VU University Amsterdam
Faculty of Sciences
Department of Mathematics

RESEARCH PAPER

Melania Călinescu

Decentralized Networks: the Impact on Return Policies for an Assemble-to-Order System

Supervisors: Dr. Sandjai Bhulai
Dr. Iris F.A. Vis

Study Programme: Business Mathematics and Informatics

Amsterdam
2008

ABSTRACT

The last years have displayed an impressive e-commerce boom. To protect the customer stricter policies have been imposed to e-tailers, most important of which is the return policy. Companies have to deal now with more complicated management of inventories due to uncertainty in the reverse flow of materials. Moreover, returns do not re-enter the supply chain immediately after leaving the customer. Although there is a broad literature on returns management and on how companies should handle returns, mathematical models have not yet addressed efficiently the customer's needs. Assemble-to-order systems build products to customer's requirements. However, returns in such systems cause losses due to the uniqueness of the products. The present paper implements a customer-friendly return policy, at a component level, for an assemble-to-order system. Finished products are assembled in response to customer demands; the inventory is kept at a component level; once the customer reports the intention of returning a product because of a particular component, the company will take care of the dissembling and shipment processes and costs and pays the customer a return credit; no time limits are imposed for returns because any returned product crosses refurbishing processes and quality checks. We describe a simple network (manufacturer, two suppliers, third-party logistics provider, customers) and derive both the optimized profit and the optimal base-stock inventory level for different return rates. Through a simulation study we show that a return policy as such brings improvements to the company's profits, for certain levels of return rates.

Contents

Introduction

In view of today's impressive expansion of the e-commerce, serious legal boundaries have been imposed to companies that sell products via Internet. One of the main requirements each e-tailer[1] has to fulfill is the return policy. By return policy we understand a contract between the manufacturer and forward positions in the supply chain (retailers, suppliers, customers), concerning the procedure of accepting back products after having sold them, either used or in a as-good-as-new state. The way management handles returns plays an important role in the company's strategy to success, especially in the area of e-commerce.

Several reasons can be laid out to justify this last statement. First of all, every customer has the right to return the product they have purchased if their expectations are not met (this has even a higher impact on sales within e-commerce since no physical contact is involved). Second, environmental concerns have led to organized collections of used products, either due to legally required take-backs (automobiles, electronic goods, packaging), or voluntary campaigns, more or less regular, that help companies recover residual value. Regardless of why they occur, product returns introduce a level of uncertainty in the reverse flow of materials, which seriously complicates the management of an inventory system (DeCroix et al. (2008)). The need for strategies that will optimize the returns management for an e-business motivates the present research paper.

Customer returns of as-good-as-new products have increased dramatically lately. Growth in mail-order[2] and transactions over the Internet has increased the volume of product returns - customers are unable to see and touch the items they decide to buy, so they are more likely to return them. Several studies (Arar (2008), Saskatchewan (2008)) draw attention to possible causes for high number of returns: in 2007, Americans returned between 11 and 20% of electronic goods, which adds up to the staggering amount of $13.8 billions, out of which just 5% were actually broken. The rest failed to meet the customers' expectations. Most often the customers discovered that the product they had bought did not have the functionality they expected. To conclude, retailers and vendors could save a fortune if they spent more time producing instructions that were easy to understand and marketing material that explained the set up and functionality of the products they wanted to sell, in other words implementing customer-friendly returns policies offer a company a real chance to secure a life-long customer and a word-of-mouth free advertising. These statements have motivated us to research whether profits can indeed increase when more attention is paid to the customer's needs.

The returns may consist of finished goods that can be used immediately to satisfy new customer demand or used goods from which components or subassemblies can be recovered for use. According to DeCroix et al. (2008), questions that managers need to consider regard which information level is more insightful - product or component level; the impact on inventory holding costs results in benefits for the company; which items should the company

[1] E-tailer denotes any company that sells its products via Internet.

[2] Mail-order represents the act of ordering merchandise by mail.

design to be recoverable etc.

Dealing with this type of questions, designing a quick and easy return policy for the particular case of an assemble-to-order system and focusing on the customer's comfort and satisfaction are not issues too widely treated in the literature. As a matter of fact, there has not been much research done in trying to combine these aspects. Our paper contributes to this field by building a model that minimizes the costs incurred by the inventory for an assemble-to-order system with returns. The novelty offered through this model regards practical constraints: returns shipment handled by the company, lower delays within the network, optimized inventory base levels, cheaper production and an overall increase in the net profit of the company.

The remainder of the paper is as follows. Chapter 1 introduces the main concepts that define the context of our problem and presents a short literature overview on the topic. In Chapter 2, we explain the model and compute optimal base-stock policies. Chapter 3 explores the feasibility of the model through data simulations and offers comments on the results. Conclusions are briefly noted at the end of the paper.

1 Background

In recent years, companies have faced major changes in their management, due to increased competition in e-business. Online fast comparisons between similar products, high quality customer service, increasingly diversifying customer needs, higher volume of returns for as-good-as-new goods are only few of the main reasons why e-tailers find themselves in a tight competition against each other. To continuously adapt their supply chain management, e-tailers have turned toward assemble-to-order systems, rather than the traditional make-to-order systems.

1.1 Assemble-to-Order Systems

In an assemble-to-order system, products are designed around interchangeable modules and the company makes and stocks only the modules and other major components. When a customer order arrives requesting a specific kit of modules and components, the company will just assemble those components and deliver the end product to the customer (SONG ET AL. (1999)). Consequently, according to MUKHOPADHYAY AND SETOPUTRO (2005), the benefits for the company include speeding up the product development process, increasing the range of the product variations, rapid technological upgrading of products, reducing the number of suppliers and reducing the costs of development and production.

Yet another benefit of the modularity in the assemble-to-order systems, particulary desirable in our case, is the ease with which the product can be dismantled (just as easy as it was assembled), producing a number of components that are standard products keeping their full value. A returned product would then give back a large salvage value to the firm, therefore cutting down its loss due to the return.

Still, component inventory management brings up some operational difficulties. According to SONG ET AL. (1999), the problems that inventory managers deal with in such an environment are substantially different from the issues that may arise in a make-to-order system. First of all, the assumed independence of demand across items is not valid anymore and it is replaced by joint management of inventories and production capacities across various items. Also, the determination of the stock level of one item should take into account the stock level of other items, since each customer order requires the simultaneous availability of several items.

However, any model that aims to find the joint optimal inventory levels would involve evaluation of multidimensional probabilities and optimization of nonseparable functions, which is computationally demanding. This partly explains why in practice most systems simply apply single-item inventory planning tools to each component, ignoring the connections between components. This item-based approach brings up the natural question of to what extent is this good enough to manage the inventories in a reliable manner? How can managers design an effective policy, based on the solutions of these models?

LU AND SONG (2005) try to answer these questions, formulating a model that determine the joint optimal base-stock levels, minimizing the inventory costs. Based on their work, DeCroix et al. (2008) introduce the returns for an assemble-to-order system. As it will be presented in Section 1.4, we have extended these models by applying a decentralized network to an assemble-to-order system that eliminates backorders and leads to less demanding computations to get the optimal base-stock levels.

1.2 Handling Returns

Returns management is one of the supply chain management processes that gains increasing attention globally for its promising financial potentials and the environmental positive impact it could have. Generally, it is associated with returns, reverse logistics, gatekeeping[1] and avoidance[2] that are managed within the firm and across key members of the supply chain (see Rogers et al. (2002)).

There are many types of returns that need to be managed within this process, each of which poses unique challenges: consumer returns (due to buyers' remorse or defects), marketing returns (due to slow sales, quality issues or the need to reposition inventory at a position forward in the supply chain), asset returns (reusable assets), product recalls (due to safety or quality issues) and environmental returns (disposal of hazardous materials or law reinforced processes). In this paper, we focus our attention on the consumer returns, especially those caused by customers' remorse, since it is one of the most recent and popular aspects raised by management. Efficiently handling returned products leads to increased profit and improved customer service, which is highly important in e-business, as previously said. Other types of returns are fairly little present in e-business, which is why our model does not include them. For models that apply to any of the other types of returns, the reader can refer to the literature reviews presented by Rogers et al. (2002) or DeCroix et al. (2008).

