Rate1 at the start of a cycle and changes to Rate2 when the inventory level at the location reaches 0. For location 1 Rate1=λ_1 and Rate2=$\lambda_1(1-\phi z)$ while for location 2 Rate1=$\lambda_2 +$

$\lambda_1 z\theta$ and Rate2$=\lambda_2$.

Let $G(\lambda)$ be a randomly generated instance of the inter-arrival time for a Poisson demand process with rate λ. The cycle ends when the next delivery occurs and the final action is to set the initial inventory level for the next cycle. We also define $t\,$EV, $t\,$Dem and $t\,$Del to be the time of next event, time of next delivery and time of next demand respectively. Initially, simulation during a single cycle kicks off from $i = i_0, t = 0, d\,\text{Rate} = \text{Rate1}, t\,\text{Dem} = G(\text{Rate1})$, $t\,\text{Del} = \infty$. Next, we check the next demand event is either time of next delivery or time of next demand. For the case when the time of next event is a next demand, we update the holding cost and backorder cost for the situations when current inventory level is more than zero and below zero. When current inventory i equals to reorder point R, the time of next delivery $t\,$Del is updated to $t + L$. Then the system continues until the condition $t\,\text{Dem} \geqslant t\,\text{Del}$ has been met.

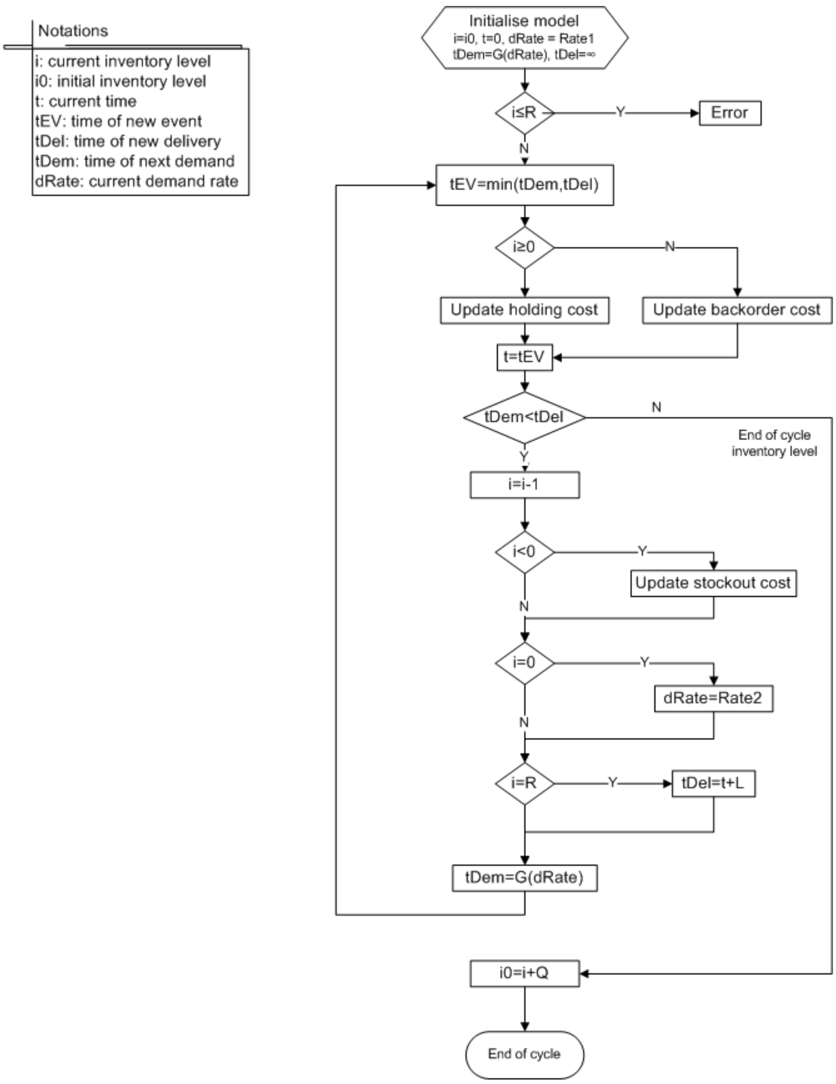

Figure 5.8. Flow chart for the simulation of a single cycle

Verification of the single cycle time period for the TAP model

Let $T_k, C_{h_k}, C_{\hat{b}_k}, C_{b_k}$ and $E[X_k(R_k)]$ be the mean cycle time, the expected average holding cost, the expected average stockout cost, the expected average backorder cost and the expected value of $X_k(R_k)$ during a cycle at location $k\,(k=1,2)$. We use the same parameter set for both locations, namely $L_k=1, h_k=0.8, \hat{b}_k=60, b_k=5, (k=1,2), \phi=0.8$ and $\theta=0.1$.

Tables 5.1 and 5.2 show the results from simulation (columns headed "sim") and the results from our **JAVA** implementation (columns headed "an") for different values of λ_k, R_k, Q_k and z. The simulation results are based on the simulation of 1,000,000 cycles. In each case, the standard errors for simulation are very small and so have not been included in the tables.

					T_1		C_{h_1}		$C_{\hat{b}_1}$		C_{b_1}		$E[X_1(R_1)]$	
λ_1	R_1	Q_1	z	ϕ	sim	an	sim	an	sim	an	sim	an	sim	an
10	10	40	0.0	0.8	4.00	4.00	16.44	16.44	18.79	18.76	0.26	0.26	0.00	0.00
10	10	40	0.3	0.8	4.03	4.03	16.54	16.54	14.15	14.22	0.20	0.20	0.30	0.30
10	10	40	0.7	0.8	4.07	4.07	16.67	16.68	8.11	8.11	0.11	0.11	0.70	0.70
10	10	40	1.0	0.8	4.10	4.10	16.78	16.78	3.65	3.66	0.05	0.05	1.00	1.00
10	10	60	0.0	0.8	6.00	6.00	24.43	24.43	12.51	12.51	0.17	0.17	0.00	0.00
10	10	60	0.3	0.8	6.03	6.03	24.54	24.53	9.45	9.50	0.13	0.13	0.31	0.30
10	10	60	0.7	0.8	6.07	6.07	24.68	24.68	5.43	5.44	0.08	0.08	0.71	0.70
10	10	60	1.0	0.8	6.10	6.10	24.79	24.79	2.46	2.46	0.03	0.03	1.01	1.00
10	20	40	0.0	0.8	4.00	4.00	24.40	24.40	0.04	0.04	0.00	0.00	10.00	10.00
10	20	40	0.3	0.8	4.00	4.00	24.40	24.40	0.03	0.03	0.00	0.00	10.00	10.00
10	20	40	0.7	0.8	4.00	4.00	24.40	24.40	0.02	0.02	0.00	0.00	10.00	10.00
10	20	40	1.0	0.8	4.00	4.00	24.40	24.40	0.01	0.01	0.00	0.00	10.00	10.00
10	10	60	0.0	0.8	6.00	6.00	24.43	24.43	12.51	12.51	0.17	0.17	0.00	0.00
10	10	60	0.3	0.8	6.03	6.03	24.54	24.53	9.45	9.50	0.13	0.13	0.31	0.30
10	10	60	0.7	0.8	6.07	6.07	24.68	24.68	5.43	5.44	0.08	0.08	0.71	0.70
10	10	60	1.0	0.8	6.10	6.10	24.79	24.79	2.46	2.46	0.03	0.03	1.01	1.00
20	10	60	0.0	0.8	3.00	3.00	17.13	17.13	200.27	200.17	4.59	4.58	-10.00	-10.00
20	10	60	0.3	0.8	3.12	3.12	18.02	18.02	146.30	146.28	3.35	3.35	-7.60	-7.60
20	10	60	0.7	0.8	3.28	3.28	19.23	19.23	80.59	80.56	1.85	1.84	-4.40	-4.40
20	10	60	1.0	0.8	3.40	3.40	20.15	20.12	35.34	35.32	0.81	0.81	-1.99	-1.99
20	10	80	0.0	0.8	4.00	4.00	24.95	24.95	150.04	150.13	3.43	3.44	-9.99	-10.00
20	10	80	0.3	0.8	4.12	4.12	25.87	25.86	110.68	110.77	2.53	2.54	-7.59	-7.60
20	10	80	0.7	0.8	4.28	4.28	27.09	27.09	61.69	61.74	1.41	1.41	-4.39	-4.40
20	10	80	1.0	0.8	4.40	4.40	28.02	28.02	27.30	27.30	0.63	0.63	-1.99	-1.99
20	20	60	0.0	0.8	3.00	3.00	24.46	24.46	35.62	35.53	0.37	0.37	-0.00	0.00
20	20	60	0.3	0.8	3.02	3.02	24.60	24.60	26.87	26.81	0.28	0.28	0.42	0.43
20	20	60	0.7	0.8	3.05	3.05	24.80	24.80	15.41	15.37	0.16	0.16	0.99	1.00
20	20	60	1.0	0.8	3.07	3.07	24.95	24.95	6.98	6.94	0.07	0.07	1.42	1.42
20	20	80	0.0	0.8	4.00	4.00	32.45	32.44	26.58	26.65	0.27	0.28	0.01	0.00
20	20	80	0.3	0.8	4.02	4.02	32.60	32.59	20.07	20.14	0.21	0.21	0.43	0.43
20	20	80	0.7	0.8	4.05	4.05	32.80	32.80	11.53	11.58	0.12	0.12	1.00	1.00
20	20	80	1.0	0.8	4.07	4.07	32.96	32.95	5.21	5.23	0.05	0.05	1.42	1.42

Table 5.1. Verification results for **TAP** model at location 1

						T_2		C_{h_2}		$C_{\hat{b}_2}$		C_{b_2}		$E[X_2(R_2)]$	
λ_1	λ_2	R_2	Q_2	z	θ	sim	an	sim	an	sim	an	sim	an	sim	an
10	10	10	40	0.0	0.1	4.00	4.00	16.44	16.44	18.79	18.76	0.26	0.26	0.00	0.00
10	10	10	40	0.3	0.1	3.89	3.89	16.22	16.22	21.29	21.26	0.30	0.30	-0.26	-0.26
10	10	10	40	0.7	0.1	3.75	3.75	15.94	15.94	24.85	24.83	0.36	0.36	-0.59	-0.59
10	10	10	40	1.0	0.1	3.65	3.65	15.73	15.73	27.70	27.68	0.41	0.41	-0.83	-0.83
10	10	20	40	0.0	0.1	4.00	4.00	24.40	24.40	0.04	0.04	0.00	0.00	10.00	10.00
10	10	20	40	0.3	0.1	3.88	3.88	24.16	24.16	0.06	0.06	0.00	0.00	9.70	9.70
10	10	20	40	0.7	0.1	3.74	3.74	23.84	23.84	0.10	0.10	0.00	0.00	9.30	9.30
10	10	20	40	1.0	0.1	3.64	3.64	23.60	23.60	0.14	0.13	0.00	0.00	9.00	9.00
10	30	20	90	0.0	0.1	3.00	3.00	28.94	28.93	200.70	200.99	3.33	3.33	-9.99	-10.00
10	30	20	90	0.3	0.1	2.98	2.97	28.76	28.76	206.31	206.63	3.46	3.46	-10.18	-10.20
10	30	20	90	0.7	0.1	2.94	2.94	28.53	28.53	213.89	214.14	3.64	3.64	-10.44	-10.46
10	30	20	90	1.0	0.1	2.92	2.92	28.36	28.36	219.48	219.76	3.77	3.77	-10.63	-10.64
10	30	40	90	0.0	0.1	3.00	3.00	44.41	44.40	1.90	1.91	0.01	0.01	10.02	10.00
10	30	40	90	0.3	0.1	2.97	2.97	44.17	44.16	2.19	2.20	0.01	0.01	9.72	9.70
10	30	40	90	0.7	0.1	2.93	2.93	43.85	43.84	2.65	2.65	0.01	0.01	9.32	9.30
10	30	40	90	1.0	0.1	2.90	2.90	43.61	43.60	3.03	3.03	0.02	0.02	9.02	9.01
20	10	10	40	0.0	0.1	4.00	4.00	16.44	16.44	18.79	18.76	0.26	0.26	0.00	0.00
20	10	10	40	0.3	0.1	3.78	3.78	16.01	16.01	23.94	23.92	0.35	0.35	-0.51	-0.51
20	10	10	40	0.7	0.1	3.53	3.53	15.46	15.46	31.70	31.67	0.48	0.48	-1.14	-1.14
20	10	10	40	1.0	0.1	3.37	3.37	15.06	15.06	38.02	38.04	0.60	0.60	-1.57	-1.57
20	10	20	40	0.0	0.1	4.00	4.00	24.40	24.40	0.04	0.04	0.00	0.00	10.00	10.00
20	10	20	40	0.3	0.1	3.77	3.77	23.92	23.92	0.09	0.09	0.00	0.00	9.40	9.40
20	10	20	40	0.7	0.1	3.51	3.51	23.28	23.28	0.21	0.20	0.00	0.00	8.61	8.60
20	10	20	40	1.0	0.1	3.33	3.33	22.80	22.80	0.36	0.35	0.00	0.00	8.01	8.00
20	30	20	90	0.0	0.1	3.00	3.00	28.94	28.93	200.70	200.99	3.33	3.33	-9.99	-10.00
20	30	20	90	0.3	0.1	2.95	2.95	28.59	28.59	212.02	212.26	3.59	3.59	-10.38	-10.39
20	30	20	90	0.7	0.1	2.88	2.88	28.14	28.14	226.93	227.24	3.95	3.96	-10.88	-10.89
20	30	20	90	1.0	0.1	2.84	2.84	27.82	27.81	238.07	238.41	4.23	4.23	-11.23	-11.25
20	30	40	90	0.0	0.1	3.00	3.00	44.41	44.40	1.90	1.91	0.01	0.01	10.02	10.00
20	30	40	90	0.3	0.1	2.94	2.94	43.93	43.92	2.53	2.53	0.01	0.01	9.42	9.40
20	30	40	90	0.7	0.1	2.87	2.87	43.29	43.29	3.61	3.62	0.02	0.02	8.63	8.61
20	30	40	90	1.0	0.1	2.81	2.81	42.82	42.81	4.66	4.69	0.03	0.03	8.03	8.01

Table 5.2. Verification results for **TAP** model at location 2

We would use the percentage difference to compare the results of the simulation model and the **JAVA** implementation. However, as some of the values are close to zero, the percentage difference can be large even though the values are close. We will use the percentage difference when the estimate values are greater than 1 and the difference otherwise. The estimates of mean cycle time are within 0.041% and 0.036% of each other at location 1 and 2 respectively. The estimates of expected holding cost are within 0.018% and 0.025% of each other respectively. For values greater than 1, the estimates of expected stockout cost are within 0.608% and 0.702% of each other for location 1 and location 2 respectively. For smaller values, the estimates are within 0.00 and 0.01 of each other for locations 1 and 2 respectively. For values greater than 1, the estimates of expected backorder cost are within 0.112% and 0.081% of each other for location 1 and location 2 respectively. For smaller values, the estimates are within 0.01 and 0.00 of each other locations 1 and 2 respectively.

Verdict: The verification results above reveal strong evidence that our **JAVA** implementation of the analytic approximations to the performance measures for the **TAP** model is accurate. Hence we are confident that we can use our **JAVA** implementation in a numerical investigation to evaluate the proposed solution algorithm.

5.4.7 Numerical experiments

The purpose of our numerical experiments is to determine whether the proposed solution algorithm can be used to find the optimal holdout policy for the two-location inventory system with unidirectional transshipment. From the numerical results in Chapter 4, we learned that the parameters b_k and $\hat{b_k}$ can have a significant effect on the optimal transshipment policy and the optimal average total cost. Therefore, we will investigate the relationship between the parameters $\hat{b_1}$ and $\hat{b_2}$ to find the optimal average total cost. To demonstrate these optimal average total cost features, we choose two snapshot numerical test sets.

In the following result tables, z denotes the transshipment agreement probability, C denotes the average total cost for the two-location system and C_t denotes the average transshipment cost from location 2 to location 1. Then C_k, C_{o_k}, C_{h_k}, $C_{\hat{b_k}}$, C_{b_k} denote the average cost, average order cost, average holding cost, average stockout cost and average backorder cost at location $k\,(k=1,2)$ respectively. In addition, α_k, β_k denotes the fill rate and backorder fill rate at location $k\,(k=1,2)$ and γ the transshipment fill rate from location 2 to location 1. Since $\alpha_1+\beta_1+\gamma=1$ and $\alpha_2+\beta_2=1$. We only provide the results for fill rate α_1, α_2 and γ in the result tables.

For the simulation implementation, in order to make a good balance between computational time and accuracy, we performed 500 independent simulation runs of 50,000 time units with a warm-up period of 500 time units. In all cases, the standard errors are small and so are not included in the result tables.

Snapshot numerical test set 1

To demonstrate how parameters $\hat{b_1}$ or $\hat{b_2}$ affect the optimal average total cost, we use the 12 snapshots described in Table 5.3 corresponding to 12 different combinations on $\hat{b_1}$ and $\hat{b_2}$. In each snapshot, the other parameters are given values as follows: $\lambda_1=20$, $c_1=10$, $h_1=1$, $b_1=25$, $R_1=20$, $Q_1=40$, $L_1=1$, $\lambda_2=5$, $c_2=10$, $h_2=0.5$, $b_2=70$, $R_2=5$, $Q_2=10$, $L_2=1$ and $t=10$. Snapshot numerical test set 1 reflects the situation in which there is a higher demand rate at location 1 than location 2. Hence stockouts are more likely to occur at location 1 and transshipment is likely to be attractive as there is a less dense customer demand at location 2. As well as the values of $\hat{b_1}$ and $\hat{b_2}$, Table 5.3 also provides the optimal transshipment agreement probability, z^*, for each snapshot.

In the first 6 snapshots the value of $\hat{b_1}$ increases relative to $\hat{b_2}$ and the system is more likely to favour transshipment, hence z^* increases accordingly. Meanwhile, for the last 6 snapshots, the value of $\hat{b_2}$ increases relative to $\hat{b_1}$ and the transshipment is less attractive, hence z^* decreases accordingly.

s/n	$\hat{b_1}$	$\hat{b_2}$	z^*	s/n	$\hat{b_1}$	$\hat{b_2}$	z^*
1	17	20	0.0	7	20	15	0.9
2	18	20	0.0	8	20	16	0.8
3	19	20	0.2	9	20	17	0.7
4	20	20	0.5	10	20	18	0.6
5	21	20	0.7	11	20	19	0.5
6	22	20	0.9	12	20	20	0.5

Table 5.3. Summary of the snapshot test set 1 of **TAP** model

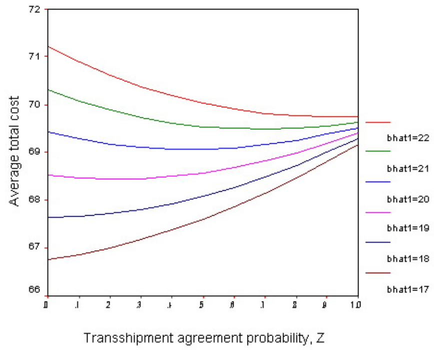

Figure 5.9. Test set 1 of **TAP** approximation on $\hat{b_1}$

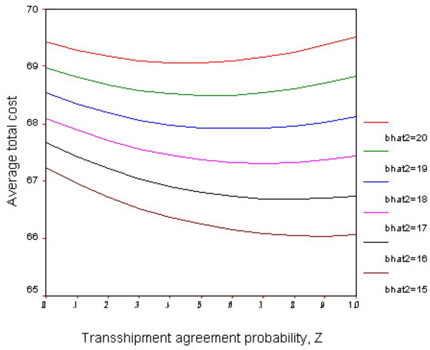

Figure 5.10. Test set 1 of **TAP** approximation on $\hat{b_2}$

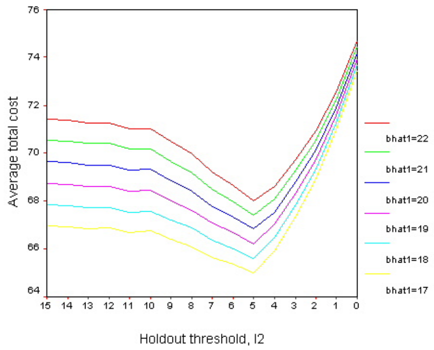

Figure 5.11. Test set 1 of original **TAP** simulation on $\hat{b_1}$

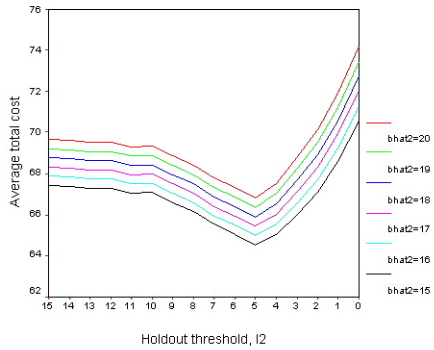

Figure 5.12. Test set 1 of original **TAP** simulation on $\hat{b_2}$

Figures 5.9 and 5.10 show the **TAP** models' estimate of the average total cost against transshipment agreement probability for different values of $\hat{b_1}$ and $\hat{b_2}$. Figure 5.9 suggests that, for the snapshots 1 to 6, the optimal pooling policy is ranging from no-pooling policy to a partial pooling policy which is very close to the complete pooling policy with the increases on $\hat{b_1}$ from 17 to 22. Partial pooling is most clearly the optimal pooling policy when $\hat{b_1}$ is equal to 19 and 20 respectively. For snapshots 7 to 12, Figure 5.10 suggests that partial pooling is always the optimal pooling policy when the value of $\hat{b_2}$ is ranging from 15 to 20. From our observations, we find that the **TAP** model predicts that partial pooling can deliver cost savings over the simple policies of no pooling and complete pooling under some situations.

Figures 5.11 and 5.12 show average total cost for a range of holdout transshipment policies based on simulations of the original system for each snapshot. It is apparent from the figures that partial pooling is the optimal transshipment policy for all snapshots. It is interesting to note that the optimal holdout value is the same for all snapshots ($I_2 = 5$) and that the actual cost savings from partial pooling are up to 5% compared to no pooling (I_2=15) and up to 13% compared to complete pooling (I_2=0). These findings appear contrary to the results of the **TAP** model. However, the **TAP** model could still be of use if it correctly predicts the optimal holdout value or predicts a holdout value which results in average total cost which is comparable to optimal.

z	C	C_1	C_{o_1}	C_{h_1}	$C_{\hat{b}_1}$	C_{b_1}	C_2	C_{o_2}	C_{h_2}	$C_{\hat{b}_2}$	C_{b_2}	C_t	α_1	γ	α_2
0	69.42	46.13	5.00	20.61	17.76	2.76	23.29	5.00	2.80	8.77	6.72	0.00	0.96	0.00	0.91
0.1	69.27	44.23	4.98	20.67	16.09	2.50	24.24	5.08	2.76	9.26	7.14	0.80	0.96	0.00	0.91
0.2	69.17	42.37	4.96	20.73	14.44	2.24	25.20	5.16	2.72	9.75	7.57	1.59	0.96	0.01	0.90
0.3	69.09	40.55	4.94	20.79	12.83	1.99	26.18	5.24	2.68	10.25	8.01	2.36	0.96	0.01	0.90
0.4	69.06	38.76	4.92	20.85	11.24	1.74	27.18	5.31	2.65	10.76	8.46	3.12	0.96	0.01	0.89
0.5*	69.06	37.01	4.90	20.91	9.69	1.50	28.18	5.39	2.61	11.27	8.92	3.87	0.96	0.02	0.89
0.6	69.09	35.29	4.89	20.97	8.17	1.27	29.20	5.46	2.58	11.78	9.38	4.59	0.96	0.02	0.87
0.7	69.15	33.61	4.87	21.03	6.67	1.04	30.23	5.53	2.54	12.30	9.86	5.31	0.96	0.03	0.88
0.8	69.25	31.97	4.85	21.10	5.21	0.81	31.27	5.60	2.51	12.82	10.34	6.01	0.96	0.03	0.87
0.9	69.37	30.35	4.83	21.16	3.78	0.59	32.32	5.67	2.48	13.35	10.83	6.70	0.96	0.03	0.87
1.0	69.52	28.78	4.82	21.22	2.37	0.37	33.38	5.74	2.45	13.87	11.32	7.37	0.96	0.04	0.86

Table 5.4. Snapshot 4 in test set 1 of **TAP** model

Table 5.4 provides the result of snapshot 4 from Table 5.3 in detail and suggests that the optimal pooling policy is the partial pooling policy corresponding to transshipment agreement probability equal to 0.5. However, it is worth noting that the predicted improvements on the cost from the optimal partial pooling policy are only 0.67% and 0.52% with respect to complete pooling ($z=1$) and no pooling ($z=0$) respectively.

We can derive the optimal holdout threshold value at location 2 suggested by the model by searching on the value of I_2 for which the optimal transshipment agreement probability $z \simeq P[X_2 > I_2 | X_2 \geqslant 1]$. In Table 6.5, we provide a comparison of the optimal holdout value I_2 as predicted by the **TAP** model and as determined by simulation of the original system. We also provide the actual total cost for the predicted I_2 calculated by simulation of the original system and optimal total cost also from simulation for each snapshot. These values are shown in columns C_{TAP} and C_{SIM} respectively.

For snapshot 4 in test set 1, the optimal z is equal to 0.5, the optimal holdout value of I_2 is predicted as 5 which is equal to the optimal holdout threshold value determined by the simulation of the original system (see Figures 5.11 and 5.12). Hence, we can say that the **TAP** model correctly predicts the optimal holdout value for snapshot 4. However, for other snapshots in test set 1, there are gaps between the predicted I_2 and optimal I_2.

		I_2						I_2			
s/n	z^*	TAP	SIM	C_{TAP}	C_{SIM}	s/n	z^*	TAP	SIM	C_{TAP}	C_{SIM}
1	0.0	11	5	66.67	65.00	7	0.9	1	5	68.58	64.48
2	0.0	11	5	67.54	65.60	8	0.8	2	5	67.69	64.95
3	0.2	8	5	67.62	66.20	9	0.7	3	5	67.13	65.41
4	0.5	5	5	66.81	66.81	10	0.6	4	5	66.53	65.88
5	0.7	3	5	69.27	67.41	11	0.5	5	5	66.34	66.34
6	0.9	1	5	72.55	68.01	12	0.5	5	5	66.81	66.81

Table 5.5. Comparison of optimal z, predicted and optimal I_2 for snapshot test set 1

Moreover, when we compare costs C_{TAP} and C_{SIM} in Table 6.5 for each snapshot in test set 1, the difference between the actual total cost for the predicted I_2 and the optimal total cost is ranging from 0.00% to 3.92%.

Snapshot numerical test set 2

Similarly, we use the 14 snapshots described in Table 5.6 corresponding to 14 different combinations on $\hat{b_1}$ and $\hat{b_2}$. In each snapshot, the other parameters are given values as follows: $\lambda_1 = 10$, $c_1 = 10$, $h_1 = 0.5$, $b_1 = 25$, $R_1 = 1$, $Q_1 = 40$, $L_1 = 1$, $\lambda_2 = 25$, $c_2 = 10$, $h_2 = 0.5$, $b_2 = 340$, $R_2 = 25$, $Q_2 = 40$, $L_2 = 1$ and $t = 10$. Snapshot numerical test set 2 reflects the situation in which there is a higher demand rate at location 2 than location 1. Hence, stockout is less likely to occur at location 1 and transshipment is less likely to be attractive as there is a more dense customer demand at location 2. As well as the values of $\hat{b_1}$ and $\hat{b_2}$, Table 5.6 also provides the optimal transshipment agreement probability, z^* for each snapshot.

In the first 7 snapshots the value of $\hat{b_1}$ increases relative to $\hat{b_2}$ and the system is more likely to favour transshipment, hence z^* increases accordingly. Meanwhile, for the last 7 snapshots, the value of $\hat{b_2}$ increases relative to $\hat{b_1}$ and the transshipment is less attractive, hence z^* decreases accordingly.

s/n	$\hat{b_1}$	$\hat{b_2}$	z^*	s/n	$\hat{b_1}$	$\hat{b_2}$	z^*
1	17	20	0.2	8	20	14	0.8
2	18	20	0.3	9	20	15	0.7
3	19	20	0.4	10	20	16	0.6
4	20	20	0.5	11	20	17	0.6
5	21	20	0.6	12	20	18	0.6
6	22	20	0.7	13	20	19	0.5
7	23	20	0.9	14	20	20	0.5

Table 5.6. Summary of the snapshot test set 2 of **TAP** model

We use Figures 5.13 and 5.14 to show the **TAP** model's estimate of the average total cost against transshipment agreement probability for different values of $\hat{b_1}$ and $\hat{b_2}$. Figure 5.13 suggests that, for the snapshots 1 to 7, partial pooling is the optimal pooling policy when $\hat{b_1}$ is ranging from 17 to 23. Meanwhile, for snapshots 8-14, Figure 5.14 suggests that the partial pooling is always the optimal pooling policy when $\hat{b_2}$ is ranging from 13 to 20.

Figure 5.15 and 5.16 show average total cost for a range of holdout transshipment policies based on simulations of the original system for each snapshot. It is apparent from the figures that partial pooling is the optimal transshipment policy for all snapshots. It is interesting to note that the optimal holdout value is the same for all snapshots (I_2=25) and that the actual cost savings from partial pooling are up to 12.03% compared to no pooling (I_2=65) and up to 27.36% compared to complete pooling (I_2=0).

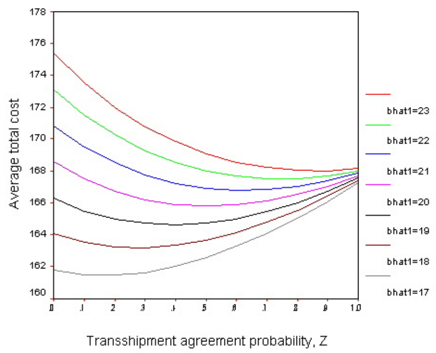

Figure 5.13. Test set 2 of **TAP** approximation on $\hat{b_1}$

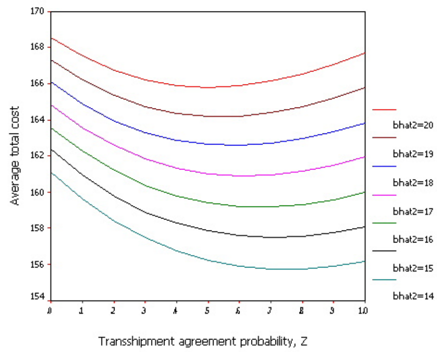

Figure 5.14. Test set 2 of **TAP** approximation on $\hat{b_2}$

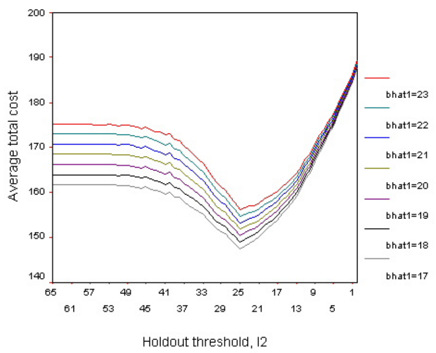

Figure 5.15. Test set 2 of original **TAP** simulation on $\hat{b_1}$

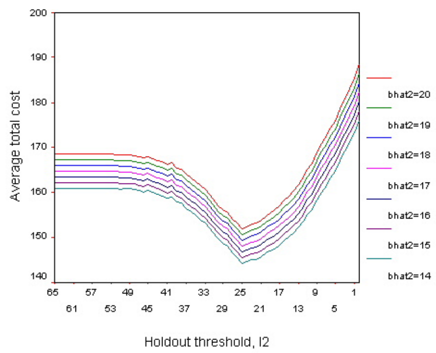

Figure 5.16. Test set 2 of original **TAP** simulation on $\hat{b_2}$

In Table 5.7, we provide the result of snapshot 4 from Table 5.6 in detail which suggests that the optimal pooling policy is the partial pooling policy corresponding to transshipment agreement probability equal to 0.5. However, the predicted improvement on the cost from the optimal partial pooling are only 1.15% and 1.67% with respect to complete pooling ($z =$

1) and no pooling ($z=0$) respectively.

z	C	C_1	C_{o_1}	C_{h_1}	$C_{\hat{b}_1}$	C_{b_1}	C_2	C_{o_2}	C_{h_2}	$C_{\hat{b}_2}$	C_{b_2}	C_t	α_1	γ	α_2
0	168.58	79.63	2.50	6.26	45.15	25.72	88.95	6.25	10.32	24.87	47.51	0.00	0.77	0.00	0.95
0.1	167.50	71.72	2.45	6.45	40.02	22.80	93.70	6.30	10.22	26.31	50.87	2.09	0.78	0.02	0.95
0.2	166.74	64.18	2.40	6.65	35.13	20.01	98.48	6.35	10.12	27.76	54.24	4.08	0.78	0.04	0.94
0.3	166.20	57.00	2.35	6.84	30.46	17.35	103.22	6.40	10.03	29.17	57.62	5.98	0.79	0.06	0.94
0.4	165.91	50.15	2.31	7.04	25.99	14.81	107.97	6.44	9.94	30.59	61.00	7.79	0.79	0.08	0.94
0.5*	165.81	43.60	2.26	7.24	21.73	12.37	112.68	6.49	9.86	31.97	64.36	9.53	0.80	0.10	0.94
0.6	165.90	37.34	2.22	7.43	17.64	10.05	117.37	6.53	9.78	33.34	67.72	11.19	0.80	0.11	0.93
0.7	166.14	31.34	2.18	7.62	13.72	7.82	122.01	6.57	9.70	34.68	71.07	12.79	0.80	0.13	0.93
0.8	166.53	25.60	2.14	7.82	9.96	5.67	126.62	6.61	9.63	36.00	74.39	14.32	0.81	0.14	0.93
0.9	167.06	20.09	2.11	8.01	6.35	3.62	131.19	6.64	9.55	37.29	77.70	15.78	0.81	0.16	0.93
1.0	167.71	14.80	2.07	8.20	2.88	1.64	135.72	6.68	9.49	38.57	80.98	17.19	0.81	0.17	0.92

Table 5.7. Snapshot 4 in test set 2 of **TAP** model

Following the same prediction approach on the holdout threshold value I_2 in snapshot test set 1, in Table 5.8, we provide a comparison of the optimal holdout value I_2 as predicted by the **TAP** model and as determined by simulation of the original system. We also provide the actual total cost for the predicted I_2 calculated by simulation of the original system and optimal total cost also from simulation for each snapshot. These values are shown in column C_{TAP} and C_{SIM} respectively.

		I_2						I_2			
s/n	z^*	TAP	SIM	C_{TAP}	C_{SIM}	s/n	z^*	TAP	SIM	C_{TAP}	C_{SIM}
1	0.2	28	25	150.69	147.49	9	0.8	9	25	158.41	144.32
2	0.3	26	25	150.07	148.97	10	0.7	12	25	155.27	145.59
3	0.4	24	25	150.92	150.45	11	0.6	16	25	152.29	146.59
4	0.5	20	25	154.85	151.93	12	0.6	16	25	153.80	148.12
5	0.6	16	25	159.35	153.41	13	0.6	16	25	155.32	149.39
6	0.7	12	25	165.07	154.89	14	0.5	20	25	153.45	150.66
7	0.9	5	25	177.34	156.37	15	0.5	20	25	154.85	151.93

Table 5.8. Summary of the snapshot test set 2 of **TAP** model

For snapshot 4 in test set 2, the optimal z is equal to 0.5, the optimal holdout value of I_2 is predicted as 20 which is close to the optimal holdout threshold value determined by the simulation of the original system (see Figure 5.15 and 5.16). For other snapshots in the test set 2, there are also differences between the predicted I_2 and the optimal I_2.

Moreover, we compare cost C_{TAP} and C_{SIM} in Table 5.8 for each snapshot in test set 2, the difference between the actual total cost for the predicted I_2 and the optimal total cost for the optimal I_2 is ranging between 0.31% to 13.41% respectively.

5.4.8 Conclusions

We have extended the approach of Chapter 4 by decomposing the two-location system into two independent locations with non-constant demand rates. By this new approach, we have been able to derive explicit expressions for a range of system performance measures under the assumptions of the decomposition approach. There is some evidence from numerical results that the **TAP** model more closely reflects the behaviour of the system than the approximations from Chapter 4.

In addition, the type of decomposition approach has a good framework which could be extended to other systems with more than two locations. For those multi-location systems, we only need to derive explicit expressions for a range of system performance measures by reflecting transshipment interactions among them. However, we should emphasize that this approach is based on the assumption that there is at most one outstanding order any time.

