

MANAGING WARRANTIES: FUNDING A WARRANTY RESERVE AND  
OUTSOURCING PRIORITIZED WARRANTY REPAIRS

by  
Peter S. Buczkowski

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill  
2004

Approved by

---

Advisor: Professor Vidyadhar G. Kulkarni

---

Reader: Dr. Suheil Nassar

---

Reader: Professor Jayashankar M. Swaminathan

---

Reader: Professor Eylem Tekin

---

Reader: Professor Jon W. Tolle

© 2004  
Peter S. Buczkowski  
ALL RIGHTS RESERVED

## ABSTRACT

Peter S. Buczkowski: MANAGING WARRANTIES: FUNDING A WARRANTY  
RESERVE AND OUTSOURCING PRIORITIZED WARRANTY REPAIRS

(Under the direction of Professor Vidyadhar G. Kulkarni)

We consider two problems central to the managing of warranty costs by the manufacturer. First, we consider funding an interest-bearing warranty reserve with contributions after each sale. The problem for the manufacturer is to determine the initial level of the reserve fund and the amount to be put in after each sale, so as to ensure that the reserve fund covers all the warranty liabilities with a prespecified probability over a fixed period of time. We assume a non-homogeneous Poisson sales process, random warranty periods, and an exponential failure rate for items under warranty. We derive the mean and variance of the reserve level as a function of time and provide a heuristic to aid the manufacturer in its decision.

We also consider the problem of outsourcing warranty repairs to outside vendors when items have priorities in service. The manufacturer has a contract with a fixed number of repair vendors. The manufacturer pays a fixed fee for each repair done by a vendor which is independent of the repair type and priority class but depends on the vendor. There are a fixed number of items under warranty, and each item belongs to one of a fixed number of priority classes. The manufacturer also pays for holding costs incurred when the items are at the vendors, the holding cost being higher for the higher priority items. The vendors provide a pre-emptive priority for an item over all other items of lower priority. We focus on static allocation of the warranty repairs; that is, we assign

all items to the vendors at the beginning of the warranty period. We give the known algorithm to optimally solve the one priority class problem and solve the multi-priority class problem by formulating it as a convex minimum cost network flow problem. Then, we give numerical examples to illustrate the cost benefits of a multi-priority structure.

## ACKNOWLEDGMENTS

I am sincerely grateful to my advisor, Dr. Vidyadhar Kulkarni, for his knowledge, guidance, time, and support through my studies at Chapel Hill. His comments greatly improved this dissertation from beginning to end.

I would also like to express my gratitude to my committee members: Dr. Suheil Nassar, Dr. Jayashankar M. Swaminathan, Dr. Eylem Tekin, and Dr. Jon W. Tolle, for their help and suggestions on my research. Special thanks to Dr. Mark Hartmann for donating his time and providing invaluable suggestions, without which this dissertation would not be complete. Let me also extend thanks to my classmates, especially Wei Huang, Bala Krishnamoorthy, Michelle Opp, and Rob Pratt. Their advice and support is evident in many areas of this thesis.

Finally, I would like to thank my wife Gretchen for her support and understanding over the last four years.

## CONTENTS

LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Literature Review . . . . .	2
1.3 Organization of the Dissertation . . . . .	6
<b>2 Funding a Warranty Reserve</b>	<b>8</b>
2.1 Overview . . . . .	8
2.2 Notation and Assumptions . . . . .	8
2.3 Probability Distribution of the Number of Items Under Warranty . . . . .	11
2.3.1 Distribution of $X^n(t)$ . . . . .	11
2.3.2 Distribution of $X^o(t)$ . . . . .	12
2.4 Differential Equations for Moments of $R(t)$ . . . . .	15
2.5 Solution for Moments of $R(t)$ . . . . .	24
2.5.1 Example: Constant Warranty Period and Constant Sales Rate . . . . .	25
2.6 Deciding the Values of $c$ and $R_0$ . . . . .	28
2.6.1 Distribution of $R(t)$ . . . . .	28
2.6.2 Heuristic for Deciding $c$ and $R_0$ . . . . .	30
2.7 Numerical Computations . . . . .	33