Customers may return items for a variety of reasons: sudden change of mind after purchase, legally required take-backs (in Europe for electronic goods, automobiles, packaging) due to environmental concerns, companies voluntarily collect of used products (single-use Kodak cameras, IBM personal computers, Canon toner cartridges). More interesting to study are the returns that occur shortly after the time of purchase, since they introduce an uncertain flow of goods in the supply chain. They are common for goods that may not be easy to assess fully at the time of purchase, such as clothing or gifts. In mail-order and e-business channels, returns are especially problematic; if a customer is unable to touch an item, he is even less able to assess it at the time of purchase and is thus more likely to return it. For a review of models regarding other categories of customer returns, the reader can refer to Frankel (1996).

However, returns must be kept within certain limits, because no seller desires for returns. To systematically handle returns, the companies or third party logistics operators have developed applications with which the use of web-enabled databases contribute to higher visibility of data and interoperability of IT systems. Kokkinaki et al. (2002) pointed out that to have a proactive minimization of the returns, companies have to increase the efficiency of

[1] Gatekeeping allows management to control and reduce returns without damaging customer service because it eliminates the cost associated with returning products with no defects (due to inadequate customer first use, e.g., lack of experience with installing software or a hardware component) or the cost of products returned to an inappropriate destination.

[2] Avoidance can include, for instance, ensuring that the quality of the product and its user friendliness are at the highest attainable level before the product is sold and shipped.

forward logistics (online tracking and tracing of the orders, web-based tools to cross-examine each order for incompatibilities). To minimize the returns' uncertainty, web-interfaces are used to either collect data about the product so that preliminary support can already be planned or to offer financial incentives to the customer to follow the optimal alternative for their returns.

Moreover, by extending their customer services, through avoidance and gatekeeping, companies can minimize their returns. For example, Dell Computers Corp. has a liberal return policy (less restrictions are imposed on the customer before returning a product), but a product can be returned only after the customer is walked through their avoidance and gatekeeping stages, to make sure that the product is indeed defective or it is not satisfactory enough for the customer (Banker (2001), Kokkinaki et al. (2002)).

Another issue that business-to-customer e-business has to face when it comes to returns is customer satisfaction. Since the customers can change their mind about purchasing online a certain product from a certain seller much faster than offline (due to possibilities of comparison available with just one mouse click), customer satisfaction and loyalty are the goals to achieve for both short and long run. Besides the restrictions in the return policy, which may keep the customer close to the company's catalogue, is the ease with which products can be returned. Recently, companies have developed Internet technologies that assign the customer a specific destination for his/her return, according to the merchant's bar code. This way, customers are more likely to buy from and be loyal to an online seller.

We understand how important these aspects can be for an e-business, so our model for an assemble-to-order system will take them into account, as it will be explained in Section 1.4.

1.3 Decentralized Networks

Summarizing briefly the matters discussed above, we conclude that companies need to learn how to wisely handle their customer returns in order to recover as much value as possible, i.e., both product value and increased customer satisfaction. The large fraction of product value is lost due to long processing delays. The longer it takes to retrieve a returned product, the lower the likelihood of economically viable reuse options. Souza et al. (2006) outline opportunities to create competitive advantage by improving the design of the reverse supply chain.

Many reverse supply chain networks are designed to minimize logistics costs through central product returns depots. Nevertheless, cost-efficient supply chain networks are not always appropriate. Products with high return rates, considerable recovery value and high value decay over time[3], especially if a high percentage of products require remanufacturing or a redesigned flow inside the supply chain to make better use of their features. Faster response in business processes can be a source of competitive advantage and few of the main triggers of higher responsiveness in the supply chain are the return rate, the time value decay parameters, the percentage of new returns, and the returned product recoverable value.

In their paper, Souza et al. (2006) show the main characteristics for a supply chain with different levels of responsiveness. They also develop an alternative model that reduces the costs of time decay value at each stage of the return process: a decentralized network. Opposite to the centralized one (where all commercial returns are shipped to a central facility for economies of scale), this innovative design allows new returns to be sorted and immediately

[3] Significant value deterioration / decrease due to time lapse.

re-stocked at the retailer / manufacturer. Thus, transportation costs to/from the central evaluation facility and consequently the delays involved are reduced. Although extra costs might be experienced (additional work at the retailer/manufacturer to handle and repackage the returns), the financial benefits in comparison with a centralized network are significant.

Due to these advantages, our model explores the impact of a decentralized network applied to the supply chain of an assemble-to-order system.

1.4 The Novelty of the Model

Assemble-to-order systems, short lead times, fast responsive networks, high customer service levels are few of the most popular topics among both researchers and practitioners. Perfecting these settings acts as an incentive for developing more complex structures, through combining two or more of the previously mentioned concepts. Bernstein et al. (2005) consider Dell Computers Corp. as the best example of these more complex systems. Dell operates in an assemble-to-order system, with a highly proficient returns management system at the same time. In our model we want to explore whether applying a decentralized network to a general assemble-to-order system with returns will significantly improve the profit.

We start with a simple network: one manufacturer with two suppliers and one third-party logistics provider (3PL). We have decided that the presence of a 3PL is an important detail in our model since it is obviously closer to a real supply chain design and additionally, it will ease the computations and clear up the model. We analyze an assemble-to-order system, in which finished products are assembled in response to stochastic customer demand. The system is managed over an infinite horizon using a component-level base-stock policy and it experiences stochastic returns of subsets of components. These returns can be used to satisfy subsequent demand, therefore the business process has a more complicated behavior. We demonstrate that these complexities can be more easily handled through the use of a new model.

We construct a decentralized network for our assemble-to-order system and a profit function per ordered product, in order to outline the lower number of factors that produce costs, in comparison with a regular centralized network. The new design significantly reduces the transportation and distribution costs and also increases the customer satisfaction, due to the use of a web-interface for returns specifications and the home pick-up strategy for the claimed items. Moreover, we present a method for the derivation of the optimal number of items each supplier should keep in inventory. In order to compute the optimal base-stock level, we construct an inventory cost function that adds all terms depending on the stock level from the main profit function. The optimal solution is computed by setting the first derivative of this cost function to 0. To support our conclusions, we test the performance of our model within a computer simulation study.

2 Model Description

The model presented in this chapter builds on those of DeCroix et al. (2008) and Souza et al. (2006) and where possible, their notation is used. Consider a multi-product, multi-component Assemble-to-Order system in a decentralized network. Demands are filled on a first-come-first-served (FCFS) basis. Both demand and returns processes are Poisson processes, independent of each other.

2.1 Concepts and Notation

We will start the description of the model by introducing the main elements of the system and the assumptions that govern the flow of products among different positions in the supply chain. Figures 2.1 and 2.2 present the design of the decentralized network (the flow of goods and the costs involved, respectively) and the notation is defined in Table 2.1. Although the problem can be extended to include multiple elements, in order to illustrate a complex real-life network, our model is based on the simplest setting that allows us to explore the efficiency of our solution. Moreover, it offers the reader the possibility to easily understand complex mathematical aspects and to follow the computations without a significant effort.

2.1.1 Elements in the Setting

The elements of the decentralized network are: one main assembler (the factory), two suppliers ($s_{(1)}$ and $s_{(2)}$), a third-party logistics provider (3PL) and customers.

The assembler's task regards the process of assembling the components he receives from the suppliers into the end product, according to the customers' requirements. No additional delays are assigned to the assembling process.

The suppliers manufacture different components. Their production range will be denoted by two disjoint sets. We have chosen this approach to eliminate the dilemma arisen when the returns have to be distributed in between the suppliers. The suppliers' sites include remanufacturing facilities for those returns that require small changes in order to reach again the as-good-as-new state (reflected in the remanufacturing/refurbishing costs).

The third-party logistics provider is responsible for the transportation across the network, the dissembling process at the customer's location and the returns quality evaluation. Figure 2.2 illustrates the routes and Section 2.1.4 describes the processes flow in the network.

The customers give the orders to the factory, specifying online a certain assemble (through the factory's website), receive the end products and can return components. Moreover, for the returns, the customers can send the technical specifications via Internet, even before the component is checked and collected by the logistics provider.