The analysis of the predicted holdout threshold values I_2 from the **TAP** model for two snapshot test sets show that the predicted values are only close to optimal for a few snapshots. For other snapshots, there can be large differences between the predicted I_2 and the optimal I_2. Furthermore, the comparison of the actual total cost of the policy predicted by the **TAP** model and the optimal total cost shows that the predicted policy can lead to costs that are significantly higher than optimal. Hence, we conclude that the **TAP** model does not provide reliable method of determining an optimal holdout transshipment policy in general.

One reason which might explain the differences between the model's predictions and simulations is the lack of accuracy from the transshipment agreement probability. So far, we only use the transshipment agreement probability to model the degree of transshipment from location 2 to location 1. The numerical results suggest this approximation is not accurate enough to capture the true system performance. Hence, in Section 5.5, we will develop a new approximation approach with the similar decomposition method we used in Section 5.4 and introduce an explicit holdout transshipment decision variable to replace the transshipment agreement probability. Hopefully, this approach will lead to a better approximation model.

5.5 Explicit holdout transshipment policy (HOT) model

As before the aim of this new model is to find an optimal holdout transshipment policy for a two-location system with unidirectional transshipment. Again, we decompose the two-location system into two independent single-depot systems with non-constant demand rates. However, in this approach we try to model the holdout threshold at location 2 explicitly rather than using the transshipment agreement probability. We call this model the **HOT** model. In theory, when the holdout threshold is equal to zero, the holdout transshipment policy is equivalent to a complete pooling; when it is infinite, the holdout transshipment policy is equivalent to no pooling; and when it is between these two extremes, the holdout transshipment policy represents a partial pooling policy.

Following a similar decomposition approach as we applied in the analysis of the **TAP** model, we derive a range of system performance measures for the **HOT** model. Firstly, we provide the specific assumptions for the **HOT** model in Section 5.5.1. The distribution of the inventory level at the end of a cycle and mean cycle time during a cycle at location 1 and location 2 are derived in Section 5.5.2 and Section 5.5.3 respectively.

Subsequently, we provide a range of performance measures including average cost rate and direct fill rate at location 1 and location 2 in Section 5.5.4. In Section 5.5.5, we present an algorithm that can be used to find an approximation to the optimal transshipment policy.

In Section 5.5.6, we verify the correctness of our approximations to the performance measures at the locations using simulation of a single location with non-constant demand rate. Finally, a series of numerical experiments designed to evaluate the **HOT** model are presented in Section 5.5.7.

5.5.1 Assumptions

In addition to the general assumptions of Section 5.2, we define following assumptions for the **HOT** model. We use the decision variable I_2 to represent the holdout threshold at location 2. Define X_k to be the inventory level at location k $(k = 1, 2)$. Hence, demand that occurs at location 1 is satisfied from local stock if $X_1 > 0$; by transshipment from location 2 if $X_1 \leqslant 0$

and $X_2 > I_2$; and by backorder at location 1 if $X_1 \leqslant 0$ and $X_2 \leqslant I_2$. While at location 2, demand is satisfied from local stock if $X_2 > 0$ and by backorder at location 2 if $X_2 \leqslant 0$. If $I_2 = 0$, the holdout transshipment policy is equivalent to complete pooling; if I_2 is large enough, the holdout transshipment policy approximates no pooling; within this range, the holdout transshipment policy corresponds to partial pooling.

We introduce $\omega = P[X_2 > I_2]$ representing the probability that location 2 meets a transshipment request from location 1, $\theta = P[X_1 \leqslant 0]$ representing the probability that demand at location 1 results in a transshipment request and $\phi = P[X_2 \geqslant 1]$ representing the probability that location 2 has stock. We use these probabilities to decompose the two-location system in the following manner.

When the inventory level at location 1 is greater than zero, location 1 is able to meet all of its demands from its local stock. Hence, the inventory level at location 1 falls due to a Poisson demand process with rate λ_1. When the inventory level of location 1 reaches R_1, an order of Q_1 items is placed which will arrive after a fixed time L_1. When the inventory level at location 1 is less than or equal to zero, location 1 makes transshipment requests at a rate of λ_1 per time unit. The probability that location 2 meets a transshipment request is ω. Hence, the probability that a transshipment request at location 1 results in a backorder is $1 - \omega$. Therefore, when the inventory level at location 1 is less than or equal to zero, inventory level at location 1 falls due to a Poisson demand process with rate $\lambda_1(1 - \omega)$. The situation at location 1 is illustrated in Figure 5.17.

Meanwhile, when the inventory level at location 2 is greater than I_2, location 2 meets all of its demand and all transshipment requests from its local stock. Since location 1 only makes transshipment request when it is out of stock, transshipment requests occur at an average rate of $\lambda_1\theta$ per time unit. Hence, when the inventory level at location 2 is greater than I_2, the inventory level at location 2 falls due to a Poisson demand process with rate $\lambda_2 + \lambda_1\theta$. When the inventory level at location 2 is less than or equal to I_2, location 2 meets all of its demand (either from local stock or by backorders), but refuses all transshipment requests from location 1. Hence, the inventory level at location 2 falls due to a Poisson demand process with rate λ_2. When the inventory level at location 2 reaches R_2, an order of Q_2 items is placed and will arrive after a fixed time L_2. There are three cases to consider depending on whether I_2 is less than, equal to or greater than R_2. The situation at location 2 from illustrated in Figure 5.18 to 5.21.

Hence, using the probabilities ω and θ, we decompose the two-location system into two independent single-depot systems with non-constant demand rates. By analysing the time between the delivery of successive orders (which we refer to as a cycle) at each location, we can derive approximation to performance measures of the two-location system and so find an approximation to the optimal holdout transshipment policy. Unlike our previous approximation models which used the transshipment agreement probability to approximate the holdout threshold, the **HOT** model predicts the optimal threshold value directly.

5.5.2 Distribution of inventory level & mean cycle time for location 1

Under the assumption that there is never more than one outstanding order at each location, whenever an order arrives at location 1, the inventory level is restored to a level greater than R_1. Figure 5.17 depicts the inventory level as a function of time during a cycle for location 1. Comparing Figure 5.6 at location 1 in the **TAP** model and Figure 5.17, we note that the situation at location 1 under the assumption of the **HOT** model is equivalent to the situation at location 1 under the assumptions of the **TAP** model with $z\phi$ replaced by ω. Hence, we can deduce expressions for the distribution of the end of cycle inventory level and the mean cycle time at location 1 for the **HOT** model.

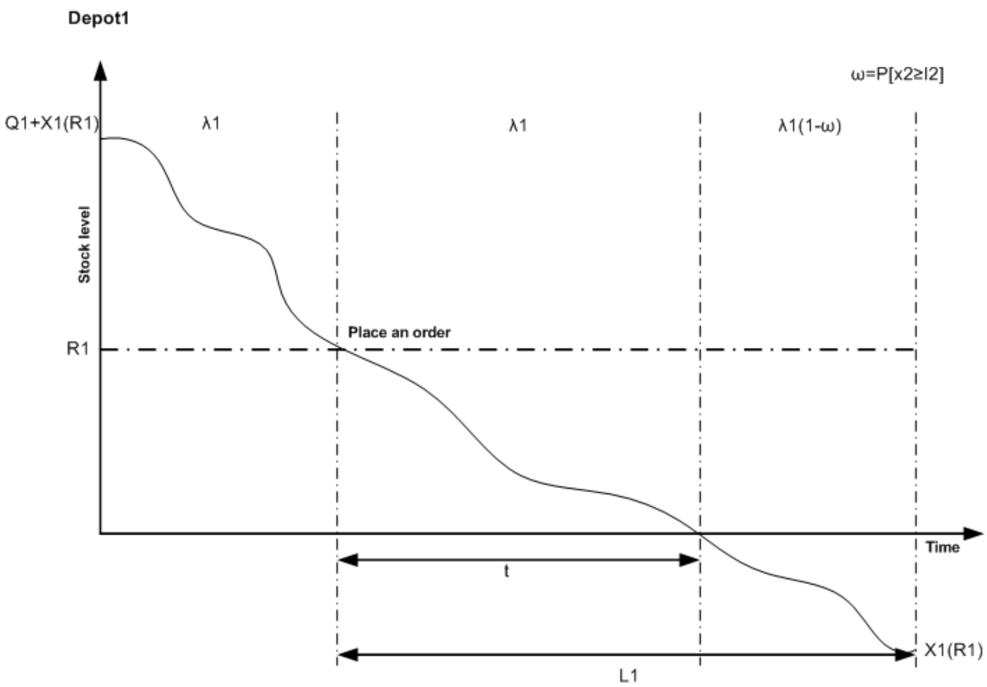

Figure 5.17. Inventory level at location 1 as a function of time during a single cycle

Distribution of $X_1(R_1)$

From analysis of the **TAP** model, the probability distribution of $X_1(R_1)$ is as follows

For $1 \leqslant j \leqslant R_1$

$$P[X_1(R_1) = j] = \frac{(\lambda_1 L_1)^{R_1 - j}}{(R_1 - j)!} e^{-\lambda_1 L_1} \tag{5.35}$$

While for $j \leqslant 0$

$$P[X_1(R_1) = j] = \frac{\lambda_1^{R_1}[\lambda_1(1-\omega)]^{-j}}{(R_1 - 1)!(-j)!} e^{-\lambda_1 L_1(1-\omega)} \int_0^{L_1} t^{R_1 - 1}(L_1 - t)^{-j} e^{-\lambda_1 \omega t} dt \tag{5.36}$$

We then have

$$E[X_1(R_1)] = \sum_{j=-\infty}^{R_1} j P[X_1(R_1) = j] \tag{5.37}$$

$$E[X_1(R_1)^2] = \sum_{j=-\infty}^{R_1} j^2 P[X_1(R_1) = j] \tag{5.38}$$

Cycle time

From analysis of the **TAP** model, we have the mean cycle time as such

$$\frac{(Q_1 - R_1)}{\lambda_1} + \frac{E[X_1(R_1)]}{\lambda_1} + L_1 \tag{5.39}$$

5.5.3 Distribution of inventory level & mean cycle time for location 2

We now derive the mean cycle time and the probability distribution of $X_2(R_2)$, the inventory level at the end of a cycle at location 2. Under the assumption that there is never more than one outstanding order at each location, whenever an order arrives at location 2, the inventory level is restored to a level greater than R_2. The analysis is more complicated than in the **TAP** model because we need to consider three cases: $R_2 > I_2$, $R_2 = I_2$ and $R_2 < I_2$ respectively. Figures from 5.18 and 5.21 show the inventory level at location 2 as a function of time during a cycle for these three cases. We give the separate derivations for each case as follows.

When $R_2 > I_2$

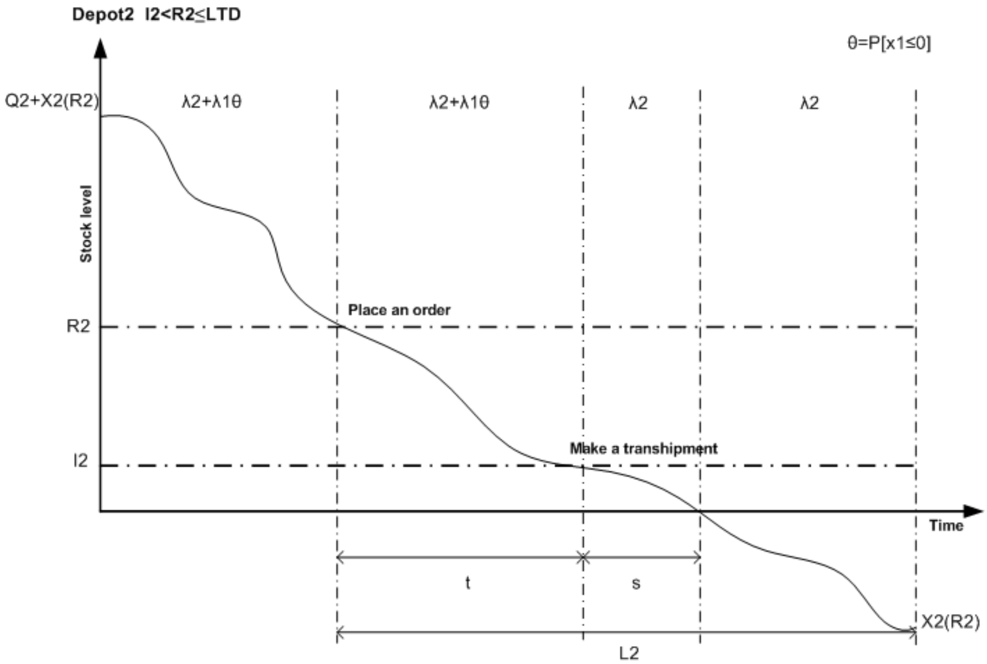

Figure 5.18. Inventory level at location 2 during a cycle when $R_2 > I_2$

Distribution of $X_2(R_2)$

Define D_2 to be a random variable representing the lead time demand (**LTD**). We first consider two cases for the demand during the lead time. For the case when the lead time demand is less than to $R_2 - I_2$, then the rate of the Poisson demand process is equal to $\lambda_2 + \lambda_1\theta$ throughout the cycle. Hence, for $0 \leqslant d < R_2 - I_2$, the probability that the lead time demand is equal to d is given by a simple Poisson probability as follows.

$$P[D_2 = d] = \frac{[(\lambda_2 + \lambda_1\theta)L_2]^d}{d!} e^{-(\lambda_2 + \lambda_1\theta)L_2} \tag{5.40}$$

For the case when the lead time demand is greater than or equal to $R_2 - I_2$, the time at which the inventory level reaches I_2 and hence the rate of the Poisson demand process changes, has an **Erlang** distribution with shape parameter $R_2 - I_2$ and scale parameter $(\lambda_2 + \lambda_1\theta)L_2$. If t is the time at which the rate changes, the demand during the remainder of the lead time is a Poisson random variable with mean $\lambda_2(L_2 - t)$. Conditioning on the time at which the inventory level reaches I_2, the probability that the lead time demand equals d, where $d \geqslant R_2 - I_2$ is as follows.

$$P[D_2 = d] = \int_0^{L_2} \frac{(\lambda_2 + \lambda_1\theta)^{R_2 - I_2} t^{R_2 - I_2 - 1}}{(R_2 - I_2 - 1)!} e^{-(\lambda_2 + \lambda_1\theta)t} \frac{[\lambda_2(L_2 - t)]^{d - R_2 + I_2}}{(d - R_2 + I_2)!} e^{-\lambda_2(L_2 - t)} dt$$

$$= \frac{\lambda_2^{d-R_2+I_2}(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!(d-R_2+I_2)!}\int_0^{L_2} e^{-(\lambda_2+\lambda_1\theta)t}t^{R_2-I_2-1}(L_2-t)^{d-R_2+I_2}e^{-\lambda_2(L_2-t)}dt$$

$$= \frac{\lambda_2^{d-R_2+I_2}(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!(d-R_2+I_2)!}e^{-\lambda_2 L_2}\int_0^{L_2} e^{-\lambda_1\theta t}t^{R_2-I_2-1}(L_2-t)^{d-R_2+I_2}dt$$

(5.41)

Since $X_2(R_2) = R_2 - D_2$ and $D_2 \geqslant 0$, we have the probability distribution of $X_2(R_2)$ as follows.

For $I_2 < j \leqslant R_2$

$$P[X_2(R_2)=j] = \frac{[(\lambda_2+\lambda_1\theta)L_2]^{R_2-j}}{(R_2-j)!}e^{-(\lambda_2+\lambda_1\theta)L_2} \tag{5.42}$$

While for $j \leqslant I_2$

$$P[X_2(R_2)=j] = \frac{\lambda_2^{I_2-j}(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!(I_2-j)!}e^{-\lambda_2 L_2}\int_0^{L_2} t^{R_2-I_2-1}(L_2-t)^{I_2-j}e^{-\lambda_1\theta t}dt \tag{5.43}$$

Then we derive the mean of $X_2(R_2)$ and $X_2(R_2)^2$

$$E[X_2(R_2)] = \sum_{j=-\infty}^{R_2} jP[X_2(R_2)=j] \tag{5.44}$$

$$E[X_2(R_2)^2] = \sum_{j=-\infty}^{R_2} j^2P[X_2(R_2)=j] \tag{5.45}$$

Having established the probability distribution of the inventory level at the end of a cycle, we are now able to derive the expected cycle time.

Cycle time

As illustrated in Figure 5.18, the mean cycle time at location 2 when $R_2 > I_2$ consists of two parts, the first part is for the period when the inventory level falls from $Q_2 + X_2(R_2)$ to R_2 according to a Poisson process with rate $\lambda_2 + \lambda_1\theta$ before the lead time L_2, and the second part is the lead time L_2, hence the mean cycle time is given by

$$\frac{E[Q_2+X_2(R_2)-R_2]}{\lambda_2+\lambda_1\theta} + L_2 = \frac{(Q_2-R_2)+E[X_2(R_2)]}{\lambda_2+\lambda_1\theta} + L_2 \tag{5.46}$$

When $R_2 = I_2$

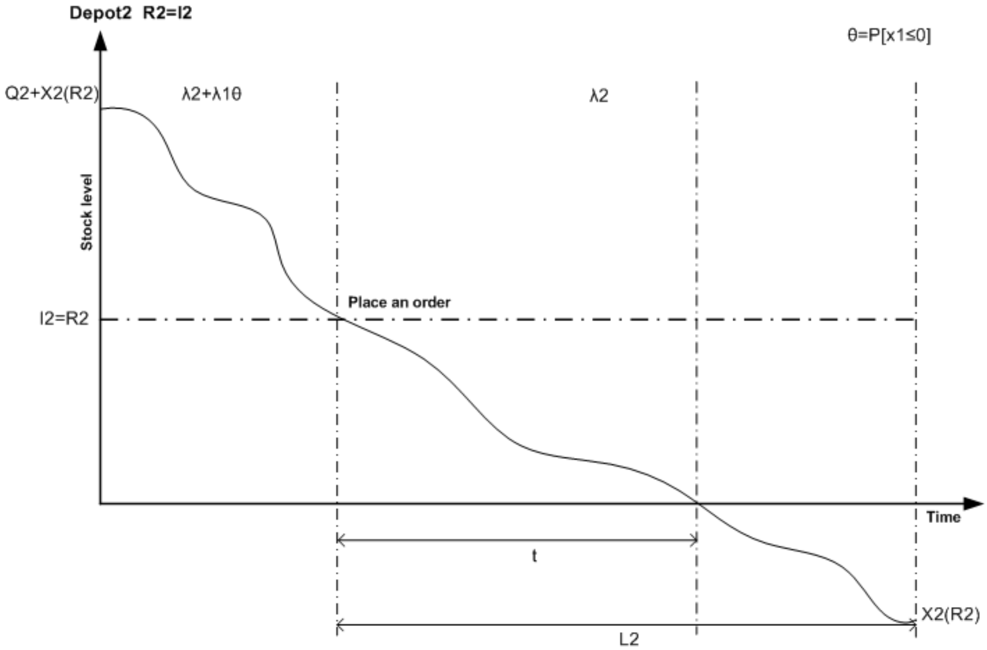

Figure 5.19. Inventory level at location 2 during a cycle when $R_2 = I_2$

As illustrated in Figure 5.19 when $R_2 = I_2$, before the start of lead time, the inventory level falls from $Q_2 + X_2(R_2)$ to R_2 according to a Poisson demand process with rate $\lambda_2 + \lambda_1\theta$, and then continue to fall from R_2 to zero according to a Poisson demand process with rate λ_2 for the duration of the lead time.

Distribution of $X_2(R_2)$

Because the inventory level only falls from R_2 according to a Poisson demand process with rate λ_2 during the lead time, we first consider the probability of the lead time demand which is a simple Poisson demand process as follows, for $d \geqslant 0$

$$P[D_2 = d] = \frac{(\lambda_2 L_2)^d}{d!} e^{-\lambda_2 L_2} \tag{5.47}$$

Then we have the distributions of $X_2(R_2)$, for $j \leqslant R_2$

$$P[X_2(R_2) = j] = \frac{(\lambda_2 L_2)^{R_2 - j}}{(R_2 - j)!} e^{-\lambda_2 L_2} \tag{5.48}$$

Then we infer the mean of $X_2(R_2)$ and $X_2(R_2)^2$ from equation (5.44) and (5.45). Having established the probability distribution of the inventory level at the end of a cycle, we are now able to derive the expected cycle time.

Cycle time

During the first part of a cycle, the inventory level falls from $Q_2 + X_2(R_2)$ to R_2

according to a Poisson demand process with rate of $\lambda_2 + \lambda_1\theta$. As this point, an order is placed and the cycle ends after the lead time L_2. Hence, the mean cycle time is given by

$$\frac{E[Q_2 + X_2(R_2) - R_2]}{\lambda_2 + \lambda_1\theta} + L_2 = \frac{(Q_2 - R_2) + E[X_2(R_2)]}{\lambda_2 + \lambda_1\theta} + L_2 \tag{5.49}$$

When $R_2 < I_2$

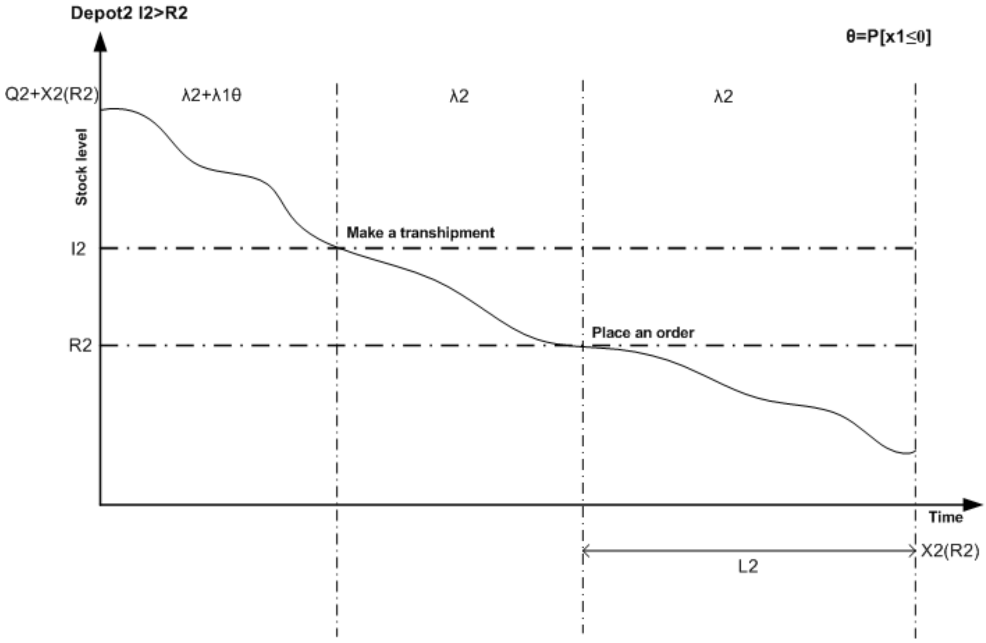

Figure 5.20. Inventory level at location 2 during a cycle when $R_2 < I_2 < Q_2 + X_2(R_2)$

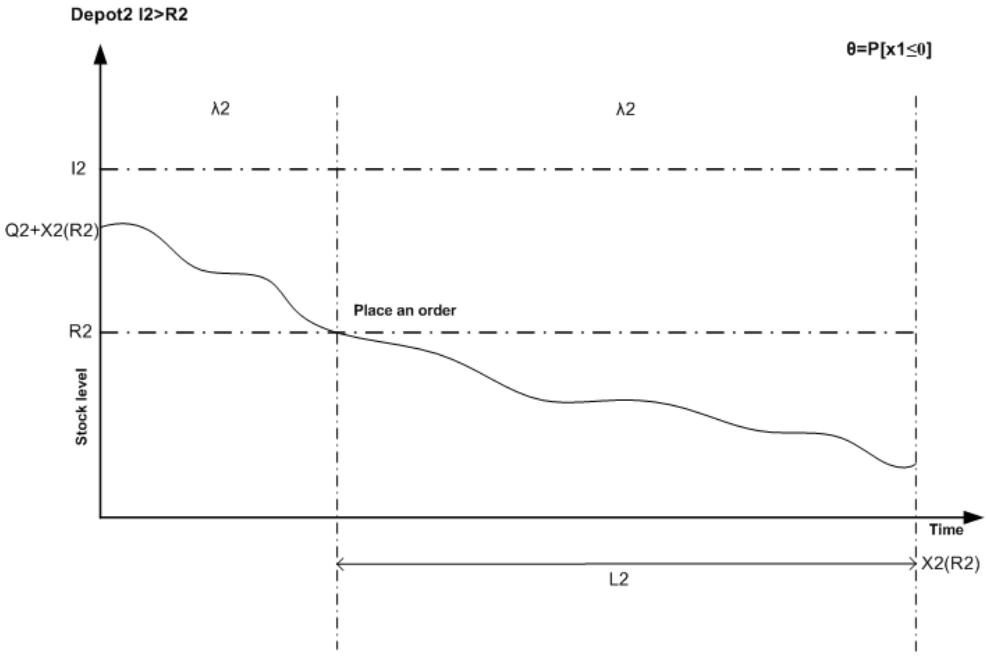

Figure 5.21. Inventory level at location 2 during a cycle when $R_2 < Q_2 + X_2(R_2) < I_2$

Distribution of $X_2(R_2)$

This case is simpler because the lead time demand is a simple Poisson random variable with mean $\lambda_2 L_2$. Hence, for $j \leqslant R_2$, $P[X_2(R_2) = j]$ equals to the probability that the lead time demand is equal to $R_2 - j$ which is given by

$$\frac{(\lambda_2 L_2)^{R_2 - j}}{(R_2 - j)!} e^{-\lambda_2 L_2} \tag{5.50}$$

Then we infer the expected mean of $X_2(R_2)$ and $X_2(R_2)^2$ from equation (5.44) and (5.45). Having established the probability distribution of the inventory level at the end of a cycle, we are now able to derive the expected cycle time.

Cycle time

Shown in Figure 5.20 and 5.21, for the case when $R_2 < I_2$, the inventory level at the start of a cycle, $Q_2 + X_2(R_2)$, maybe above, below or equal to I_2. We need to take account of additional situations when $Q_2 + X_2(R_2)$ is below I_2 or over I_2 for two periods with two rates before the lead time L_2. More specific, we need to consider the situations when the inventory level falls down from $Q_2 + X_2(R_2)$ to I_2 at rate $\lambda_2 + \lambda_1\theta$ and the situation when the inventory level falls down from I_2 to R_2 at rate λ_2. Therefore, the general mean cycle time is given by

$$\begin{aligned}
&= E\{\frac{\max\left(Q_2 + X_2(R_2) - I_2, 0\right)}{\lambda_2 + \lambda_1\theta} + \frac{\min\left(I_2, Q_2 + X_2(R_2)\right) - R_2}{\lambda_2} + L_2\} \\
&= E\{\frac{Q_2 + X_2(R_2) - \min\left(I_2, Q_2 + X_2(R_2)\right)}{\lambda_2 + \lambda_1\theta} + \frac{\min\left(I_2, Q_2 + X_2(R_2)\right) - R_2}{\lambda_2} + L_2\} \\
&= \frac{Q_2 + E\{X_2(R_2)\} - E\{\min\left(I_2, Q_2 + X_2(R_2)\right)\}}{\lambda_2 + \lambda_1\theta} + \frac{E\{\min\left(I_2, Q_2 + X_2(R_2)\right)\} - R_2}{\lambda_2} + L_2
\end{aligned} \tag{5.51}$$

5.5.4 System performance measure approximations

Again, we use the result from the renewal theory that

$$\text{Average cost at location } k = \frac{\text{mean cost over } a \text{ single cycle at location } k}{\text{mean time of } a \text{ single cycle at location } k} \tag{5.52}$$

Because we have already computed the mean cycle time for each location, in order to estimate the average cost rate at a location, we need to find an expression for the mean cost over a cycle at the location k, $k = 1, 2$. Typically, we consider four cost components at each location: **ORDER COST, HOLDING COST, STOCKOUT COST AND BACKORDER COST.** The average transshipment cost per time unit is estimated using a similar approach and added to obtain the average cost rate for the system as a whole.

Note that for location 2, we derive the holding cost, stockout cost and backorder cost during a single cycle for three cases: $R_2 > I_2, R_2 = I_2, R_2 < I_2$ respectively.

Order cost over a cycle

During a cycle at location k, $k = 1, 2$, exactly one replenishment order is placed. Hence the order cost over a cycle at location k is given by c_k.

Holding cost over a cycle at location 1

Because the situation at location 1 under the assumption of the **HOT** model is equivalent to the situation at location 1 under the assumptions of the **TAP** model with $z\phi$ replaced by ω. Hence, we deduce the expected holding cost during the period preceding the point at which the order is placed, inventory level falls from $Q_1 + X_1(R_1)$ to R_1 due to a Poisson demand process with rate λ_1 as such.

$$\frac{h_1}{2\lambda_1}\{Q_1^2 + Q_1 - R_1^2 - R_1 + (2Q_1 + 1)E[X_1(R_1)] + E[X_1(R_1)^2]\} \tag{5.53}$$

The expected holding cost during the lead time of length L_1 is given by

$$h_1 L_1 R_1 F(R_1 - 1, \lambda_1 L_1) - \frac{h_1 \lambda_1 L_1^2}{2} F(R_1 - 2, \lambda_1 L_1) + \frac{h_1 R_1 (R_1 + 1)}{2\lambda_1} \overline{F}(R_1, \lambda_1 L_1) \tag{5.54}$$

Holding cost over a cycle at location 2

When $R_2 > I_2$

Between the start of a cycle and the point at which an order is placed, the inventory level falls from $Q_2 + X_2(R_2)$ to R_2 due to a Poisson demand process with constant rate $\lambda_2 + \lambda_1\theta$. By Lemma 5.1a, the expected holding cost during this period is given by

$$\frac{h_2}{2(\lambda_2 + \lambda_1\theta)}[Q_2 + X_2(R_2) - R_2][Q_2 + X_2(R_2) + R_2 + 1]$$

Hence the expected holding cost is given by

$$E\{\frac{h_2}{2(\lambda_2+\lambda_1\theta)}[Q_2+X_2(R_2)-R_2][Q_2+X_2(R_2)+R_2+1]\}$$

$$=\frac{h_2}{2(\lambda_2+\lambda_1\theta)}\{Q_2^2+Q_2-R_2^2-R_2+(2Q_2+1)E[X_2(R_2)]+E[X_2(R_2)^2]\} \tag{5.55}$$

During the lead time L_2, we consider the situations where the inventory level falls from R_2 to I_2 due to a Poisson demand process with rate $\lambda_2+\lambda_1\theta$, then the inventory level falls from I_2 due to a Poisson demand process with rate λ_2 until it reaches zero, or where the inventory level falls from R_2 to I_2 due to a Poisson demand process with rate $\lambda_2+\lambda_1\theta$. Therefore, by Lemma 5.3, put these situations together, the expected holding cost occurred during the lead time L_2 is given by

$$\begin{aligned}
&= h_2L_2R_2F(R_2-I_2-1,(\lambda_2+\lambda_1\theta)L_2)-h_2\frac{(\lambda_2+\lambda_1\theta)L_2^2}{2}F(R_2-I_2-2,(\lambda_2+\lambda_1\theta)L_2)+\\
&\quad h_2\frac{(R_2+I_2+1)(R_2-I_2)}{2(\lambda_2+\lambda_1\theta)}\overline{F}(R_2-I_2,(\lambda_2+\lambda_1\theta)L_2)+\\
&\quad h_2\frac{(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!}\int_0^{L_2}t^{R_2-I_2-1}e^{-(\lambda_2+\lambda_1\theta)t}\{(L_2-t)I_2F(I_2-1,\lambda_2(L_2-t))-\\
&\qquad \frac{\lambda_2(L_2-t)^2}{2}F(I_2-2,\lambda_2(L_2-t))+\frac{I_2(I_2+1)}{2\lambda_2}\overline{F}(I_2,\lambda_2(L_2-t))\}dt
\end{aligned} \tag{5.56}$$

When $R_2 = I_2$

Shown in Figure 5.19, between the start of a cycle and the point at which an order is placed, the inventory level falls from $Q_2+X_2(R_2)$ to R_2 due to a Poisson demand process with constant rate $\lambda_2+\lambda_1\theta$. By Lemma 5.1a, the holding cost till the start of the lead time L_2 is $\frac{h_2}{2(\lambda_2+\lambda_1\theta)}[Q_2+X_2(R_2)-R_2][Q_2+X_2(R_2)+R_2+1]$

Hence the expected holding cost during the period is given by

$$E\{\frac{h_2}{2(\lambda_2+\lambda_1\theta)}[Q_2+X_2(R_2)-R_2][Q_2+X_2(R_2)+R_2+1]\}$$

$$=\frac{h_2}{2(\lambda_2+\lambda_1\theta)}\{Q_2^2+Q_2-R_2^2-R_2+(2Q_2+1)E[X_2(R_2)]+E[X_2(R_2)^2]\} \tag{5.57}$$

During the lead time L_2, the inventory level starts from R_2 and falls due to a Poisson process with rate λ_2 until it reaches zero. After this point, no further holding cost is incurred. Hence by Lemma 5.2, the expected holding cost occurred during the lead time L_2 is given by

$$h_2L_2R_2F(R_2-1,\lambda_2L_2)-\frac{h_2\lambda_2L_2^2}{2}F(R_2-2,\lambda_2L_2)+\frac{h_2R_2(R_2+1)}{2\lambda_2}\overline{F}(R_2,\lambda_2L_2) \tag{5.58}$$

When $R_2 < I_2$

Shown in Figures 5.20 and 5.21, the inventory level at the beginning of the single cycle could start above or below I_2, therefore, we need to consider two situations in which $I_2 < Q_2+X_2(R_2)$ and $I_2 \geqslant Q_2+X_2(R_2)$ over a period before the lead time respectively.