<b>3</b>	<b>Warranty Reserve: Extensions</b>	<b>36</b>
3.1	Random contribution to the reserve after each sale . . . . .	36
3.2	Multiple products using a single reserve . . . . .	37
3.3	$X^o(t)$ is known during $[0, T]$ . . . . .	38
<b>4</b>	<b>Outsourcing Prioritized Warranty Repairs</b>	<b>42</b>
4.1	Overview . . . . .	42
4.2	Notation and Assumptions . . . . .	43
4.3	Problem Formulation . . . . .	45
4.4	Single Priority Class . . . . .	48
4.5	Minimum Cost Network Flow Problems . . . . .	52
4.5.1	Convex Network Problems . . . . .	55
4.6	Network Flow Formulation . . . . .	57
4.6.1	Single Priority Class . . . . .	57
4.6.2	Two Priority Classes . . . . .	59
4.6.3	Multiple Priority Classes . . . . .	61
4.7	Computational Issues . . . . .	64
4.8	An Example . . . . .	67
4.9	Cost Benefits of the Multi-Priority Approach . . . . .	68
4.10	Selecting the Values of $K_i$ . . . . .	70
<b>5</b>	<b>Conclusions and Future Work</b>	<b>74</b>
	<b>Bibliography</b>	<b>77</b>

## LIST OF TABLES

2.1	Suggested Values for $q$ . . . . .	32
2.2	Simulation Results . . . . .	35
4.1	Arc Properties for Two-priority Network . . . . .	59
4.2	Arc Properties for $m$ -priority Network . . . . .	63
4.3	Costs and Service Rates for Each Vendor . . . . .	68
4.4	Average Holding Costs for Vendors . . . . .	69
4.5	Vendor Properties for Similar Cost Example . . . . .	70
4.6	Arc Properties for the Reward Network . . . . .	72
4.7	Costs and Service Rates for Each Vendor . . . . .	73

## LIST OF FIGURES

2.1	Example of Warranty Reserve Account . . . . .	9
2.2	Examples of Confidence Bands for $R(t)$ . . . . .	29
2.3	Expected Reserve for Various Values of $X(0)$ . . . . .	34
4.1	Network Model of Single Priority Problem . . . . .	57
4.2	Network Model of Two-priority Problem . . . . .	59
4.3	Network Model of $m$ Priority Problem . . . . .	62
4.4	Network Model of Reward Problem . . . . .	72

# Chapter 1

## Introduction

### 1.1 Overview

Since the Magnuson-Moss Warranty Act of 1975 [33], manufacturers are required to provide a warranty for all consumer goods which cost more than \$15. Warranties play an important role in the consumer-manufacturer relationship. They offer assurance to the consumer that their purchase will achieve certain performance standards through at least the warranty period. The manufacturers use warranties as a marketing tool and they limit their liability.

When designing product warranties, the manufacturers must decide on many issues, such as warranty policy, length of warranty period, repair policy, and quality control. They also have to plan to cover the costs associated with the warranty. An issue of critical importance to the manufacturers is managing the costs associated with the warranty effectively. Our research investigates two key questions of planning for these costs.

The first is of funding a warranty reserve account with contributions made after each sale. A warranty reserve is used to accommodate all of the costs associated with the servicing of a warranty of a product. We model a policy that is currently implemented in industry; that of adding a fraction of each sale to the reserve fund. There are a variety

of goals that a manufacturer may have regarding its warranty reserve. Two general goals are to keep the reserve above some target dollar amount  $B > 0$  and to not have an excessive amount of money in the reserve. The reasoning behind these goals is simple: a shortage requires extra administrative costs and may even have legal ramifications, while an excessive surplus locks money in the reserve that may be more useful for other business interests. Achieving these goals requires careful planning.