Section 2.1.3 will explain the demand and return rates that define the communication lines between different elements of the network, as depicted in Figure 2.1.

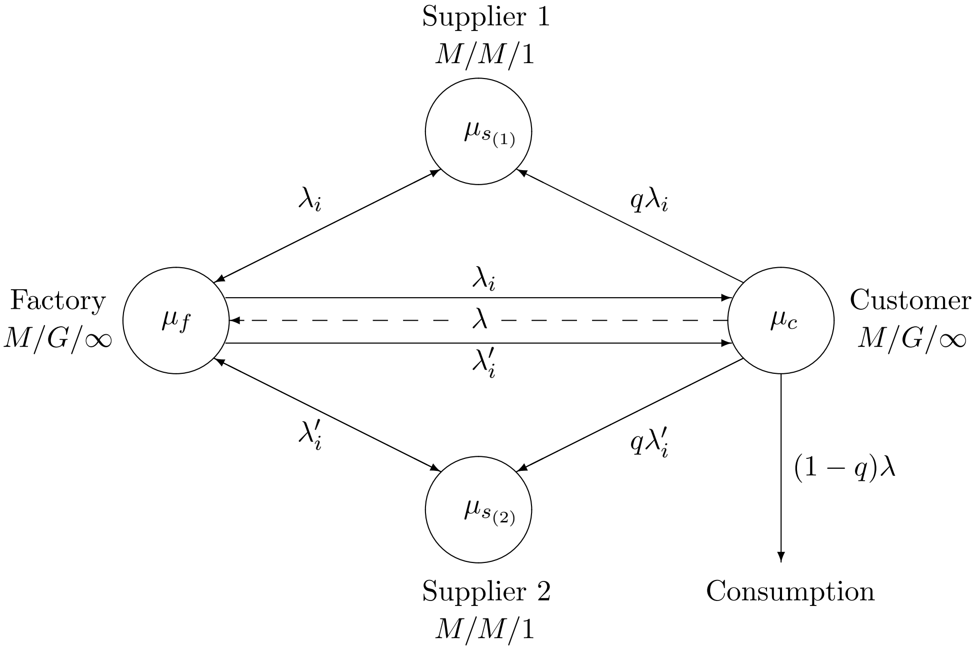

Figure 2.1: The flow of goods for a decentralized network in an ATO system

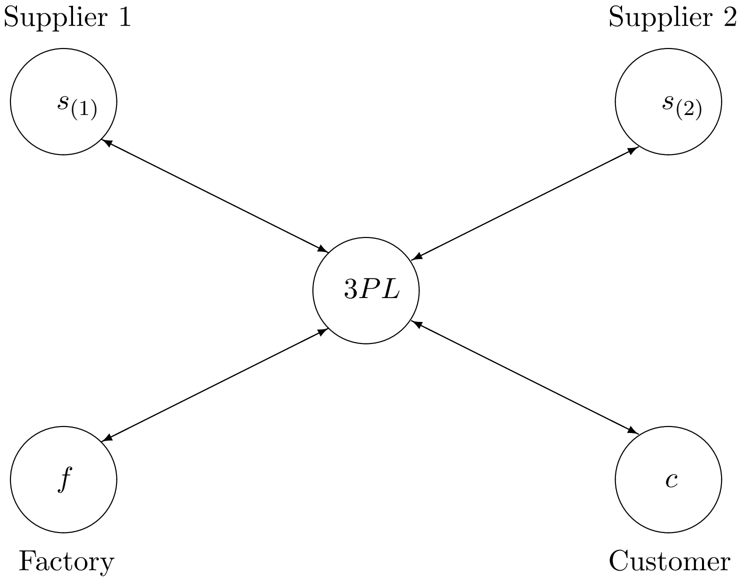

Figure 2.2: The transportation routes for a decentralized network in an ATO system

2.1.2 Assumptions and Notation

(A.1) *The processes on the forward network (customers - factory and factory - suppliers) are modeled as $M/G/\infty$ queues and the returns as $M/M/1$ queues.*

(A.2) *We assume independence among probabilities that a certain component is required within one customer order.*

(A.3) *The product price function $P(A)$ depends on the ordered assemble A of components. Each order requires at most one unit of each item. The variable cost functions for both new and remanufactured/ refurbished items depend only on the type of the component.*

(A.4) *Handling costs and transportation costs are assumed to be constant over time.*

(A.5) *(**Tractability**)*[1] *The actual flow times in the network of Figure 2.1 are approximated by their expected values W_{ab}.*

The assumptions above are meant to construct a framework for simpler computations. To develop a more efficient method in handling returns, we need to focus on how this process affects the performance of the network. One way to observe the existing implications is to force high waiting times at the suppliers, on the backward network. This leads to modeling the suppliers nodes as $M/M/1$ queues. In order to avoid trying to compute a solution for a multidimensional Markov decision process, we model the factory and the customers as $M/G/\infty$ queues.

Although time is an important factor in evaluating cost and price functions, the time index as well as dependence of returns on past sales are suppressed. Thus, assumptions *(A.2)* - *(A.5)* are added to our exposition to serve clarity purposes. Moreover, a return component crosses the network only once, we do not admit multiple returns for the same individual component.

2.1.3 Demand, Return and Production Rates

In order to determine the rates with which demand and returns from customers reach our network, we follow the method from DeCroix et al. (2008).

First, we need to explain the notation we will use. Consider the information available on components produced by each of the suppliers. We will denote by $\boldsymbol{x} = (x_i)_{1 \leq i \leq m}$ the components produced by supplier $s_{(1)}$. Analogously, $\boldsymbol{y} = (y_i)_{1 \leq i \leq n}$ represents the production range for the second supplier. For the sake of simplicity in our mathematical exposition, we construct a vector $\boldsymbol{z}$ as follows:

$$\begin{aligned} z_i &= x_i, \quad \text{for} \quad i = 1, \ldots, m, \\ z_{m+j} &= y_j, \quad \text{for} \quad j = 1, \ldots, n. \end{aligned}$$

Consider $\mathcal{A} = \mathcal{P}(\{1, 2, \ldots, m+n\})$, the set of all possible subsets of $\{1, 2, \ldots, m+n\}$, which represents the set of indices of $\boldsymbol{z}$. Then each ordered assembly can be represented by a set $A \in \mathcal{A}$.

From the data records we can estimate the frequency that component z_i is ordered by taking the weighted average of customer orders for a certain period of time. Denote this estimated frequency by p_i, where $i = 1, \ldots, m+n$. Thus, according to assumption *(A.2)*,

[1] See Souza et al. (2006) for a detailed explanation.

a, b	Subscripts for nodes: f(factory), $s_{(1)}$ (supplier 1), $s_{(2)}$ (supplier 2), c (customer)
$\boldsymbol{x}$, $\boldsymbol{y}$	Components produced by the first and second supplier, respectively: $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)^T$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^T$
$\boldsymbol{s}$	Base-stock inventory level at suppliers, $\boldsymbol{s} = (s_1, s_2, \ldots, s_{m+n})^T$
$\boldsymbol{q_p}$	$\boldsymbol{q_p} = (q_1, q_2, \ldots, q_{m+n})^T$, with q_i the probability that a customer orders component i
$\boldsymbol{p}$	$\boldsymbol{p} = (p_1, p_2, \ldots, p_{m+n})^T$, with p_i the estimated probability that a customer orders component i
$\boldsymbol{C_{ab}}$	$\boldsymbol{C_{ab}} = (c_{ab}^1, c_{ab}^2, \ldots, c_{ab}^{m+n})^T$, with c_{ab}^i the unit delay cost for component i between nodes a and b
τ_{ab}	Average transportation time between nodes a and b
W_{ab}	Expected flow time between the beginning of processing at node a and end of processing at node b
$C_s(A)$	Unit shipment cost per assemble A from the factory to the customer
$P(A)$	Unit price for the end-product
$C_f(A)$	Modules assembling costs at the factory
$\pi(A)$	Expected profit rate for the end-product
Π	Total expected discounted profit over the set of orders
q	Probability that the customer will return a component
$P_r(i)$	Credit paid to the customer for returning component i
$v(i)$	Variable production cost for component i
$v_r(i)$	Variable remanufacturing cost
h_i	Handling costs for component i
ρ_i	Load to the system, per component
λ, λ_i	Average demand at the factory and suppliers
$\eta(s_i)$	Average number of items in inventory, before new order arrives
μ_i	Average processing rate of items (new/returns) at suppliers

Table 2.1: Notation

the estimated probability that a certain assemble A would be ordered, is $p_A = \prod_{i \in A} p_i$, for all $A \in \mathcal{A}$. Then, if the expected profit obtained by assembling the ordered components is denoted by $\pi(A)$, the total profit over the considered interval of time is $\Pi = \sum_{A \in \mathcal{A}} p_A \pi(A)$.