For the situation in Figure 5.20, the inventory level falls from $Q_2+X_2(R_2)$ to I_2 due to a Poisson demand process with rate $\lambda_2 + \lambda_1\theta$, then falls from I_2 to R_2 according to another Poisson demand process at rate λ_2. For the situation in Figure 5.21, the inventory level falls from $Q_2 + X_2(R_2)$ to R_2 due to a Poisson demand process with rate λ_2. We can combine these two situations by thinking of the inventory level first falling from $Q_2+X_2(R_2)$ to $\min(I_2,Q_2+X_2(R_2))$ according to a Poisson demand process with rate $\lambda_2+\lambda_1\theta$ and then falling from this level to R_2 according to a Poisson demand process with rate λ_2. By Lemma 5.1a, the holding cost from the beginning of a cycle to the start of the lead time is given by

$$\frac{h_2}{2(\lambda_2+\lambda_1\theta)}[Q_2+X_2(R_2)-\min(I_2,Q_2+X_2(R_2))][Q_2+X_2(R_2)+\min(I_2,Q_2+X_2(R_2)) \\ +1]+\frac{h_2}{2\lambda_2}[\min(I_2,Q_2+X_2(R_2))-R_2][\min(I_2,Q_2+X_2(R_2))+R_2+1]$$

Hence the expected holding cost during the period is given by

$$\begin{aligned}
&= E\{\tfrac{h_2}{2(\lambda_2+\lambda_1\theta)}[Q_2+X_2(R_2)-\min(I_2,Q_2+X_2(R_2))][Q_2+X_2(R_2)+\min(I_2,Q_2+X_2(R_2)) \\
&\qquad +1]+\tfrac{h_2}{2\lambda_2}[\min(I_2,Q_2+X_2(R_2))-R_2][\min(I_2,Q_2+X_2(R_2))+R_2+1]\} \\
&= \tfrac{h_2}{2(\lambda_2+\lambda_1\theta)}\{Q_2^2+Q_2-E[\min(I_2,Q_2+X_2(R_2))^2]-E[\min(I_2,Q_2+X_2(R_2))]+ \\
&\quad (2Q_2+1)E[X_2(R_2)]+E[X_2(R_2)]^2\}+\tfrac{h_2}{2\lambda_2}\{E[\min(I_2,Q_2+X_2(R_2))^2]+ \\
&\quad E[\min(I_2,Q_2+X_2(R_2))]-R_2^2-R_2\} \\
&= \tfrac{h_2}{2(\lambda_2+\lambda_1\theta)}\{Q_2^2+Q_2+(2Q_2+1)E[X_2(R_2)]+E[X_2(R_2)]^2\}+\tfrac{h_2\lambda_1\theta}{2\lambda_2(\lambda_2+\lambda_1\theta)} \\
&\quad \{E[\min(I_2,Q_2+X_2(R_2))]^2+E[\min(I_2,Q_2+X_2(R_2))]\}-\tfrac{h_2}{2\lambda_2}\{R_2^2+R_2\}
\end{aligned} \tag{5.59}$$

During the lead time L_2, the inventory level falls from R_2 due to a Poisson process with rate λ_2. Hence, by Lemma 5.2, the expected holding cost occurred during the lead time L_2 is given by

$$h_2 L_2 R_2 F(R_2-1,\lambda_2 L_2) - \frac{h_2\lambda_2 L_2^2}{2} F(R_2-2,\lambda_2 L_2) + \frac{h_2 R_2(R_2+1)}{2\lambda_2}\overline{F}(R_2,\lambda_2 L_2) \tag{5.60}$$

Stockout cost over a cycle

The inventory level at a location is only replenished at the beginning of a cycle. Therefore, any backorder placed at a location during a cycle will still be outstanding at the end of that cycle. It follows that the number of backorders placed at location k, $k=1,2$ during a cycle is equal to max $\{-X_k(R_k),0\}$. Hence, the expected stockout cost during a cycle at location k is given by

$$\sum_{j=1}^{\infty} j\hat{b}_k P[X_k(R_k)=-j] \tag{5.61}$$

Backorder cost over a cycle at location 1

As the situation at location 1 of the **HOT** model is similar to the situation at location 1 of the **TAP** model with $z\phi$ replaced by ω. Hence, by Lemma 5.3c, the expected backorder cost over a cycle is given by

$$b_1 \frac{\lambda_1^{R_1+1}(1-\omega)}{2(R_1-1)!}\int_0^{L_1} t^{R_1-1}e^{-\lambda_1 t}(L_1-t)^2 dt \tag{5.62}$$

Backorder cost over a cycle at location 2

When $R_2 > I_2$

Shown in Figure 5.18, during the lead time, the inventory level falls from R_2 to I_2 according to a Poisson process with rate $\lambda_2+\lambda_1\theta$. After this point, the inventory level continues to fall according to a Poisson process with rate λ_2. By Lemma 5.3c, the expected backorder cost over a cycle is given by

$$\begin{aligned} &b_2\tfrac{(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!}\int_0^{L_2} t^{R_2-I_2-1}e^{-(\lambda_2+\lambda_1\theta)t}\{\tfrac{\lambda_2(L_2-t)^2}{2}\overline{F}(I_2-1,\lambda_2(L_2-t)) - \\ &\quad (L_2-t)I_2\overline{F}(I_2,\lambda_2(L_2-t)) + \tfrac{I_2(I_2+1)}{2\lambda_2}\overline{F}(I_2+1,\lambda_2(L_2-t))\}dt \end{aligned} \tag{5.63}$$

When $R_2 = I_2$

Illustrated in Figure 5.19, the inventory level falls from R_2 at the beginning of the lead time L_2 due to Poisson demand process with rate λ_2. By Lemma 5.2b, the expected backorder cost over the lead time is given by

$$b_2[\frac{\lambda_2 L_2^2}{2}\overline{F}(R_2-1,\lambda_2 L_2) - L_2 R_2 \overline{F}(R_2,\lambda_2 L_2) + \frac{R_2(R_2+1)}{2\lambda_2}\overline{F}(R_2+1,\lambda_2 L_2)] \tag{5.64}$$

When $R_2 < I_2$

Illustrated in Figure 5.20 or 5.21, the inventory level falls from R_2 at the beginning of the lead time L_2 due to Poisson demand process with rate λ_2. By Lemma 5.2b, the expected backorder cost over the lead time is given by

$$b_2[\frac{\lambda_2 L_2^2}{2}\overline{F}(R_2-1,\lambda_2 L_2) - L_2 R_2 \overline{F}(R_2,\lambda_2 L_2) + \frac{R_2(R_2+1)}{2\lambda_2}\overline{F}(R_2+1,\lambda_2 L_2)] \tag{5.65}$$

Transshipment cost over a cycle

Recall that we assume the unidirectional transshipment from location 2 to location 1 in the **HOT** model. Under this assumption, a demand at location 1 results in a transshipment when location 1 has no stock and the inventory level at location 2 at the meantime is greater than I_2. Hence, the indirect fill rate, or transshipment fill rate, at location 1 is approximated as $\theta\omega$ and transshipment occurs at a rate of $\lambda_1\theta\omega$. Hence the transshipment cost over a single cycle is given by

$$t\lambda_1\theta\omega \tag{5.66}$$

Average cost rates

The expected total cost rate at location 1 over a single cycle is equal to the sum of c_1 and equations (5.53), (5.54), (5.61) and (5.62). The average cost rate at location 1 is obtained by dividing the result by equation (5.39).

At location 2, we have the expected total cost rate for three cases: $R_2 > I_2$, $R_2 = I_2$ and $R_2 < I_2$ respectively. For the case $R_2 > I_2$, the expected total cost rate at location 2 over a single cycle is equal to the sum of c_2 and equations (5.55), (5.56), (5.61) and (5.63). The average cost rate at location 1 is obtained by dividing the result by equation (5.46).

For the case $R_2 = I_2$, the expected total cost rate at location 2 over a single cycle is equal to the sum of c_2 and equations (5.57), (5.58), (5.61) and (5.64). The average cost rate at location 1 is obtained by dividing the result by equation (5.49).

For the case $R_2 < I_2$, the expected total cost rate at location 2 over a single cycle is equal to the sum of c_2 and equations (5.59), (5.60), (5.61) and (5.65). The average cost rate at location 1 is obtained by dividing the result by equation (5.51).

As a whole, the expected total cost for the two-location system is the sum of the expected total cost rate at location 1 and location 2 plus the expected transshipment cost rate from equation (5.66).

Direct fill rates

By the definitions of θ and ϕ, we derive the direct fill rate α_1 at location 1 as $1 - \theta$ and the direct fill rate α_2 at location 2 as ϕ. Note that a method of estimating θ, and ϕ is derived in Section 5.5.5.

Backorder fill rates

Any unmet demand will be either met by a backorder or a transshipment delivery at location 1, therefore, the backorder fill rate β_1 at location 1 is derived as $\theta(1 - \omega)$. At location 2, an unmet demand will be backordered, hence, the backorder fill rate β_2 at location 2 is derived as $1 - \phi$. Note that a method of estimating ω is derived in Section 5.5.5.

Indirect fill rate at location 1

As explained above, the indirect fill rate, or transshipment fill rate, at location 1 is derived as $\theta\omega$.

5.5.5 Solution algorithm

The decomposition approach and the approximations to the system performance measure depend on θ, ϕ and ω. Therefore before we use the expressions derived in the sections above, we need a method of estimating θ, ϕ and ω. In this section, we propose an iterative scheme that starts with estimates based on the assumption that the locations are independent. While we offer no proof of convergence, the scheme is found to converge after relatively few iterations in all examples we have considered. We first derive expressions for θ, ϕ and ω.

Factor $\boldsymbol{\omega}$

We address the question of estimating the transshipment factor ω using renewal type arguments as follows.

$$\omega = P[X_2 > I_2] = 1 - P[X_2 \leqslant I_2] = 1 - \frac{\text{mean time spent with } x_2 \leqslant I_2 \text{ in the cycle}}{\text{mean cycle time}}$$

We need to consider three cases: R_2 greater than I_2, R_2 equal to I_2 and R_2 less than I_2 respectively.

For the case when $R_2 > I_2$, when the lead time demand is greater than or equal to $R_2 - I_2$, the time at which the inventory level reach I_2 has an **Erlang** distribution with shape parameter $R_2 - I_2$ and scale parameter $(\lambda_2 + \lambda_1\theta)L_2$. Since inventory level is decreasing during a cycle, if t is the time at which the inventory level reaches I_2, the time for which the inventory level is less than or equal to I_2 during the cycle is $L_2 - t$. Conditioning on the time at which the inventory level reaches I_2, the mean time in which the inventory level is less than or equal to I_2 during a single cycle is estimated as follows

$$\int_0^{L_2} \frac{(\lambda_2 + \lambda_1\theta)^{R_2 - I_2} t^{R_2 - I_2 - 1}}{(R_2 - I_2 - 1)!} e^{-(\lambda_2 + \lambda_1\theta)t} (L_2 - t) dt$$
$$= L_2 \overline{F}\left(R_2 - I_2 - 1, (\lambda_2 + \lambda_1\theta)L_2\right) - \frac{R_2 - I_2}{\lambda_2 + \lambda_1\theta} \overline{F}\left(R_2 - I_2, (\lambda_2 + \lambda_1\theta)L_2\right)$$

Thus we can further derive ω as such.

$$1 - \frac{L_2 \overline{F}\left(R_2 - I_2 - 1, (\lambda_2 + \lambda_1\theta)L_2\right) - \frac{R_2 - I_2}{\lambda_2 + \lambda_1\theta} \overline{F}\left(R_2 - I_2, (\lambda_2 + \lambda_1\theta)L_2\right)}{\text{mean cycle time from equation (5.46)}} \tag{5.67}$$

For the case when $R_2 = I_2$, as the inventory level starts with R_2 at the beginning of the lead time, the mean time for the inventory level is less than or equal to I_2 is equal to the length of the lead time L_2. Hence, ω is derived as such

$$1 - \frac{L_2}{\text{mean cycle time from equation (5.49)}} \tag{5.68}$$

For the case when $R_2 < I_2$, the inventory level at the start of a cycle, $Q_2 + X_2(R_2)$, maybe above, below or equal to I_2. The time for which the inventory level is less than or equal to I_2 has two parts, one part before the lead time starts and the other part is the lead time. During the first part, the inventory level falls either from I_2 to R_2 or from $Q_2 + X_2(R_2)$ to R_2 depending on whether the inventory level at the beginning of the cycle is greater than I_2 or less than or equal to I_2 respectively. The fall occurs according to a Poisson process with rate λ_2. Thus the ω is given by

$$1 - \frac{\frac{E\{\min\left(I_2, Q_2 + X_2(R_2)\right)\} - R_2}{\lambda_2} + L_2}{\text{mean cycle time from equation (5.51)}} \tag{5.69}$$

Factor θ

We address the question of estimating the factor θ, the probability of location 1 needing to make a transshipment request. Again, by renewal-type arguments we have

$$\theta = P[X_1 \leqslant 0] = \frac{\text{mean time spent with } X_1 \leqslant 0 \text{ in the cycle}}{\text{mean cycle time from equation (5.39)}}$$

As illustrated in Figure 5.17, when the lead time demand is greater than R_1, the time at which the inventory level reaches zero has an **Erlang** distribution with shape parameter R_1 and scale parameter $\lambda_1 L_1$. Orders are never delivered during a cycle, so the inventory level is decreasing during a cycle. If t is the time at which the inventory level reaches zero, the time for which the inventory level is less than or equal to zero during the cycle is $L_1 - t$. Conditioning on the time at which the inventory level reaches zero, the mean time at which the inventory level is less than or equal to zero during a single cycle is as follows.

$$\int_0^{L_1} \frac{\lambda_1^{R_1} t^{R_1-1}}{(R_1-1)!} e^{-\lambda_1 t}(L_1 - t)dt = L_1 \int_0^{L_1} \frac{\lambda_1^{R_1} t^{R_1-1} e^{-\lambda_1 t}}{(R_1-1)!} dt - \frac{R_1}{\lambda_1} \int_0^{L_1} \frac{\lambda_1^{R_1+1} t^{R_1} e^{-\lambda_1 t}}{R_1!} dt$$

$$= L_1 \sum_{j=R_1}^{\infty} \frac{(\lambda_1 L_1)^j}{j!} e^{-\lambda_1 L_1} - \frac{R_1}{\lambda_1} \sum_{j=R_1+1}^{\infty} \frac{(\lambda_1 L_1)^j}{j!} e^{-\lambda_1 L_1}$$

$$= L_1 \overline{F}(R_1 - 1, \lambda_1 L_1) - \frac{R_1}{\lambda_1} \overline{F}(R_1, \lambda_1 L_1)$$

Hence we derive θ as follows

$$\frac{L_1 \overline{F}(R_1 - 1, \lambda_1 L_1) - \frac{R_1}{\lambda_1} \overline{F}(R_1, \lambda_1 L_1)}{\frac{(Q_1 - R_1)}{\lambda_1} + \frac{E[X_1(R_1)]}{\lambda_1} + L_1} \tag{5.70}$$

Factor ϕ

Finally we address the question of estimating the factor ϕ, the probability that location 2 has stock. Again, we use the renewal-type arguments as follows.

$$\phi = P[X_2 \geqslant 1] = 1 - P[X_2 < 1] = 1 - \frac{\text{mean time spent with } x_2 < 1 \text{ in the cycle}}{\text{mean cycle time}}$$

We need to consider three cases: R_2 greater than I_2, R_2 equal to I_2 and R_2 less than I_2 respectively.

For the case when $R_2 > I_2$, we need to approximate the time at which the inventory level reaches I_2, which has an **Erlang** distribution with shape parameter $R_2 - I_2$ and scale parameter $(\lambda_2 + \lambda_1\theta)L_2$. Let t denote this time. We then approximate the time at which the inventory level drops from I_2 to zero, which has an **Erlang** distribution with shape parameter I_2 and scale parameter $\lambda_2(L_2 - t)$. Let s denote this time. Since the inventory level is decreasing during a cycle, the time for which the inventory level is less than or equal to zero during the cycle is $L_2 - t - s$. Conditioning on t and s, the mean time for which the inventory level is less than or equal to zero during a single cycle is estimated as follows.

$$\int_0^{L_2} \frac{(\lambda_2+\lambda_1\theta)^{R_2-I_2} t^{R_2-I_2-1}}{(R_2-I_2-1)!} e^{-(\lambda_2+\lambda_1\theta)t} \int_0^{L_2-t} \frac{\lambda_2^{I_2} s^{I_2-1}}{(I_2-1)!} e^{-\lambda_2 s}(L_2-t-s)\,ds\,dt$$
$$= \frac{(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!} \int_0^{L_2} t^{R_2-I_2-1} e^{-(\lambda_2+\lambda_1\theta)t}\{(L_2-t)\overline{F}(I_2-1,\lambda_2(L_2-t))$$
$$-\frac{I_2}{\lambda_2}\overline{F}(I_2,\lambda_2(L_2-t))\}dt$$

Thus we can further derive ϕ as such.

$$1 - \frac{\frac{(\lambda_2+\lambda_1\theta)^{R_2-I_2}}{(R_2-I_2-1)!} \int_0^{L_2} t^{R_2-I_2-1} e^{-(\lambda_2+\lambda_1\theta)t}\{(L_2-t)\overline{F}(I_2-1,\lambda_2(L_2-t)) - \frac{I_2}{\lambda_2}\overline{F}(I_2,\lambda_2(L_2-t))\}dt}{\text{mean cycle time from equation (5.46)}} \tag{5.71}$$

For the case when $R_2 = I_2$, when the lead time demand is greater than R_2, the time at which the inventory level reaches zero has an **Erlang** distribution with shape parameter R_2 and scale parameter $\lambda_2 L_2$. Since inventory level is decreasing during a cycle, if t is the time at which the inventory level reaches zero, the time for which the inventory level is less than or equal to zero during the cycle is $L_2 - t$. Conditioning on the time at which the inventory level reaches zero, the mean time for which the inventory level is less than or equal to zero during a single cycle is estimated as follows.

$$\int_0^{L_2} \frac{\lambda_2^{R_2} t^{R_2-1}}{(R_2-1)!} e^{-\lambda_2 t}(L_2-t)dt = L_2\overline{F}(R_2-1,\lambda_2 L_2) - \frac{R_2}{\lambda_2}\overline{F}(R_2,\lambda_2 L_2).$$

Thus we can further derive ϕ as such.

$$1 - \frac{L_2\overline{F}(R_2-1,\lambda_2 L_2) - \frac{R_2}{\lambda_2}\overline{F}(R_2,\lambda_2 L_2)}{\text{mean cycle time from equation (5.49)}} \tag{5.72}$$

For the case when $R_2 < I_2$, shown in Figures 5.20 and 5.21, the mean time for which the inventory level is less than or equal to zero is exactly the same as in the case when $R_2 = I_2$. Hence, we can derive ϕ as such

$$1 - \frac{L_2\overline{F}(R_2-1,\lambda_2 L_2) - \frac{R_2}{\lambda_2}\overline{F}(R_2,\lambda_2 L_2)}{\text{mean cycle time from equation (5.51)}} \tag{5.73}$$

Computation algorithm

From the expressions derived for ω, θ and ϕ, we learn that ω is dependent on $E[X_2(R_2)]$ and θ, θ is dependent on $E[X_1(R_1)]$ and ϕ is dependent on θ and $E[X_2(R_2)]$. Meanwhile, $E[X_1(R_1)]$ is dependent on ω and $E[X_2(R_2)]$ is dependent on θ. Hence, we devise an iterative approach on two pairs $\{\omega, \theta\}$ and $\{E[X_1(R_1)], E[X_2(R_2)]\}$. Specifically, at iteration n we first calculate the estimate $E[X_1(R_1)]^n$ and $E[X_2(R_2)]^n$ based on known values ω^n and ϕ^n. We then calculate the estimate ω^{n+1} from θ and $E[X_2(R_2)]^n$ and the estimate θ^{n+1} from $E[X_2(R_2)]^n$. To initialise the iteration we set ω^0 equal to the transshipment fill rate at location 1 and θ^0 equal to the backorder fill rate at location 1 under the assumption that the locations are independent. We continue the calculation until the values ω^n and θ^n converge in the sense that $\max(|\omega^n - \omega^{n-1}|,|\theta^n - \theta^{n-1}|) < \epsilon$, where ϵ a is pre-defined tolerance.

$$\begin{Bmatrix} \omega^{(0)} \\ \theta^{(0)} \end{Bmatrix} \to \begin{Bmatrix} E[X_1(R_1)]^{(0)} \\ E[X_2(R_2)]^{(0)} \end{Bmatrix} \to \begin{Bmatrix} \omega^{(1)} \\ \theta^{(1)} \end{Bmatrix} \to \begin{Bmatrix} E[X_1(R_1)]^{(1)} \\ E[X_2(R_2)]^{(1)} \end{Bmatrix} \cdots\cdots \to \text{continue until converged} \begin{Bmatrix} \omega^* \\ \theta^* \end{Bmatrix}$$

Then using the converged ω^* and θ^*, we work out all the expressions for the average cost rates at the two locations in Section 5.5.4.

Optimal total cost algorithm

Given the fixed values of (R_k, Q_k) at location $k, k = 1, 2$. We perform an exhaustive search to find the value of holdout threshold I_2. which minimises the approximation to the average total cost rate. Specifically, for each value of I_2 ranging from 0 to $Q_2 + R_2$, we first estimate ω and θ using the iterative algorithm described above and then calculate the average total cost rate for the two–location system.

5.5.6 Model verifications

For the same reasons we explained in Section 5.4.6, we need to verify the correctness of the cost approximations against the simulation results with the explicit holdout transshipment policy during a cycle. Hence, we reuse the simulation algorithm for the system performance measures during a cycle in Section 5.4.6. Generally, T_k, C_{h_k}, $C_{\hat{b}_k}$, C_{b_k} and $E[X_k(R_k)]$ denote the mean cycle time, expected average holding cost, expected average stockout cost, expected average backorder cost and expected end of cycle inventory level at location k, $k = 1, 2$. We use the same parameters set for both locations, namely $L_k = 1, h_k = 0.8, \hat{b}_k = 60, b_k = 5, (k = 1, 2)$, $\omega = 0.2, 0.8$ and $\theta = 0.05, 0.10$.

Tables 5.9 and 5.10 show the results from simulation (column headed "sim") and the results from our **JAVA** implementation (column headed "an") for different values of λ_k, R_k, Q_k and I_2. The simulation results are based on the simulation of 1,000,000 cycles. In each case, the standard errors for simulation are very small and so are omitted in the result tables.

				T_1		C_{h_1}		$C_{\hat{b}_1}$		C_{b_1}		$E[X_1(R_1)]$	
λ_1	R_1	Q_1	ω	sim	an	sim	an	sim	an	sim	an	sim	an
10	10	60	0.2	6.03	6.03	24.51	24.52	9.98	9.96	0.14	0.14	0.25	0.25
10	10	60	0.8	6.10	6.10	24.79	24.79	2.46	2.46	0.03	0.03	1.00	1.00
10	20	60	0.2	6.00	6.00	32.40	32.40	0.02	0.02	0.00	0.00	10.00	10.00
10	20	60	0.8	6.00	6.00	32.40	32.40	0.01	0.01	0.00	0.00	10.00	10.00
20	10	60	0.2	3.10	3.10	17.87	17.87	155.10	154.97	3.55	3.55	-8.00	-8.00
20	10	60	0.8	3.40	3.40	20.14	20.15	35.35	35.32	0.81	0.81	-1.99	-1.99
20	20	60	0.2	3.02	3.02	24.57	24.58	28.36	28.25	0.29	0.29	0.35	0.36
20	20	60	0.8	3.07	3.07	24.95	24.95	6.96	6.94	0.07	0.07	1.42	1.42

Table 5.9. Verification results for **HOT** model at location 1

						T_2		C_{h_2}		$C_{\hat{b}_2}$		C_{b_2}		$E[X_2(R_2)]$	
λ_1	λ_2	I_2	R_2	Q_2	θ	sim	an	sim	an	sim	an	sim	an	sim	an
20	10	1	10	40	0.05	3.66	3.66	15.75	15.75	26.90	26.86	0.40	0.40	-0.77	-0.78
20	10	5	10	40	0.05	3.69	3.68	15.90	15.89	23.74	23.70	0.34	0.34	-0.45	-0.48
20	10	10	10	40	0.05	3.73	3.73	16.14	16.14	20.18	20.14	0.28	0.28	0.00	0.00
20	10	15	10	40	0.05	3.77	3.77	16.07	16.07	19.93	19.90	0.28	0.28	0.00	0.00
20	10	20	10	40	0.05	3.82	3.82	16.05	16.05	19.69	19.66	0.27	0.27	0.00	0.00
20	10	1	10	40	0.10	3.38	3.38	15.11	15.10	36.06	36.08	0.56	0.56	-1.44	-1.47
20	10	5	10	40	0.10	3.43	3.43	15.40	15.38	28.85	28.86	0.43	0.43	-0.83	-0.86
20	10	10	10	40	0.10	3.50	3.50	15.86	15.86	21.49	21.45	0.30	0.30	0.00	0.00
20	10	15	10	40	0.10	3.58	3.58	15.73	15.73	20.99	20.95	0.29	0.29	0.00	0.00
20	10	20	10	40	0.10	3.67	3.67	15.70	15.70	20.51	20.47	0.29	0.28	0.00	0.00
20	30	1	20	60	0.05	1.95	1.95	16.68	16.68	327.57	327.86	5.61	5.62	-10.61	-10.61
20	30	5	20	60	0.05	1.95	1.95	16.74	16.73	323.04	323.30	5.50	5.50	-10.48	-10.48
20	30	10	20	60	0.05	1.96	1.96	16.82	16.81	317.37	317.69	5.35	5.36	-10.32	-10.32
20	30	15	20	60	0.05	1.96	1.96	16.91	16.91	311.76	312.14	5.21	5.22	-10.16	-10.17
20	30	20	20	60	0.05	1.97	1.97	17.01	17.01	306.19	306.43	5.07	5.07	-9.99	-10.00
20	30	25	20	60	0.05	1.97	1.97	17.02	17.01	305.36	305.59	5.06	5.06	-9.99	-10.00
20	30	30	20	60	0.05	1.98	1.98	17.03	17.03	304.53	304.76	5.04	5.05	-9.99	-10.00
20	30	1	20	60	0.10	1.90	1.90	16.19	16.18	353.60	353.84	6.26	6.26	-11.18	-11.19
20	30	5	20	60	0.10	1.91	1.91	16.30	16.29	344.42	344.66	6.01	6.01	-10.93	-10.94
20	30	10	20	60	0.10	1.92	1.92	16.45	16.45	333.11	333.43	5.71	5.72	-10.62	-10.63
20	30	15	20	60	0.10	1.93	1.93	16.63	16.63	321.96	322.35	5.42	5.43	-10.31	-10.32
20	30	20	20	60	0.10	1.94	1.94	16.83	16.82	310.97	311.21	5.15	5.15	-9.99	-10.00
20	30	25	20	60	0.10	1.95	1.95	16.84	16.83	309.31	309.55	5.12	5.13	-9.99	-10.00
20	30	30	20	60	0.10	1.96	1.96	16.87	16.86	307.67	307.90	5.10	5.10	-9.99	-10.00

Table 5.10. Verification results for **HOT** model at location 2

We would use the percentage difference to compare the results of the simulation model and the **JAVA** implementation. However, as some of the values are close to zero, the percentage difference can be large even though the values are close. Hence, we only use the percentage difference when the estimate values are greater than 1 and the difference otherwise.

The estimates of mean cycle time are within 0.180% and 1.649% of each other at location 1 and location 2 respectively. The estimates of expected holding cost are within 0.008% and 0.083% of each other respectively. For values greater than 1, the estimates of expected stockout cost are within 0.003% and 0.098% of each other for location 1 and location 2 respectively. For smaller values, the estimates are within 0.00 of each other for location 1. For values greater than 1, the estimates of expected backorder cost are within 0.000% and 0.176% of each other for location 1 and location 2 respectively. For smaller values, the estimates are within 0.01 and 0.03 of each other for location 1 and location 2 respectively.

Verdict: The verification results above reveal strong evidence that our **JAVA** implementation of the analytic approximations to the performance measures for the **HOT** model is accurate. Hence we are confident that we can use our **JAVA** implementation in a numerical investigation to evaluate the proposed solution algorithm.

5.5.7 Numerical experiments

The aim of our numerical experiments is to determine whether the proposed solution algorithm can be used to find the optimal holdout policy for the two-location inventory system with unidirectional holdout transshipment policy. Because $\hat{b_1}$ and $\hat{b_2}$ were identified as two parameters to which the optimal average total cost is sensitive in our numerical experiments on the **TAP** model in Section 5.4.7, we continue to examine how the optimal average total cost depends $\hat{b_1}$ and $\hat{b_2}$.

In the following result tables, I_2 denotes the holdout threshold, C denotes the average total cost for the two-location system and C_t denotes the average transshipment cost from location 2 to location 1. Then C_k, C_{o_k}, C_{h_k}, $C_{\hat{b_k}}$ and C_{b_k} denote the average cost, average order cost, average holding cost, average stockout cost and average backorder cost at location $k\,(k = 1, 2)$ respectively. In addition, α_k, β_k denotes the fill rate and backorder fill rate at location $k\,(k = 1, 2)$ and γ the transshipment fill rate from location 2 to location 1 respectively. Since $\alpha_1 + \beta_1 + \gamma = 1$ and $\alpha_2 + \beta_2 = 1$. We only provide the fill rate α_1, α_2 and γ in the result tables.

For the simulation implementation, in order to make a good balance between computational time and accuracy, we performed 500 independent simulation runs of 50,000 time units with a warm-up period of 500 time units. In all cases, the standard errors are small and so are not included in the result tables. Firstly, we would like to compare the results by using the same parameter set for the snapshot test 1 and 2 which we used in the **TAP** model.

Snapshot numerical test set 1

To demonstrate how parameters $\hat{b_1}$ or $\hat{b_2}$ affect the optimal average total cost, we use 12 snapshots described in Table 5.11 corresponding to 12 combinations on $\hat{b_1}$ and $\hat{b_2}$. In each snapshot, the other parameters are given values as follows: $\lambda_1 = 20, c_1 = 10, h_1 = 1, \hat{b_1} = 17, \ldots, 22, b_1 = 25, R_1 = 20, Q_1 = 40, L_1 = 1, \lambda_2 = 5, c_2 = 10, h_2 = 0.5, \hat{b_2} = 15, \ldots, 20, b_2 = 70, R_2 = 5, Q_2 = 10, L_2 = 1$ and $t = 10$. In the first 6 snapshots, the value of $\hat{b_1}$ increases relative to $\hat{b_2}$. Meanwhile, for the last 6 snapshots, the value of $\hat{b_2}$ increases relative to $\hat{b_1}$. Snapshot numerical test set 1 shows the situation in which there is a bigger demand rate at location 1 than location 2. So stockout is more likely to occur at location 1 and transshipment is more likely to be made from location 2 to location 1 as there is a lower customer demand at location 2.

s/n	$\hat{b_1}$	$\hat{b_2}$	s/n	$\hat{b_1}$	$\hat{b_2}$
1	17	20	7	20	15
2	18	20	8	20	16
3	19	20	9	20	17
4	20	20	10	20	18
5	21	20	11	20	19
6	22	20	12	20	20

Table 5.11. Summary of the snapshot test set 1 of **HOT** model

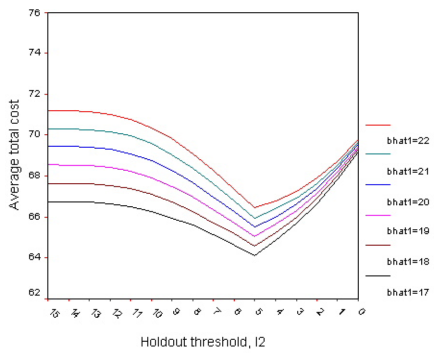

Figure 5.22. Test set 1 of **HOT** approximation on $\hat{b_1}$

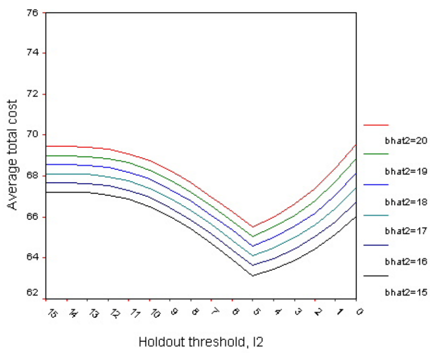

Figure 5.23. Test set 1 of **HOT** approximation on $\hat{b_2}$

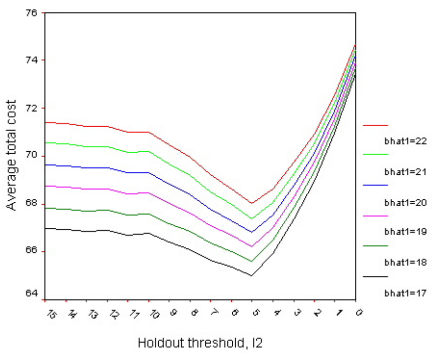

Figure 5.24. Test set 1 of original **HOT** simulation on $\hat{b_1}$

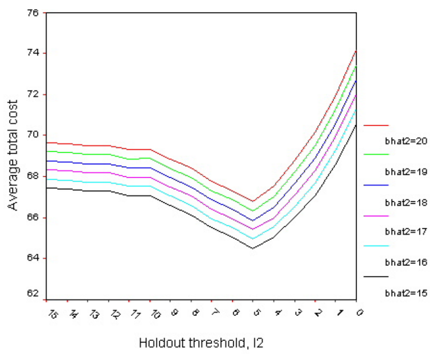

Figure 5.25. Test set 1 of original **HOT** simulation on $\hat{b_2}$

We use Figures 5.22 and 5.23 to show how the **HOT** model's estimates of the average total cost depend on the parameters $\hat{b_1}$ and $\hat{b_2}$. Figure 5.22 suggests that, for the snapshots 1 to 6, partial pooling with a holdout threshold of 5 (I_2=5) is always the optimal pooling policy when the value of $\hat{b_1}$ is ranging from 17 to 22. Meanwhile, for the snapshots 7 to 12 in Figure 5.23, partial pooling with a holdout threshold of 5 is also the optimal pooling policy when the value of $\hat{b_2}$ is ranging between 15 and 20. Hence, the **HOT** model predicts that partial pooling can deliver cost savings under the situations defined by the parameter test set 1.

Figures 5.24 and 5.25 show the average total cost for a range of holdout transshipment policies based on simulations of the original system for each snapshot. It is apparent from the figures that partial pooling is the optimal transshipment policy for all snapshots. It is interesting to note that the optimal holdout value is the same for all snapshots (I_2=5), and that this is the same value predicted by the **HOT** approximation model. For the **HOT** approximation model, predicted cost savings from partial pooling are up to 7.16% compared to no pooling (I_2=15) and up to 7.92% compared to complete pooling (I_2=0). For the simulation of the original holdout transshipment system, the actual cost savings from partial pooling are up to 5.03% (I_2=15) and up to 13.00% compared to complete pooling (I_2=0). Compared to the numerical results of the **TAP** model, the results of the **HOT** model show good prediction of the optimal holdout threshold values and improved predictions of the cost savings.

I_2	C	C_1	C_{o_1}	C_{h_1}	$C_{\hat{b}_1}$	C_{b_1}	C_2	C_{o_2}	C_{h_2}	$C_{\hat{b}_2}$	C_{b_2}	C_t	α_1	γ	α_2
15	69.42	46.13	5.00	20.61	17.77	2.76	23.29	5.00	2.80	8.77	6.72	0.00	0.96	0.00	0.91
14	69.42	46.12	5.00	20.61	17.76	2.75	23.29	5.00	2.80	8.77	6.72	0.01	0.96	0.00	0.91
13	69.39	46.05	5.00	20.61	17.69	2.74	23.30	5.00	2.79	8.78	6.73	0.04	0.96	0.00	0.91
12	69.29	45.83	5.00	20.62	17.50	2.71	23.34	5.01	2.79	8.80	6.74	0.13	0.96	0.00	0.91
11	69.09	45.35	4.99	20.63	17.08	2.65	23.40	5.03	2.78	8.83	6.77	0.33	0.96	0.00	0.91
10	68.75	44.55	4.98	20.66	16.37	2.54	23.52	5.07	2.76	8.89	6.81	0.67	0.96	0.00	0.91
9	68.27	43.42	4.97	20.69	15.37	2.38	23.70	5.11	2.74	8.97	6.87	1.15	0.96	0.01	0.91
8	67.68	42.00	4.96	20.74	14.12	2.19	23.93	5.17	2.72	9.08	6.96	1.75	0.96	0.01	0.91
7	67.00	40.36	4.94	20.80	12.66	1.96	24.19	5.24	2.70	9.20	7.05	2.44	0.96	0.01	0.91
6	66.26	38.55	4.92	20.86	11.06	1.72	24.50	5.32	2.69	9.34	7.15	3.21	0.96	0.02	0.91
5*	65.50	36.68	4.90	20.92	9.40	1.46	24.82	5.40	2.68	9.48	7.26	4.00	0.96	0.02	0.91
4	66.01	34.97	4.88	20.98	7.88	1.22	26.31	5.47	2.62	10.26	7.96	4.73	0.96	0.02	0.87
3	66.63	33.25	4.86	21.05	6.35	0.99	27.92	5.55	2.56	11.10	8.71	5.46	0.96	0.03	0.87
2	67.40	31.58	4.85	21.11	4.87	0.76	29.64	5.62	2.51	11.99	9.53	6.17	0.96	0.03	0.87
1	68.35	30.06	4.83	21.17	3.52	0.55	31.47	5.68	2.47	12.91	10.40	6.82	0.96	0.03	0.86
0	69.53	28.78	4.82	21.22	2.37	0.37	33.38	5.74	2.45	13.88	11.32	7.37	0.96	0.04	0.86

Table 5.12. Snapshot 4 in test set 1 of **HOT** model

In Table 5.12, we provide the result of snapshot 4 from Table 5.11 in detail which suggests that the optimal pooling policy is the partial pooling policy when the value of the holdout threshold I_2 equals to 5. However, it is worth noting that the predicted improvements on the cost from the optimal partial pooling policy are 6.14% and 5.99% with respect to complete pooling (I_2=0) and no pooling (I_2=15) respectively.

Snapshot numerical test set 2

Similarly, we use the 14 snapshots described in Table 5.13 corresponding to 14 different combinations on $\hat{b_1}$ and $\hat{b_2}$. In each snapshot, the other parameters are given values as follows: $\lambda_1 = 10, c_1 = 10, h_1 = 0.5, b_1 = 25, R_1 = 1, Q_1 = 40, L_1 = 1, \lambda_2 = 25, c_2 = 10, h_2 = 0.5, b_2 = 340, R_2 = 25, Q_2 = 40, L_2 = 1$ and $t = 10$. In the first 7 snapshot the value of $\hat{b_1}$ increases relative to $\hat{b_2}$. Meanwhile, for the last 7 snapshots, the value of $\hat{b_2}$ increases relative to $\hat{b_1}$. Snapshot numerical test set 2 reflects the situation in which there is a higher demand rate at location 2 than location 1. Hence, transshipment is less likely to be attractive as there is a higher customer demand at location 2.

s/n	$\hat{b_1}$	$\hat{b_2}$	s/n	$\hat{b_1}$	$\hat{b_2}$
1	17	20	8	20	14
2	18	20	9	20	15
3	19	20	10	20	16
4	20	20	11	20	17
5	21	20	12	20	18
6	22	20	13	20	19
7	23	20	14	20	20

Table 5.13. Summary of the snapshot test set 2 of **HOT** model

We use Figures 5.26 and 5.27 to show how the **HOT** model's estimates of the average total cost depend on the parameters $\hat{b_1}$ and $\hat{b_2}$. Figure 5.26 suggests that, for the snapshots 1 to 7, the partial pooling with holdout threshold 25 is always the optimal pooling policy when the value of $\hat{b_1}$ is ranging from 17 to 23. Meanwhile, for the snapshots 8 to 14 in Figure 5.27, the partial pooling with holdout threshold 25 is also the optimal pooling policy when the value of $\hat{b_2}$ is ranging between 14 and 20. Hence, the **HOT** model predicts that partial pooling can deliver cost savings under the situations defined by the parameter test set 2.