We also consider the problem of outsourcing warranty repairs to outside vendors when items have priority levels. For example, some warranty contracts specify the repair turnaround time (e.g. 1 day, 3 days, or 7 days). With careful management, repair outsourcing can be a major benefit to the manufacturer. A smooth operation can improve customer satisfaction and turnaround times, while allowing the manufacturer to maintain its focus on production. While the manufacturer may have a central repair depot, it often is not effective to ship items to the depot due to time and cost constraints. Thus it might be beneficial to choose repair vendors distributed geographically so as to be close to the customers. The manufacturer must seek a balance between cost savings and customer service. If not, some customers will be lost because of poor service. Repair outsourcing is an especially important problem when considering priorities because high priority customers will typically inflict greater loss if the manufacturer does not meet their expectations.

## 1.2 Literature Review

Warranty theory has been heavily studied over the past two decades. Blischke and Murthy [5] wrote a comprehensive reference for the subject. They discuss many different types of warranty policies, including many warranty policies currently implemented in industry. Numerous cost and optimization models are developed from both the consumer's and the manufacturer's point of view, including life cycle and long-run average

cost models. We use these models to compute the expected warranty cost of a product in our numerical examples.

Many of the early papers on warranty theory discuss the costs and other effects that are associated with warranties. Glickman and Berger [13] consider the effect of warranty on demand by assuming that demand increases as the warranty period increases.

Warranty costs affect both the buyer and the seller. Mamer [23] wrote the first paper to provide a comprehensive model of both the buyer's and seller's expected costs and long-run average costs for the free replacement warranty. Our research focuses on the manufacturers' view of warranty costs.

The concept of a warranty reserve is a topic of many research works. The initial papers on warranty reserves discussed here consider a fixed product lot size throughout the life cycle of a product (or equivalently, a fixed cumulative failure rate). Menke [25] wrote one of the first papers to address the warranty reserve problem. He concentrates on calculating the expected warranty cost over a given warranty period for two types of pro-rata warranty policies (linear rebate and lump-sum rebate) assuming a constant product failure rate. Amato and Anderson [2] extend Menke's model by allowing the reserve fund to accrue interest, requiring the consideration of discounted costs. A comparison to Menke's results is made, concluding that discounting significantly reduces the expected warranty reserve over longer periods of time. Both models are rather limited in scope because they only consider pro-rata warranty policies and an exponential failure distribution.

Balcer and Sahin [4] derive the moments of the total replacement cost for both the free-replacement and pro-rata warranty policies during the product life cycle. They assume that successive failure times form a renewal process.

Mamer [24] uses renewal theory to model repeated product failures over a life cycle of the product. He incorporates discounting in his model and allows for a general failure distribution. However, he does not consider the sales process nor compute a reserve.

Tapiero and Posner [32] allow for a portion of each sale to be set aside for future warranty costs. The contributions to the reserve fund and the items sold occur at a constant rate. The claims are generated according to a compound Poisson Process and they use a sample path technique to compute the long-run probability distribution of the warranty reserve.

Eliashberg, Singpurwalla, and Wilson [12] calculate the reserve for a product whose failure rate is indexed by two scales, time and usage. They allow for a general failure rate and assume a form of imperfect repair. The warranty reserve is computed to minimize a loss function for the manufacturer.

Ja, et al. [18] compute the distribution of the total discounted warranty cost over the life cycle of the product. They analyze the discounted warranty cost of a single sale under many different policies and then consider different stochastic sales processes. A single contribution to the reserve is made at the beginning of the life cycle. However, the subtractions from the reserve due to warranty costs are tracked as a function of time.

Another application related to the warranty reserve problem is the insurance premium problem. An insurance company must decide on the monthly premium to charge a certain class of customer. Low premiums result in loss to the insurer, while high premiums result in loss of business to the competition. A discussion of this can be found in [30]. There are other related problems, including the funding of a company's pension plan. Many of these problems are solved using actuarial models, particularly collective risk (loss) models (see [22] and [9] for references on this subject). However, the current models do not incorporate the number of policies insured by the company at any given time.

The works described above illustrate many different models to compute the warranty reserve. However, they assume that the reserve is either funded at the beginning of the product sales period or at a constant rate. We extend this research by modeling

contributions to the reserve after each sale and allowing the cumulative warranty claim rate to depend on the sales process.