In order to compute $\pi(A)$ we first need some preparatory steps. As previously mentioned, the demand and the return process are both Poisson distributed, with rates λ and $q\lambda$, respectively, with the rates estimated from the available data. To compute the component demand rate we can use p_i, the above estimate, or, for a more precise result, the true component demand probability can be computed as follows. Take $\mathcal{A}_i \subseteq \mathcal{A}$ to be the subset of all ordered assemblies that contain component z_i. Then, the probability q_i that a customer order will contain component z_i is $q_i = \frac{|\mathcal{A}_i|}{|\mathcal{A}|}$, where $|X|$ represents the cardinality of the set $|X|$. The probability that a certain assemble is ordered is $q_A = \Pi_{i \in A} q_i$, for all $A \in \mathcal{A}_i$.

Orders $z_1, \ldots, z_{m+n}$ will be sent to the suppliers according to Poisson processes with rates $\lambda_i = q_i \lambda$, where $i = 1, \ldots, m+n$. The rates $\lambda_1, \ldots, \lambda_m$ correspond to the first supplier $s_{(1)}$ and $\lambda_{m+1}, \ldots, \lambda_{m+n}$ correspond to the second supplier $s_{(2)}$. From these relations it follows that the return rates are $q\lambda_i$, for $i = 1, 2, \ldots, m + n$. Moreover, we assume that each order is independent of the other orders and of all other events. For the sake of simplicity, we will compute the profit using the formula dependent on p_i.

The production rates are influenced by the inventory level. If the inventory level is positive, then the suppliers will have to produce only a smaller number of items than constitutes a received order. At a moment when a supplier receives an order from the factory, the number of the i-th component (i.e., z_i) that already exists in the supplier's inventory is given by

$$\eta(s_i) = \sum_{k=0}^{s_i} (1 - \rho_i)\rho_i^k = 1 - \rho_i^{s_i+1} \quad \text{with} \quad \rho_i = \frac{\lambda_i}{\mu_i + q\lambda_i}, \quad \text{for all} \quad i = 1, 2, \ldots, m + n,$$

where s_i represents the base-stock level for item z_i and ρ_i is the load to the node, which follows from the $M/M/1$ construction of the nodes and the presence of returns that occur at the suppliers' sites. Note that we consider the load *per component.*

2.1.4 Processes Flow Description

This subsection describes in detail the flow of products between the nodes in Figure 2.1 and the transportation routes presented in Figure 2.2.

The product flow consists of the following:

Flow 1: Customer orders arrive at the factory with arrival rate λ. Factory sends demand requests for components to the suppliers, λ_i.

Flow 2: Components arrive at the factory from the suppliers: new components, produced with cost $v(i)$ and returns (if remanufactured, with cost $v_r(i)$). They are assembled into the end-product A, which costs $C_f(A)$.

Flow 3: The customer receives the end-product at price $P(A)$. Figure 2.1 illustrates which components are part of the end-product, by splitting the flow from the factory to the customer into the individual components flows.

Flow 4: With probability q, customers will return components, that are part of the end-product; returns arrive at suppliers with rates $q\lambda_i$.[2] Customers receive credit for their returns of $P_r(i)$.

Figure 2.2 describes the transportation routes that exist within our decentralized network. All the transportation tasks are a responsibility of the third-party logistics provider: the physical communication between the factory and its suppliers regarding items delivery, end-products shipment to the customers from the factory and returns delivery from the customers to the suppliers.

One aspect that still concerns us is the component dissembly process, at the customer's site. Although we think this is a task that the 3PL could handle (and it is reflected in our model in this way), technical issues require a certain level of accuracy in dissembling and proper engineering skills so that other components are not affected during this process. This is still a question that asks for thorough decision making before implementing such a return policy.

The expected delays W_{ab} are computed as follows:

$$W^i_{s_{(j)}f} \;=\; \frac{1}{\mu_i - q\lambda_i} + \tau^i_{s_{(j)}f} + \frac{1}{\mu_f}\,, \qquad W^i_{cs_{(j)}} \;=\; \frac{1}{\mu_c} + \tau^i_{cs_{(j)}} + \frac{1}{\mu_i - q\lambda_i}\,,$$

[2] If there are no return specifications recorded on the factory's website, the 3PL provider has to decide which supplier will handle the return.

for all $i = 1, 2, \ldots, m+n$ and $j = 1, 2$.

2.1.5 Objectives

The expected profit rate for one end-product A, following the above described steps, is:

$$\begin{aligned} \textit{Profit} \;=\; & \textit{Revenue} - [\; \textit{Factory costs (assembling)} + \textit{Distributors costs (production} \\ & \textit{and remanufacturing costs, inventory holding and delay costs,} \\ & \textit{returns credit)} + \textit{Shipment costs} \;], \end{aligned}$$

which in our notation is as follows:

$$\begin{aligned} \pi(A) \;=\; & \lambda P(A) - C_f(A) - \sum_{i \in A} \Bigg[\lambda_i \eta(s_i) \cdot 0 + \lambda_i (1 - \eta(s_i)) v(i) + q \lambda_i \big(v_r(i) + P_r(i) \big) \\ & + h_i \left(s_i + \frac{\rho_i^{s_i}}{(1-\rho_i)\mu_i} \right) + \sum_{j=1}^{2} \Big(c^i_{s_{(j)}f} W^i_{s_{(j)}f} + c^i_{cs_{(j)}} W^i_{cs_{(j)}} \Big) \Bigg] - C_s(A)\tau_{fc}. \end{aligned} \tag{2.1}$$

The total expected discounted profit over the considered time period is given by

$$\Pi = \textstyle\sum_{A \in \mathcal{A}} \pi(A) p_A,$$

and can be easily computed, considering an input data set.

In the further sections, we will describe the steps that will lead us to compute optimal base-stock policies that will minimize the costs across the network, incurred by inventory holding.

2.2 The Cost-Minimization Problem

In this section, we present the formulation for the optimization problem, taking into consideration that a significant difference in the total profit across the network is obtained by implementing a better inventory cost strategy (as we will see from data simulations).

Since there exists no known form for the optimal policy for a general assemble-to-order system, we assume that independent base-stock policies are used to control the item inventories (as it can already be seen in the model described above). This means that there is a target base-stock level for each item and the replenishment decision for this item is solely determined by its inventory position relative to the target level. Due to its simplicity, this type of policy is widely adopted in industry, but also in theoretical papers (see Lu and Song (2005). It is well known to be optimal if there are no economies of scale in replenishment.

2.2.1 Model Formulation

Consider again the decentralized network in Figure 2.1 and the profit function from Section 2.1.5. Two of the main goals we have achieved by developing such a profit function is reduced transportation costs and increased efficiency in handling returns from customers, from the point of view of reusing them instead of producing new items. At this stage, we have observed that we can still improve the profit, by optimizing the base-stock policies per item.

In order to find the optimal base-stock policies, we need to minimize all the costs generated by the level of the inventory, per item. Reconsider the profit Equation (2.1) and all terms that depend on the inventory levels s_i. Then the costs generated by holding inventory are as follows:

$$C(s_i) = p_i\Big[\lambda_i(1-\eta(s_i))v_i + h_i\Big(s_i + \frac{\rho_i^{s_i}}{(1-\rho_i)\mu_i}\Big)\Big],$$

for $i \in A$, and all $A \in \mathcal{A}$, and we are interested in

$$\min_{s_i} C(s_i).$$

Such a minimum is guaranteed to exist since the exponential functions with a base smaller than 1 are decreasing functions and the linear function is increasing due to a positive coefficient. Moreover, the minimum we find is the global minimum: the intersection point of all the functions.