Figures 5.28 and 5.29 show the average total cost for a range of holdout transshipment policies based on simulations of the original system for each snapshot. It is apparent from the figures that partial pooling is the optimal transshipment policy for all snapshots. It is interesting to note that the optimal holdout value is the same for all snapshots (I_2=25), and this is the same value predicted by the **HOT** approximation model. For the **HOT** approximation model, cost savings from partial pooling are up to 13.75% compared to no pooling (I_2=65) and up to 14.66% compared to complete pooling (I_2=0). For the simulation of the original holdout transshipment system, the actual cost savings from partial pooling are up to 13.60% compared to no pooling (I_2=65) and up to 11.79% compared to complete pooling (I_2=0). Compared the numerical results of the **TAP** model, the results of the **HOT** model show good predictions of the optimal holdout threshold values and improved prediction of the cost approximations.

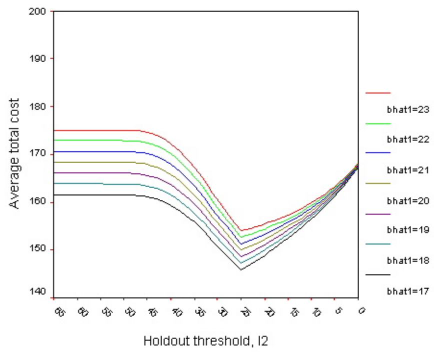

Figure 5.26. Test set 2 of **HOT** approximation on $\hat{b_1}$

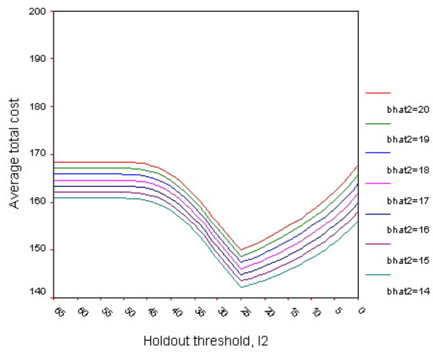

Figure 5.27. Test set 2 of **HOT** approximation on $\hat{b_2}$

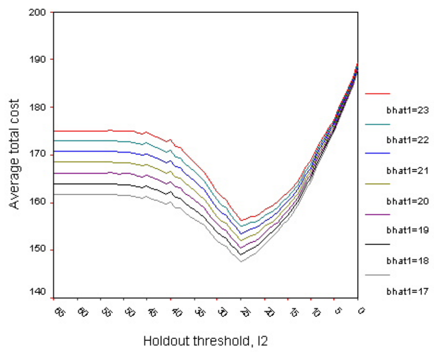

Figure 5.28. Test set 2 of original **HOT** simulation on $\hat{b_1}$

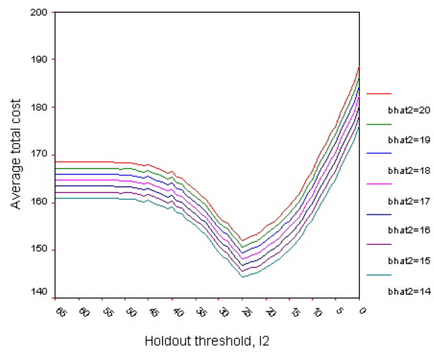

Figure 5.29. Test set 2 of original **HOT** simulation on $\hat{b_2}$

I_2	C	C_1	C_{o_1}	C_{h_1}	$C_{\hat{b}_1}$	C_{b_1}	C_2	C_{o_2}	C_{h_2}	$C_{\hat{b}_2}$	C_{b_2}	C_t	α_1	γ	α_2
40	165.76	75.54	2.47	6.36	42.51	24.21	89.20	6.28	10.27	24.95	47.70	1.02	0.78	0.01	0.94
39	165.06	74.48	2.47	6.38	41.82	23.81	89.27	6.28	10.26	24.98	47.75	1.30	0.78	0.01	0.94
38	164.26	73.27	2.46	6.41	41.03	23.37	89.36	6.29	10.25	25.01	47.81	1.62	0.78	0.02	0.94
37	163.37	71.93	2.45	6.45	40.16	22.87	89.46	6.30	10.24	25.05	47.88	1.98	0.78	0.02	0.94
36	162.41	70.48	2.44	6.48	39.22	22.33	89.57	6.31	10.23	25.09	47.95	2.36	0.78	0.02	0.94
35	161.39	68.93	2.43	6.52	38.21	21.76	89.69	6.32	10.21	25.13	48.03	2.77	0.78	0.03	0.94
34	160.32	67.30	2.42	6.57	37.16	21.16	89.81	6.33	10.20	25.17	48.11	3.20	0.78	0.03	0.94
33	159.21	65.62	2.41	6.61	36.06	20.53	89.94	6.34	10.19	25.21	48.20	3.65	0.78	0.04	0.94
32	158.07	63.88	2.40	6.66	34.94	19.89	90.07	6.35	10.18	25.26	48.28	4.11	0.78	0.04	0.94
31	156.91	62.12	2.39	6.70	33.79	19.24	90.21	6.36	10.17	25.31	48.37	4.58	0.79	0.05	0.94
30	155.74	60.34	2.37	6.75	32.63	18.58	90.35	6.38	10.16	25.35	48.46	5.05	0.79	0.05	0.94
29	154.57	58.55	2.36	6.80	31.47	17.92	90.49	6.39	10.15	25.40	48.55	5.53	0.79	0.06	0.94
28	153.39	56.76	2.35	6.85	30.30	17.25	90.63	6.40	10.14	25.45	48.64	6.00	0.79	0.06	0.94
27	152.22	54.96	2.34	6.90	29.13	16.59	90.77	6.41	10.13	25.49	48.73	6.48	0.79	0.06	0.94
26	151.06	53.18	2.33	6.95	27.97	15.93	90.92	6.42	10.13	25.54	48.82	6.95	0.79	0.07	0.94
25*	149.89	51.40	2.31	7.00	26.81	15.27	91.06	6.44	10.12	25.59	48.91	7.42	0.79	0.07	0.95
24	150.31	49.76	2.30	7.05	25.75	14.66	92.68	6.45	10.08	26.09	50.06	7.86	0.79	0.08	0.95
23	150.68	48.13	2.29	7.10	24.68	14.06	94.25	6.46	10.04	26.57	51.18	8.29	0.79	0.08	0.95
22	151.14	46.51	2.28	7.15	23.62	13.45	95.91	6.47	10.01	27.07	52.36	8.73	0.79	0.09	0.95
21	151.61	44.89	2.27	7.20	22.57	12.85	97.56	6.48	9.97	27.57	53.54	9.16	0.80	0.09	0.94
20	152.10	43.29	2.26	7.25	21.52	12.26	99.23	6.49	9.94	28.07	54.73	9.58	0.80	0.10	0.94
19	152.60	41.69	2.25	7.29	20.48	11.66	100.91	6.50	9.90	28.58	55.93	10.01	0.80	0.10	0.94
18	153.14	40.10	2.24	7.34	19.45	11.07	102.61	6.51	9.87	29.09	57.14	10.43	0.80	0.10	0.94
17	153.70	38.52	2.23	7.39	18.41	10.49	104.32	6.52	9.84	29.60	58.37	10.85	0.80	0.11	0.94
16	154.28	36.95	2.22	7.44	17.39	9.90	106.05	6.53	9.81	30.11	59.61	11.27	0.80	0.11	0.94
15	154.88	35.39	2.21	7.49	16.37	9.32	107.80	6.54	9.78	30.62	60.86	11.69	0.80	0.12	0.94
14	155.50	33.83	2.20	7.54	15.35	8.74	109.56	6.55	9.75	31.14	62.12	12.10	0.80	0.12	0.94
13	156.14	32.29	2.19	7.59	14.34	8.17	111.34	6.56	9.72	31.66	63.39	12.51	0.80	0.13	0.94
12	156.81	30.76	2.18	7.64	13.34	7.60	113.13	6.57	9.69	32.18	64.68	12.92	0.80	0.13	0.94
11	157.50	29.23	2.17	7.69	12.34	7.03	114.93	6.58	9.67	32.71	65.97	13.33	0.80	0.13	0.93
10	158.19	27.72	2.16	7.74	11.35	6.47	116.73	6.59	9.65	33.23	67.26	13.73	0.81	0.14	0.93
9	158.94	26.23	2.15	7.80	10.38	5.91	118.58	6.60	9.62	33.76	68.60	14.13	0.81	0.14	0.93
8	159.71	24.75	2.14	7.85	9.41	5.36	120.43	6.61	9.60	34.29	69.93	14.53	0.81	0.15	0.93
7	160.50	23.30	2.13	7.90	8.46	4.82	122.29	6.62	9.58	34.82	71.27	14.92	0.81	0.15	0.93
6	161.34	21.89	2.12	7.95	7.53	4.29	124.16	6.63	9.56	35.35	72.62	15.29	0.81	0.15	0.93
5	162.22	20.52	2.11	7.99	6.63	3.78	126.05	6.64	9.55	35.88	73.98	15.66	0.81	0.16	0.93
4	163.16	19.20	2.10	8.04	5.77	3.29	127.95	6.65	9.53	36.42	75.35	16.01	0.81	0.16	0.93
3	164.17	17.96	2.09	8.09	4.96	2.82	129.86	6.66	9.52	36.95	76.73	16.34	0.81	0.16	0.93
2	165.25	16.80	2.08	8.13	4.20	2.39	131.79	6.67	9.50	37.49	78.13	16.65	0.81	0.17	0.93
1	166.42	15.74	2.08	8.17	3.50	1.99	133.74	6.67	9.49	38.03	79.54	16.94	0.81	0.17	0.93
0	167.68	14.79	2.07	8.20	2.88	1.64	135.70	6.68	9.49	38.57	80.96	17.19	0.81	0.17	0.92

Table 5.14. Snapshot 4 in test set 2 of **HOT** model

In Table 5.14, we provide the result of snapshot 4 from Table 5.13 in detail which suggests that the optimal pooling policy is the partial pooling policy when the value of the holdout threshold I_2 equals to 25. However, it is worth noting that the predicted improvements on the cost from the optimal partial pooling policy are 11.87% and 12.30% with respect to complete pooling (I_2=0) and no pooling (I_2=65) respectively.

Snapshot numerical test set 3

Furthermore, we consider another situation in which location 1 has more intensive demands than location 2 and the stockout cost at location 2 is much greater than at location 1 ($\hat{b_2}$=90 while $\hat{b_1} \leqslant 45$). We use 8 snapshots described in Table 5.15 corresponding to 8 different combinations of $\hat{b_1}$ and $\hat{b_2}$. In each snapshot, the other parameters are given values as follows: $\lambda_1 = 20, c_1 = 7, h_1 = 5, b_1 = 15, L_1 = 1, \lambda_2 = 10, c_2 = 10, h_2 = 8, \hat{b_2} = 90, b_2 = 12, L_2 = 1,$ $R_1 = 24, Q_1 = 40, R_2 = 13, Q_2 = 20$ and $t = 20$. Snapshot numerical test set 3 reflects the situation in which there is a higher demand rate at location 1 than location 2, which could be an incentive to transshipment, and a higher stockout cost at location 2 that location 1, which could be a deterrent to transshipment. The stockout cost at location 1 increases from snapshot 1 to 8 and, hence, the system will become increasingly likely to favour transshipment. Hence, we might expect the optimal holdout threshold to be non-increasing from snapshot 1 to 8.

Snapshot S/N	$\hat{b_1}$	I_2^*	Snapshot S/N	$\hat{b_1}$	I_2^*
1	10	18	5	30	6
2	15	17	6	35	5
3	20	11	7	40	4
4	25	11	8	45	3

Table 5.15. Summary of snapshot test set 3 of **HOT** model

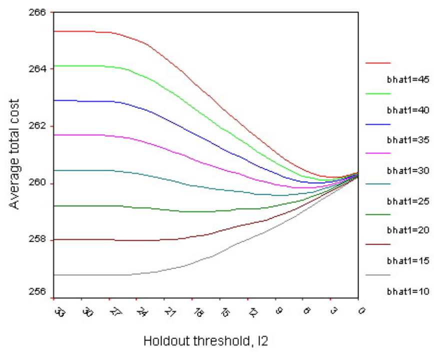

Figure 5.30. Test set 3 of **HOT** approximation on $\hat{b_1}$

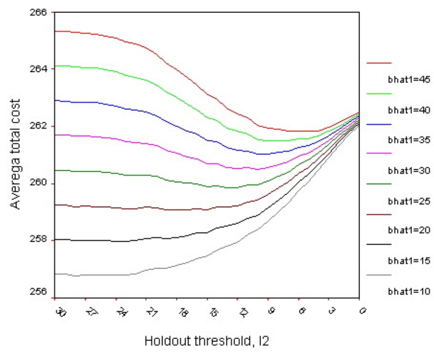

Figure 5.31. Test set 3 of original **HOT** simulation on $\hat{b_1}$

We use the Figure 5.30 to show how the **HOT** model's estimates of the average total cost depend on the parameter $\hat{b_1}$. This suggests that, for all snapshots, partial pooling is the optimal pooling policy when $\hat{b_1}$ is ranging from 10 to 45. Table 5.15 shows that the **HOT** model's prediction of the optimal holdout threshold for each snapshot. As expected from the choice of problem parameters in the snapshots, the holdout threshold decreases from 18 in snapshot 1 to 3 in snapshot 8. Figure 5.31 shows the average total cost based on simulations of the original system for each snapshot. It is apparent that partial pooling is the optimal transshipment pooling for all snapshots. For the **HOT** approximation model, the predicted cost savings from partial pooling are only up to 1.97% compared to complete pooling (I_2=33) and up to 1.33% compared to complete pooling (I_2=0). For the simulation of the original holdout transshipment system, the actual cost savings from partial pooling are up to 1.35% compared to no pooling (I_2=33) and up to 2.06% compared to complete pooling (I_2=0).

I_2	C	C_1	C_{o_1}	C_{h_1}	$C_{\hat{b}_1}$	C_{b_1}	C_2	C_{o_2}	C_{h_2}	$C_{\hat{b}_2}$	C_{b_2}	C_t	α_1	γ	α_2
30	260.46	132.55	3.50	122.61	6.09	0.34	127.91	5.00	108.16	14.51	0.24	0.00	0.99	0.01	0.98
29	260.45	132.54	3.50	122.61	6.09	0.34	127.91	5.00	108.16	14.51	0.24	0.00	0.99	0.01	0.98
28	260.45	132.54	3.50	122.61	6.08	0.34	127.90	5.00	108.15	14.51	0.24	0.01	0.99	0.01	0.98
27	260.44	132.52	3.50	122.62	6.06	0.34	127.90	5.00	108.14	14.51	0.24	0.03	0.99	0.01	0.98
26	260.42	132.48	3.50	122.62	6.02	0.34	127.88	5.00	108.13	14.52	0.24	0.06	0.99	0.01	0.98
25	260.39	132.42	3.50	122.63	5.96	0.33	127.86	5.00	108.10	14.52	0.24	0.11	0.99	0.01	0.98
24	260.34	132.32	3.50	122.64	5.86	0.33	127.83	5.00	108.06	14.52	0.24	0.19	0.99	0.01	0.98
23	260.29	132.19	3.50	122.66	5.72	0.32	127.80	5.01	108.02	14.53	0.24	0.30	0.99	0.01	0.98
22	260.22	132.03	3.50	122.68	5.54	0.31	127.76	5.01	107.97	14.54	0.24	0.44	0.99	0.01	0.98
21	260.15	131.83	3.49	122.70	5.33	0.30	127.72	5.02	107.91	14.56	0.24	0.60	0.99	0.01	0.98
20	260.07	131.60	3.49	122.73	5.09	0.28	127.68	5.02	107.85	14.57	0.24	0.79	0.99	0.01	0.98
19	260.00	131.36	3.49	122.76	4.83	0.27	127.65	5.02	107.80	14.58	0.24	1.00	0.99	0.01	0.98
18	259.93	131.10	3.49	122.80	4.55	0.25	127.62	5.03	107.75	14.60	0.24	1.22	0.99	0.01	0.98
17	259.87	130.83	3.49	122.84	5.26	0.24	127.60	5.04	107.71	14.62	0.24	1.45	0.99	0.01	0.98
16	259.82	130.55	3.49	122.87	3.97	0.22	127.59	5.04	107.68	14.63	0.24	1.68	0.99	0.01	0.98
15	259.78	130.27	3.48	122.91	3.67	0.20	127.59	5.05	107.65	14.65	0.24	1.92	0.99	0.01	0.98
14	259.75	129.99	3.48	122.95	3.37	0.19	127.60	5.05	107.64	14.67	0.24	2.16	0.99	0.01	0.98
13	259.72	129.71	3.48	122.99	3.07	0.17	127.62	5.06	107.63	14.69	0.24	2.39	0.99	0.01	0.98
12	259.68	129.43	3.48	123.03	2.77	0.15	127.62	5.07	107.46	14.85	0.24	2.63	0.99	0.01	0.98
11	259.63	129.16	3.47	123.07	2.47	0.14	127.61	5.07	107.27	15.01	0.25	2.87	0.99	0.00	0.98
10	259.61	128.88	3.47	123.12	2.17	0.12	127.62	5.08	107.11	15.18	0.25	3.10	0.99	0.00	0.98
9	259.59	128.61	3.47	123.16	1.88	0.10	127.65	5.08	106.96	15.35	0.25	3.33	0.99	0.00	0.98
8*	259.59	128.34	3.47	123.20	1.58	0.09	127.69	5.09	106.82	15.52	0.26	3.57	0.99	0.00	0.98
7	259.61	128.08	3.47	123.24	1.30	0.07	127.74	5.09	106.69	15.69	0.26	3.79	0.99	0.00	0.98
6	259.64	127.84	3.47	123.28	1.04	0.06	127.81	5.10	106.58	15.87	0.26	4.00	0.99	0.00	0.98
5	259.70	127.62	3.46	123.31	0.80	0.04	127.90	5.10	106.49	16.04	0.27	4.19	0.99	0.00	0.98
4	259.78	127.43	3.46	123.34	0.59	0.03	128.00	5.11	106.41	16.21	0.27	4.35	0.99	0.00	0.98
3	259.88	127.27	3.46	123.37	0.42	0.02	128.12	5.11	106.36	16.38	0.27	4.49	0.99	0.00	0.98
2	260.00	127.15	3.46	123.39	0.28	0.02	128.26	5.11	106.32	16.55	0.28	4.59	0.99	0.00	0.98
1	260.14	127.06	3.46	123.41	0.18	0.01	128.41	5.12	106.29	16.72	0.28	4.67	0.99	0.00	0.98
0	260.29	126.99	3.46	123.42	0.11	0.01	128.57	5.12	106.28	16.89	0.28	4.73	0.99	0.00	0.98

Table 5.16. Snapshot 5 of test set 4 of **HOT** model

In Table 5.16, we provide the results of snapshot 4 from Table 5.15 in detail which suggests that the optimal pooling policy is the partial pooling policy corresponding to holdout transshipment threshold I_2 equal to 8. However, the predicted improvements on the cost from the optimal partial pooling policy are only 0.27% and 0.33% with respect to complete pooling (I_2=0) and no pooling (I_2=33) respectively.

Table 5.17 compares the predicted optimal holdout threshold from the **HOT** approximation (I_2-HOT) with the optimal holdout threshold from simulation (I_2-SIM) for 8 snapshots. This shows that the predicted value does not coincide to the optimal value in many cases. The table also shows the simulated average cost rate using the predicted holdout value (C_{HOT}) and the optimal holdout value (C_{SIM}).

	I_2					I_2			
s/n	HOT	SIM	C_{HOT}	C_{SIM}	s/n	HOT	SIM	C_{HOT}	C_{SIM}
1	29	28	256.80	256.77	5	6	10	259.85	260.48
2	23	23	258.00	257.95	6	4	10	260.02	261.00
3	17	15	259.02	259.07	7	3	7	260.13	261.47
4	8	12	259.59	259.86	8	3	5	260.21	261.80

Table 5.17. Comparison of predicted and optimal I_2 for snapshot test set 3

We conclude that the **HOT** model has a poor prediction accuracy on the optimal I_2 under the situations of test set 3. However, when we compare the costs C_{HOT} and C_{SIM} in Table 5.17 for each snapshot in test set 3, the difference between the actual total cost for the optimal holdout threshold predicted by the **HOT** model and the optimal total cost from simulation with between 0.00% and 0.61%. This suggests that although the predicted optimal holdout threshold is not always accurate, the **HOT** model is a good approximation model with an improved accuracy.

5.6 Conclusions

In Section 5.5, we develop an approximation model which reflects the holdout transshipment policy explicitly. Instead of the transshipment agreement probability in the **TAP** model, we introduce the decision variable I_2 to represent the holdout threshold in the **HOT** model. Meanwhile, we still use the approach of decomposing the two-location system into two independent locations with non-constant demand rates.

The **HOT** approximation model provides a more insightful study on the transshipment interactions between the two locations due to the given holdout transshipment policy. The verification results of the **HOT** model show that the derived explicit expressions for a range of system performance measures are highly consistent with estimates of these system performance measures by simulation. More importantly, by using the same parameters for the snapshot test set 1 and 2, the numerical results for the **TAP** and **HOT** models show that the **HOT** model can be a more accurate approximation model than the two approximation models based on the transshipment agreement probability.

In addition, as we reported in snapshot test set 2, there is evidence that partial pooling can lead to significant cost savings compared to no pooling and complete pooling. (up to 13.75% and 14.66% respectively) and the **HOT** model can identity the optimal holdout threshold for partial pooling. Hence, the **HOT** approximation does help to deliver great cost savings for the two-location system under some situations.

Meanwhile, the framework developed using the property of renewal theory to analyse a single cycle can be applied to other systems. For example, we can extend this modelling framework to a system which has more then two locations. For multi-location systems which could have more than two locations, we only need to derive the demand and inventory level distributions during a single cycle at each location, then modify those demand rates and inventory level distributions due to the transshipment interactions between the locations.

The approximation models developed in this chapter are based on a number of assumptions which may not always hold. Firstly, for the **TAP** and **HOT** model, we give a strong assumption that we require that there is at most one outstanding order at any time in a cycle. This assumption is too strong and not realistic for the real problem. Nevertheless, this modelling approach will be difficult if we consider more than one outstanding orders during the cycle. Because we use the Poisson demand process to model customer demands at each location, an **Erlang** distribution is used to model the time taken for the inventory level to fall by a certain amount due to a specific Poisson demand process. If we choose a different demand process, this time condition needs to be reviewed.

Secondly, from the numerical results in **HOT** approximation model, we learn that the estimation of the optimal holdout threshold I_2 and approximations of a range of system performance measures are not accurate with the simulation counterparts under some situations. Therefore, we need to develop further our approximation approach to improve the accuracy of the model.

Thirdly, it is uncommon in our results to find an optimal partial pooling policy contributing to a significant cost saving. To get more evidence of significant cost savings, we need to explore further numerical experiments. The results shown could mean that we would not expect to see huge cost savings with the holdout transshipment policy commonly. However, the average total cost would not lead to more costs to the system under the holdout transshipment policy. We could say that this holdout transshipment policy has no harm to the general inventory control management but provides more flexibility.

Overall, we conclude that the **HOT** model provides a fresh and better approach to understand the impact of the holdout transshipment policy to the two-location inventory control system. It is an improved accurate approximation with an extendable framework. However there are opportunities for further enhancement to make it more robust. So far, all approximation models we have developed are based on the decomposition approach, it is worth to consider a different modelling technique to help us get more insights of the system performance and behaviours under the holdout transshipment policy.

Glossary

TAP model: the approximation model where the transshipment decision is still made randomly according to the transshipment agreement probability
HOT model: the approximation model which assumes a holdout transshipment policy explicitly
R_k: reorder point at location k
Q_k: replenishment order quantity at location k
L_k: lead time at location k
$F(x, \lambda)$: the probability that a Poisson random variable with mean λ does not exceed x
z: transshipment agreement probability
X_k: the inventory level at location k
$X_k(R_k)$: the inventory level at the end of a cycle at location k
ϕ: the probability that location 2 can consider a transshipment request
θ: the probability that location 1 needs to make a transshipment request
I_2: the decision variable of holdout transshipment threshold at location 2
ω: the probability that the inventory level at location 2 is greater than the threshold value I_2

Chapter 6
Markov decision process approximation

6.1 Introduction

Until now we have developed a series of approximation models for a two-location inventory system with unidirectional transshipment via decomposition approaches. Although some interesting results have been obtained, these models only consider holdout transshipment policies and, even then, might not find the optimal holdout level because they do not exactly capture all interactions between two locations in the system. In this chapter, we aim to examine the potential benefits of general transshipment policies and develop a methodology that captures interactions between locations more closely. Instead of decomposing the two-location system into two independent locations, we model the entire system as a semi-Markov decision process (**SMDP**).

The key strength of the **SMDP** approach for the analysis of the two-location system is that the transshipment decision can be allowed to depend on the actual inventory levels at two locations. Hence, it is possible to consider a wider class of transshipment policy than the simple holdout policy. The **SMDP** approach also makes it possible to consider the replenishment decision in more details. However, it is not practical to model fixed replenishment lead times using the **SMDP** approach due to the complexity of the resulting models. Hence, we consider two stochastic models of replenishment lead time.

Firstly we assume that the replenishment lead time has an exponential distribution. This is convenient for a **SMDP** model because of the memory-less property, i.e. the time until an order is delivered does not depend on the time that has passed since the order was placed. However this does not model the fixed replenishment lead time effectively, so we develop a second model in which we assume that the replenishment lead time has a phase-type distribution. More precisely, we assume that the replenishment lead time is the sum of a fixed number of independent and identically distributed (**IID**) exponential random variables. The number of variables in the sum is referred to as the number of phases. As the number of phases increases, the phase-type distribution more closely approximates a fixed replenishment lead time. The phase-type distribution is also convenient for a **SMDP** model because the time until the end of a phase does not depend on the time that has passed since the start of the phase. Hence, to model the time until the delivery of an order, it is only necessary to know how many phases have been completed since the order was placed.

Using a **SMDP** approach, it is not possible to derive explicit expressions for each component of the expected total cost rate. Rather, using the principle of optimality from Bellman (1957) , the expected total cost rate is calculated in an iterative manner through the optimality equation. Due to the popularity of the notations used by Tijms and Puterman in their books on stochastic modelling: Tijms (1986, 1994, 1995 and 2003) and Puterman (1994), we use their notations in our **SMDP** models.

The remainder of the Chapter is organised as follows. We provide the general assumptions for all **SMDP** models of this chapter in Section 6.2. For the exponential lead time, we present two **SMDP** models with general transshipment policy and holdout transshipment policy in Sections 6.3. Some numerical examples of the **SMDP** models with exponential lead time are presented in Section 6.3.3. For the phase-type lead time, we present two **SMDP** models with general transshipment policy and holdout transshipment policy in sections 6.4. Some numerical examples of the **SMDP** models with phase-type lead time are presented in Section 6.4.3. In Section 6.5, we give a conclusion for all **SMDP** models in this chapter.

6.2 Assumptions

We consider a two-location inventory system with unidirectional transshipment from location 2 to location 1. At location k, k = 1, 2, a Poisson demand process with rate λ_k is used to model the customer demand. The fixed order cost at location $k\,(k=1,2)$ is c_k and the cost of holding inventory at location k is h_k per item per time unit. Demand that cannot be met from on-hand inventory may be backordered. When a backorder is placed at location k, there is one-off stockout cost $\hat{b_k}$ and a further backorder cost of b_k per time unit until the backorder is satisfied. Alternatively unmet demands at location 1 may be met by transshipment from location 2. Transshipment is assumed to be instantaneous and involves a cost of t per item. In addition, we have following explicit assumptions for the **SMDP** model.

At each location k, k = 1, 2, we assume that the inventory level at location k can not exceed the storage capacity M_k and the maximum number of backorders that can be placed is N_k. This assumption is necessary for a finite state model. If a demand occurs at a location after the maximum number of backorders has been reached and, in the case of location 1, transshipment is not possible, the demand results in a lost sale at a cost of B_k per item. Ideally, we would like to choose values of N_k and M_k large enough not to impose any practical restrictions on the choice of policy. However, increasing N_k and M_k increases the number of states in the model and hence the computational complexity of the model. We therefore experiment with the choice of N_k and M_k to find a balance between the tractability of the model and the impact of N_k and M_k on the optimal cost and policy.

Although it is possible to consider an $(R,\ Q)$ replenishment order policy in our **SMDP** models, we introduce a more general form of the replenishment order policy, because it is straightforward to do this for a **SMDP** model while it is not for the earlier approximation models. We still assume that the system only allows one outstanding order at each location. Whenever a replenishment order is placed at location k, $k = 1, 2,$ the order quantity $q_k > 0$ must be one of the finite number of values in Q_k, the set of possible order quantities for location k. The storage capacity imposes a further constraint on the size of a replenishment order. The current inventory level plus the order quantity must not exceed the storage capacity. Otherwise it is possible (for example, if there is no demand during the replenishment lead time) that the storage capacity will be violated when the order is delivered.

The assumption of fixed replenishment lead time, which we used in our earlier approximation models is not suitable for a **SMDP** model. This is because we would need to define a continuous state variable for each location to keep track of the time until the delivery of any outstanding replenishment order. Due to the complexity of the resulting model, it is unlikely that analysis would lead to any useful insight about the transshipment and replenishment decisions in the original inventory system.

Instead of the fixed replenishment lead time, we consider two stochastic models of replenishment lead time which overcome this problem by exploiting the memory-less property of the exponential distribution. Define $\frac{1}{\mu_k}$ to be the mean replenishment lead time at location k. In the first model, the replenishment lead time at location k is assumed to be an exponential random variable with mean $\frac{1}{\mu_k}$. The second model uses a phase-type distribution (see P261, Tijms (1995)) and the lead time at location k is assumed to be the sum of W_k independent exponential random variables each with mean $\frac{1}{W_k\mu_k}$. We call it the phase-type distribution of the lead time. Each random variable represents a phase of the lead time and W_k is referred to as the number of phases.

In both cases, due to the memory-less property of exponential random variable, the delivery of outstanding orders can be modelled without knowledge of when the orders were placed. The assumption of the exponential lead time results in a simple model, but it is not always well-suited to model the lead time and it is certainly very different from the fixed lead time assumed in the previous models. The assumption of phase-type lead time can sometimes be more appropriate, but it leads to models which are computationally more demanding. Of particular relevance to this work is the property that the phase-type model of lead time more closely approximates a fixed lead time as the number of phases increases.

For the models with holdout transshipment policy, the decision variable I_2 denotes the holdout threshold at location 2. To help us to define relevant terms, we denote the indicator function $\delta(x) = \{^{0 \text{ if } x \leqslant 0}_{1 \text{ if } x > 0}$ and its complement function $\hat{\delta}(x) = 1 - \delta(x) = \{^{1 \text{ if } x \leqslant 0}_{0 \text{ if } x > 0}$. Throughout this chapter the terms decision and action are interchangeable.

6.3 SMDP models with exponential lead time

In a **SMDP** model, the state of the process is observed at random points in time known as decision epochs. At each decision epoch, the state of the system is observed and a decision is made. The state and decision may influence the time until the next decision epoch, the costs incurred until the next decision epoch and the state of the process at the next decision epoch.

6.3.1 General transshipment policy formulation

State space

For each location, we need one state variable to model the inventory process and one state variable to model the replenishment order process. Let i_k denote the inventory level at location k, $k = 1, 2$, where negative values indicate outstanding backorders. Let $q_k = 0$ represent the situation where there is no outstanding replenishment order at location k. Let $q_k > 0$ represent the situation where there is one outstanding replenishment order for q_k items at location $k, k = 1, 2$.

Definition 6.1. *Under our assumptions, the state space is given by*

$$I = \{(i_1, i_2, q_1, q_2) \colon -N_k \leqslant i_k \leqslant M_k, i_k + q_k \leqslant M_k \text{ and } q_k \in \{0\} \cup Q_k \text{ for } k = 1, 2\}$$

From the assumption, we conclude that the state space I is finite.

Action space

We need two decision variables to model the order decisions at the two locations and one decision variable to model the transshipment decision from location 2 to location 1. If there is no outstanding order at location k (i.e. $q_k = 0$), we can choose not to order or to place an order for any quantity in the set Q_k that will not violate the storage capacity constraint. Because we have the assumption which only allows one outstanding order, if there is already an outstanding order at location k (i.e. $q_k > 0$), we cannot place another order. We model this by setting the replenishment order decision equal to the quantity of the outstanding order (i.e. q_k). Let $X_k(i)$ denote the set of possible replenishment order decisions at location k when the state of the process is $i = (i_1, i_2, q_1, q_2)$. It follows that

$$X_k(i) = \{q_k\} \text{ if } q_k > 0 \quad \text{and } X_k(i) = \{x \colon i_k + x \leqslant M_{k,} x \in \{0\} \cup Q_k\} \text{ if } q_k = 0$$

Let $q_k^{'}$ represent the decision taken regarding the order process at location $k, k = 1, 2$. For the given state $i \in I$, $q_k^{'}$ can take any value in the set $X_k(i)$.

Let $i = (i_1, i_2, q_1, q_2)$ be the current state of the system. We consider the decision of how to satisfy a demand at location 1. Due to the memory-less property of all the random variables in the model, this decision remains valid until one of the state variables changes. The inventory levels at the two locations determine whether or not transshipment is a feasible way of satisfying a demand at location 1. If $i_1 > 0$, location 1 would meet the demand from local stock and so there is no possibility of a transshipment. If $i_2 \leqslant 0$, location 2 has no stock to tranship and again transshipment is impossible. If $i_1 = -N_1$ and $i_2 > 0$, then transshipment would be essential to avoid a lost sale at location 1. Otherwise a transshipment decision would arise. It is convenient for some of the later expressions to use 1 to denote no transshipment and 0 to denote transshipment. Let $D(i)$ denote the set of possible transshipment decisions when the system is in state i. It follows that

$$D(i) = \{1\} \quad \text{if } i_1 > 0 \text{ or } i_2 \leqslant 0,\ D(i) = \{0,1\} \text{ if } -N_1 < i_1 \leqslant 0 \text{ and } i_2 > 0$$

$$\text{and } D(i) = \{0\} \text{ if } -N_1 = i_1 \text{ and } i_2 > 0 \qquad \text{for } i \in I \tag{6.1}$$

Let d represent the decision with regard to transshipment from location 2 to location 1. For given state $i \in I$, d can take any value in the set of $D(i)$.

Definition 6.2. *Under our assumptions, the action space in state i is given by*

$$A(i) = \{(q_1^{'}, q_2^{'}, d) \colon q_k^{'} \in X_k(i) \text{ for } k = 1, 2, \, d \in D(i)\}$$

From the assumption, we conclude that the action space $A(i)$ is finite.

Decision epoch

Decisions need to be taken when a demand occurs at one of the two locations and when a delivery of a replenishment order is made to one of the two locations. It is convenient to introduce the concept of an event to mean any occurrence of a demand or a delivery in the system. We can then say that the next decision epoch occurs at the time of the next event.