We now turn to the warranty repair outsourcing problem. At its most basic structure, the static allocation model reduces to a resource allocation problem with integer variables. Without considering priorities, the problem has a separable objective function. This problem has been widely studied in the literature. Gross [15] first proposed a simple greedy algorithm to find the optimal solution if the objective is convex.

Several authors have since expanded the problem. Ibarki and Katoh [16] provide a comprehensive review of resource allocation problems and algorithms to solve them. Their bibliography provides a review of the literature up to 1988. Bretthauer and Shetty [6], [7] also give a survey of a generalization: the nonlinear knapsack problem. They provide a proof of the greedy algorithm by the generalized Lagrange multiplier method. Zaporozhets [34] gives an alternate proof of the greedy algorithm. Opp, et al. [28] describes the greedy algorithm in detail for the convex separable resource allocation problem and its application to our problem without priorities. Also discussed are some computational issues associated with the application, mostly regarding the expected queue length.

Once priorities are considered, the objective is no longer separable. We extend the previous research by providing an algorithm to optimally solve the closed static allocation problem with priorities. We have developed a new proof of the greedy algorithm when there is only one priority class, and give a new algorithm to handle the special structure of the objective when there are multiple priority classes. Finally, we investigate the benefits of a multi-priority structure for the manufacturer.

## 1.3 Organization of the Dissertation

In Chapter 2, we address the problem of funding a warranty reserve. In the first two sections, we provide an overview of the problem and the notations and assumptions used throughout Chapters 2 and 3. In Section 2.3, we derive the probability distribution for the number of items under warranty at time  $t$ . We follow that with differential equations for the first and second moment of the reserve level in Section 2.4. The general solutions to these equations are provided in the following section along with the special case of a constant warranty period. We provide a heuristic for determining the values of the contribution amount after each sale and the initial reserve level in Section 2.6 and some simulation results in Section 2.7.

We consider three extensions of the warranty reserve problem in Chapter 3:

- The reserve contribution after the  $j$ th sale is a random variable. (Section 3.1)
- The manufacturer maintains a single reserve fund for multiple products or multiple warranties. (Section 3.2)
- The remaining lifetimes of the items sold prior to time  $t$  are known. (Section 3.3)

Next, we turn to the warranty repair outsourcing problem in Chapter 4. After a brief problem overview, we state the notation and assumptions of the problem in Section 4.2. In Section 4.3, we derive the cost function and state the optimization problem for the model. We provide the known algorithm to solve the single priority problem in Section 4.4 and give a new proof of the algorithm. Our algorithm to solve the  $m$ -priority problem uses network concepts. We give a brief overview of minimum cost network flow problems in Section 4.5. Then we reformulate the optimization problem as a convex minimum cost flow problem and provide the algorithm to solve the problem. We provide the simplified algorithm for the one- and two-priority case and give the general algorithm for the  $m$ -priority case. In Section 4.7, we discuss the computational issues that arise in

the problem and provide an example in the following section. In Section 4.9, we illustrate the cost benefits of the priority structure. We provide two examples: the first with very different holding costs between the high and low priority customers and the second with relatively similar holding costs between the high and low class customers. We complete the discussion of the outsourcing problem in Section 4.10 by presenting an optimization problem for the manufacturer when the customer pays additional monies for priority in service.

# Chapter 2

## Funding a Warranty Reserve

### 2.1 Overview

In this chapter, we consider the problem of funding a warranty reserve account. We consider a manufacturer who adjusts its warranty reserve at a series of fixed time points (e.g. at times  $0, T, 2T, \dots$ ). In this dissertation, we consider a single period  $[0, T]$ . The manufacturer must decide on the initial amount in the reserve at the beginning of the period and the contribution amount from each sale. We derive the mean and variance of the reserve level as a function of time and provide a heuristic to aid the manufacturer in its decision.