2.2.2 The Exact Optimal Solution

We will derive the solution for the optimal base-stock policies $\boldsymbol{s}$. From *(A.2)*, we infer that computing the optimum per item will suffice for our goal, of optimizing the profit across the network. To find this optimum, we take the derivative of $C(s_i)$ with respect to s_i and set it to 0.

Recall that $\eta(s_i) = 1 - (\rho_i)^{s_i+1}$. It follows that

$$\frac{dC(s_i)}{ds_i} = p_i\big(\lambda_i v_i(\rho_i)^{s_i+1}\ln\rho_i + h_i + \frac{h_i}{(1-\rho_i)\mu_i}\rho_i^{s_i}\ln\rho_i\big) \overset{!}{=} 0,$$

for all $i = 1,\dots,m+n$, where $\overset{!}{=} 0$ denotes that we take the derivative to be equal to 0. Then

$$\rho_i^{s_i} = \frac{-h_i}{\lambda_i v_i\rho_i\ln\rho_i + \dfrac{h_i}{(1-\rho_i)\mu_i}\ln\rho_i}.$$

Since $0 < \rho_i < 1$, the right hand side of the equality stays positive. Apply the natural logarithm to both sides and get

$$s_i\ln\rho_i = \ln\frac{-h_i}{\lambda_i v_i\rho_i\ln\rho_i + \frac{h_i}{(1-\rho_i)\mu_i}\ln\rho_i}.$$

It follows that

$$s_i^* = (\ln\rho_i)^{-1}\ln\frac{-h_i}{\lambda_i v_i\rho_i\ln\rho_i + \frac{h_i}{(1-\rho_i)\mu_i}\ln\rho_i}, \tag{2.2}$$

for all $i = 1,\dots,m+n$, with $0 < \dfrac{-h_i}{\lambda_i v_i\rho_i\ln\rho_i + \frac{h_i}{(1-\rho_i)\mu_i}\ln\rho_i} < 1$ and $0 < \rho_i < 1$.

We conclude that in order to maximize our profit Π across the network, the suppliers should maintain a base-stock inventory level of $\boldsymbol{s}^* = (s_1^*,\dots,s_{m+n}^*)$.

3 Data Simulation Study

To test the performance of our model and gain insight in the key drivers of our system, we have conducted a simulation study for a sensitivity analysis. For the sake of clarity, we have decided to randomly generate just two parameters of the model: the order frequencies and the service rates. The other parameters are kept rather fixed so that a general trend can be interpreted more easily. We focus on the influence over the profit function and the optimal base-stock inventory level induced by various values of the probability of return and order arrival rates. In other words, we follow the behavior of our functions (2.1) and (2.2), respectively, for several values of those variables that cannot be entirely influenced by the company, i.e., return probability and arrival order rate. When analyzing the amount of inventory, we also draw in two distinctive graphs the percentage of inventory occurred due to returns and the percentage of inventory held due to newly produced items. The main code can be found in Appendix A.

We have made the following extra assumptions:

1. The model uses a uniformly distributed variable on the interval $[0, 1]$ to generate the order frequencies p_i for each component $i = 1, \ldots, m + n$;
2. The service rates are generated so that stability is conferred to the system (i.e., $\rho < 1$);
3. We fix the production cost at the amount of 100 and all other cost / price functions are taken proportional to the production costs;
4. The probability of return varies between 0.01 and 0.9;
5. The general order arrival rate λ varies between 5 and 50;
6. All fixed costs incurred by transport and assembling are taken equal to 0.05.

The general case is presented in Figure 3.1. By general case we mean the setup in which the arrival rate for orders from the customer (for end-products) is 5 and we take the service rate sufficiently big so that stability in the system is assured, for a load smaller than or equal to 0.5. As we can see, the profit decreases with the increase of the return probability. It means that the returns hurt the current configuration of the system and obviously the best solution will be to eliminate them. However, as presented in Chapter 1, businesses in the e-commerce area experience returns, whether they like or not, sometimes for no logical reason. How can this situation be changed? What parameter in this setup can be influenced so that our network handles returns in a more profitable way? We claim that for a lower service rate the profit will increase for some probabilities of returns.

One of the parameters that can be influenced by the company (i.e., each of the two suppliers) is the service rate. Thus, the most interesting part of the simulation results refers to what happens in the system when the load to the system is close to 1. For this purpose, μ_i is taken to be $1.04 \cdot \lambda_i$ for all $i = 1, 2, \ldots, m + n$ and the arrival rate for orders from the customer (for end-products) is 5.

In this setting, Figure 3.3 shows the impact on the profit function and on the inventory base-stock level of various values for the return probability. Both of the graphs show that returns are beneficial for the entire network. As a comparison with the previous network state, the overall profit increases even if higher stock needs to be held in order to meet the demand. In this model, we prove that, for certain probabilities of return, the company can actually boost its profit by accepting returns from its customers. The improvement observed for a return probability between 0 and 0.1 comes from decreasing the production costs due to the use of the returned components.

Moreover, we can assume that the customer's behavior is influenced by the returns policy and by how the company handles and these returns, which will be translated in a change in the arrival rate on the longer term. For return rates higher than 0.1 the increasing inventory holding costs will determine a steady profit reduction. The company can influence the return rates by "educating" the customers: providing instructions that are easy to understand and marketing material that explains the set up and functionality of the products. This way, they can control the number of items that come back and maximize the profits. The inventory base-stock level displays the gradual decrease when returns come with higher rates: there is no need for previously established inventory when the arrival rate of the returns is high.

Figures 3.4 shows the ratios between returns and the total inventory (first graph) and between newly produced items and the total inventory. As expected, once the probability of return increases, then the ratio for the returns in the total inventory increases and at the same time the ratio for newly produced items decreases. In other words, there will be no need for further production once there is a high number of returns.

For higher/lower general arrival rate λ, similar behavior can be exhibited. To obtain realistic results, whenever the arrival rate changes by an order of 10 ($5 \rightarrow 50$ or viceversa), the input parameters of the model, such as the production costs, sales price etc., should be adjusted accordingly.