We denote $\tau_i(a)$ as the expected time until the next decision epoch when decision a is chosen in state i. If there are outstanding orders at both locations, then the next event could be a demand at location 1, a demand at location 2, a delivery at location 1 or a delivery at location 2. By the memory-less property of the exponential distribution, the time until the next decision epoch is equal to the minimum of four independent exponentially distributed random variables, namely the time until the next demand at location 1, the time until the next demand at location 2, the time until the next delivery at location 1 and the time until the next delivery at location 2. It follows from further properties of the exponential distribution that the time until the next decision epoch has an exponential distribution with scale parameter $\lambda_1 + \lambda_2 + \mu_1 + \mu_2$. Hence, if $q_1^{'} > 0$ and $q_2^{'} > 0$, then $\tau_i(a) = \frac{1}{\lambda_1 + \lambda_2 + \mu_1 + \mu_2}$. If there is an outstanding order at location 1 and no outstanding order at location 2, the next event could be a demand at location 1, a demand at location 2 or a delivery at location 1. Following the same argument as above, the time until the next decision epoch has an exponential distribution with scale parameter $\lambda_1 + \lambda_2 + \mu_1$. Hence, if $q_1^{'} > 0$ and $q_2^{'} = 0$, then $\tau_i(a) = \frac{1}{\lambda_1 + \lambda_2 + \mu_1}$. Similarly, if $q_1^{'} = 0$ and $q_2^{'} > 0$, then $\tau_i(a) = \frac{1}{\lambda_1 + \lambda_2 + \mu_2}$. Finally, if $q_1^{'} = 0$ and $q_2^{'} = 0$, the time until the next decision epoch has an exponential distribution with scale parameter $\lambda_1 + \lambda_2$ and $\tau_i(a) = \frac{1}{\lambda_1 + \lambda_2}$. These four cases can be combined into a single expression as follows.

$$\tau_i(a) = \frac{1}{\lambda_1 + \lambda_2 + \delta(q_1^{'})\mu_1 + \delta(q_2^{'})\mu_2} \tag{6.2}$$

The value iteration algorithm that we use to solve the **SMDP** model is more straightforward when the expected time until the next decision epoch does not depend on the state or action. For the two-location inventory process, it is possible to do this with the introduction of fictitious decision epochs. When there is no outstanding order at location k (i.e. $q_k = 0$), we introduce a fictitious event after a time that has an exponential distribution with mean $\frac{1}{\mu_k}$. This event can be thought of as the delivery of a zero replenishment order at location k and will leave the state unchanged. Hence, whatever the state and action, the next event can always be a demand at location 1, a demand at location 2, a delivery (of a possible zero) replenishment order at location 1 or a delivery (of a possible zero) replenishment order at location 2. The time until each of the possible events is exponential with scale parameter independent of the state and action. Hence, the expected time until next decision epoch no longer depends on state i or decision a. We can now define τ as the expected time until the next decision epoch where

$$\tau = \frac{1}{\lambda_1 + \lambda_2 + \mu_1 + \mu_2} \tag{6.3}$$

Transition probabilities

We need to define the probability that the process is in state j at the next decision epoch given that action a is chosen in state i at the current decision epoch. We define the transition probabilities for the model implicitly by considering which of the four possible events occurs at the next decision epoch. Each event results in a transition to a single state at the next decision epoch, but it is possible for several events to result in a transition to the same state. Therefore the probability of the events occurring do not always correspond directly to transition probabilities. However, the transition probabilities can be deduced from the probabilities of the events occurring.

Consider a decision epoch at which the system is in state $i = (i_1, i_2, q_1, q_2)$ and action $a = (q_1^{'}, q_2^{'}, d)$ is chosen. The probability that the next event is a demand at location 1 is $\lambda_1\tau$. If $d = 0$, this demand will be met by transshipment from location 2 and the inventory levels at the two locations at the next decision epoch will be i_1 and $i_2 - 1$ respectively. If $d = 1$ and $i_1 > -N_1$, the demand will be met by local stock or a backorder at location 1. In both cases, the inventory level at location 1 at the next decision epoch will be $i_1 - 1$. However, if $d = 1$ and $i_1 = -N_1$, the demand will result in a lost sale and the inventory level at location 1 at the next decision epoch will be unchanged. If $d = 1$, the inventory level at location 2 at the next decision epoch will be unchanged. If there is an outstanding order at location k, then $q_k^{'} = q_k > 0$ and the order will still be outstanding at the next decision epoch. If an order is placed at location k, then $q_k^{'} > q_k = 0$ and this order will be outstanding at the next decision epoch. If there is no outstanding order and no order is placed, then $q_k^{'} = q_k = 0$ and there will be no outstanding order at the next decision epoch. Hence, at the next decision epoch, the state variable describing the replenishment order process at location k will be equal to $q_k^{'}$. Therefore, when the next event is a demand at location 1, the state of the process at the next decision epoch is $(\max(-N_1, i_1 - d),\ i_2 - (1-d), q_1^{'}, q_2^{'})$.

The probability that the next event is a demand at location 2 is $\lambda_2 \tau$. As there is no transshipment into location 2, all demand has to be dealt with locally. If $i_2 > -N_2$, the demand will be met by local stock or a backrorder at location 2 and the inventory level at location 2 at the next decision epoch will be $i_2 - 1$. If $i_2 = -N_2$, the demand will result in a lost sale and no change in inventory level. As above the state variable describing the replenishment order process at location 1 will be $q_k^{'}$ at the next decision epoch. Therefore, when the next event is a demand at location 2, the state of the process at the next decision epoch is $(i_1, \max(-N_2, i_2 - 1), q_1^{'}, q_2^{'})$.

The probability that the next event is a delivery at location 1 is $\mu_1 \tau$. If $q_1^{'} = 0$, this is a fictitious delivery and there is no change in state. If $q_1^{'} > 0$, q_1 items are added to the inventory at location 1 and, due to our assumptions, there can not be any outstanding orders at location 1. A delivery at location 1 will not affect the state variables relating to location 2. Therefore, when the next event is a delivery at location 1, the state of the process at the next decision epoch is $(i_1 + q_1,\ i_2,\ 0,\ q_2^{'})$. The probability that the next event is a delivery at location 2 is $\mu_2 \tau$ and, following a similar argument to the above, the state of the process at the next decision epoch is $(i_1, i_2 + q_2, q_1', 0)$.

Immediate cost

We need to define the expected cost incurred until the next decision epoch when action a is chosen in state i at the current decision epoch. We denote this expected cost by $c_i(a)$. This so called immediate cost is the short-run economic consequence of making a given decision in a given state. In our model, the cost consists of the fixed order cost, holding cost, backorder cost, stockout cost, lost sale penalty cost and transshipment cost.

Under our assumptions, a replenishment order is placed at location k at the current decision epoch if and only if $q_k^{'} > q_k$. Replenishment orders can only be placed at decision epochs, so the fixed order cost incurred until the next decision epoch is equal to $c_k \delta(q_k^{'} - q_k)$ at location $k, k = 1, 2$.

The inventory levels at the two locations do not change between decisions epochs. Hence, if $i_k > 0$, holding cost is incurred at location k at a constant rate of $i_k h_k$ per time unit. Similarly, if $i_k < 0$, backorder cost is incurred at location k at a rate of $-i_k b_k$ per time unit. Therefore, the expected holding cost incurred at location k until the next decision epoch is equal to $\delta(i_k) i_k \tau h_k$. and the expected backorder cost incurred at location k until the next decision epoch is equal to $-\hat{\delta}(i_k) i_k \tau b_k$.

From the definition of a decision epoch, there can be at most one demand in the system in the time until the next decision epoch and this demand can only occur at the time of the next decision epoch. The probability that the next event is a demand at location 1 is $\lambda_1 \tau$. Assume that the next event is a demand at location 1. If $i_1 \leqslant 0$ and $d = 1$, the demand will result in a stockout, and hence a stockout cost of $\hat{b_1}$ at location 1. Otherwise (i.e. if $i_1 > 0$ or $d = 0$), the demand will be satisfied and there is no stockout cost. Therefore, the expected stockout cost incurred at location 1 until the next decision epoch is equal to $\hat{\delta}(i_1)\lambda_1 \tau d \hat{b_1}$. If $d = 0$, the demand will be met by transshipment from location 2 at a cost of t. While if $d = 1$, there will be no transshipment cost incurred. Therefore, the expected transshipment cost until the next decision epoch is equal to $\lambda_1 \tau (1 - d) t$.

The probability that the next event is a demand at location 2 is $\lambda_2 \tau$. Assume now that the next event is a demand at location 2. The demand will result in a stockout, and hence a cost of $\hat{b_2}$ at location 2 if and only if $i_2 \leqslant 0$. Therefore, the expected stockout cost incurred at location 2 until the next decision epoch is equal to $\hat{\delta}(i_2)\lambda_2 \tau \hat{b_2}$.

When $i_k = -N_k$, a stockout at location k will result in a lost sale, and hence an additional cost of $B_k - \hat{b_k}$ at location k. Lost sales cannot occur at location k under any other circumstances (i.e. when there is no stockout at location k or $i_k > -N_k$). Note that $N_k + i_k > 0$ if $i_k > -N_k$ and $N_k + i_k = 0$ if $i_k = -N_k$. Therefore, the expected additional cost due to lost sales until the next decision epoch is equal to $\hat{\delta}(N_1 + i_1)\lambda_1 \tau d (B_1 - \hat{b_1})$ and $\hat{\delta}(N_2 + i_2)\lambda_2 \tau (B_2 - \hat{b_2})$ at location 1 and location 2 respectively.

Combining the cost components above, we conclude that the expected cost until the next decision epoch when action a is chosen in state i at the current decision epoch is as follows.

$$c_i(a) = \{\sum_{k=1}^{2} \{c_k \delta(q_k^{'} - q_k) + [h_k \delta(i_k) - b_k \hat{\delta}(i_k)] i_k \tau\} + \lambda_1 \tau d \{\hat{\delta}(i_1)\hat{b_1} + \hat{\delta}(N_1 + i_1)(B_1 - \hat{b_1})\} +$$

$$\lambda_2 \tau \{\hat{\delta}(i_2)\hat{b_2} + \hat{\delta}(N_2 + i_2)(B_2 - \hat{b_2})\} + \lambda_1 \tau (1 - d) t \tag{6.4}$$

Value iteration algorithm

Following standard techniques for **SMDP** models (see for example Tijms (2003)), the optimal long-run average cost per time unit can be calculated using the value iteration algorithm. This approach applies a data transformation to the **SMDP** model to create a discrete-time Markov decision process (**DMDP**) model. Under mild assumptions, which we will assume hold for the model being examined, the **DMDP** model has the same class of stationary policies and the same long-run average expected cost per time unit as the original **SMDP** model.

When the expected time until the next decision epoch does not depend on the state or action, the data transformation simply involves dividing the immediate costs by the expected time between successive decision epochs. Hence, we transform the immediate costs as follows.

$$C_i(a) = \frac{c_i(a)}{\tau} \text{ for } i \in I \text{ and } a \in A(i)$$

Define $P_i^e(a)$ to be the probability that, when action a is taken in state i, the next event is a demand at location 1, a demand at location 2, a delivery to location 1 or delivery to location 2 for e = 1, 2, 3 or 4 respectively. Define $J_i^e(a)$ to be the state of the process at the next decision epoch when action a is taken in state i and the next event is a demand at location 1, a demand at location 2, a delivery to location 1 or delivery to location 2 for $e = 1, 2, 3$ or 4 respectively.

A value-iteration algorithm for the **SMDP** model of the inventory system with unidirectional transshipment and exponential lead time can be stated as follows.

Step 1: Let $V_0(i) = 0 \ \forall i \in I$ and let $n = 1$

Step 2: Compute the function $V_n(i), \ \forall i \in I$, using

$$V_n(i) = \min_{a \in A(i)} [C_i(a) + \sum_{e=1}^{4} P_i^e(a) V_{n-1}(J_i^e(a))] \tag{6.5}$$

and determine $R(n)$ as a stationary policy whose actions minimise the right side of equation (6.5)

Step 3: Compute

$$m_n = \min_{j \in I} \{V_n(j) - V_{n-1}(j)\}, \ \ M_n = \max_{j \in I} \{V_n(j) - V_{n-1}(j)\}$$

Step 4: if $M_n - m_n > \epsilon m_n$ (where ϵ is a specified tolerance), add 1 to n and repeat from step 2. Otherwise stop the iteration: $R(n)$ approximates an optimal stationary policy and $(M_n + m_n)/(2\tau)$ approximates the optimal long-run average cost per time unit.

6.3.2 Holdout transshipment policy formulation

For comparison we formulate the problem under the assumption of a holdout transshipment policy. Because there is only a change to the transshipment policy in this model, most definitions including the state space, decision epoch, transition probabilities, immediate cost and value-iteration algorithm remain as the same as in Section 6.3.1. Therefore, we just redefine the action space for this model as follows.

Action space

Let $X_k(i)$ denote the set of possible replenishment order decisions at location k when the state of the process is $i = (i_1, i_2, q_1, q_2)$. It follows that

$X_k(i) = \{q_k\}$ if $q_k > 0$ and $X_k(i) = \{x : i_k + x \leqslant M_k, x \in \{0\} \cup Q_k\}$ if $q_k = 0$

Let $q_k^{'}$ represent the decision taken regarding the order process at location k, $k = 1, 2$. For any given state $i \in I$, $q_k^{'}$ can take any value in the set $X_k(i)$.

Let $i = (i_1,\ i_2,\ q_1,\ q_2)$ be the current state of the system and let the decision variable I_2 denote the holdout threshold at location 2. We consider the decision of how to satisfy a demand at location 1. The inventory levels at the two locations determine whether or not transshipment is used to satisfy a demand at location 1. If $i_1 > 0$, location 1 would meet the demand from local stock and so there is no possibility of a transshipment. If $i_2 \leqslant I_2$, location 2 will not tranship and again transshipment is impossible. If $i_1 \leqslant 0$ and $i_2 > I_2$, then transshipment would be used to satisfy the demand. We again use 1 to denote no transshipment and 0 to denote transshipment. Let $D(i)$ denote the set of possible transshipment decisions when the system is in state i. Hence, a holdout transshipment policy can be modelled by revising equation (6.1) as follows.

$$\begin{aligned} &D(i) = \{1\} \text{ if } i_1 > 0 \text{ or } i_2 \leqslant I_2 \text{ and} \\ &D(i) = \{0\} \text{ if } i_1 \leqslant 0 \text{ and } i_2 > I_2, \quad i \in I \end{aligned} \tag{6.6}$$

Let d represent the decision with regard to be the transshipment decision from location 2 to location 1. For any given state $i \in I$, d can take any value in the set of $D(i)$.

Definition 6.3. *Under our assumptions, the action space in state i is given by*

$$A(i) = \{(q_1^{'}, q_2^{'}, d): q_k^{'} \in X_k(i) \text{ for } k = 1, 2, d \in D(i)\}$$

From the assumption, we conclude that the action space $A(i)$ is finite.

6.3.3 Numerical experiments

The purpose of our numerical experiments is to find the optimal average total cost rate under the holdout transshipment policy and to compare this to the optimal average total cost rate under a general transshipment policy. However, the choice of the limit on back-orders and storage capacity (N_k, M_k) impact on the computation of the average total cost rate. Therefore, as the first step, we need to determine suitable values for N_k and M_k in our numerical experiments.

In the following result tables, column "Iter. No." and "Av. Total Cost" denote the number of iterations and the average total cost rate respectively. In all the examples considered in this section, we only allow one possible order quantity at each location. For convenience, we use Q_k to represent the one possible order quantity at location k rather than a set. The problem parameters are given values as follows: $\lambda_1 = 10, Q_1 = 30, \mu_1 = 1, c_1 = 20, h_1 = 0.5, \hat{b_1} = 10, b_1 = 5, B_1 = 500, t = 1, \lambda_2 = 10, Q_2 = 30, \mu_2 = 1, c_2 = 20, h_2 = 0.5, \hat{b_2} = 10, b_2 = 5, B_2 = 500$.

N_k	M_k	Iter. No.	Av. Total Cost	N_k	M_k	Iter. No.	Av. Total Cost
-40	40	3384	77.74	-40	90	4006	49.26
-50	50	3896	52.27	-50	90	4014	47.71
-60	60	4010	47.24	-60	90	4018	46.95
-70	70	4018	46.61	-70	90	4018	46.61
-80	80	4016	46.47	-80	90	4016	46.47
-90	90	4016	46.41	-90	90	4016	46.41
-100	100	4016	46.39	-100	90	4016	46.39
-110	110	4016	46.38	-110	90	4016	46.38
-120	120	4016	46.37	-120	90	4016	46.37

Table 6.1. Choice of N_k and M_k

The average total cost rates of the two-location system for different choices of N_k and M_k are provided in Table 6.1. On the left hand side of Table 6.1, we start the search on the large range from (-40,40) to (-120,120). These computational results for the average total cost rate suggest that the average total cost rate is largely unaffected when the bound on the absolute inventory level is increased beyond 90. From the further searching on the right hand side of Table 6.1, we identify that a suitable choice of N_k and M_k is (-80,90). Hence we define $N_k = -80$ and $M_k = 90$ for $k = 1, 2$ as the limit on backorders and storage capacity in our numerical experiments of Section 6.3.

From the numerical results in Chapter 5, we learnt that b_k and $\hat{b_k}$ at location k, $k = 1, 2$ are two parameters to which the performance of the system can be particularly sensitive. Hence, in these experiments we vary b_k and $\hat{b_k}$ separately and observe the optimal average total cost rate under a general transshipment policy. In Table 6.2, we provide the number of iterations and average total cost rate for each value of b_1 and $\hat{b_1}$.

b_1	$\hat{b_1}$	Iter. No.	Av. Total Cost	b_1	$\hat{b_1}$	Iter. No.	Av. Total Cost
1	10	3970	46.08	5	2	3978	45.45
3	10	3992	46.29	5	4	3996	45.80
5	10	4016	46.47	5	6	4006	46.07
7	10	4036	46.62	5	8	4012	46.29
9	10	4050	46.76	5	10	4016	46.47

Table 6.2. Average total cost rate for optimal transshipment policy with different values of b_1 and $\hat{b_1}$

We are keen to know these results compare to the optimal average total cost rate under the holdout transshipment policy. We are also interested to learn whether the optimal holdout policy is a partial pooling policy rather than one of complete pooling or no pooling.

The results in Table 6.3 show that, when $b_1 = 1, 3, 5$ and 7, 9, the optimal holdout transshipment policy is the partial pooling policy with holdout threshold I_2 equal to 6, 5, 3 and 2 respectively. However, when b_1=9, the optimal holdout transshipment policy is complete pooling. The predicted improvements in the average total cost from the optimal holdout transshipment policy are 0.34%, 0.18%, 0.08%, 0.02% and 0.00% with respect to complete pooling (I_2=0) and 1.89%, 2.44%, 2.97%, 3.46% and 3.93% with respect to no pooling ($I_2 = M_2$) respectively. Further, the difference between the average total cost rates under the optimal holdout transshipment policy and the optimal general transshipment policy is

0.004% , 0.003%, 0.002%, 0.002% and 0.000% respectively.

Similarly, when $\hat{b_1} = 2, 4, 6,$ and $8, 10$, the optimal holdout transshipment policy is the partial pooling policy with holdout threshold I_2 equal to 12, 9, 6, 5 and 3 respectively. The predicted improvements in the average total cost from the optimal holdout transshipment policy are 1.06%, 0.62%, 0.34%, 0.18% and 0.08% with respect to complete pooling (I_2=0) and 0.71%, 1.29%, 1.88%, 2.44% and 2.97% with respect to no pooling (I_2=M_2) respectively. Further, the difference between the average total cost rates under the optimal holdout transshipment policy and the optimal general transshipment policy is 0.003%, 0.002%, 0.003%, 0.003% and 0.002% respectively.

	$\hat{b_1}$=10					b_1=5				
I_2	$b_1=1$	$b_1=3$	$b_1=5$	$b_1=7$	$b_1=9$	$\hat{b_1}=2$	$\hat{b_1}=4$	$\hat{b_1}=6$	$\hat{b_1}=8$	$\hat{b_1}=10$
0	46.23	46.37	46.50	46.63	46.76*	45.93	46.09	46.23	46.37	46.50
1	46.19	46.34	46.49	46.62	46.76	45.86	46.03	46.19	46.34	46.49
2	46.16	46.32	46.47	46.62*	46.77	45.80	45.98	46.15	46.32	46.47
3	46.12	46.30	46.47*	46.63	46.78	45.73	45.93	46.12	46.30	46.47*
4	46.10	46.29	46.47	46.65	46.81	45.68	45.89	46.10	46.29	46.47
5	46.09	46.29*	46.49	46.67	46.85	45.62	45.86	46.08	46.29*	46.49
6	46.08*	46.30	46.51	46.71	46.89	45.58	45.84	46.07*	46.30	46.51
7	46.08	46.32	46.54	46.75	46.95	45.54	45.82	46.07	46.31	46.54
8	46.09	46.34	46.58	46.81	47.02	45.50	45.81	46.08	46.34	46.58
9	46.11	46.38	46.63	46.87	47.10	45.48	45.80*	46.10	46.38	46.63
10	46.13	46.42	46.69	46.95	47.18	45.46	45.81	46.13	46.42	46.69
11	46.17	46.48	46.77	47.03	47.28	45.45	45.83	46.16	46.48	46.77
12	46.22	46.55	46.85	47.13	47.39	45.45*	45.85	46.21	46.54	46.85
13	46.28	46.62	46.94	47.24	47.51	45.46	45.88	46.27	46.62	46.94
14	46.34	46.71	47.04	47.35	47.64	45.47	45.93	46.34	46.70	47.04
15	46.42	46.81	47.16	47.48	47.78	45.50	45.98	46.41	46.80	47.16
16	46.51	46.91	47.28	47.61	47.93	45.54	46.05	46.50	46.91	47.28
17	46.60	47.03	47.41	47.76	48.08	45.58	46.12	46.59	47.02	47.41
18	46.71	47.15	47.55	47.91	48.25	45.64	46.20	46.70	47.15	47.55
19	46.83	47.28	47.70	48.07	48.42	45.70	46.30	46.82	47.28	47.70
20	46.95	47.42	47.85	48.24	48.59	45.77	46.40	46.94	47.42	47.85

Table 6.3. Average total cost rate under the holdout transshipment policy with different holdout thresholds and values of b_1 and $\hat{b_1}$

The Figures 6.1 and 6.2 demonstrate an interesting property of the holdout transshipment policy. At location 1, when the expected stockout or backorder cost rate is likely to increase because of increases in b_1 or $\hat{b_1}$, under the holdout transshipment policy, the optimal strategy is to lower the holdout threshold I_2 at location 2 in order to give more transshipment support. The evidence is clear from the movement of optimal holdout threshold in these two figures.

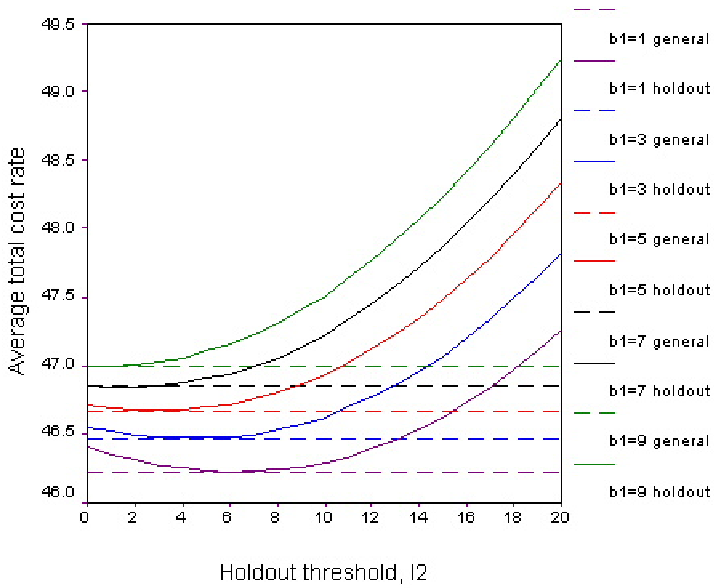

Figure 6.1. Average total cost rate for b_1 and $\hat{b_1} = 10$

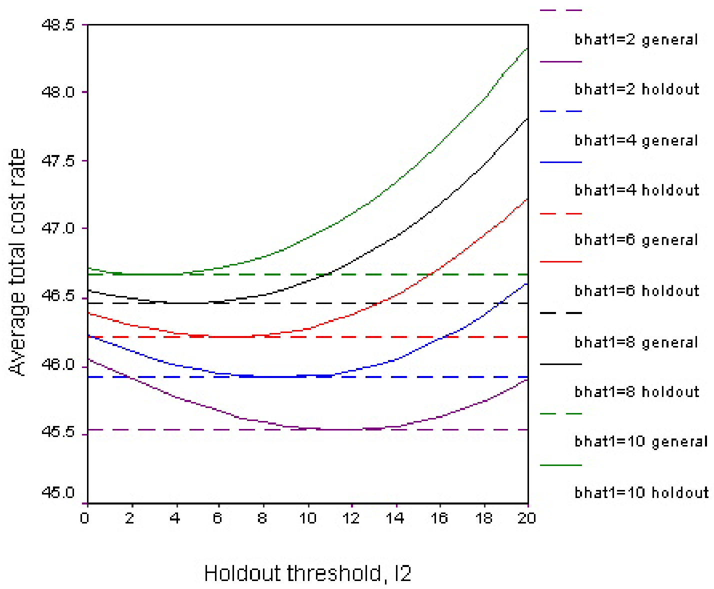

Figure 6.2. Average total cost rate for $\hat{b_1}$ and b_1=5

These results show that there exist situations where partial pooling is the optimal holdout transshipment policy, though the cost savings, compared to complete pooling or no pooling, may not always be very significant. More importantly, in each case we considered, the optimal average total cost rate with the holdout transshipment policy is very close to the optimal average total cost rate with the general transshipment policy. This suggests that there often exists a holdout transshipment policy that effectively perform as well as the optimal general transshipment policy. This has practical implications, because holdout transshipment policies are easier to implement.

Note that we do not compare the numerical results with those in the previous **TAP** or **HOT** approximations, because in our **SMDP** models, we assume the more general replenishment order policy rather than the $(\boldsymbol{R}, \boldsymbol{Q})$ replenishment order policy.

6.3.4 Optimal replenishment and transshipment decisions

In this section we consider the optimal replenishment and transshipment decisions for the model with general transshipment policy. We are keen to know how the optimal holdout transshipment policy compares to the optimal general transshipment policy. We also demonstrate that the optimal general transshipment policy does not depend on the state of the replenishment order process. We also examine the form of the optimal replenishment policy.

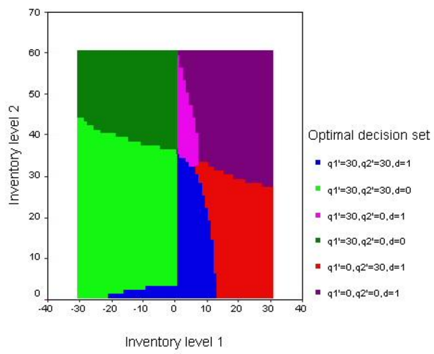

Figure 6.3. Optimal actions when $b_1 = 5, \hat{b_1} = 10, q_1 = 0, q_2 = 0$

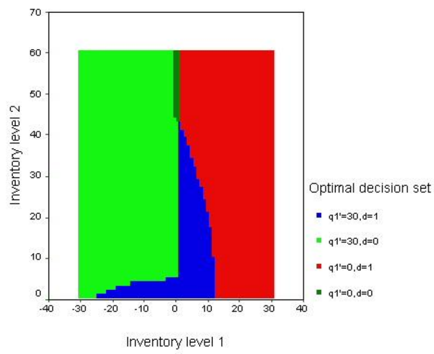

Figure 6.4. Optimal actions when $b_1 = 5, \hat{b_1} = 10, q_1 = 0, q_2 = 30$

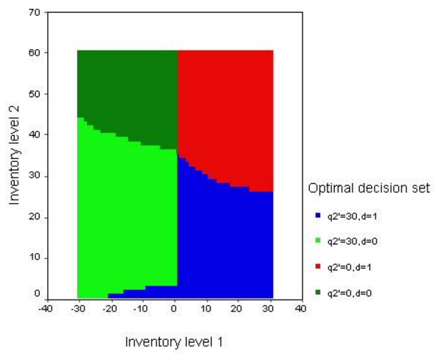

Figure 6.5. Optimal actions when $b_1 = 5, \hat{b_1} = 10, q_1 = 30, q_2 = 0$

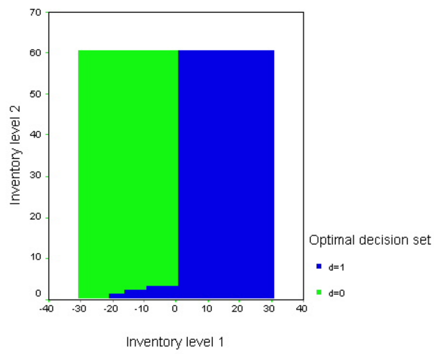

Figure 6.6. Optimal actions when $b_1 = 5, \hat{b_1} = 10, q_1 = 30, q_2 = 30$

Note that we use the x-axis and y-axis to represent the inventory level at location 1 and location 2 respectively. The areas covered by green colour represent situations in which transshipment is optimal to meet demand at location 1. Other areas coorrespond to situations where transshipment is not the optimal decision.

Figures 6.3 to 6.6 plot the optimal decisions as a function of the inventory levels at the two locations when $b_1 = 5$ and $\hat{b_1} = 10$. Each plot represents a different state of the replenishment order process. Different decisions are represented by different colours, so the plots clearly show the regions in which different decisions are optimal. By the choice of M_k, it is only feasible to place a replenishment order at a location when the inventory level is less than or equal to 60. In the figures it is never optimal to place a replenishment order when the inventory level is greater than 12 at location 1 and greater than 45 at location 2. This suggests that the replenishment decision is not being constrained by the storage capacity of the locations.

The two green areas to the left of the figures show the situations in which transshipment is optimal to meet demand at location 1. The other areas show the situations in which it is optimal to meet demand at location 1 from local stock or by backorders. The most striking feature about the figures is that the optimal transshipment decision does not depend on the state of the replenishment order process (the green colours cover the same area in each plot). This is largely explained by the memory-less property of the exponential lead time for replenishment orders. The fact there is an outstanding order at a location does not affect the expected time until the earliest replenishment of the location and so has little affect on the transshipment decision. It is also interesting to note that the optimal transshipment policy is not a simple holdout policy. Rather there is a transition from complete pooling to partial pooling as the inventory level at location 1 increases from -30 to 0 and the inventory level at location 2 is at most 3. This means that location 2 becomes more willing to share inventory as the number of outstanding backorders at location 1 increases.

When the inventory level at location 2 is greater than 3, transshipment is always optimal when location 1 has no local stock. It is interesting to note that this value is consistent with the threshold value in the optimal holdout transshipment policy for this case from Table 6.3. In fact if the inventory level at location 1 is between -9 and 0, it is optimal to use transshipment to meet demand at location 1 if and only if the inventory level at location 2 is greater than 3. In other words, in such situations the optimal transshipment policy is a holdout policy with threshold 3. If the inventory level at location 1 rarely falls below -9, the performance of a holdout transshipment policy with threshold 3 would be very similar to that of the optimal general transshipment policy. This perhaps explains the results in Figure 6.1.

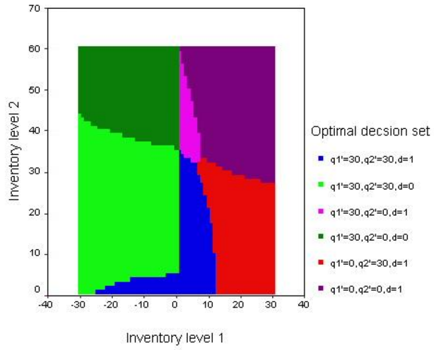

Figure 6.7. Optimal actions when $b_1 = 3$, $\widehat{b_1} = 5$, $q_1 = 0$, $q_2 = 0$

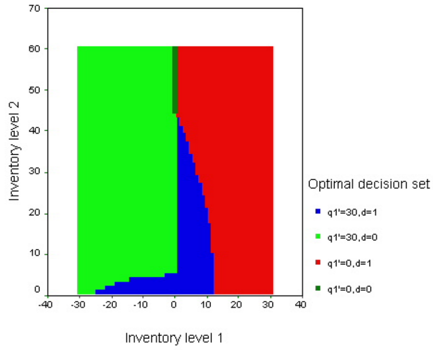

Figure 6.8. Optimal actions when $b_1 = 3$, $\widehat{b_1} = 5$, $q_1 = 0$, $q_2 = 30$

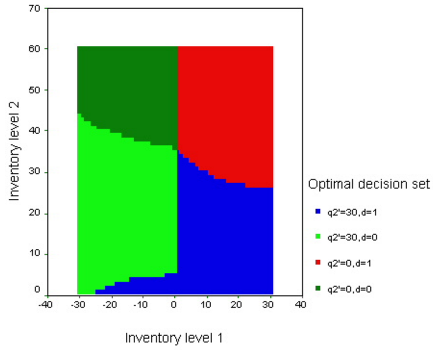

Figure 6.9. Optimal actions when $b_1 = 3$, $\widehat{b_1} = 5$, $q_1 = 30$, $q_2 = 0$

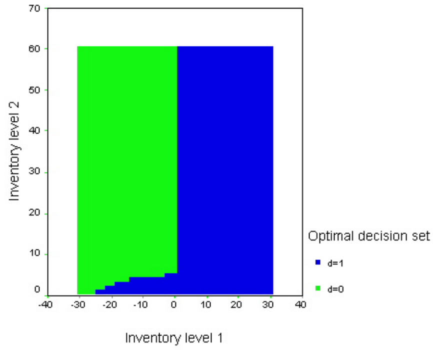

Figure 6.10. Optimal actions when $b_1 = 3$, $\widehat{b_1} = 5$, $q_1 = 30$, $q_2 = 30$

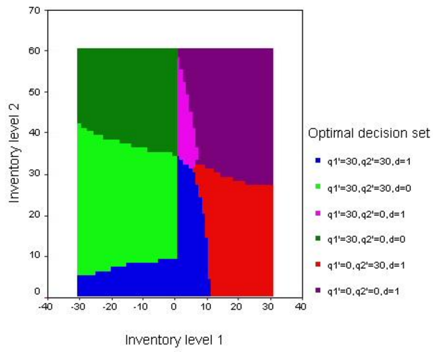

Figure 6.11. Optimal actions when $b_1 = 5$, $\widehat{b_1} = 4$, $q_1 = 0$, $q_2 = 0$

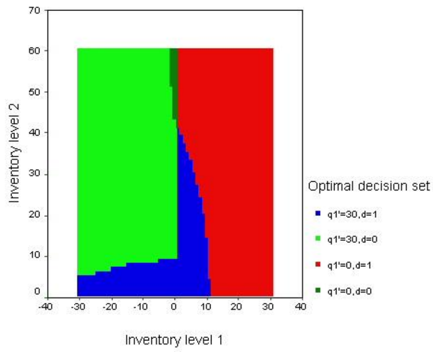

Figure 6.12. Optimal actions when $b_1 = 5$, $\widehat{b_1} = 4$, $q_1 = 0$, $q_2 = 30$

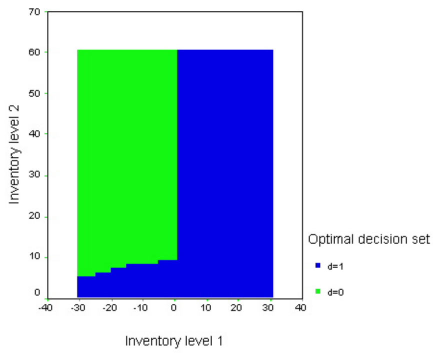

Figure 6.13. Optimal actions when $b_1 = 5$, $\widehat{b_1} = 4$, $q_1 = 30$, $q_2 = 0$

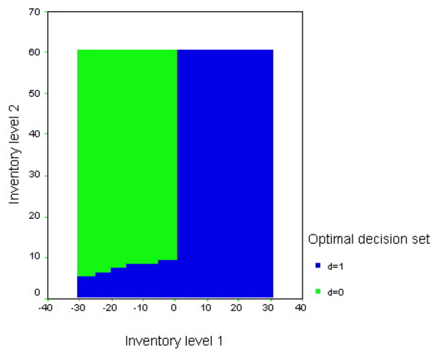

Figure 6.14. Optimal actions when $b_1 = 5$, $\widehat{b_1} = 4$, $q_1 = 30$, $q_2 = 30$

Figure 6.3 shows the situation where there are no outstanding orders. In the figure it is optimal to place a replenishment order for location 2 when its inventory level is less than some threshold level which depends on the inventory level at location 1. The threshold level is between 28 and 45 and exhibits a downward trend as the inventory level at location 1 increases. However, the threshold does not decrease monotonically as the inventory level at location 1 increases. Similarly, it is optimal to place a replenishment order at location 1 when its inventory level is less than a threshold value between 0 and 12 depending on the inventory level at location 2. The threshold value exhibits a downward trend as the inventory level at location 2 increases, but the increase is not monotonic.

Figures 6.4 and 6.5 show the situations where there is one outstanding order in the system at location 2 and location 1 respectively. It is apparent that the situations in which it is optimal to replenish location 1 are different in Figures 6.3 and 6.4. Similarly the situations in which it is optimal to replenish location 1 are different in Figures 6.3 and 6.5. We conclude that the optimal replenishment decision is a complex function of the state of the system.