### 2.2 Notation and Assumptions

We begin by introducing some notation and assumptions. We define  $R(t)$  as the amount in the reserve at time  $t$ , where  $t = 0$  represents the beginning of the period. The reserve fund accrues interest at constant rate  $\alpha > 0$ . At each sale, an amount  $c$  is contributed to the account. The manufacturer must decide on the initial reserve level,  $R_0$ , and the contribution amount to the reserve from each sale,  $c$ , at the beginning of the period. Let  $S(t)$  be the total number of sales up to time  $t$ . We assume that  $\{S(t), t \geq 0\}$  is

a nonhomogeneous Poisson Process with a known rate function  $\theta(\cdot)$  (we call this an *NPP*( $\theta(\cdot)$ )). Each item is under warranty for a random amount of time. The warranty durations are independent and identically distributed with common cdf  $F(\cdot)$  and mean  $w$ . Also, the warranty durations are independent of any future failures. Note that this allows for a constant warranty period. The customer always makes a warranty claim at each product failure. We assume instantaneous repair and that the repair times of a given item follow a Poisson Process with rate  $\lambda$ . The repair cost of the  $i$ th failure (at time  $Y_i$ ) is  $D_i$ , a random variable. The  $D_i$ 's are i.i.d. and are independent of the failure time. Let  $D(t)$  be the total undiscounted cost of all claims up to time  $t$ ; hence

$$D(t) = \sum_{i:Y_i \leq t} D_i.$$

Let  $X(t)$  denote the number of items under warranty at time  $t$  and  $S_j$  denote the time of the  $j$ th sale. The manufacturer observes the number of items under warranty at time 0 to aid in his determination of  $R_0$  and  $c$ . The manufacturer may or may not know the remaining warranty lifetimes of the items under warranty at time 0; we consider both cases. Figure 2.1 illustrates the evolution of the warranty reserve over time.

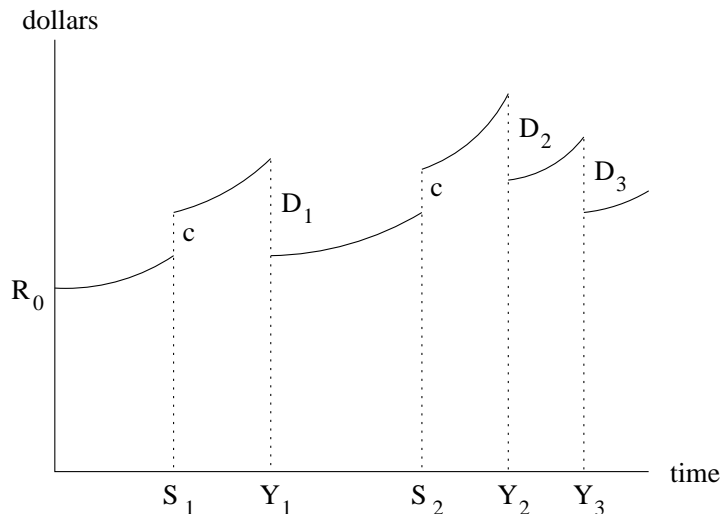


Figure 2.1: Example of Warranty Reserve Account

For computational purposes, it is helpful to distinguish between the effects of the items sold since time 0 from the items sold before time 0. We will break  $X(t)$  into two parts: let  $X^n(t)$  represent the number of items under warranty at time  $t$  that were sold after time 0, and let  $X^o(t)$  represent the number of items under warranty at time  $t$  that were sold prior to time 0. We write

$$R(t) = R^n(t) + R^o(t),$$

where  $R^n(t)$  is the portion of the reserve related to the new items  $X^n(t)$ , and  $R^o(t)$  is the portion of the reserve related to the old items  $X^o(t)$ . Thus, in  $R^n(t)$ , we add contributions from new purchases and only subtract the claims generated by new items. In  $R^o(t)$ , there are no new contributions, so we only subtract claims generated by old items. Similarly, we define  $D^n(t)$  ( $D^o(t)$ ) as the total undiscounted claims from time 0 to  $t$  generated by the new (old) items. It is convenient to define  $R^n(0) = 0$  and  $R^o(0) = R_0$ . In our model we track both  $R^n(t)$  and  $R^o(t)$  for ease in computation, while the manufacturer just tracks  $R(t)$ .