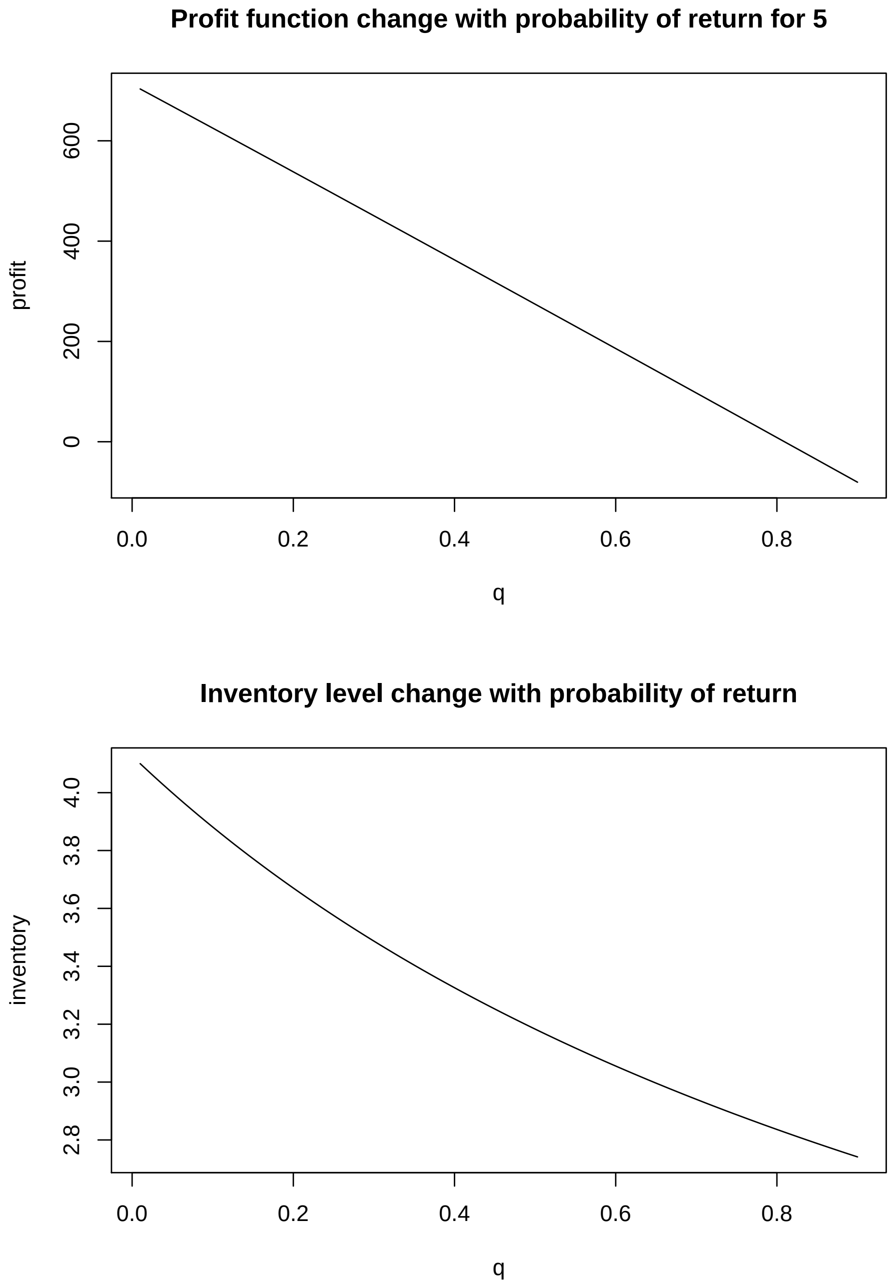

Figure 3.1: Profit and inventory base-stock level behavior for $\lambda = 5$

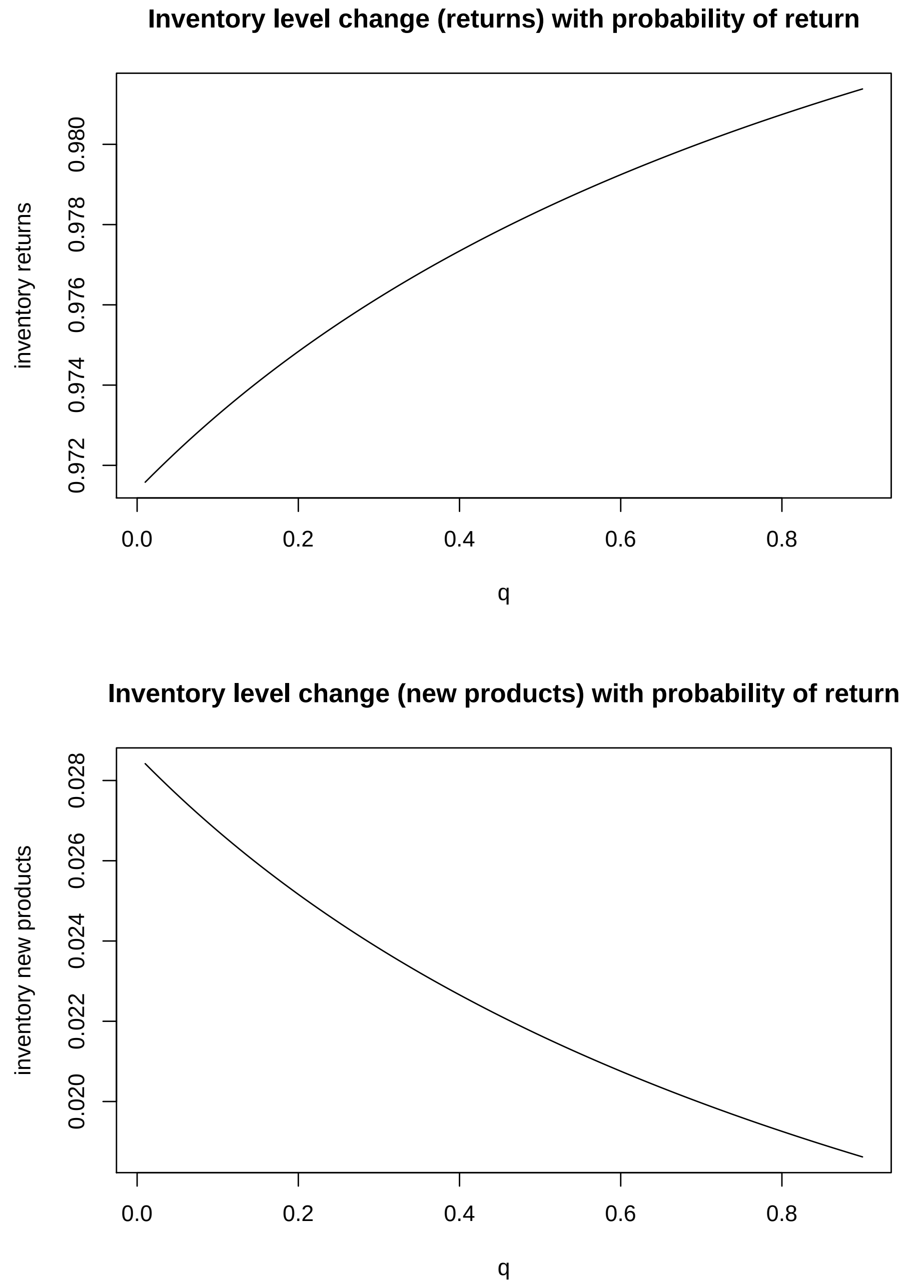

Figure 3.2: The evolution of the returns and new products ratios for $\lambda = 5$