Similarly, Figures 6.7 to 6.10 plot the optimal decisions as a function of the inventory levels at the two locations when $b_1 = 3$ and $\hat{b_1} = 10$ and Figures 6.11 to 6.14 plot this information when $b_1 = 5$ and $\hat{b_1} = 4$. These figures demonstrate similar features in terms of replenishment and transshipment decisions to Figures 6.3 to 6.6. One important difference is the extent to which transshipment is used. The green area in these graphs is smaller than in Figures 6.3 to 6.6, indicating transshipment is used less often to meet demand at location 1. This is because either the backorder cost (Figures 6.7 to 6.10) or stockout cost (Figures 6.11 to 6.14) is lower.

When $b_1 = 3$, $\hat{b_1} = 10$ and the inventory level at location 2 is greater than 5, transshipment is always optimal when location 1 has no local stock. When $b_1 = 5$, $\hat{b_1} = 4$ and the inventory level at location 2 is greater than 9, transshipment is always optimal when location 1 has no local stock. Again these values are consistent with the threshold values in the optimal holdout transshipment policy for these cases from Table 6.3. We hypothesise that a suitable threshold value for a holdout transshipment policy can be estimated by the largest inventory level at location 2 for which it is optimal to reject a transshipment request in an optimal general transshipment policy.

6.3.5 Conclusions

In Section 6.3, we developed two **SMDP** models with exponential lead time to reflect the general transshipment and holdout transshipment policies. The numerical results in Section 6.3.3 illustrate how these models can be used to find the optimal average total cost rate and optimal policy for a given set of parameters. In particular, we investigate the effect of the parameters $\hat{b_1}$ and b_1 on the optimal holdout threshold for a holdout transshipment policy. When b_1 and $\hat{b_1}$ are increasing the optimal strategy for the system is to lower the holdout threshold in order to give more transshipment support. In the cases we consider, the optimal average total cost rate under the holdout transshipment policy is very close to that under the general transshipment policy. This suggests that, under the assumption of exponential lead time, the benefits of transshipment can often be captured by a simple holdout transshipment policy.

In addition, we use the optimal general transshipment policy to estimate the holdout threshold for the optimal holdout transshipment policy. Therefore, the **SMDP** modelling technique provides another way to understand the two-location system with unidirectional holdout transshipment policy. To some extent, analysis of the optimal decisions provides some insights into how the optimal replenishment decisions depend on the inventory level of the locations and the state of the replenishment order process.

Overall, we can conclude that the optimal average total cost rate can be found using the **SMDP** models with exponential lead time. The difference between the optimal average total cost rate under the general and holdout transshipment policy appears to be small. However, because of our assumption of exponential lead time, the optimal transshipment decision does not depend on the state of the replenishment order process due to the memory-less property of the exponential distribution. Intuitively one might expect that location 2 would become more willing to share inventory as its next replenishment approaches. One might also expect that, due to the form of the backorder cost, the incentive for transshipment would decrease as the next replenishment at location 1 approaches. These features cannot be captured by the **SMDP** model with exponential lead time. Therefore, the results of the **SMDP** model with exponential lead time might not be applicable to, for example, the situation of fixed lead time. Therefore, we develop another **SMDP** model with phase-type lead time in which the optimal transshipment decision is not only dependent on inventory level at the locations, but also dependent on the state of the replenishment order process.

6.4 SMDP models with phase-type lead time

In Section 6.3, we examined two **SMDP** models of the two location inventory system with the assumption of exponential lead time. The numerical results demonstrate some interesting features of the transshipment policy. Although the **SMDP** models benefit from the memoryless property of the exponential lead time, the exponential random variable does not model the fixed replenishment lead time effectively because of its long-tail property.

In this section, instead of assuming exponential lead time, we develop another **SMDP** model assuming a phase-type distribution for the lead time. Rather than modelling the lead time by a single exponential random variable, the idea is to model the lead time as the sum of a fixed number of **IID** exponential random variables. As the number of phases increases, this phase-type model of replenishment lead time more closely approximates the fixed replenishment lead time. Meanwhile, to compensate for the increased complexity due to the phase-type model, we restrict the choice of order quantity at each location to a single value for the **SMDP** models in this section instead of allowing a set of possible order quantities as for the **SMDP** models in Section 6.3. We denote the number of phases in the model of the lead time at location k by W_k and the order quantity at location k by Q_k. Following the structure of Section 6.3, we first develop a **SMDP** model of the system with general transshipment policy and then modify it for holdout transshipment policy.

6.4.1 General transshipment policy formulation

State space

For each location, we need one state variable to model the inventory process and, under the assumption above, one state variable to model the replenishment order process. Note that if we allowed more than one possible order quantity at each location, we would need two state variables at each location to model the replenishment order process. Such a model is too demanding computationally to be considered in our research.

Let i_k denote the inventory level at location k, $k = 1, 2$, where negative values indicate outstanding backorders. Let $w_k = 0$ represent the situation where there is no outstanding replenishment order at location k. Let $w_k > 0$ represent the situation where there is one outstanding replenishment order for Q_k items at location k and that this order will arrive after

w_k phases (i.e. after a time equal to the sum of w_k **IID** exponential random variables each with mean $\frac{1}{W_k \mu_k}$).

Definition 6.4. *Under our assumptions, the state space is given by*

$$I = \{(i_1, i_2, w_1, w_2);\ -N_k \leqslant i_k \leqslant M_k,\ 0 \leqslant w_k \leqslant W_k \text{ for } k = 1, 2\}$$

From the above assumptions, we conclude that the state space I is finite.

Action space

We need two decision variables to model the order decisions at the two locations and one decision variable to model the transshipment decision from location 2 to location 1. When there is no outstanding order at location k (i.e. w_k=0), we can choose to place a replenishment order provided the delivery of Q_k items would not violate the capacity constraint (i.e. $i_k + Q_k \leqslant M_k$). We model this by setting the replenishment order decision equal to W_k. When there is no outstanding order at location k, we model the decision not to place an order by setting the replenishment order decision equal to 0. When there is already an outstanding order at location k (i.e. $w_k > 0$), we cannot place another one and we set the replenishment order decision equal to the number of phases remaining in the lead time (i.e. w_k). Let $X_k(i)$ denote the set of the possible replenishment order decisions at location k, $k = 1, 2$ when the system is in state $i = (i_1, i_2, w_1, w_2)$. It follows that

$$X_k(i) = \{w_k\} \text{ if } w_k > 0,\ X_k(i) = \{0\} \text{ if } w_k = 0 \text{ and } i_k + Q_k > M_k$$

$$\text{and } X_k(i) = \{0, W_k\} \text{ if } w_k = 0 \text{ and } i_k + Q_k \leqslant M_k$$

Let $w_k^{'}$ represent the decision taken regarding the replenishment order at location k, $k = 1, 2$. For any given state $i \in I$, $w_k^{'}$ can take any value in the set $X_k(i)$.

Let $i = (i_1, i_2, w_1, w_2)$ be the current state of the system. Because we have the same unidirectional transshipment policy as we have defined in Section 6.3, we just reuse the definition of the set of the transshipment decisions (where 0 represents transshipment and 1 represents no transshipment) as such.

$$\begin{aligned} D(i) &= \{1\} \text{ if } i_1 > 0 \text{ or } i_2 \leqslant 0; \\ D(i) &= \{0, 1\} \text{ if } -N_1 < i_1 \leqslant 0 \text{ and } i_2 > 0; \text{ and} \\ D(i) &= \{0\} \text{ if } -N_1 = i_1 \text{ and } i_2 > 0 \text{ for } i \in I \end{aligned} \tag{6.7}$$

Let d represent the decision with regard to transshipment from location 2 to location 1. For any given state $i \in I$, d can take any value in the set of $D(i)$.

Definition 6.5. *Under our assumptions, the action space in state i is given by*

$$A(i) = \{(w_1^{'}, w_2^{'}, d);\ w_k^{'} \in X_k(i),\, d \in D(i) \text{ for } k = 1, 2\}$$

From the above assumptions, we conclude that the action space $A(i)$ is finite.

Decision epoch

Decisions need to be taken when a demand occurs at each location and when the system reaches the end of a phase of the lead time at each location. We define an event to be any occurrence of a demand or the end of any phase of a replenishment order lead time in the system. With this definition, the time of the next event depends on the state of the system. As before we introduce fictitious decision epochs so that the time of the next event is independent of the state of the process and the action chosen.

When there is no outstanding order at location k, we introduce a fictitious event after a time that has an exponential distribution with mean $\frac{1}{W_k \mu_k}$. This can be thought of as the end of the final phase of the lead time of a zero replenishment order at location k and will leave the state unchanged. Hence, whatever the state and action, the next event can always be a demand at location 1, a demand at location 2, the end of a phase of the lead time of a (possible fictitious) replenishment order at location 1 or location 2. The time until each of the possible events is exponential with scale parameter independent of the state and action. Hence, the expected time until next decision epoch no longer depends on state or decision. We can now define τ as the expected time until the next decision epoch where

$$\tau = \frac{1}{\lambda_1 + \lambda_2 + W_1\mu_1 + W_2\mu_2} \tag{6.8}$$

Transition probabilities

We need to define the probability that the process is in state j at the next decision epoch given that action a is chosen in state i at the current decision epoch. As before we define the transition probabilities for the model implicitly by considering which of the four possible events occurs at the next decision epoch.

Consider a decision epoch in which the system is in state $i=(i_1,i_2,w_1,w_2)$ and action $a=(w_1^{'},w_2^{'},d)$ is chosen. Assume the next event is a demand. If there is an outstanding order at location k, then $w_k^{'}=w_k>0$ and the order will still be outstanding at the next decision epoch. If an order is placed at location k, then $w_k^{'}>w_k=0$ and this order will be outstanding at the next decision epoch. If there is no outstanding order and no order is placed, then $w_k^{'}=w_k=0$ and there will be no outstanding order at the next decision epoch. Hence, at the next decision epoch, the state variable describing the replenishment order process at location k will be equal to $w_k^{'}$. The change in the state variable describing the inventory process follows from a similar argument as in Section 6.3. Therefore, when the next event is a demand at location 1, the state of the process at the next decision epoch is $(\max(-N_1,i_1-d),i_2-(1-d),w_1^{'},w_2^{'})$. Similarly, when the next event is a demand at location 2, the state of the process at the next decision epoch is $(i_1,\max(-N_2,i_2-1),w_1^{'},w_2^{'})$. The probability that the next event is a demand at location k is $\lambda_k\tau$.

The probability that the next event is the end of a phase of a replenishment order lead time at location 1 is $W_1\mu_1\tau$. If $w_1^{'}=0$, this is a fictitious event and there is no change in state. If $1<w_1^{'}\leqslant W_1$, the lead time of the outstanding replenishment order at location 1 now has $w_1^{'}$-1 phases remaining. If $w_1^{'}=1$, this is the end of the lead time of the outstanding order at location 1. Q_1 items are added to the inventory at location 1 and there cannot any more outstanding orders at location 1. This event will not affect the state variables relating to location 2. Therefore, when the next event is the end of a lead time phase at location 1, the state of the process at the next decision epoch is $(i_1+Q_1,i_2,0,w_2')$ if $w_1^{'}=1$ and $(i_1,i_2,\min(0,w_1^{'}-1),w_2^{'})$ otherwise. The probability that the next event is the end of a phase of a replenishment order lead time at location 2 is $W_2\mu_2\tau$ and, following a similar argument to the above, the state of the process at the next decision epoch is $(i_1,i_2+Q_2,w_1',0)$ if $w_2^{'}$=1 and $(i_1,i_2,w_1^{'},\min(0,w_2^{'}-1))$ otherwise.

Immediate cost

We now define $c_i(a)$ the expected cost incurred until the next decision epoch when action a is chosen in state i at the current decision epoch. To reiterate, this cost consists of the fixed order cost, holding cost, backorder cost, stockout cost, lost sale penalty cost and transshipment cost.

Under our assumptions, a replenishment order is only placed at location k at a decision epoch when $w_k^{'}>w_k$. Hence, the fixed order cost incurred until the next decision epoch is equal to $c_k\delta(w_k^{'} - w_k)$ at location $k, k=1,2$.

Because introduction of phase-type replenishment lead time only affects the expected replenishment order cost, we can reuse the expected costs for the holding cost, backorder cost, stockout cost, lost sale penalty cost and transshipment cost from Section 6.3.1. Hence, we conclude that the expected cost until the next decision epoch when action a is chosen in state i at the current decision epoch is as follows.

$$c_i(a) = \{\sum_{k=1}^{2}\{c_k\delta(w_k^{'} - w_k) + [h_k\delta(i_k) - b_k\hat{\delta}(i_k)]i_k\tau\} + \lambda_1\tau d\{\hat{\delta}(i_1)\hat{b_1} + \hat{\delta}(N_1 + i_1)(B_1 - \hat{b_1})\} +$$

$$\lambda_2\tau\{\hat{\delta}(i_2)\hat{b_2} + \hat{\delta}(N_2 + i_2)(B_2 - \hat{b_2})\} + \lambda_1\tau(1-d)t \tag{6.9}$$

Value-iteration algorithm

Because the **SMDP** model developed in this section has a similar structure to the **SMDP** model in Section 6.3.1, we can reuse the value-iteration algorithm from that Section.

6.4.2 Holdout transshipment policy formulation

For the **SMDP** model with the phase-type lead time and holdout transshipment policy, most definitions including the state, decision epoch, transition probabilities, immediate cost and value-iteration algorithm are the same as those in the Section 6.4.1. Therefore, we just redefine the action space for this model as such.

Action space

Let X_k denote the set of possible decision replenishment order decisions at location k when the state of the process is $i = (i_1, i_2, w_1, w_2)$. It follows that

$$X_k(i) = \{w_k\} \text{ if } w_k > 0,\ X_k(i)=\{0\} \text{ if } w_k = 0 \text{ and } i_k + Q_k > M_k$$

$$\text{and } X_k(i)=\{0, W_k\} \text{ if } w_k = 0 \text{ and } i_k + Q_k \leqslant M_k$$

Let $w_k^{'}$ represent the decision taken regarding the replenishment order at location $k, k=1,2$. For any given state $i \in I$, $w_k^{'}$ can take any value in the set $X_k(i)$.

Let I_2 denote the holdout threshold at location 2. As before, the set of the transshipment decision in state i is given by

$$D(i) = \{1\} \text{ if } i_1 > 0 \text{ or } i_2 \leqslant I_2 \text{ and } D(i) = \{0\} \text{ if } i_1 \leqslant 0 \text{ and } i_2 > I_2,\ i \in I \tag{6.10}$$

Let d represent the decision with regard to the transshipment decision from location 2 to location 1. For any given state $i \in I$, d can take any value in the set $D(i)$.

Definition 6.6. *Under our assumptions, the action space in state i is given by*

$$A(i) = \{(w_1^{'}, w_2^{'}, d);\ w_k^{'} \in W_k(i),\ d \in D(i) \text{ for } k = 1, 2\}$$

From the assumptions, we conclude that the action space $A(i)$ is finite.

6.4.3 Numerical experiments

For the **SMDP** model with phase-type lead time, the computational overheads increase exponentially with an increase in the number of phases. First, as for the model with exponential lead time, we define $N_k = -80$, and $M_k = 70$ for $k = 1, 2$ as the limit on backorders and storage capacity in our numerical experiments. In Table 6.4, we compare the estimated computing time[6.1], the number of iterations and average total cost rate for different numbers of phases and the same basic set of parameter values as in Section 6.3.3. Namely, we use $\lambda_1 = 10, Q_1 = 30, \mu_1 = 1, c_1 = 20, h_1 = 0.5, \hat{b}_1 = 10, b_1 = 5, B_1 = 500, t = 1, \lambda_2 = 10, Q_2 = 30, \mu_2 = 1,$ $c_2 = 20, h_2 = 0.5, \hat{b}_2 = 10, b_2 = 5, B_2 = 500$. These results show significant increases in the computing time when the number of phases increases from 1 to 8. For reasons of practical implementation, we choose to do our numerical experiments only for the situations where the number of phases equals to 1 and 4 to assess the impact of the phase-type lead time.

Iter. No.	Ph. No.	Av. Total Cost	Est. Computing Time
4016	1	46.47	2 mins
2472	2	39.17	6 mins
1964	3	36.78	10 mins
1734	4	35.58	17 mins
1652	5	34.84	26 mins
1656	6	34.34	42 mins
1706	7	33.97	65 mins
1790	8	33.69	97 mins

Table 6.4. Average total cost rate for optimal transshipment policy with different numbers of phases

6.1. The estimated computing time is the result from running on the desktop PC machine with Pentium IV CPU 1.2G HZ, 256M memory

As is Section 6.3.3, we do the sensitivity tests on the parameters b_1 and $\hat{b_1}$. Hence, in these experiments we vary b_1 and $\hat{b_1}$ separately and observe the optimal average total cost rate under a general transshipment policy. The average total cost rate and the number of iterations for each value of b_1 or $\hat{b_1}$ when the number of phases is equal to 4 are shown in Table 6.5.

b_1	$\hat{b_1}$	Iter. No.	Av. Total Cost	b_1	$\hat{b_1}$	Iter. No.	Av. Total Cost
1	10	1732	35.24	5	2	1734	35.34
3	10	1730	35.44	5	4	1734	35.41
5	10	1734	35.58	5	6	1734	35.48
7	10	1738	35.68	5	8	1734	35.53

Table 6.5. Average total cost rate for optimal transshipment policy with different values of b_1 and $\hat{b_1}$ when number of phases = 4

	$\hat{b_1}$=10				b_1=5			
I_2	b_1=1	b_1=3	b_1=5	b_1=7	$\hat{b_1}$=2	$\hat{b_1}$=4	$\hat{b_1}$=6	$\hat{b_1}$=8
0	35.43	35.52	35.61*	35.69*	35.46	35.50	35.54	35.57
1	35.40	35.51	35.61	35.71	35.43	35.48	35.52	35.57*
2	35.37	35.51*	35.63	35.74	35.41	35.47*	35.52*	35.58
3	35.36	35.51	35.66	35.78	35.40*	35.47	35.54	35.60
4	35.36*	35.54	35.70	35.84	35.41	35.49	35.56	35.63
5	35.37	35.57	35.76	35.92	35.42	35.51	35.60	35.68
6	35.39	35.62	35.83	36.01	35.45	35.56	35.65	35.74
7	35.42	35.69	35.91	36.11	35.49	35.61	35.72	35.82
8	35.47	35.76	36.01	36.22	35.55	35.68	35.80	35.91
9	35.53	35.85	36.12	36.35	35.62	35.76	35.89	36.01
10	35.60	35.94	36.23	36.48	35.70	35.85	35.99	36.12
11	35.67	36.05	36.36	36.62	35.79	35.95	36.10	36.23
12	35.76	36.16	36.49	36.76	35.88	36.06	36.22	36.36
13	35.85	36.28	36.62	36.91	35.99	36.17	36.34	36.49
14	35.95	36.39	36.75	37.06	36.10	36.29	36.46	36.62
15	36.04	36.51	36.89	37.20	36.21	36.41	36.59	36.74
16	36.14	36.63	37.03	37.35	36.32	36.53	36.71	36.87
17	36.24	36.75	37.16	37.49	36.43	36.65	36.83	37.01
18	36.33	36.87	37.28	37.62	36.54	36.76	36.96	37.13
19	36.42	36.98	37.41	37.75	36.64	36.87	37.07	37.24
20	36.51	37.08	37.52	37.88	36.74	36.97	37.18	37.36

Table 6.6. Average total cost rate under the holdout transshipment policy with different holdout thresholds and values of b_1 and $\hat{b_1}$ when number of phases = 4

The results in Table 6.6 show that, when $b_1 = 1$ and 3, the optimal holdout transshipment policy is the partial pooling policy with holdout threshold I_2 equal to 4 and 2 respectively.

However, when $b_1 = 5$ and 7, the optimal holdout transshipment policy is complete pooling. The predicted improvements in the average total cost from the optimal holdout transshipment policy are 0.21%, 0.05%, 0.00% and 0.00% with respect to complete pooling ($I_2 = 0$) and 3.27%, 4.44%, 5.37% and 6.13% with respect to no pooling ($I_2 = M_2$) respectively. Further, the difference between the average total cost rates under the optimal holdout transshipment policy and the optimal general transshipment policy is 0.32%, 0.18%, 0.10% and 0.03% respectively.

Similarly, when $\hat{b_1} = 2, 4, 6,$ and 8, the optimal holdout transshipment policy is the partial pooling policy with holdout threshold I_2 equal to 3, 2, 2 and 1 respectively. The predicted improvements in the average total cost from the optimal holdout transshipment policy are 0.16%, 0.09%, 0.04% and 0.02% with respect to complete pooling ($I_2 = 0$) and 3.77%, 4.24%, 4.65% and 5.02% with respect to no pooling ($I_2 = M_2$) respectively. Further, the difference between the average total cost rates under the optimal holdout transshipment policy and the optimal general transshipment policy is 0.18%, 0.15%, 0.14% and 0.11% respectively.

Figures 6.15 and 6.16 show similar properties to those observed in Section 6.3. That is, at location 1, when the expected stockout or backorder cost increases, under the holdout transshipment policy, the optimal strategy is to lower the holdout threshold I_2 at location 2 to give more transshipment support. The evidence is clear from the movement of the optimal holdout threshold in these two figures.

Compared to the results with exponential lead time, we found that the difference between the average total cost rate under the general transshipment policy and the optimal average total cost rate under the holdout transshipment policy is greater when there are more phases. The optimal holdout transshipment policy is closer to complete pooling than in the case of exponential lead time. This is apparent from the higher percentage differences between the optimal partial pooling policy and the no pooling policy ($I_2 = M_2$) and the smaller percentage differences between the optimal partial pooling policy and the complete pooling policy ($I_2 = 0$).

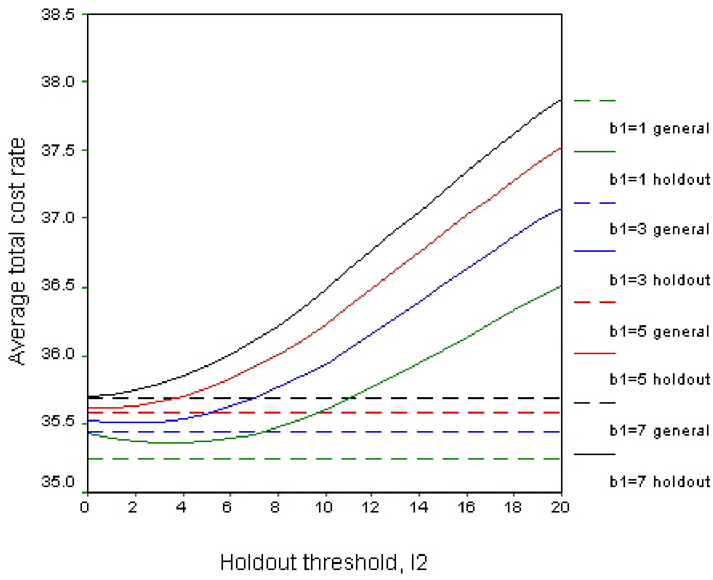

Figure 6.15. Average total cost rate for b_1, $\hat{b_1} = 10$ and 4 phases

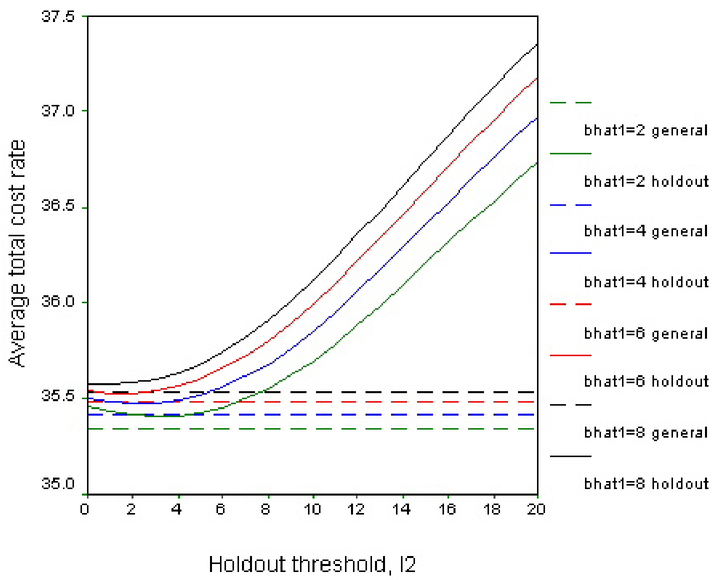

Figure 6.16. Average total cost rate for $\hat{b_1}$, $b_1 = 5$ and 4 phases

6.4.4 Optimal transshipment decisions

In this section, we investigate how optimal transshipment decisions depend on the state of the replenishment order process. Due to the memory-less property of the exponential distribution, the same transshipment decisions are taken when there is no outstanding order at location k as when there are W_k phases remaining in the lead time of an outstanding order at location k. Hence, in this section, we only consider situations where there are outstanding orders at both locations (i.e. $1 \leqslant w_k \leqslant 4$ for $k=1$ and 2). We are also keen to know how the optimal holdout transshipment policy compares to the optimal general transshipment policy.

Figures 6.17 to 6.20 plot the optimal decisions as a function of the inventory levels at the two locations when $b_1 = 5$ and $\hat{b_1} = 10$. Each plot represents a different number of remaining phases in the lead time of the outstanding order at location 2. Green colour represents the decision to use transshipment to meet demand at location 1 and blue colour represents the decision to meet demand at location 1 from local stock or by backorders. Considering the limit of the storage capacity and the order quantity for all of numerical experiments, we only draw the optimal decisions within a range where $-30 \leqslant i_1 \leqslant 30$ and $0 \leqslant i_2 \leqslant 30$.

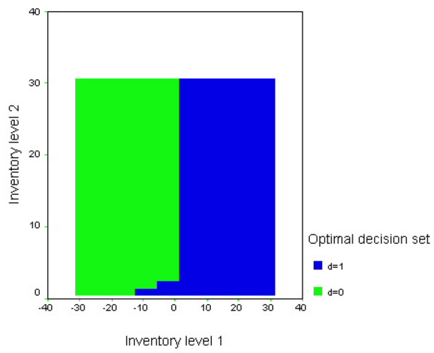

Figure 6.17. Optimal actions when $b_1 = 5, \hat{b_1} = 10, w_1 = 4, w_2 = 4$

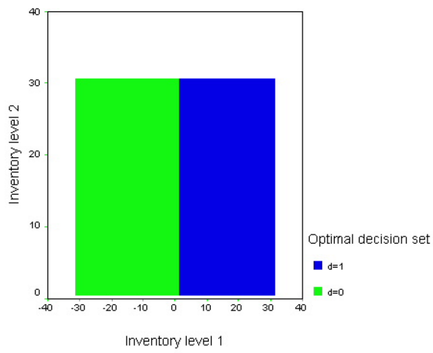

Figure 6.18. Optimal actions when $b_1 = 5, \hat{b_1} = 10, w_1 = 4, w_2 = 3$

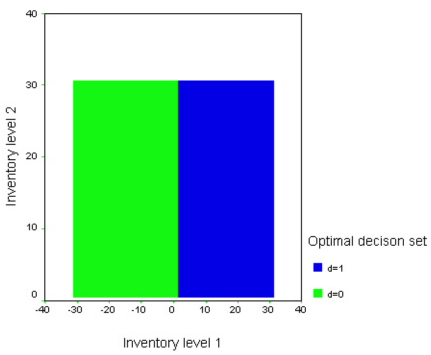

Figure 6.19. Optimal actions when $b_1 = 5, \hat{b_1} = 10, w_1 = 4, w_2 = 2$

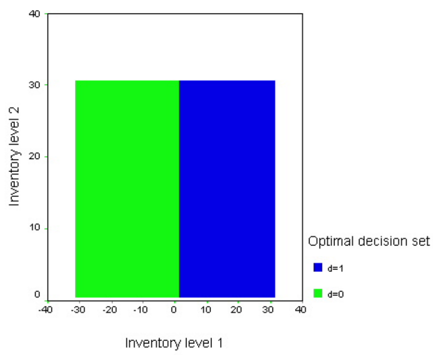

Figure 6.20. Optimal actions when $b_1 = 5, \hat{b_1} = 10, w_1 = 4, w_2 = 1$

Figures 6.17 to 6.20 show the optimal transshipment policy when there are 4 phases remaining in the lead time of the outstanding order at location 1. We see that complete pooling is optimal unless there are 4 phases remaining in the lead time of the outstanding order at location 2. In this case ($w_2 = 4$), there is some evidence of partial pooling when the inventory level at location 2 is 1 or 2. However, if the inventory level at location 2 exceeds 2, transshipment is always optimal. We conclude that, when the expected time to replenishment at location 1 is large, the level of transshipment support provided by location 2 is relatively high. Further the level of transshipment support provided by location 2 is non-increasing as the expected time until replenishment at location 2 increases.

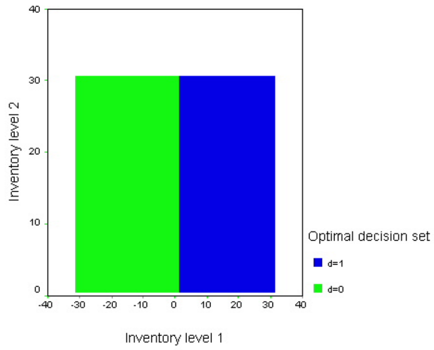

Figure 6.21. Optimal actions when $b_1 = 3, \widehat{b_1} = 10, w_1 = 4, w_2 = 2$

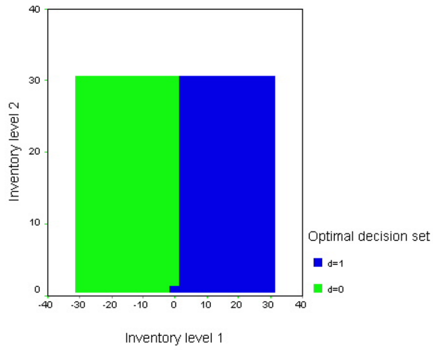

Figure 6.22. Optimal actions when $b_1 = 3, \widehat{b_1} = 10, w_1 = 3, w_2 = 2$

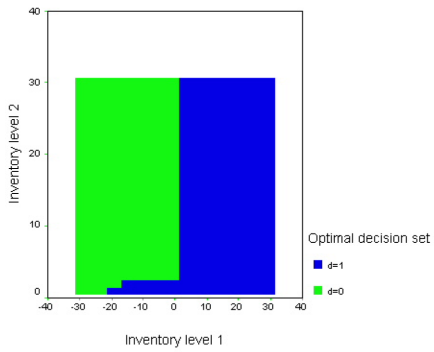

Figure 6.23. Optimal actions when $b_1 = 3, \widehat{b_1} = 10, w_1 = 2, w_2 = 2$

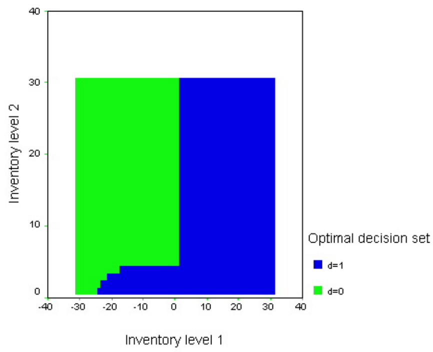

Figure 6.24. Optimal actions when $b_1 = 3, \widehat{b_1} = 10, w_1 = 1, w_2 = 2$

Similarly, Figures 6.21 to 6.24 plot the optimal transshipment policy when there are 2 phases remaining in the lead time of the outstanding order at location 2. Also in this case, b_1 is lower which means there is less incentive for transshipment. We see that, as the number of phases remaining in the lead time at location 1 decreases, location 2 provides less transshipment support. Complete pooling is optimal when $w_1 = 4$ but, if there is one outstanding backorder at location 1, location 2 will only tranship in response to a stockout at location 1 when its inventory level is greater than 1, 2 and 4 for $w_1 = 3$, 2 and 1 respectively. This suggests the use of a dynamic holdout transshipment policy in which the threshold value depends on the time until replenishment at location 1.

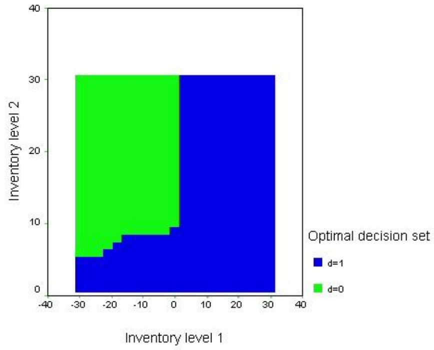

Figure 6.25. Optimal actions when $b_1 = 5$, $\hat{b_1} = 4$, $w_1 = 2$, $w_2 = 4$

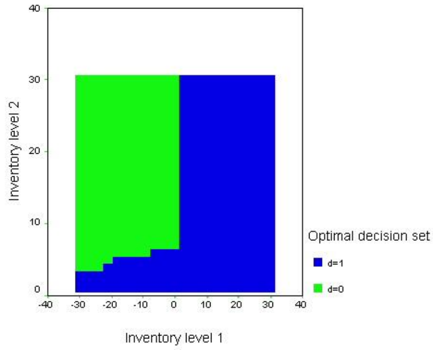

Figure 6.26. Optimal actions when $b_1 = 5$, $\hat{b_1} = 4$, $w_1 = 2$, $w_2 = 3$

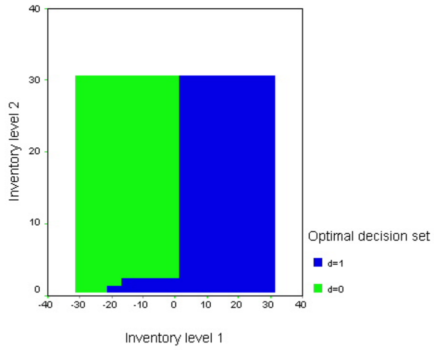

Figure 6.27. Optimal actions when $b_1 = 5$, $\hat{b_1} = 4$, $w_1 = 2$, $w_2 = 2$

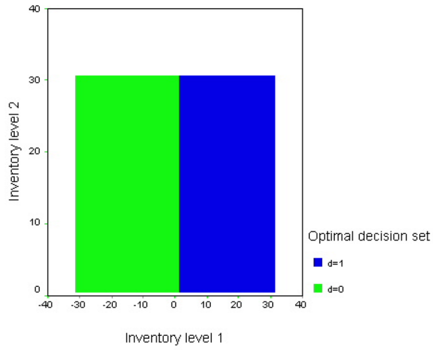

Figure 6.28. Optimal actions when $b_1 = 5$, $\hat{b_1} = 4$, $w_1 = 2$, $w_2 = 1$

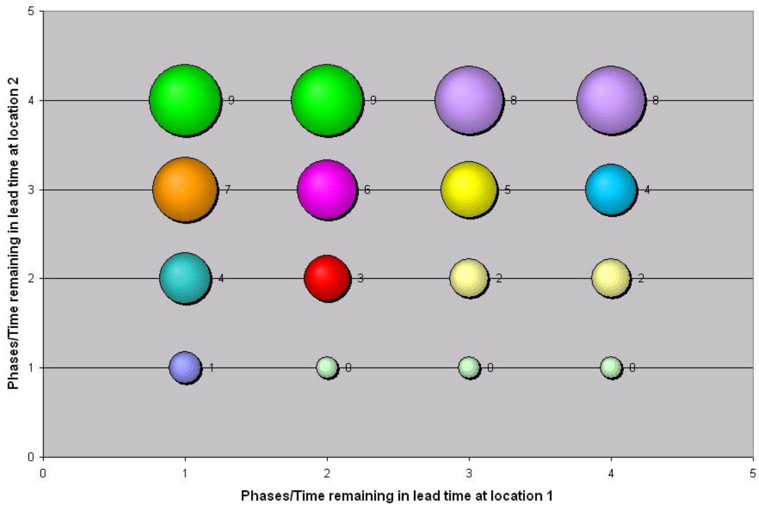

Figure 6.29. Suggested thresholds for a dynamic holdout transshipment policy when $b_1=5$ and $\hat{b_1}=4$

Additionally, Figures 6.25 and 6.28 plot the optimal transshipment policy when there are 2 phases remaining in the lead time of the outstanding order at location 1. Compared to the example used in Figures 6.17 to 6.20, $\hat{b_1}$ is lower and consequently there is less incentive for transshipment. The transshipment support provided by location 2 increases as the number of phases remaining in the lead time of the outstanding order at location 2 decreases. However, in contrast to Figures 6.17 to 6.20, complete pooling is only optimal when there is one phase remaining in the lead time. Transshipment is only optimal for all levels of inventory level at location 1, when the inventory level at location 2 is greater than 2, 6 and 9 for $w_2 = 2, 3$ and 4 respectively. This suggests the use of a dynamic holdout transshipment policy in which the threshold depends on the time until replenishment at location 2.