We will calculate first and second moments for some of the functions  $R(t)$ ,  $S(t)$ ,  $X(t)$ ,  $D(t)$  and their components ( $R^n(t)$ ,  $R^o(t)$ , etc.). We represent this by using lower case for the first moment and using lower case with a subscript of 2 for the second moment (e.g.  $r(t) = E[R(t)]$  and  $r_2(t) = E[R^2(t)]$ ). Any exception to this will be mentioned at the appropriate place throughout the thesis. Also, we will use  $\Delta_h$  to indicate the change in a function from  $t$  to  $t + h$ . For example,  $\Delta_h R(t) = R(t + h) - R(t)$ . Finally, we will use the standard  $o(h)$  notation for a function  $g(h)$  when

$$\lim_{h \rightarrow 0} \frac{g(h)}{h} = 0.$$

## 2.3 Probability Distribution of the Number of Items Under Warranty

In this section we derive the distributions for  $X^n(t)$  and  $X^o(t)$ .

### 2.3.1 Distribution of $X^n(t)$

First we explore the  $\{X^n(t), t \geq 0\}$  process. At time  $t$ , items are purchased according to an  $NPP(\theta(\cdot))$ . The amount of time an item is under warranty is a random variable with cdf  $F(\cdot)$ . We assume there is no capacity on the total number of items under warranty at any time. Therefore, we can model the  $\{X^n(t), t \geq 0\}$  process as an  $M_t/G/\infty$  queue with arrival rate  $\theta(\cdot)$  and service time distribution  $F(\cdot)$ .

The following result was established independently by Palm [29] and Khintchine [21]. Most recently, Eick, Massey, and Whitt [11] provided a simpler proof of this result and developed some further results for the  $M_t/G/\infty$  queue.

**Theorem 1** *Let  $Q(t)$  be the number of items in an  $M_t/G/\infty$  queue at time  $t$  with arrival rate  $\theta(\cdot)$  and i.i.d. service times  $S$  with cdf  $F(\cdot)$ . At time  $t$ , there are 0 items in the queue. Then, for each time point  $t \geq 0$ ,  $Q(t)$  has a Poisson distribution with mean*

$$E \left[ \int_{t-S}^t \theta(u) du \right] = \int_0^t \theta(t-u) [1 - F(u)] du.$$

Therefore, the moments of  $X^n(t)$  are

$$x^n(t) = \int_0^t \theta(t-u) [1 - F(u)] du, \text{ and} \tag{2.1}$$

$$x_2^n(t) = x^n(t) + (x^n(t))^2. \tag{2.2}$$

### 2.3.2 Distribution of $X^o(t)$

We consider two possible cases for the items sold prior to time 0: either the manufacturer fully knows the remaining warranty durations of all items under warranty at time 0 or that the remaining warranty durations are i.i.d. random variables with common cdf  $Q(t)$ . The former case is rather easy to handle – the entire sample path of  $X^o(t)$  is a deterministic function. If the remaining warranty durations are unknown, the probability that the remaining warranty duration of an item is greater than  $t$ , given that it was under warranty at time 0, is  $1 - Q(t)$ . Hence,

$$X^o(t) \sim \text{Bin}(X(0), 1 - Q(t)),$$

where  $X(0)$  is the number of items under warranty at time 0. The moments are

$$x^o(t) = X(0)(1 - Q(t)), \text{ and} \tag{2.3}$$

$$x_2^o(t) = X(0)Q(t)(1 - Q(t)) + (x^o(t))^2. \tag{2.4}$$

One choice for  $Q(t)$  is obtained from the stationary distribution of the remaining service times in an  $M/G/\infty$  queue in steady state. From Takács [31], we have the following lemma.