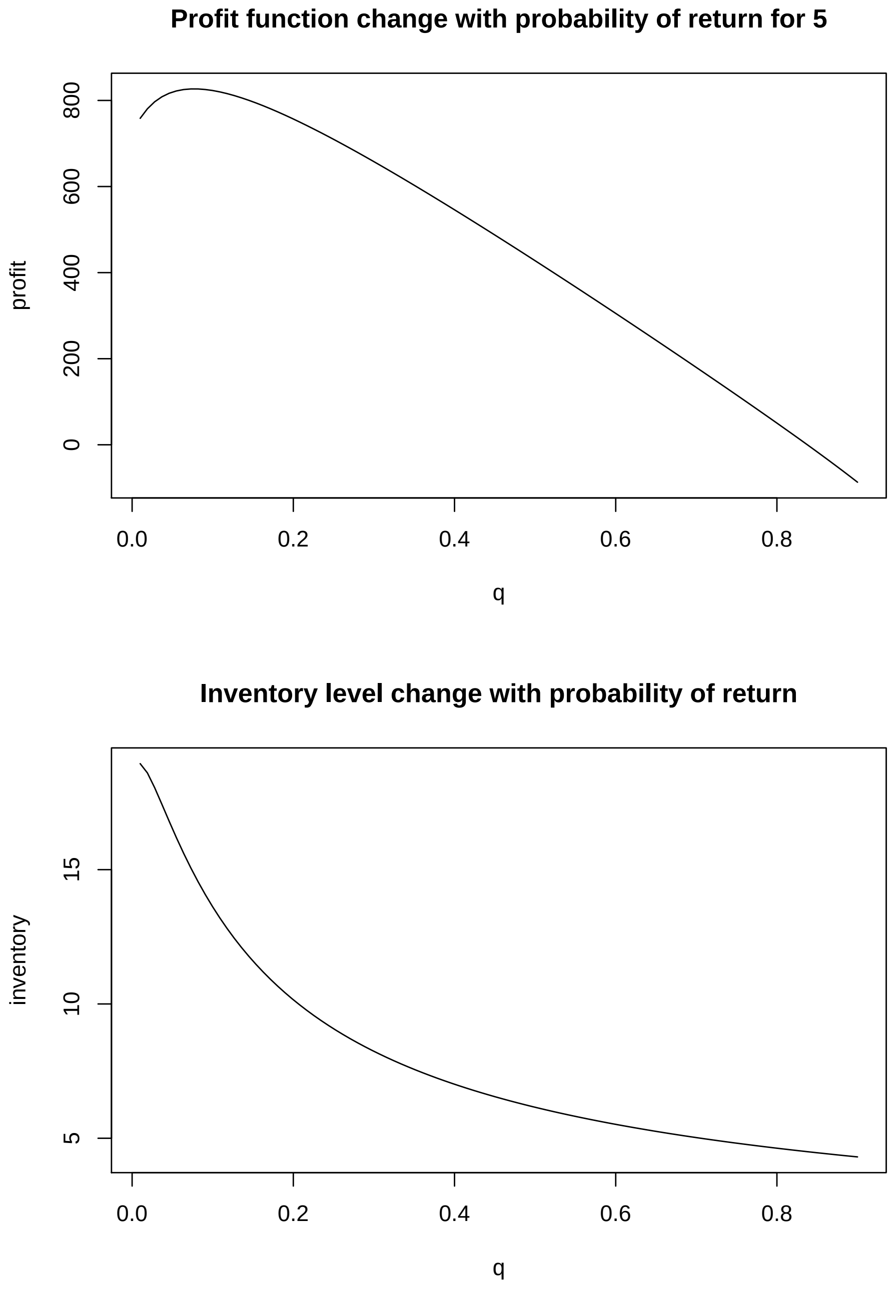

Figure 3.3: Profit and inventory base-stock level behavior for $\lambda = 5$

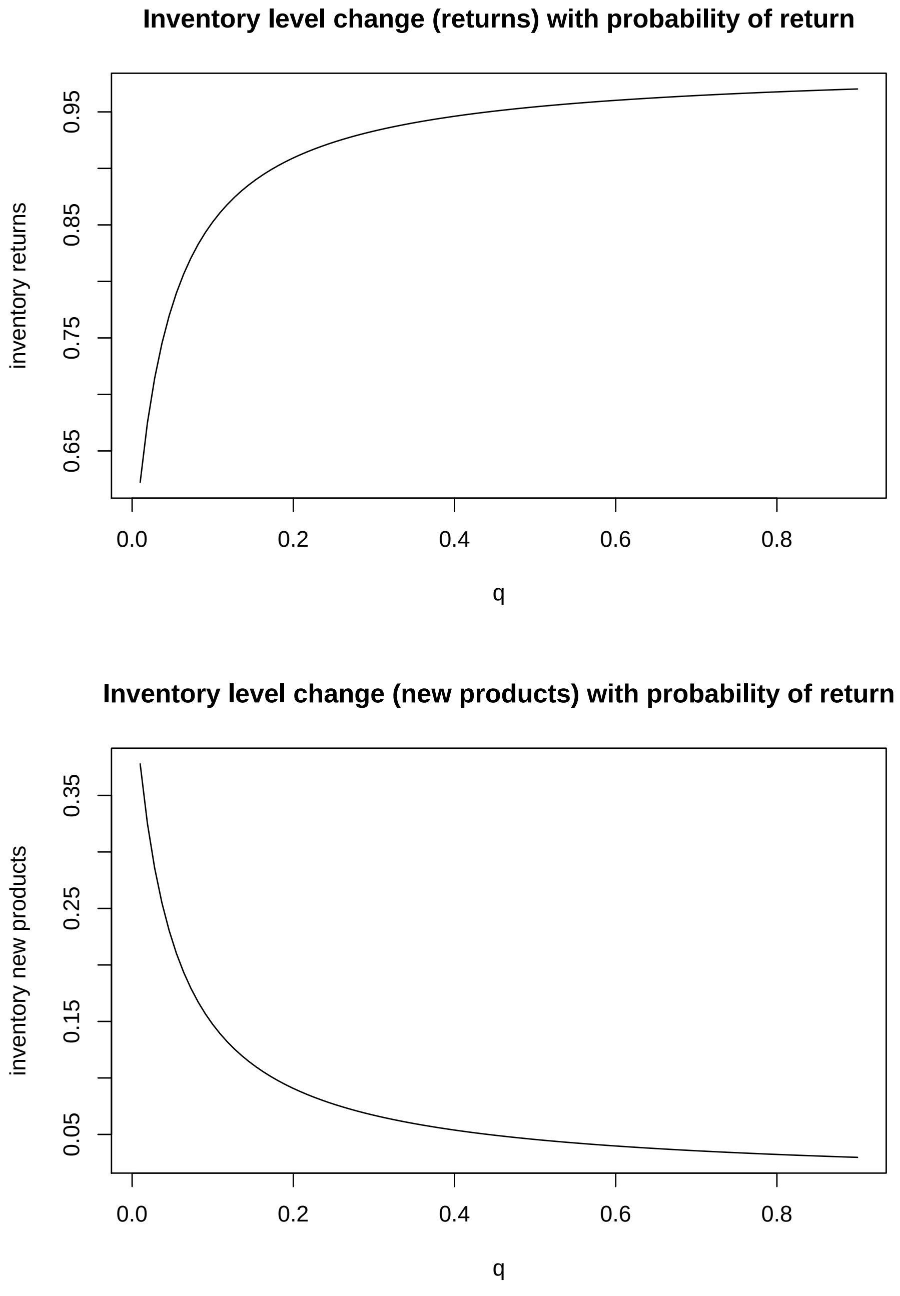

Figure 3.4: The evolution of the returns and new products ratios for $\lambda = 5$

4 Conclusions

E-commerce has always been the land of new profitable opportunities and the last years have exhibited an impressive boom. Since online customers are hard to keep, e-businesses have to create new incentives and improve already exiting policies so that they catch the interest of the potential customer visiting their website.

Returns policies are crucial for e-businesses since customers cannot physically check the item before the purchase. In addition to this, once a product is returned, the faster it re-enters the supply chain, the higher the chances for having it sold to another customer.

Assemble-to-order systems and the interchangeable modules offer the opportunity of satisfying a greater deal of customers at the same time. However, experiencing returns for customer-made products diminishes significantly the efficiency of these systems.

The solution we have proposed in the paper was to develop a component return strategy, which yields an overall solution to the above questions. We build a simple network, consisting of one manufacturer with two suppliers that receives orders from customers and forwards individual component orders to the right supplier. After assembling the components into the end-product as required, the manufacturer ships it to the customer (through a third-party logistics provider), which has the choice of returning the particular component he/she does not consider satisfactory enough. We focus on the company's profit behavior for different levels of returns as well as their impact on the inventory level, by finding the optimal base-stock policy.

By means of a simulation study we confirm our theoretical findings and analyze the significance of small changes of several parameters in the output of the model. We find, on one hand, that indeed returns are beneficial (the profit increases), which is in line with the economical expectations (the customers appreciate a strong return policy as an element of high quality in the customer service, and the decentralized network decreases extra transportation and handling costs and puts new returns back in the chain much faster. On the other hand, too high return rates will affect the inventory holding costs, causing a steady drop in the profit and they will also spread a negative perception among potential customers of the company. We also indicate how the inventory should be handled, in order to minimize the incurred costs.

The current paper is one step closer to building optimized and more profitable strategies that take the customer as the center of their actions. There are still few questions that remain open: the implementation of a component exchange in case of return, what physical distances should be considered so that the model can still provide reliable solutions etc., issues that come from practice but which impose heavy notation on mathematical models or high complexity in computations.