We conclude that the difference between the average total cost rates for the optimal general transshipment policy and the optimal holdout transshipment policy is most likely due to the fact that the holdout threshold is fixed and cannot vary with the times to replenishment at the locations. We propose a dynamic holdout transshipment policy with up to 16 different threshold values each corresponding to a different combination of (w_1,w_2) where $1 \leqslant w_k \leqslant 4$. The case $w_k = 0$ is like $w_k = 4$ due to memory-less property of the exponential distribution. In Section 6.4.3, we observed that, for exponential lead time, the threshold for the optimal holdout transshipment policy was equal to the largest inventory level for which it is optimal for location 2 to refuse a transshipment request. Extending this observation to the model with phase-type lead time, we propose using this method to choose the threshold value for a given state of the replenishment order process in a dynamic holdout transshipment policy.

Figure 6.29 plots the estimated threshold values for different states of the replenishment order process when $b_1 = 5$ and $\hat{b_1} = 4$. Both the size and colour of bubbles represent the size of the holdout threshold which is also shown by the numbers next to the bubbles. For example, the red colour bubble in the middle represents the threshold value 3 while the two green colour bubbles on the top represent the threshold value 9. The value of the threshold is non-increasing in the time remaining in the lead time at location 1 and non-decreasing in the time remaining in the lead time at location 2.

6.5 Conclusions

In this chapter, we have examined four **SMDP** models with exponential lead time and phase-type lead time for the general and holdout transshipment policy. Numerical experiments with the models demonstrate interesting properties of the optimal transshipment decisions, although, compared to complete pooling and no pooling, the total cost savings from selective transshipment in the examples we considered were not very significant.

The numerical experiments for the **SMDP** model with exponential lead time show that complete pooling is not the optimal transshipment policy. In all the examples we considered, the optimal policy when the inventory level at location 1 is zero is one of partial pooling and this gradually changes to complete pooling as the inventory level at location 1 decreases. The optimal total cost rate with the holdout transshipment policy is very close to the optimal total cost rate with the general transshipment policy. To approximate the fixed lead time more effectively, we developed the **SMDP** model with phase-type lead time. Unlike with exponential lead time, the optimal transshipment decisions depend on the state of the replenishment order process. We observed that it is optimal for location 2 to provide more transshipment support as the number of phases remaining in the lead time at location 2 decreases and as the number of phases remaining in the lead time at location 1 increases. These are intuitively appealing properties.

While, for the examples considered, the cost savings from optimal transshipment are less with the phase-type lead time, there is a greater difference between the optimal cost rates of the general transshipment and the holdout transshipment policy. We suggest that this is due to the use of a fixed holdout threshold which does not depend on the times until replenishment of the locations. We propose a dynamic holdout transshipment policy in which the threshold is a function of the state of the replenishment order process. The policy is specified by defining a threshold for each pair of value (w_1,w_2) satisfying $1 \leqslant w_k \leqslant W_k$. Determining an optimal set of thresholds for such a policy would be computationally difficult due to the large number of variables. However, we propose a method to estimate the set of threshold values from the optimal general transshipment policy.

Compared to general transshipment policies, the proposed dynamic holdout transshipment policy is easy to implement and to explain. These properties make it an attractive policy to use in practice. Further, we believe that the optimal cost rate for the proposed dynamic holdout transshipment policy would be close to the optimal cost rate for the general transshipment policy.

The thresholds in the dynamic holdout transshipment policy can be interpreted for a fixed lead time as follows. Let t be the time until replenishment at location k. If $\frac{(n-1)L_k}{W_k} < t \leqslant \frac{nL_k}{W_k}$, then use the threshold corresponding to $w_k = n$. We believe this dynamic holdout transshipment policy for fixed lead time would result in a lower average cost rate than the optimal holdout transshipment policy considered in chapter 4 and 5.

The modelling approach considered in this chapter offers benefits over the approximation models of earlier chapters. Firstly, rather than modelling each location as an independent location, we model the two-location system as a whole system. Hence, it is possible to capture all the interactions which occur between the two locations. Such an approach provides us with a powerful tool to improve our modelling of a system with strong interactions. Secondly, as we programme the decision policy, therefore we can monitor how the optimal decision depends on the state of the process. Hence, we benefit from insights on optimal transshipment decisions for the **SMDP** model. For example, the use of the **SMDP** model to predict optimal holdout thresholds. These properties of the **SMDP** models prove **SMDP** modelling to be an effective approach for our problem domain.

However, the **SMDP** modelling approach has its weaknesses. Firstly, we need a large number of states and actions to model the problem especially in the case of phase-type lead time. As a consequence of this, the solution method is computationally burdensome. It is unlikely that the method could be extended to more than two locations due to the exponential increase in the number of states and actions. Secondly, the **SMDP** approach does not provide explicit expressions for the average total cost rate and other performance measures such as direct fill rate and backorder fill rate.

In summary, the **SMDP** modelling technique provides a method of examining general transshipment policies for the two-location inventory system with unidirectional transshipment. In turn this has led to the development of a dynamic holdout transshipment policy that is easy to implement yet potentially very efficient in terms of cost. However, we have to pay significant computational prices.

Glossary

SMDP: semi-Markov decision process
IID: independent and identically distributed
phase-type distribution: the sum of a fixed number of **IID** exponential random variables
$\frac{1}{\mu_k}$: the mean replenishment lead time at location k
i_k: the inventory level at location k
$\tau_i(a)$: the expected time until the next decision epoch when decision a is chosen in state i

Chapter 7
Summary & Discussion

7.1 Research overview

In the previous chapters, we investigate the general unidirectional transshipment policy to the two-location inventory system by examining a variety of modelling approaches. Firstly, we develop three approximation models. For each model, explicit expressions for a range of performance measures are derived. We evaluate and compare these approximations with corresponding simulations for verification and numerical experiments.

Furthermore, two **SMDP** models are developed to get more accurate and deep insights of the interactions between the locations in the system. We compare the general and holdout transshipment policies and use the **DP** technique to predict the optimal holdout threshold by analysing the optimal decision distributions.

The models all show that partial pooling can help to improve the whole system performance measured, for example, by the average total cost. However, each modelling approach has its own strength and weakness. Subsequently, we give a summary and discussion of each modelling approach in our research as follows.

7.2 Research summary and discussion

7.2.1 Single-depot system

Prior to our series modelling on the two-location inventory control system, we begin our study on the single-depot system as our decomposition modelling approach for the two-location system is dependent on the single-depot system.

At the first step, we define the steady-state distribution of inventory level for the single-depot system in which an $(R,\ Q)$ replenishment order policy is applied. We derive the function $p(x)$ to be the probability that the inventory level is x at any time when the system is in steady-state. Then we derive all performance measures of this single-depot system straightforwardly.

To verify our derivations of all performance measures, we compare results based on our derivations with those from the single-depot simulation. This comparison shows that there is a high degree of agreement between the expressions derived and simulation. Thus, we can trust our analysis of the single-depot model and develop an approximation of the two-location model based on this single-depot model.

In addition, we provide numerical experiments to search for the optimal average total cost using an exhaustive search algorithm. These results suggest that the average total cost $C(R,\ Q)$ could be a convex function of Q.

7.2.2 Decomposition approach model with constant demand rates

This is our first analytic approximation model and it uses a decomposition approach. Rather than handling the explicit holdout transshipment policy, we first consider the two-location system in which the transshipment decision is made randomly by the transshipment agreement probability, z. In other words, location 2 is more likely to deliver a transshipment in response to a stockout at location 1 as the value of z increases.

We decompose the two-location system into two independent single-depot systems with constant Poisson demand rates. Given the value of z between 0 and 1, the two locations have independent, but modified demand rates. Based on these two modified demand rates, we formulate each location as a single-depot system, and derive cost approximations and system performance measures respectively.

The merit of this kind of modelling is that it is very intuitive and proves a quick approach to get a glimpse of the system performance. We perform numerical experiments on the backorder costs at the two locations. Results demonstrate that those cost parameters are important factors in determining the form of the optimal transshipment policy. The approximation model suggests that partial pooling is optimal for a number of the examples considered. Furthermore, estimates of holdout thresholds I_2 which correspond to the optimal transshipment agreement probability can be obtained. Hence, we can infer that the approximation model could be applied to inform the holdout transshipment policy.

In order to convince us of the accuracy of our decomposition approximation model, we check the test results against a simulation of the two-location system with holdout transshipment policy. Unfortunately, the simulation results are not consistent with the findings of the approximation model. However, there is evidence of consistency between these simulation results and the simulation of a two location system with random transshipment decisions.

From these numerical experiment results and analysis, we conclude that the decomposition approach model with the transshipment agreement probability does provide us with a quick reference to understand the impact of the holdout transshipment policy on the two-location inventory control system. The approximation using the modified demand rates might explain the difference between the decomposition approach and the simulation of the original system. Because of the intensive interactions between the two locations when the unidirectional transshipment policy is employed, decomposition using constant demand rates at the two locations is not enough to capture all moments.

7.2.3 Decomposition TAP model with non-constant demand rates

In order to establish an improved approximation model of the holdout transshipment policy applied to the two-location inventory system, we developed a new decomposition model in Chapter 5 with non-constant demand rates.

In Chapter 4, we found that the transshipment agreement probability can be used to reflect the holdout transshipment, however we need to approximate the demand process more closely to improve the approximation accuracy. Therefore, our new approach acknowledges that transshipment is only possible when location 1 has no stock and location 2 has some stock, and focuses on each location over a cycle. We derive the mean cycle time and distribution of inventory level at the end of a cycle at each location. An iterative algorithm is developed to estimate the average total cost and other performance measures for a given transshipment agreement probability.

To check the accuracy of our approximation, we compare the approximation expressions against the simulation of a single location with non-constant demand rate. The verification results demonstrate that our approximations for all terms show a high degree of consistency with the simulation.

In order to investigate the optimal holdout transshipment policy for the two-location system, we conduct numerical experiments on the stockout costs at the two locations. The results show that the **TAP** model predicts that partial pooling could be the optimal transshipment policy. However, those partial pooling policies predicted by the **TAP** model are not consistent with the optimal policies suggested by the simulation of the original system with the holdout transshipment policy.

Furthermore, we derive the optimal holdout threshold value at location 2 suggested by the **TAP** model by searching on the value of I_2. We compare this threshold value and optimal threshold determined by simulation of the original system with holdout transshipment policy. The results show that **TAP** model correctly predicts the optimal holdout value for some cases, however, there are some discrepancies between the predicted and optimal thresholds for other cases.

One advantage of this decomposition approach is the simple technique of modelling the system during a single cycle. It becomes possible to capture more moments due to transshipment interactions and reflect these interactions by more accurately modified demand rates. By this new approach, we are able to derive explicit expressions for a range of system performance measures under the assumptions of the decomposition approach.

Another advantage of this modelling is the extendable framework. Because there are similar derivations on the relevant cost approximations both for the **TAP** and the subsequent **HOT** model, we derive three lemmas in Section 5.3 of Chapter 5 to avoid duplicating the derivations. These three lemmas can be used commonly to derive the cost approximations for any single-depot system where customer demand is modelled by Poisson demand process and an (R, Q) replenishment order policy is employed.

The discrepancies between the numerical results of the **TAP** and simulation models could be due to the lack of accuracy from the transshipment agreement probability. So far, we only use the transshipment agreement probability to model the degree of transshipment from location 2 to location 1. Hence, it is worth developing a new decomposition approximation approach with an explicit holdout transshipment decision variable to replace the transshipment agreement probability.

7.2.4 Decomposition HOT model with non-constant demand rates

The holdout transshipment policy (**HOT**) model is our first approximation model which handles the holdout transshipment policy explicitly. Evolving from the **TAP** model, we also approximate all cost components during a cycle.

The main difference compared to the **TAP** model is that the transshipment decision is now dependent on the holdout threshold I_2 instead of the transshipment agreement probability z. Therefore, we are able to more accurately approximate distributions of inventory levels, demand rates and mean cycle times during the cycle. For the **HOT** model, we must consider more scenarios depending on the relationship between inventory level and holdout threshold at location 2. As before, explicit expressions for average total cost and other performance measures are derived.

To check the accuracy of our approximation, we compare the approximation expressions against the simulation of a single location with non-constant demand rate. The verification results demonstrate that our approximations for all terms show a high degree of consistency with the simulation.

We investigate the form of the optimal holdout transshipment policy by performing numerical experiments on the stockout cost rates at the two locations. The results from some cases show that the partial pooling policy could lead to significant cost savings compared to the policies of complete pooling and no pooling. This suggests that the whole system performance could be improved from the optimal holdout transshipment policy. In addition, we also compare the results with simulation of the original system with holdout transshipment policy. The results show that the **HOT** model often correctly predicts the optimal holdout threshold for the examples considered. Further, for the cases where the holdout thresholds predicted by the **HOT** model and determined by simulation are different, the average total cost rate when the **HOT** model threshold is applied is close to the optimal total cost rate. Hence, we conclude that the decomposition approach with explicit holdout transshipment decision variable provides an effective modelling approach to reflect the transshipment interactions between the two locations in the system.

7.2.5 SMDP models with exponential lead time

The reason we choose the **SMDP** approach is due to the nature of dynamic programming. The state and action space can be defined to capture all transshipment interactions between the two locations in the system. For the two-location inventory control system, this approach is appropriate and feasible because the transshipment interactions between the two locations occur dynamically and the scale of the state and action spaces is reasonable for computation. By applying the **SMDP** approach, the transshipment decisions can been reviewed with more consideration than with the analytic approximation approaches.

In the first part of Chapter 6, we developed two **SMDP** models with exponential lead time under the general and holdout transshipment policies. Instead of the assumption of fixed lead time which we used in the **TAP** and **HOT** models, we assume the exponential lead time. Under such assumption, the time until an order is delivered does not depend on the time that has passed since the order was placed because of the memory-less property of the exponential random variable.

These bespoke **SMDP** models are developed by defining the state space, action space, transition structure and immediate cost. We then investigate the effect of the backorder and stockout cost rates at location 1 on the optimal holdout threshold for a holdout transshipment policy. When these cost parameters are increasing, the optimal strategy for the system is to lower the holdout threshold in order to give more transshipment support. We found that the optimal average total cost rate under the holdout transshipment policy is very close to that with the general transshipment policy.

We also take advantage of information from the optimal general transshipment policy which can be determined from the **DP** approach. More interestingly, from plots of the optimal policy, we observe similar features between the optimal holdout threshold values and the optimal general transshipment policies. We conclude that the **SMDP** model is an effective approach to the two-location system with holdout transshipment policy under the assumption of exponential lead time.

7.2.6 SMDP models with phase-type lead time

Under the assumption of exponential lead time, the transshipment decision is not dependent on whether any replenishment orders are outstanding due to the memory-less property of exponential random variables. However, this assumption needs to be modified if we need a model to approximate fixed lead time effectively. Hence, we develop another **SMDP** model with phase-type lead time. The idea of phase-type lead time model is intuitive, rather than assuming a single exponential lead time, we assume that the lead time as the sum of a number of **IID** exponential random variables. As the number of phases increases, this phase-type model of replenishment lead time more closely approximates the fixed replenishment lead time. Accordingly, we redefine the state space, action space, transition structure and immediate cost for the two-location system.

To compensate for the increased complexity due to the phase-type model, we restrict the choice of order quantity at each location to a single value instead of allowing a set of possible order quantities as in the **SMDP** model with exponential lead time. For computational reasons, the limits on the number of outstanding backorders and the inventory level are restricted and the largest number of phases is limited to 4 for all numerical experiments.

Again, we conduct the numerical experiments on the backorder and stockout cost rates at location 1. Unlike with exponential lead time, the optimal transshipment decisions depend on the state of the replenishment order process. From the numerical results of the **SMDP** model with phase-type lead time, the most interesting property which we observe is that it is optimal for location 2 to provide more transshipment supports as the number of phases remaining in the lead time at location 2 decreases and as the number of phases remaining in the lead time at location 1 increases. However, there is a greater difference between the optimal cost rates of the general and holdout transshipment policies, which is most likely due to the use of a fixed holdout threshold which does not depend on the times until replenishment of the locations.

Alternatively, we propose a dynamic holdout transshipment policy in which the threshold is a function of the state of the replenishment order process. The policy is specified by defining a holdout threshold which depends on the number of phases remaining in the lead times of outstanding orders at the two locations. We propose a quick method to estimate the set of threshold values from the optimal general transshipment policy. Compared to the general transshipment policy, we suggest that dynamic holdout transshipment policy is easy to implement and to explain. Further, we believe that the optimal cost rate for the proposed dynamic holdout transshipment would be close to the optimal cost rate for the general transshipment policy.

Therefore, we have more confidence that the **SMDP** model with phase-type lead time is an effective approach to approximate the original inventory system with a fixed lead time. The phase-type lead time assumption provides us another angle to develop deep insights of system behaviours.

7.3 Further research

Our research work on the unidirectional holdout transshipment policy demonstrates partial pooling features of the two-location system. Through a series of modelling approaches from decomposition approximation models to the **SMDP** models, under some circumstances, our numerical experiments show that holdout partial pooling could be the optimal cost policy. Hence, holdout partial pooling can provide easily understood solutions for the system without costing too much on the total system operation cost. That is a good option for the inventory manager. Meanwhile, there are other subjects and directions we might explore in our future research study.

In chapter 6, we introduced a quick approach to derive the dynamic holdout threshold by plotting the optimal transshipment decisions. However, we still consider a fixed holdout transshipment policy in our **SMDP** models. Hence, it is worth to develop our **SMDP** models by applying the dynamic holdout thresholds which are dependent on the times until replenishment of the locations.

So far, we only review the two-location inventory control system with unidirectional holdout transshipment policy. It is worthwhile to review three or more locations system in our future research as multi-location inventory control system is more applicable to the real problem domain. For systems with three or more locations, it is not appropriate to apply the **SMDP** modelling technique because of the dramatic increases in the computational workload from the huge increases in the scale of the state and action spaces. However, we can apply the decomposition modelling techniques from the **TAP** and **HOT** models to systems with three or more locations.

In addition to the optimal total cost for the two-location system, it is also interesting to study other system performance measures such as direct fill rate, backorder fill rate, transshipment fill rate in the future. The relations between optimal total cost and other system performance measures might provide us with more information on the impact from the given holdout transshipment policy to the multi-location system.

In our models, we have assumed an unidirectional transshipment policy, which can be relaxed to allow bidirectional transshipment in a multi-location system. There are other assumptions which we might consider to review, for instance, the assumption which only allows at most one outstanding replenishment order at each location, and the assumption which only permit one item for each transshipment delivery. We also assume all demands occur according to Poisson demand processes, which is very common in our research field. However, it is interesting to consider other probability distributions, for example, geometric random variable. If the new demand distribution is defined, the **TAP** and **HOT** models need a fundamental review because we apply the **Erlang** distribution as the condition to approximate the time period during which inventory level falls down.

Bibliography

[Alfredsson and Verrijdt(1999)] Alfredsson, P., Verrijdt, J., 1999. Modeling emergency supply flexibility in a two-echelon inventory systems. Management Science 45, 1416–1431.

[Archibald et al.(1997)] Archibald, T., Sassen, S., Thomas, L., February 1997. An optimal policy for a two depot inventory problem with stock transfer. Management Science 43 (2), 173–183.

[Axsater(1990)] Axsater, S., 1990. Modelling emergency lateral transshipments in inventory systems. Management Science 36, 1329–1338.

[Axsater(2003a)] Axsater, S., 2003a. Evaluation of unidirectional lateral transshipments and substitutions in inventory systems 149, 438–447.

[Axsater(2003b)] Axsater, S., 2003b. A new decision rule for lateral transshipments in inventory systems. Management Science 49, 1168–1179.

[Bellman(1957)] Bellman, R., 1957. Dynamic programming. Princeton University Press, Princeton, New Jersey.

[Das(1975)] Das, C., 1975. Supply and redistribution rules for two-location inventory systems: One-period analysis. Management Science 21, 765–776.

[Dong and Rudi(2004)] Dong, L. X., Rudi, N., 2004. Who benefits from transshipment? exogenous vs. endogenous wholesale prices. Management Science 50, (5), 645–657.

[Economist(2006)] Economist, 17th, June 2006. A survey of logistics. Economist, 3–20.

[Evers(2001)] Evers, P. T., 2001. Heuristics for assessing emergency transshipments. European Journal of Operational Research 129, 311–316.

[Grahovac and Chakravarty(2001)] Grahovac, J., Chakravarty, A., 2001. Sharing and lateral transshipment of inventory in a supply chain with expensive low-demand items. Management Science 47, (4), 579–594.

[Hadley and Whitin(1963)] Hadley, G., Whitin, T., 1963. Analysis of Inventory Systems. Prentice-Hall, Englewood Cliffs, N.J.

[Herer et al.(2006)] Herer, Y. T., Tzur, M., Yucesan, E., 2006. The multilocation transshipment problem. Iie Transactions 38, (3), 185–200.

[Howard(1971)] Howard, R. A., 1971. Dynamic Probabilistic Systems. John Wiley & Sons, Inc.

[Hu et al.(2005)] Hu, J., Watson, E., Schneider, H., 2005. Approximate solutions for multi-location inventory systems with transshipments. International journal of production economics 97, 31–43.

[Hu et al.(2007)] Hu, X., Duenyas, I., Kapuscinski, R., 2007. Existence of coordinating transshipment prices in a two-location inventory model. Management Science 53, 1289–1302.

[Jonsson and Silver(1987)] Jonsson, H., Silver, E. A., 1987. Analysis of a two-echelon inventory control system with complete redistribution. Management Science 33, 215–227.

[Karmarkar(1987)] Karmarkar, U. S., 1987. The multilocation multiperiod inventory problem: Bounds and approximations. Management Science 33, 86–94.

[Krishnan and Rao(1965)] Krishnan, Rao, K. V., 1965. Inventory control in n warehouses. J. Indust. Engrg XVI, 212–215.

[Kukreja et al.(2001)] Kukreja, A., Schmidt, C. P., Miller, D. M., October 2001. Stocking decisions for low-usage items in a multilocation inventory system. Management Science 47 (10), 1371–1383.

[Kutanoglu(2008)] Kutanoglu, E., 2008. Insights into inventory sharing in service parts logistics systems with time-based service levels. Computers & Industrial Engineering 54, 341–358.

[Lee(1987)] Lee, H. L., 1987. A multi-echelon inventory model for repairable items with emergency lateral transshipments. Management Science 33, 1302–1316.

[Minner and Silver(2005)] Minner, S., Silver, E. A., 2005. Evaluation of two simple extreme transshipment strategies. International Journal Of Production Economics 93-94, 1–11.

[Minner et al.(2003)] Minner, S., Silver, E. A., Robb, D. J., 2003. An improved heuristic for deciding on emergency transshipments. European Journal Of Operational Research 148,(2), 384–400.

[Netessine and Rudi(2006)] Netessine, S., Rudi, N., 2006. Supply chain choice on the internet. Management Science 52,(6), 844–864.

[Ozdemir et al.(2006)] Ozdemir, D., Yucesan, E., Herer, Y. T., 2006. Multi-location transshipment problem with capacitated transportation. European Journal Of Operational Research 175,(1), 602–621.

[Puterman(1994)] Puterman, M. L., 1994. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc.

[Robinson(1990)] Robinson, L. W., March-April 1990. Optimal and approximate policies in multi-period, multilocation inventory models with transshipments. Operations Research 38 (2), 278–295.

[Rossman(1996)] Rossman, A., 1996. Workshop Statistics: Discovery with Data. Springer-Verlag.

[Rudi et al.(2001)] Rudi, N., Kapur, S., Pyke, D. F., 2001. A two-location inventory model with transshipment and local decision making. Management Science 47,(12), 1668–1680.

[Seifert et al.(2006a)] Seifert, R. W., Thonemann, U. W., Sieke, M. A., 2006a. Integrating direct and indirect sales channels under decentralized decision-making. International Journal Of Production Economics 103,(1), 209–229.

[Seifert et al.(2006b)] Seifert, R. W., Thonemann, U. W., Sieke, M. A., 2006b. Relaxing channel separation: Integrating a virtual store into the supply chain via transshipments. Iie Transactions 38,(11), 917–931.

[Sherbrooke(1992)] Sherbrooke, C. C., 1992. Multiechelon inventory systems with lateral supply. Naval Research Logistics 39, 29–40.

[Silver and Peterson(1985)] Silver, E. A., Peterson, R., 1985. Decision Systems for Inventory Management and Production Planning, 2nd Edition. John Wiley & Sons.

[Silver et al.(1998)] Silver, E. A., Pyke, D. F., Peterson, R., 1998. Inventory Management and Production Planning and Scheduling. John Wiley & Sons.

[Tagaras and Cohen(1992)] Tagaras, G., Cohen, M. A., 1992. Pooling in two-location inventory systems with non-negligible replenishment lead times. Management Science 38, 1067–1083.

[Tagaras and Vlachos(2001)] Tagaras, G., Vlachos, D., March 2001. A periodic review inventory system with emergency replenishment. Management Science 47 (3), 415–429.

[Tagaras and Vlachos(2002)] Tagaras, G., Vlachos, D., 2002. Effectiveness of stock transshipment under various demand distributions and nonnegligible transshipment times. Production And Operations Management 11,(2), 183–198.

[Tijms(1986)] Tijms, H. C., 1986. Stochastic Modelling and Analysis A Computational Approach. John Wiley & Sons.

[Tijms(1994)] Tijms, H. C., 1994. Stochastic models : an algorithmic approach. Wiley series in probability and mathematical statistics. Wiley.

[Tijms(1995)] Tijms, H. C., 1995. Stochastic Models An Algorithmic Approach. John Wiley & Sons.

[Tijms(2003)] Tijms, H. C., 2003. A First Course in Stochastic Models. John Wiley & Sons.

[Wee and Dada(2005)] Wee, K. E., Dada, M., 2005. Optimal policies for transshipping inventory in a retail network. Management Science 51,(10), 1519–1533.

[Winston(2004)] Winston, W. L., 2004. Operations Research, 4th Edition. Thomson Learning.

[Wong et al.(2005a)] Wong, H., Cattrysse, D., Oudheusden, D. V., 2005a. Inventory pooling of repairable spare parts with non-zero lateral transshipment time and delayed lateral transshipments - production, manufacturing and logistics. European Journal Of Operational Research 165,(1), 207–218.

[Wong et al.(2005b)] Wong, H., Cattrysse, D., Oudheusden, D. V., 2005b. Stocking decisions for repairable spare parts pooling in a multi-hub system. International Journal Of Production Economics 93-94, 309–317.

[Wong et al.(2005c)] Wong, H., Houtum, G. J. V., Cattrysse, D., Oudheusden, D. V., 2005c. Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints. Journal Of The Operational Research Society 56,(12), 1419–1430.

[Wong et al.(2006)] Wong, H., Houtum, G. J. V., Cattrysse, D., Oudheusden, D. V., 2006. Multi-item spare parts systems with lateral transshipments and waiting time constraints. European Joural of Operational Research 171, 1071–1093.

[Xu et al.(2003)] Xu, K. F., Evers, P. T., Fu, M. C., 2003. Estimating customer service in a two-location continuous review inventory model with emergency transshipments. European Journal Of Operational Research 145,(3), 569–584.

[Yang and Qin(2007)] Yang, J., Qin, Z., 2007. Capacitated production control with virtual lateral transshipments. Operations Research 55, 1104–1119.

[Zhao et al.(2005)] Zhao, H., Deshpande, V., Ryan, J. K., 2005. Inventory sharing and rationing in decentralized dealer networks. Management Science 51,(4), 531–547.

[Zhao et al.(2008)] Zhao, H., Deshpande, V., Ryan, J. K., 2008. Optimal dynamic production and inventory transshipment policies for a two-location make-to-stock system. Operations Research 56,(2), 400–410.

Appendix A
Simulation implementations

For further reference, we enclose the **JAVA** codes of our simulation models cooresponding to the **TAP** and **HOT** models. Only main methods and classes are provided, the simulation implementation of the single depot system and the rest of classes are omitted.

Short-term references

LAM: Poisson demand rate

C: order cost rate per each order

B: backorder cost rate

SO: stockout cost rate

H: holding cost per item per time unit

R,Q: values of Reorder position and order quantity.

IS: initial stock level

IT: initial time

nSim: simulation times

WT: warmup time

ET: end time

VT: valid time period

Tsd: next system demand arrival time

SL: stock level

SP: stock position

VDA: all valid demands

FR: direct fill rate

FRB: backorder fill rate

TC: average cost for the whole system

OC: average order cost

BC: average backorder cost

SOC: average stockout cost

HC: average holding cost

A.1 Simulation implementation for the TAP model

```
/*
 * Simulation for two-depot system with transshipment factor Z
 * under (R,Q) replenishment policy
 */

import java.io.*;
import java.util.*;

public class simuTAP
{
public static final int IT1=0, LT1=1, IT2=0, LT2=1;
public static final double WT=500, ET=5000, VT=(ET-WT);
public static final int nSim=500;

public static void main(String [] args) throws IOException
{
initialise the parameter set;

// initialise the inventory stock both at two locations;

int     IS1     = R1+Q1;
int     IS2     = R2+Q2;

double  LAM     = LAM1+LAM2;

for (double z=1; z>=0; z=z-0.1)
{
//initialise the variable array for standard error;

double [] FR1   = new double [nSim];
double [] FRB1  = new double [nSim];

double [] TC1   = new double [nSim];
double [] OC1   = new double [nSim];
double [] SOC1  = new double [nSim];
double [] BC1   = new double [nSim];
double [] HC1   = new double [nSim];

double [] FR2   = new double [nSim];
double [] FRB2  = new double [nSim];

double [] TC2   = new double [nSim];
double [] OC2   = new double [nSim];
double [] SOC2  = new double [nSim];
double [] BC2   = new double [nSim];
double [] HC2   = new double [nSim];

double [] FRTR21= new double [nSim];
double [] TRC21 = new double [nSim];
```

```
double [] TCALL = new double [nSim];

for (int j=0; j<nSim; j++)
{
/*
 *  counters for Order number, back order number
 *  and transshipment number
 */

double ON1=0, BON1=0, TRON21=0,
   ON2=0, BON2=0;

double FR1_=0, FRB1_=0, FR2_=0, FRB2_=0, FRTR21_=0;

// Initialise two-depot system

Depot depot1 = new Depot();
Vector OO1   = depot1.OOS;

double SL1   = IS1;
double SP1   = IS1;

depot1.addST(IT1);
depot1.addSL(SL1);

Depot depot2 = new Depot();
Vector OO2   = depot2.OOS;

double SL2   = IS2;
double SP2   = IS2;

depot2.addST(IT2);
depot2.addSL(SL2);

Random rgn = new Random(new Integer(1500+100*j).longValue());

/*
 * Tsd: next system arrival time
 * VDA: all demands during the valid time period
 */

double Tsd = 0;
double VDA1= 0, VDA2= 0;

while (Tsd<ET)
{
Tsd = Tsd-Math.log(rgn.nextDouble())/LAM;

/*
 *  Check the outstanding orders
 */

while(!OO1.isEmpty())
{
double OOT1   = ((Double) OO1.firstElement()).doubleValue();
```

```
if(Tsd < OOT1)
        break;
else
{
        SL1  = SL1 + Q1;
        OO1.remove(OO1.firstElement());
        depot1.addST(OOT1);
        depot1.addSL(SL1);
        continue;
}

}

while(!OO2.isEmpty())
{
double OOT2   = ((Double) OO2.firstElement()).doubleValue();
if(Tsd < OOT2)
break;
else
{
SL2  = SL2 + Q2;
OO2.remove(OO2.firstElement());
depot2.addST(OOT2);
depot2.addSL(SL2);
continue;
}
}

/*
*  --------
*  Depot 1
*  --------
*/

if (rgn.nextDouble()<LAM1/LAM)
{
if (Tsd>=WT)
VDA1++;

// When stock level is above zero

if (SL1>0)
{
SL1 = SL1-1;
SP1 = SP1-1;
depot1.addST(Tsd);
depot1.addSL(SL1);

if (Tsd>=WT)
FR1_++;

if (SP1==R1)
{
// places the order
SP1 = SP1+Q1;
depot1.addOO((double)(Tsd+LT1));
```

```
if (Tsd>=WT)
ON1++;
}
continue;
}

else // SL1<=0, transshipment is considered
{
/*
* when depot 2 has enough stock to transship
*/

if (rgn.nextDouble()<z)
{
SL2 = SL2-1;
SP2 = SP2-1;
depot2.addST(Tsd);
depot2.addSL(SL2);

depot1.addTrans((double)Tsd);

if (Tsd>=WT)
{
TRON21++;
FRTR21_++;
}

if(SP2==R2)
{
// depot 2 places the order
SP2 = SP2+Q2;
depot2.addOO((double)(Tsd+LT2));

if (Tsd>=WT)
        ON2++;
}
continue;
}

/*
* no transshipment made, depot 1 places
* the backorder only
*/

else // no transshipment made
{
SL1 = SL1-1;
SP1 = SP1-1;

depot1.addST(Tsd);
depot1.addSL(SL1);

depot1.addBO((double)Tsd);

if (Tsd >= WT)
```

```
{
BON1++;
FRB1_++;
}

if (SP1==R1)
{
// depot 1 places the order
SP1 = SP1+Q1;
depot1.addOO((double)(Tsd+LT1));

if (Tsd>=WT)
        ON1++;
}
continue;
}
}
}

/*
* --------
* depot 2
* --------
*/

else
{
if (Tsd>=WT)
VDA2++;

if (SL2>0)
{
SL2 = SL2-1;
SP2 = SP2-1;
depot2.addST(Tsd);
depot2.addSL(SL2);

if (Tsd>=WT)
FR2_++;

if (SP2==R2)
{
// depot 2 places the order
SP2 = SP2+Q2;
depot2.addOO((double)(Tsd+LT2));

if(Tsd>=WT)
ON2++;
}
continue;
}

else // SL2<=0
{
SL2 = SL2-1;
SP2 = SP2-1;
```

```
depot2.addST(Tsd);
depot2.addSL(SL2);
depot2.addBO((double)Tsd);

if (Tsd>=WT)
{
BON2++;
FRB2_++;
}

if (SP2==R2)
{
// depot 2 places the order
SP2 = SP2+Q2;
depot2.addOO((double)(Tsd+LT2));

if (Tsd>=WT)
ON2++;
}
continue;
}
}
}

double OC1_             = C1*ON1/VT;
double SOC1_    = SO1*BON1/VT;
double BC1_     = B1*Lib.ABL(depot1.STV,depot1.SLV, WT,ET);
double HC1_     = H1*Lib.AIL(depot1.STV,depot1.SLV,WT,ET);
double TRC21_   = TR21*TRON21/VT;
double TC1_     = OC1_+SOC1_+BC1_+HC1_;

double OC2_     = C2*ON2/VT;
double SOC2_    = SO2*BON2/VT;
double BC2_     = B2*Lib.ABL(depot2.STV,depot2.SLV, WT,ET);
double HC2_     = H2*Lib.AIL(depot2.STV,depot2.SLV,WT,ET);
double TC2_     = OC2_+SOC2_+BC2_+HC2_;

double TCALL_   = TC1_+TC2_+TRC21_;

Print result;
}
```

A.2 Simulation implementation for the HOT model

```
/*
* Simulation for two-depot system with holdout transshipment
* under (R,Q) replenishment policy
*/

import java.io.*;
import java.util.*;
```

```
public class simuHOT
{
//Simulation parameters

public static final double WT = 5000, DT = 50000, RT = DT - WT;

//Variables to control simulation
static int nxtEvType;
static double nxtEvTime,tSE,tSD;
static Depot d1,d2;

public static final int nSim=500;

public static void main(String [] args) throws IOException
{

initialise the parameter sets;

int     IS1     = ROP1+OQ1;

int     IS2     = ROP2+OQ2;

double LAM = LAM1+LAM2, SPLIT = LAM1/LAM;

//Simulation parameters

double [] TC1   = new double [nSim];
double [] OC1   = new double [nSim];
double [] SOC1  = new double [nSim];
double [] BC1   = new double [nSim];
double [] HC1   = new double [nSim];

double [] TC2   = new double [nSim];
double [] OC2   = new double [nSim];
double [] SOC2  = new double [nSim];
double [] BC2   = new double [nSim];
double [] HC2   = new double [nSim];
double [] TRC21 = new double [nSim];
double [] TCALL = new double [nSim];

double [] FR1   = new double [nSim];
double [] FRB1  = new double [nSim];
double [] FR2   = new double [nSim];
double [] FRB2  = new double [nSim];
double [] FRTR21= new double [nSim];

Depot.tInf = 2*DT;

for (int HO2=1; HO2<=2*OQ2; HO2++)
{
//Initialise the simulation for a holdout value

//int HO = ROP2 + OQ2;
Random rgn = new Random(new Integer (1500).longValue());
```

```
for (int j=0; j<nSim; j++)
{
//Initialise a simulation run for this holdout value
d1 = new Depot(OQ1, ROP1, IS1);
d2 = new Depot(OQ2, ROP2, IS2);
double cT = 0;
tSE = WT;

//Perform a simulation run
while (cT < DT)
{
tSD = cT - Math.log(rgn.nextDouble())/LAM;
nextEvent();
d1.areaAboveBelow(cT,nxtEvTime);
d2.areaAboveBelow(cT,nxtEvTime);
if (nxtEvType == 0)
if (tSE > WT)
{
//System.out.println("Run "+i+" over");
nxtEvTime = Depot.tInf;
}
else
{
//System.out.println("Warm-up "+i+" over");
d1.zeroAll();
d2.zeroAll();
tSE = DT;
}

else if (nxtEvType == 1)
{
d1.receiveOrder();
}

else if (nxtEvType == 2)
{
d2.receiveOrder();
}

else
{
if (rgn.nextDouble() < SPLIT)
{
if ((d1.sL < 1) && (d2.sL > HO2))
{
d2.removeItem(tSD);
d1.nTR = d1.nTR + 1;
}
else
{
d1.removeItem(tSD);
}
}
else
{
d2.removeItem(tSD);
```

```
}
}
cT = nxtEvTime;
}

//Print out average performance measures

OC1[j]   = C1*d1.nOO/RT;
HC1[j]   = H1*d1.aHL/RT;
SOC1[j]  = SO1*d1.nBO/RT;
BC1[j]   = B1*d1.aBO/RT;
TRC21[j] = TR21*d1.nTR/RT;

OC2[j]   = C2*d2.nOO/RT;
HC2[j]   = H2*d2.aHL/RT;
SOC2[j]  = SO2*d2.nBO/RT;
BC2[j]   = B2*d2.aBO/RT;

TC1[j]   = (C1*d1.nOO + SO1*d1.nBO + B1*d1.aBO + H1*d1.aHL + TR21*d1.nTR)/RT;
TC2[j]   = (C2*d2.nOO + SO2*d2.nBO + B2*d2.aBO + H2*d2.aHL)/RT;
TCALL[j] = TC1[j]+TC2[j];
}

print the result;
}

public static void nextEvent()
{
if (tSE < tSD && tSE < d1.tOO && tSE < d2.tOO)
{
nxtEvType = 0;
nxtEvTime = tSE;
}

else if (d1.tOO < tSD && d1.tOO < d2.tOO)
{
nxtEvType = 1;
nxtEvTime = d1.tOO;
}

else if (d2.tOO < tSD)
{
nxtEvType = 2;
nxtEvTime = d2.tOO;
}

else
{
nxtEvType = 3;
nxtEvTime = tSD;
}
}
}
```

Appendix B
Approximation implementations

For the references, we give the **JAVA** codes for the major classes and methods of the cost approximations for the **TAP** model, **HOT** model and **SMDP** model with phased-type lead time. The implementations for the single depot model and all rest of classes and methods are omitted.