Appendix

```
# Simulation performed in R

# Size of the simulation
N=1000     # number of simulations
n=1        # number of components
ql=100     # length of q sequence

# Variable declaration
lambda=seq(5,50,l=1)            # general order arrival rate
q=seq(0.01,0.9,l=ql)            # probability of return
rho=matrix(NA,ncol=n,nrow=N)    # load to the system
v=matrix(NA,ncol=n,nrow=N)      # production costs
price=matrix(NA,ncol=n,nrow=N)  # sales price
lind=matrix(NA,ncol=n,nrow=N)   # component arrival rates
assem=matrix(NA,ncol=n,nrow=N)  # assembly costs
ship=matrix(NA,ncol=n,nrow=N)   # shipment costs
del=matrix(NA,ncol=n,nrow=N)    # delay costs
eta=matrix(NA,ncol=n,nrow=N)    # the existing inventory
vr=matrix(NA,ncol=n,nrow=N)     # remanufacturing costs
pr=matrix(NA,ncol=n,nrow=N)     # returns credit
tau=matrix(NA,ncol=n,nrow=N)    # transport costs
s=matrix(NA,ncol=n,nrow=N)      # optimal inventory level
mu=matrix(NA,ncol=n,nrow=N)     # service rates
h=matrix(NA,ncol=n,nrow=N)      # holding costs
profit=matrix(NA,ncol=n,nrow=N) # profit function
wsf=matrix(NA,ncol=n,nrow=N)    # delay wsf=wcs for both suppliers
eta_q=matrix(NA,ncol=n,nrow=ql) # proportion of returns in the inventory for each
                                  value of the return probability

# Constants
p.frac=3          # price-production costs proportion
assem.frac=0.05   # assembly-production costs proportion
ship.frac=0.05    # shipment-production costs proportion
del.frac=0.05     # delay-production costs proportion
vr.frac=0.2       # remanufacturing-production costs proportion
pr.frac=0.8       # returns-price proportion
h.frac=1/10       # holding-production costs proportion
tau.frac=0.05     # transport-price proportion
muc=4             # mu_factory=mu_customer
taugen=0.05       # transport costs to the customer

# Output
profit_q=matrix(NA,nrow=ql,ncol=n)    # profit function of q
s_q=matrix(NA,nrow=ql,ncol=n)         # inventory function of q
s_eta_q=matrix(NA,nrow=ql,ncol=n)     # inventory proportion of returns
```

```
s_noneta_q=matrix(NA,nrow=ql,ncol=n)  # inventory proportion of newly produced items

# Code
m=length(lambda)
par(mfrow=c(m,2))
for(i in 1:m){                 # observe behavior for each value of lambda
    par(mfrow=c(2,2))
    for(j in 1:length(q)){     # observe behavior for each value of q
        for(k in 1:N){         # each simulation
            p=runif(n,0.01,0.99)
            p=p/sum(p)         # ensure they represent probabilities
            l=1
            while(l<=n){       # for each component
               lind[k,l]=p[l]*lambda[i]   # compute the individual order rates

               # service rates
               #mu[k,l]=runif(1,1.2*lind[k,l],1.3*lind[k,l])
               mu[k,l]=1.04*lind[k,l]

               rho[k,l]=lind[k,l]/(mu[k,l]+q[j]*lind[k,l]) # load to the system
               v[k,l]=100                                  # production costs
               h[k,l]=h.frac*v[k,l]                        # holding costs

               # optimal inventory level
               s[k,l]=1/log(rho[k,l])*log(-h[k,l]/(log(rho[k,l])*(lind[k,l]*
                      v[k,l]*rho[k,l]+h[k,l]/(mu[k,l]*(1-rho[k,l])))))

               eta[k,l]=1-rho[k,l]^(s[k,l]+1)    # existing inventory

               price[k,l]=p.frac*v[k,l]     # generate sales price
               assem[k,l]=assem.frac*v[k,l] # generate the assembly costs
               vr[k,l]=vr.frac*v[k,l]       # generate the remanufacturing costs
               pr[k,l]=pr.frac*price[k,l]   # returns credit
               del[k,l]=del.frac*v[k,l]     # delay costs
               tau[k,l]=tau.frac*price[k,l] # transport costs
               wsf[k,l]=1/(mu[k,l]-q[j]*lind[k,l])+tau[k,l]+1/muc # delay
               ship[k,l]=ship.frac*v[k,l]   # shipment costs

               # profit function
               profit[k,l]=lind[k,l]*price[k,l]-lind[k,l]*assem[k,l]-
                           lind[k,l]*(1-eta[k,l])*v[k,l]-q[j]*lind[k,l]*
                           (vr[k,l]+pr[k,l])-h[k,l]*(s[k,l]+
                           rho[k,l]^s[k,l]/(mu[k,l]*(1-rho[k,l])))-
                            4*del[k,l]*wsf[k,l]-ship[k,l]*taugen

               if(s[k,l]>0) l=l+1           # go to next component only if positive
                                              inventory
            } # end while
            l=n              # enables queries over the last column of all matrices

        } # end for k
        profit_q[j,]=apply(profit,2,mean)
        s_q[j,]=apply(s,2,mean)
        eta_q[j,]=apply(eta,2,mean)

    } # end for j

s_eta_q=eta_q
s_noneta_q=1-eta_q

par(mfrow=c(1,1))
postscript("fig_profit.ps",height=5,width=7,horizontal=F)
    plot(profit_q[,1]~q,t="l", main=paste("Profit function change with probability
         of return for ",lambda, sep=""), ylab="profit")        # plot for the
                                                                 first component
dev.off()
postscript("fig_sq.ps",height=5,width=7,horizontal=F)
    plot(s_q[,1]~q,t="l", main="Inventory level change with probability of return",
         ylab="inventory")                                      # plot for the
                                                                 first component
dev.off()
postscript("fig_seta.ps",height=5,width=7,horizontal=F)
        plot(s_eta_q[,1]~q,t="l", main="Inventory level change (returns)  with
             probability of return", ylab="inventory returns")  # plot for the
                                                                 first component
dev.off()
postscript("fig_snoneta.ps",height=5,width=7,horizontal=F)
        plot(s_noneta_q[,1]~q,t="l", main="Inventory level change (new products)
             with probability of return", ylab="inventory new products")
             # plot for the first component
dev.off()
} # end for i
```

Bibliography

[1] Arar, Y. (2008). Most returned products work fine, study says. Published electronically in *PC World Magazine*, June 2008. Available at *http://www.pcworld.com/article/id,146576/article.html.*

[2] Banker, S. (2001). E-business and reverse logistics. Published electronically. Available at *http://www.ebizq.net/topics/scm/features/2589.html.*

[3] Bernstein, F., DeCroix, G., and Wang, Y. (2005). Incentives and flexibility in a decentralized multi-product assemble-to-order. Research Paper, *Manufacturing and Service Operations Management Conference.*

[4] DeCroix, G., Song, J.S., and Zipkin, P. (2008). Managing an assemble-to-order system with returns. *Manufacturing and Service Operations Management.* Published online in *Articles in Advance*, May 7, 2008, DOI: 10.1287/msom.1070.0209.

[5] Frankel, C. (1996). The environment. *IEEE Spectrum*, **33**, 76–81.

[6] Guide, V.D.R., van Wassenhove, L.N., and Teunter, R.H. (2003). Matching demand and supply to maximize profits from remanufacturing. *Manufacturing and Service Operations Management*, **5**(4), 303–316.

[7] Kokkinaki, A.I., Dekker, R., de Koster, M., Pappis, C., and Verbeke, W. (2002). E-business models for reverse logistics: Contributions and challenges. *International Conference on Information Technology: Coding and Computing*, p. 0470.

[8] Lu, Y. and Song, J.S. (2005). Order-based cost optimization in assemble-to-order systems. *Operations Research*, **53**(1), 151–169.

[9] Mukhopadhyay, S.K. and Setoputro, R. (2005). Optimal return policy and modular design for build-to-order products. *Journal of Operations Management*, **23**(5), 496–506.

[10] Rogers, D.S., Lambert, D.M., Croxton, K.L., and García-Dastugue, S.J. (2002). The returns management process. *The International Journal of Logistics Management*, **13**(2), 1–18.

[11] Saskatchewan, B.C. (2008). Handling e-commerce returns. Published electronically. Available at *http://www.e-future.ca/sask/pdf/efc_e-c_returns.pdf.*

[12] Song, J.S., Xu, S.H., and Liu, B. (1999). Order-fulfillment performance measures in an assemble-to-order system with stochastic leatimes. *Operations Research*, **47**(1), 131–149.

[13] Souza, G.C., Daniel, V., Jr., R.G., van Wassenhove, L.N., and Blackburn, J.D. (2006). Time value of commercial product returns. *Management Science*, **52**(8), 1200–1214.

[14] Swaminathan, J.M. and Tayur, S.R. (2003). Models for supply chains in e-business. *Management Science*, **49**(10), 1387–1406.

[15] Vlachos, D. and Dekker, R. (2003). Return handling options and order quantities for single period products. *European Journal of Operational Research*, **151**(1), 38–52.