B.1 Approximation implementation for the TAP model

```
public class LocationTap1
{
double lam1, L1, c1, h1, so1, b1, tr21;
int r1, q1;
static final double TOL1 = 1E-16;
static final double TOL2 = 1E-14;
static final double accuracy   = 1E-4;
static final int maxIntervals  = 10000;
double lowerLimit=0, upperLimit;
double OC1, HC1, SOC1, BC1, TRC21, mct1;

public LocationTap1(double lam1_, int r1_, int q1_, double L1_,
double c1_, double h1_, double so1_, double b1_, double tr21_)
{
lam1        = lam1_;
r1          = r1_;
q1          = q1_;
L1          = L1_;
c1          = c1_;
h1          = h1_;
so1         = so1_;
b1          = b1_;
tr21        = tr21_;
upperLimit  = L1;
}

/*********************************************************************************
********************************************************************************
* Standard basic library function to computer fac, lead time demand
distribution
* and probability distribution of X1(R1)
********************************************************************************
```

```
*********************************************************************************
*/


public double PX1R1(double alpha, double z, int j)
{
double result = 0;
if (j>=1 && j<=r1)
result = Math.pow(lam1*L1,r1-j)*Math.exp(-lam1*L1)/Lib.fac(r1-j);

else if (j<=0)
{
/*
* declare a integral function and initialise the parameter set
*/

double lam1New = lam1*(1-alpha*z);

PX1R1 f = new PX1R1();
f.lam1   = lam1;
f.r1     = r1;
f.alpha  = alpha;
f.z      = z;
f.L1     = L1;
f.j      = j;

double coefficient   = Math.pow(lam1,r1)*Math.pow(lam1New,-j)
                       *Math.exp(-lam1New*L1)/(Lib.fac(r1-1)*Lib.fac(-j));
result               = coefficient*Integration.trapezium(f,
                       lowerLimit,upperLimit,accuracy,maxIntervals);
}
return result;
}

/************************************************
***********************************************
* Computation of E[X1(R1)], E[X1(R1)~2]
***********************************************
***********************************************
*/

public double EX1R1(double alpha, double z)
{
int j = r1;
double summ      = j*PX1R1(alpha,z,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*PX1R1(alpha,z,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX1R1(alpha,z,j))> TOL2);
return summ;
}

public double EX1R1S(double alpha, double z)
```

```
{
int j = r1;
double summ      = j*j*PX1R1(alpha,z,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*j*PX1R1(alpha,z,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX1R1(alpha,z,j))> TOL2);
return summ;
}

public double beta(double alpha, double z)
{
SigmaSumBetaTap SS1  = new SigmaSumBetaTap();
SS1.lam1             = lam1;
SS1.r1               = r1;
SS1.L1               = L1;

double ssbta         = SigmaSum.sumInfP(SS1,r1);
double ssbtb         = SigmaSum.sumInfP(SS1,r1+1);
double nominator     = L1*ssbta-r1*ssbtb/lam1;
double denominator   = (q1-r1)/lam1+EX1R1(alpha,z)/lam1+L1;
return nominator/denominator;
}

/**************************************************************
**************************************************************
*    Cost Evaluation Part
**************************************************************
**************************************************************
*/

public double CostEval1(double alpha, double beta, double z)
{
double ex1r1     = EX1R1(alpha,z);
double ex1r1s    = EX1R1S(alpha,z);

// Mean Cycle Time at depot 1
mct1      = (q1-r1+ex1r1)/lam1+L1;

/*
*  Order Cost Evaluation
*/

OC1   = c1/mct1;

/*
* Holding Cost Evaluation
* HC1a stands for the holding cost during the pre lead time
* HC1b stands for the holding cost during the lead time
*/

double HC1a  = (q1*q1+q1-r1*r1-r1+(2*q1+1)*ex1r1+ex1r1s)/(2*lam1);
double HC1b  = L1*r1*Lib.F(r1-1,lam1*L1)
```

```
                -lam1*L1*L1*Lib.F(r1-2,lam1*L1)/2
                +r1*(r1+1)*Lib.Fbar(r1,lam1*L1)/(2*lam1);

HC1    = h1*(HC1a+HC1b)/mct1;

/*
* Backorder Cost Evaluation
*/

StockOut1 stockout1  = new StockOut1();
stockout1.alpha      = alpha;
stockout1.z          = z;
SOC1                 = so1*SigmaSum.sumInfP1(stockout1, 1, this)/mct1;

/*
* backorder cost for depot 1
*/

BackOrder1 backorder1    = new BackOrder1();
backorder1.lam1          = lam1;
backorder1.L1            = L1;
backorder1.r1            = r1;

double coefficient   = b1*Math.pow(lam1,r1+1)*(1-alpha*z)/(2*Lib.fac(r1-1));

BC1                  = coefficient*Integration.trapezium(backorder1,
                       lowerLimit,upperLimit,accuracy,maxIntervals)/mct1;

TRC21                = z*tr21*lam1*alpha*beta;
return (OC1+HC1+SOC1+BC1);
}
}

/*
* Cost evaluation for TAP model at location 2
*/

import java.io.*;
import java.util.*;

public class LocationTap2
{
double lam1, lam2, L2, c2, h2, so2, b2;
int r2, q2;
static final double TOL1       = 1E-16;
static final double TOL2       = 1E-14;
static final double accuracy   = 1E-4;
static final int maxIntervals  = 10000;
double lowerLimit=0, upperLimit;
double OC2, HC2, SOC2, BC2, mct2;

public LocationTap2(double lam1_, double lam2_, int r2_, int q2_, double L2_,
double c2_, double h2_, double so2_, double b2_)
{
lam1 = lam1_;
lam2 = lam2_;
```

```
r2   = r2_;
q2   = q2_;
L2  = L2_;
c2   = c2_;
h2   = h2_;
so2  = so2_;
b2   = b2_;
upperLimit = L2;
}

/*********************************************************************************
*********************************************************************************
* Standard basic library function to compute lead time demand distribution
* and probability distribution of X2(R2)
*********************************************************************************
*********************************************************************************
*/

public double PX2R2(double beta, double z, int j)
{
double result = 0;
double lam2New = lam2+lam1*z*beta;

if (j>=1 && j<=r2)
{
result = Math.pow(L2*lam2New,r2-j)*Math.exp(-lam2New*L2)/Lib.fac(r2-j);
}

else if (j<=0)
{
/*
* declare a integral function and initialise the parameter set
*/

PX2R2 f = new PX2R2();
f.lam1   = lam1;
f.r2     = r2;
f.beta   = beta;
f.z      = z;
f.L2     = L2;
f.j      = j;

double coefficient   = Math.pow(lam2New,r2)*Math.pow(lam2,-j)
                       *Math.exp(-lam2*L2)/(Lib.fac(r2-1)*Lib.fac(-j));
result               = coefficient*Integration.trapezium(f,
                       lowerLimit,upperLimit,accuracy,maxIntervals);
}
return result;
}

/**********************************************
*********************************************
* Computation of E[X2(R2)], E[X2(R2)~2]
*********************************************
*********************************************
*/
```

```
public double EX2R2(double beta, double z)
{
int j = r2;
double summ      = j*PX2R2(beta,z,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*PX2R2(beta,z,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX2R2(beta,z,j)) > TOL2);

return summ;
}

public double EX2R2S(double beta, double z)
{
int j = r2;
double summ      = j*j*PX2R2(beta,z,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*j*PX2R2(beta,z,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX2R2(beta,z,j)) > TOL2);
return summ;
}

public double alpha(double beta, double z)
{
double lam2New = lam2+lam1*z*beta;

SigmaSumAlphaTap SS1 = new SigmaSumAlphaTap();
SS1.L2               = L2;
SS1.lam2New          = lam2New;

double ssata     = SigmaSum.sumInfP(SS1,r2);
double ssatb     = SigmaSum.sumInfP(SS1,r2+1);
double nominator = Math.exp(-lam2New*L2)*(L2*ssata-r2*ssatb/lam2New);
mct2             = (q2-r2+EX2R2(beta,z))/lam2New+L2;;
return 1-nominator/mct2;
}


/***************************************************************
***************************************************************
*  Cost Evaluation Part
***************************************************************
***************************************************************
*/

public double CostEval2(double alpha, double beta, double z)
{
double lam2New = lam2+lam1*z*beta;
```

```
double ex2r2   = EX2R2(beta,z);
double ex2r2s  = EX2R2S(beta,z);

// Mean cycle time at depot 2
mct2    = (q2-r2+ex2r2)/lam2New+L2;

/*
*  Order Cost Evaluation
*/

OC2   = c2/mct2;

/*
* Holding Cost Evaluation
*/

double HC20  = (q2*q2+q2-r2*r2-r2+(2*q2+1)*ex2r2+ex2r2s)/(2*lam2New);
double HC21  = L2*r2*Lib.F(r2-1,lam2New*L2)
               -lam2New*L2*L2*Lib.F(r2-2,lam2New*L2)/2
               +r2*(r2+1)*Lib.Fbar(r2,lam2New*L2)/(2*lam2New);

HC2   = h2*(HC20+HC21)/mct2;

/*
* Stockout cost for depot 2
*/

StockOut2 stockout2      = new StockOut2();
stockout2.beta           = beta;
stockout2.z              = z;

SOC2                     = so2*SigmaSum.sumInfP2(stockout2, 1, this)/mct2;

/*
* Backorder cost for depot 2
*/

BackOrder2 backorder2    = new BackOrder2();
backorder2.L2            = L2;
backorder2.r2            = r2;
backorder2.lam2New       = lam2New;

double coefficient   = b2*Math.pow(lam2New,r2)*lam2/(2*Lib.fac(r2-1));
BC2                  = coefficient*Integration.trapezium(backorder2,
                       lowerLimit,upperLimit,accuracy,maxIntervals)/mct2;
return (OC2+HC2+SOC2+BC2);
}
}
```

B.2 Approximation implementation for the HOT model

```
public class LocationHot1
{
double lam1, L1, c1, h1, so1, b1, tr21;
int r1, q1;
static final double TOL1 = 1E-16;
static final double TOL2 = 1E-16;
static final double accuracy   = 1E-6;
static final int maxIntervals  = 10000;
double lowerLimit, upperLimit;
double OC1, HC1, SOC1, BC1, TRC21;

public LocationHot1(double lam1_, int r1_, int q1_, double L1_, double c1_,
double h1_, double so1_, double b1_, double tr21_)
{
lam1 = lam1_;
r1   = r1_;
q1   = q1_;
L1   = L1_;
c1   = c1_;
h1   = h1_;
so1  = so1_;
b1   = b1_;
tr21 = tr21_;
lowerLimit = 0;
upperLimit = L1;
}

/*********************************************************************************
********************************************************************************
* Standard basic library function to computer fac, lead time demand
distribution
* and probability distribution of X1(R1)
********************************************************************************
********************************************************************************
*/


public double PX1R1(double omega, int j)
{
double result = 0;
double lam1New = lam1*(1-omega);

if (j>=1 && j<=r1)
result = Math.pow(lam1*L1,r1-j)*Math.exp(-lam1*L1)/Lib.fac(r1-j);

else if (j<=0)
{
/*
* declare a integral function and initialise the parameter set
*/
```

```
PX1R1 f  = new PX1R1();
f.lam1   = lam1;
f.r1     = r1;
f.omega  = omega;
f.L1     = L1;
f.j      = j;

double coefficient   = Math.pow(lam1,r1)*Math.pow(lam1New,-j)
                       *Math.exp(-lam1New*L1)/(Lib.fac(r1-1)*Lib.fac(-j));
result               = coefficient*Integration.trapezium(f,
                       lowerLimit,upperLimit,accuracy,maxIntervals);
}
return result;
}

/**********************************************
**********************************************
* Computation of E[X1(R1)], E[X1(R1)~2]
**********************************************
**********************************************
*/

public double EX1R1(double omega)
{
int j = r1;
double summ      = j*PX1R1(omega,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*PX1R1(omega,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX1R1(omega,j))> TOL2);

return summ;
}

public double EX1R1S(double omega)
{
int j = r1;
double summ      = j*j*PX1R1(omega,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*j*PX1R1(omega,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX1R1(omega,j))> TOL2);
return summ;
}

public double theta(double omega)
{
// caculate the theta

double nominator     = L1*Lib.Fbar(r1-1,lam1*L1)-r1*Lib.Fbar(r1,lam1*L1)/lam1;
```

```
double mct1          = (q1-r1+EX1R1(omega))/lam1+L1;
return nominator/mct1;
}

/***********************************************************
***********************************************************
*    Cost Evaluation Part
***********************************************************
***********************************************************
*/

public double CostEval1(double omega, double theta)
{
double ex1r1    = EX1R1(omega);
double ex1r1s   = EX1R1S(omega);
double mct1     = (q1-r1+ex1r1)/lam1+L1;

/*
*  Order Cost Evaluation
*/

OC1   = c1/mct1;

/*
* Holding Cost Evaluation
*/

double HC1a  = (q1*q1+q1-r1*r1-r1+(2*q1+1)*ex1r1+ex1r1s)/(2*lam1);
double HC1b  = L1*r1*Lib.F(r1-1,lam1*L1)
               -lam1*L1*L1*Lib.F(r1-2,lam1*L1)/2
               +r1*(r1+1)*Lib.Fbar(r1,lam1*L1)/(2*lam1);

HC1          = h1*(HC1a+HC1b)/mct1;

/*
* Stock Cost Evaluation
*/

StockOut1 stockout1  = new StockOut1();
stockout1.omega      = omega;
SOC1                 = so1*SigmaSum.sumInfP1(stockout1, 1, this)/mct1;

/*
* backorder cost for depot 1
*/

BackOrder1 backorder1    = new BackOrder1();
backorder1.lam1          = lam1;
backorder1.L1            = L1;
backorder1.r1            = r1;

double coefficient   = Math.pow(lam1,r1+1)*(1-omega)/(2*Lib.fac(r1-1));
BC1                  = b1*coefficient*Integration.trapezium(backorder1,
                       lowerLimit,upperLimit,accuracy,maxIntervals)/mct1;
TRC21                = tr21*lam1*omega*theta;
```

```
return OC1+HC1+SOC1+BC1;
}
}


public class LocationHot2
{
double lam1, lam2, L2, c2, h2, so2, b2;
int r2, q2;
static final double TOL1 = 1E-16, TOL2 = 1E-16, TOL3 = 1E-16;

static final double accuracy   = 1E-6;
static final int maxIntervals  = 10000;
double lowerLimit, upperLimit;
double OC2, HC2, SOC2, BC2;

public LocationHot2(double lam1_, double lam2_, int r2_, int q2_, double L2_,
double c2_, double h2_, double so2_, double b2_)
{
lam1        = lam1_;
lam2        = lam2_;
r2          = r2_;
q2          = q2_;
L2          = L2_;
c2          = c2_;
h2          = h2_;
so2         = so2_;
b2          = b2_;
lowerLimit  = 0;
upperLimit  = L2;
}

/*********************************************************************************
*********************************************************************************
* Standard basic library function to compute lead time demand distribution
* and probability distribution of X2(R2)
*********************************************************************************
*********************************************************************************
*/

public double PX2R2(double theta, int I2, int j)
{
double result = 0;
double lam2New = lam2+lam1*theta;

if (r2>I2)
{
if (j>I2)
result = Math.pow(lam2New*L2,r2-j)*Math.exp(-lam2New*L2)/Lib.fac(r2-j);

else
{
/*
* declare a integral function and initialise the parameter set
*/
```

```
PX2R2 fx2r2 = new PX2R2();
fx2r2.lam1  = lam1;
fx2r2.theta = theta;
fx2r2.r2    = r2;
fx2r2.I2    = I2;
fx2r2.L2    = L2;
fx2r2.j     = j;

double coeffPX2R2    = Math.pow(lam2New,r2-I2)*Math.pow(lam2,I2-j)
                       *Math.exp(-lam2*L2)/(Lib.fac(r2-I2-1)*Lib.fac(I2-j));
result               = coeffPX2R2*Integration.trapezium(fx2r2,
                       lowerLimit,upperLimit,accuracy,maxIntervals);
}
}

else if (r2==I2)
result = Math.pow(L2*lam2,r2-j)*Math.exp(-lam2*L2)/Lib.fac(r2-j);

else if (r2<I2)
result = Math.pow(L2*lam2,r2-j)*Math.exp(-lam2*L2)/Lib.fac(r2-j);

return result;
}

/**********************************************
**********************************************
* Computation of E[X2(R2)], E[X2(R2)~2]
**********************************************
**********************************************
*/

public double EX2R2(double theta, int I2)
{
int j = r2;
double summ      = j*PX2R2(theta,I2,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*PX2R2(theta,I2,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX2R2(theta,I2,j)) >TOL2);
return summ;
}

public double EX2R2S(double theta, int I2)
{
int j = r2;
double summ      = j*j*PX2R2(theta,I2,j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+j*j*PX2R2(theta,I2,j);
} while ( Math.abs(summ-oldSumm)> TOL1 || Math.abs(PX2R2(theta,I2,j)) >TOL2);
```

```
return summ;
}

public double EMINI2(double theta, int I2)
{
int j = r2;
double summ      = PX2R2(theta,I2,j)*Math.min(I2,q2+j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+PX2R2(theta,I2,j)*Math.min(I2,q2+j);
} while ( Math.abs(summ-oldSumm)> TOL3 ||
          Math.abs(PX2R2(theta,I2,j)*Math.min(I2,q2+j))> TOL3);

return summ;
}

public double EMINI2S(double theta, int I2)
{
int j = r2;
double summ      = PX2R2(theta,I2,j)*Math.min(I2,q2+j)*Math.min(I2,q2+j);
double oldSumm;
do
{
oldSumm = summ;
j--;
summ      = summ+PX2R2(theta,I2,j)*Math.min(I2,q2+j)*Math.min(I2,q2+j);
} while ( Math.abs(summ-oldSumm)> TOL3 ||
          Math.abs(PX2R2(theta,I2,j)*Math.min(I2,q2+j)*Math.min(I2,q2+j))>
TOL3);

return summ;
}

public double omega(double theta, int I2)
{
double mt = 0, mct2= 0, result = 0;
double lam2New = lam2+lam1*theta;
double ex2r2   = EX2R2(theta,I2);
double eminI2  = EMINI2(theta,I2);

if (r2>I2)
{
mt           = L2*Lib.Fbar(r2-I2-1,lam2New*L2)-(r2-I2)*Lib.Fbar(r2-
I2,lam2New*L2)/lam2New;
mct2         = (q2-r2+ex2r2)/lam2New+L2;
result       = 1-mt/mct2;
}

else if (r2 == I2)
{
mct2   = (q2-r2+ex2r2)/lam2New+L2;
result = 1-L2/mct2;
}
```

```
else if (r2<I2)
{
mt       = (q2+ex2r2-eminI2)/lam2New;
mct2     = (q2+ex2r2-eminI2)/lam2New+(eminI2-r2)/lam2+L2;
result   = mt/mct2;
}
return result;
}

public double phi(double theta, int I2)
{
// caculate the phi

double mt=0, mct2=0, result=0;
double lam2New = lam2+lam1*theta;
double ex2r2   = EX2R2(theta,I2);
double eminI2 = EMINI2(theta,I2);

if (r2>I2)
{
IntegralPhi intPHI  = new IntegralPhi();
intPHI.I2           = I2;
intPHI.L2           = L2;
intPHI.r2           = r2;
intPHI.lam2         = lam2;
intPHI.lam2New      = lam2New;

double coefficient1 = Math.pow(lam2New,r2-I2)/Lib.fac(r2-I2-1);
mt                  = coefficient1*Integration.trapezium(intPHI,
                      lowerLimit,upperLimit,accuracy,maxIntervals);
mct2                = (q2-r2+ex2r2)/lam2New+L2;
result              = 1-mt/mct2;
}

else if (r2==I2)
{
mt                  = L2*Lib.Fbar(r2-1,lam2*L2)-r2*Lib.Fbar(r2,lam2*L2)/lam2;
mct2                = (q2-r2+ex2r2)/lam2New+L2;
result              = 1-mt/mct2;
}

else if (r2<I2)
{
mt                  = L2*Lib.Fbar(r2-1,lam2*L2)
                      -r2*Lib.Fbar(r2,lam2*L2)/lam2;
mct2                = (q2+ex2r2-eminI2)/lam2New+(eminI2-r2)/lam2+L2;
result              = 1- mt/mct2;
}
return result;
}

/**************************************************************
**************************************************************
*  Cost Evaluation Part
**************************************************************
```

```
*******************************************************
*/

public double CostEval2(double theta, int I2)
{
double mct2      = 0;
double lam2New   = lam2+lam1*theta;
double ex2r2     = EX2R2(theta,I2);
double ex2r2s    = EX2R2S(theta,I2);
double eminI2    = EMINI2(theta,I2);
double eminI2s   = EMINI2S(theta,I2);

if (r2>I2)
{
//Mean cycle time at depot 2

mct2 = (q2-r2+ex2r2)/lam2New+L2;

/*
*  Order Cost Evaluation
*/

OC2   = c2/mct2;

/*
* Holding Cost Evaluation
*/

IntegralHLD intGHLD = new IntegralHLD();
intGHLD.I2          = I2;
intGHLD.L2          = L2;
intGHLD.r2          = r2;
intGHLD.lam2        = lam2;
intGHLD.lam2New     = lam2New;

double HC2a         = (q2*q2+q2-r2*r2-r2+(2*q2+1)*ex2r2+ex2r2s)/(2*lam2New);
double HC2b         = L2*r2*Lib.F(r2-I2-1,lam2New*L2)
                      -lam2New*L2*L2*Lib.F(r2-I2-2,lam2New*L2)/2
                      +(r2+I2+1)*(r2-I2)*Lib.Fbar(r2-
I2,lam2New*L2)/(2*lam2New);
double coefficient1 = Math.pow(lam2New,r2-I2)/Lib.fac(r2-I2-1);
double HC2c         = coefficient1*Integration.trapezium(intGHLD,
                      lowerLimit,upperLimit,accuracy,maxIntervals);

HC2    = h2*(HC2a+HC2b+HC2c)/mct2;

/*
* Stock Cost Evaluation
*/

StockOut2 stockout2  = new StockOut2();
stockout2.theta      = theta;
stockout2.I2         = I2;
SOC2                 = so2*SigmaSum.sumInfP2(stockout2, 1, this)/mct2;

/*
```

```
* Backorder Cost Evaluation
*/

BackOrder2 backorder2   = new BackOrder2();
backorder2.lam2New      = lam2New;
backorder2.lam2         = lam2;
backorder2.I2           = I2;
backorder2.L2           = L2;
backorder2.r2           = r2;

double coefficient2     = Math.pow(lam2New,r2-I2)/Lib.fac(r2-I2-1);
BC2                     = b2*coefficient2*Integration.trapezium(backorder2,
                          lowerLimit,upperLimit,accuracy,maxIntervals)/mct2;
}

else if (r2==I2)
{
mct2 = (q2-r2+ex2r2)/lam2New+L2;

/*
*  Order Cost Evaluation
*/

OC2   = c2/mct2;

/*
* Holding Cost Evaluation
*/

double HC2a  = (q2*q2+q2-r2*r2-r2+(2*q2+1)*ex2r2+ex2r2s)/(2*lam2New);
double HC2b  = L2*r2*Lib.F(r2-1,lam2*L2)
               -lam2*L2*L2*Lib.F(r2-2,lam2*L2)/2
               +r2*(r2+1)*Lib.Fbar(r2,lam2*L2)/(2*lam2);

HC2    = h2*(HC2a+HC2b)/mct2;

/*
* Stock Cost Evaluation
*/

StockOut2 stockout2  = new StockOut2();
stockout2.theta      = theta;
stockout2.I2         = I2;
SOC2                 = so2*SigmaSum.sumInfP2(stockout2, 1, this)/mct2;

/*
* Backorder Cost Evaluation
*/

BC2     = b2*(lam2*L2*L2*Lib.Fbar(r2-1,lam2*L2)/2-L2*r2*Lib.Fbar(r2,lam2*L2)+
r2*(r2+1)*Lib.Fbar(r2+1,lam2*L2)/(2*lam2))/mct2;
}

else if (r2<I2)
{
mct2 = (q2+ex2r2-eminI2)/lam2New+(eminI2-r2)/lam2+L2;
```

```
/*
*  Order Cost Evaluation
*/

OC2   = c2/mct2;

/*
* Holding Cost Evaluation
*/

double HC2a  = (q2*q2+q2+(2*q2+1)*ex2r2+ex2r2s)/(2*lam2New)
               +(lam2New-lam2)*(eminI2s+eminI2)/(2*lam2*lam2New)
               -(r2*r2+r2)/(2*lam2);
double HC2b  = L2*r2*Lib.F(r2-1,lam2*L2)
               -lam2*L2*L2*Lib.F(r2-2,lam2*L2)/2
               +r2*(r2+1)*Lib.Fbar(r2,lam2*L2)/(2*lam2);

HC2          = h2*(HC2a+HC2b)/mct2;

/*
* Stock Cost Evaluation
*/

StockOut2 stockout2  = new StockOut2();
stockout2.theta      = theta;
stockout2.I2         = I2;
SOC2                 = so2*SigmaSum.sumInfP2(stockout2, 1, this)/mct2;

/*
* Backorder Cost Evaluation
*/

BC2     = b2*(lam2*L2*L2*Lib.Fbar(r2-1,lam2*L2)/2
          -L2*r2*Lib.Fbar(r2,lam2*L2)
          +r2*(r2+1)*Lib.Fbar(r2+1,lam2*L2)/(2*lam2))/mct2;
}
return OC2+HC2+SOC2+BC2;
}
}
```

B.3 Approximation implementation for the SMDP model with phase-type lead time

```
/*
* Main function of MDP model with holdout phased lead time
*/

import java.io.*;
import java.util.*;
```

```
public class mdplt
{
int N1, N2, M1, M2;
int Q1, Q2, nPh, HO2, nIter;
double lam1, lam2, c1, c2, b1, b2, h1, h2, mu1, mu2, bHat1, bHat2, B1, B2, T,
g, tau;
PrintWriter out;
boolean printOptAction, printOptCost;
State st;
Action optAction;
double pDem1, pDel1, pDem2, pDel2;
double mn,mx;

double lpr;
double Tol;

public mlpFLib(
double mu1_, double lam1_, double c1_, double h1_, double b1_, double B1_,
double bHat1_, int n1_, int m1_, int Q1_,
double mu2_, double lam2_, double c2_, double h2_, double b2_, double B2_,
double bHat2_, int n2_, int m2_, int Q2_,
int nPh_, double T_, State st_, int HO2_, PrintWriter OUT, boolean
printOptAction, boolean printOptCost)

{
lam1         = lam1_;
lam2         = lam2_;
mu1          = mu1_;
mu2          = mu2_;
c1           = c1_;
c2           = c2_;
h1           = h1_;
h2           = h2_;
b1           = b1_;
b2           = b2_;
bHat1        = bHat1_;
bHat2        = bHat2_;
B1           = B1_;
B2           = B2_;
N1           = n1_;
N2           = n2_;
M1           = m1_;
M2           = m2_;
Q1           = Q1_;
Q2           = Q2_;
nPh          = nPh_;
T            = T_;
HO2          = HO2_;
st           = st_;
out          = OUT;
lpr          = 1E10;
Tol          = 1E-3;
}

public void transformData()
{
```

```
//Scale rate of lead time events to allow number of phases

mu1 = mu1*nPh;
mu2 = mu2*nPh;

// tau is mean time between events in regular MDP

tau = 1/(lam1+mu1+lam2+mu2);

/*
*scale holding cost rate and back order holding cost rate to allow for
*mean time between events
*/

h1       = h1*tau;
bHat1    = bHat1*tau;
h2       = h2*tau;
bHat2    = bHat2*tau;

//pDem is probability next event is a demand at the location

pDem1 = lam1*tau;
pDem2 = lam2*tau;

//pDem is probability next event is a delivery at the location

pDel1 = mu1*tau;
pDel2 = mu2*tau;

}

public static void initV(double[] v)
{
for (int i=0; i<v.length; i++)
v[i] = 0;
}

public double iteration(double[] vOld, double[] vNew)
{
/* Initialise st to allow enumeration of the state space and i
* to reference the value functions
*/

st.resetState();

int i = -1;

//Initialise calcultion of maximum and minimum change in value function

double mnDiff = lpr;
double mxDiff = -lpr;

//Consider each state in turn

while (st.nextState())
{
```

```
i = i+1;

//Initialise calculation of value of this state at this iteration

vNew[i]=lpr;

/*Holding cost depends only on i1 and i2 so can be calculated and stored
*until arrival of the next state without consideration of actions
*/

double hCst = 0;

if (st.i1>0)
hCst = hCst+st.i1*h1;
else
hCst = hCst-st.i1*bHat1;

if (st.i2>0)
hCst = hCst+st.i2*h2;
else
hCst = hCst-st.i2*bHat2;

//Initialise a action to allow numberation of the action space for this state

Action a = new Action(st, HO2, Q1, Q2, M1, M2, N1);
a.resetAction();

//Consider each action in turn

while (a.nextAction())
{
//acCst, the cost of this action, can be initialised to hCst

double acCst = hCst;

/*
* for all available actions, all the possibilities and conditions
* have been checked
* nNull is the pointer to the next state when the inventory levels
* do not change and no order events occur
*/

int nNull = i;

//Update acCst and nNull if this action places an order at location 1

if(a.ph1>st.ph1)
{
nNull = nNull+st.ph2Hsh;
acCst = acCst+c1;
}

//Update acCst and nNull if this action places an order at location 2

if (a.ph2>st.ph2)
```

```
{
nNull = nNull+1;
acCst = acCst+c2;
}

/*nFi is the pointer of the next state when an item is
*removed from location i
*/

int nF2 = nNull - st.ph1Hsh;
int nF1 = nNull - st.i2Hsh;

//Next event is demand at location 1 which results in transshipment

if (a.d==0)
acCst = acCst+pDem1*(T+vOld[nF2]);

//More backorders than possible at location 1

else if (st.i1==st.i1Mn)
acCst = acCst+pDem1*(B1+vOld[nNull]);

//Stockout at location 1

else if (st.i1<=0)
acCst = acCst+pDem1*(b1+vOld[nF1]);

//Supply from location 1

else
acCst = acCst+pDem1*vOld[nF1];


/* Next event is demand at location 2 which results in more
* backorders than possible at location 2
*/

if (st.i2==st.i2Mn)
acCst = acCst+pDem2*(B2+vOld[nNull]);

//Stockout from location 2

else if (st.i2<=0)
acCst = acCst+pDem2*(b2+vOld[nF2]);

//Supply from location 2

else
acCst = acCst+pDem2*vOld[nF2];

/*Next event is an order event at location 1 which results in:
*Delivery of Q1 items
*/

if (a.ph1==nPh)
{
```

```
int nO1 = nNull-nPh*st.ph2Hsh+Math.min(Q1,M1-st.i1)*st.i2Hsh;
acCst   = acCst+pDel1*vOld[nO1];
}

//End of next phase of lead time

else if (a.ph1>0)
acCst = acCst+pDel1*vOld[nNull+st.ph2Hsh];
else //Fictitious decision epoch
acCst = acCst+pDel1*vOld[nNull];

/*Next event is an order event at location 2 wihich results in:
*Delivery of Q2 items
*/

if (a.ph2==nPh)
{
int nO2 = nNull-nPh+Math.min(Q2,M2-st.i2)*st.ph1Hsh;
acCst   = acCst+pDel2*vOld[nO2];
}

//End of next phase of lead time
else if (a.ph2>0)
acCst   = acCst+pDel2*vOld[nNull+1];
else //Ficitious decision epoch
acCst   = acCst+pDel2*vOld[nNull];

//Compare acCst with minimum cost for this state calculated to date
if (acCst<vNew[i])
{
vNew[i]     = acCst;
optAction   = a;
}
}

/*
* Print out the optimal actions when depot 1 runs out the stock
* and depot 2 has any stock meanwhile neither of depot 1 nor depot 2
* has the outstanding orders
*/

if (st.i1<=0 && st.i2>0 && st.ph1 == 0 && st.ph2 == 0 && printOptAction ==
true)
{
out.print("State (,"+st.i1+","+st.i2+","+st.ph1+","+st.ph2+",),");
out.println("OPT Action (,"+optAction.ph1  +","+optAction.ph2
+","+optAction.d  +",)");
}


/*
*Compare the difference in value for this state with maximum and minimum
difference
*/

double diff = vNew[i]-vOld[i];
```

```
if (diff>mxDiff)
mxDiff = diff;
if (diff<mnDiff)
mnDiff = diff;
}

//Update g and return latest error

g = (mxDiff+mnDiff)/(2*tau);
return (mxDiff-mnDiff)/mnDiff;
}

public void valueIteration()
{
double[] v1 = new double[st.i1Hsh];
double[] v2 = new double[st.i1Hsh];
initV(v1);

//initialise the iteration count and the error value

nlter       = 0;
double erVal    = lpr;

//Perform value iteration until the error no longer above the tolerance

while (erVal>Tol)
{
//At odd iterations calculate v2 from values in v1
nlter++;
erVal = iteration(v1,v2);

/*
if (nlter>=1734)
{
printOptAction = true;
printOptCost   = false;
}
*/

//At even iterations calculate v1 from values in v2
nlter++;
erVal = iteration(v2,v1);
}

return;
}
}
```