**Lemma 1 (Takács, Theorem 3.2.2)** : *Let  $X(t)$  be the number of items under warranty at time  $t$ , and let  $L_i(t)$  denote the remaining warranty period of item  $i$  under warranty. The sales process is a Poisson process. If  $w < \infty$ , we have*

$$\lim_{t \rightarrow \infty} P(L_i(t) < x_i \forall i = 1, \dots, k \mid X(t) = k) = \prod_{i=1}^k \frac{1}{w} \int_0^{x_i} [1 - F(s)] ds,$$

*and the limiting distribution is independent of the initial state.*

Therefore, under the assumption of Poisson input in steady state, we have that the remaining warranty distributions are independent of each other and the probability that an item is still under warranty at time  $t$ , given that it was under warranty at time 0 is

$$1 - Q(t) = \frac{1}{w} \int_t^{\infty} [1 - F(s)] ds, \quad (2.5)$$

where  $Q(t)$  is determined by Lemma 1, i.e.

$$Q(t) = \frac{1}{w} \int_0^t [1 - F(s)] ds. \quad (2.6)$$

This result can be extended to the case of a non-homogeneous Poisson Process, as shown in the following lemma.

**Lemma 2** *Let  $X(t)$  be the number of items under warranty at time  $t$ , and let  $L_i(t)$  denote the remaining warranty period of item  $i$  under warranty. Suppose that the sales process begins at time  $-A$ , and  $\{S(t), t \geq -A\}$  is a non-homogeneous Poisson with rate function  $\theta(\cdot)$ . We have*

$$P(L_i(t) < x_i \forall i = 1, \dots, k | X(t) = k) = \prod_{i=1}^k \frac{\int_0^{t+A} [F(s + x_i) - F(s)] \theta(t - s) ds}{\int_0^{t+A} [1 - F(s)] \theta(t - s) ds}.$$

**Proof.** Since the sales process is an  $NPP(\theta(\cdot))$ , we know that for  $t \geq -A$ ,

$$P(X(t) = k) = \exp \left( - \int_{u=0}^{t+A} (1 - F(u)) \theta(t - u) du \right) \frac{\left( \int_{s=0}^{t+A} (1 - F(s)) \theta(t - s) ds \right)^k}{k!}. \quad (2.7)$$

We compute  $P(L_i(t) < x_i \forall i = 1, \dots, k; X(t) = k)$ . Let  $\Theta(u) = \int_{s=-A}^u \theta(s)ds$ . We have

$$\begin{aligned}
& P(L_i(t) < x_i \forall i = 1, \dots, k; X(t) = k) \\
&= \sum_{n=k}^{\infty} e^{-\Theta(t)} \frac{\Theta(t)^n}{n!} \binom{n}{k} \left[ \frac{1}{\Theta(t)} \int_0^{t+A} F(s)\theta(t-s)ds \right]^{n-k} \\
&\quad \cdot \prod_{i=1}^k \left[ \frac{1}{\Theta(t)} \int_0^{t+A} [F(x_i+u) - F(u)]\theta(t-u)du \right] \\
&= e^{-\Theta(t)} \frac{\Theta(t)^k}{k!} \sum_{n=k}^{\infty} \frac{1}{(n-k)!} \left[ \int_0^{t+A} F(s)\theta(t-s)ds \right]^{n-k} \\
&\quad \cdot \prod_{i=1}^k \left[ \frac{1}{\Theta(t)} \int_0^{t+A} [F(x_i+u) - F(u)]\theta(t-u)du \right] \\
&= \frac{e^{-\Theta(t)}}{k!} \exp \left( \int_0^{t+A} F(s)\theta(t-s)ds \right) \prod_{i=1}^k \int_0^{t+A} [F(x_i+u) - F(u)]\theta(t-u)du \\
&= \exp \left( - \int_0^{t+A} (1-F(s))\theta(t-s)ds \right) \frac{1}{k!} \prod_{i=1}^k \int_0^{t+A} [F(x_i+u) - F(u)]\theta(t-u)du. \quad (2.8)
\end{aligned}$$

The conditional probability  $P(L_i(t) < x_i \forall i = 1, \dots, k | X(t) = k)$  is Equation 2.8 divided by Equation 2.7. We get

$$\prod_{i=1}^k \frac{\int_0^{t+A} [F(x_i+u) - F(u)]\theta(t-u)du}{\int_0^{t+A} (1-F(s))\theta(t-s)ds}.$$

This completes the proof. ■

The above lemma implies that for an  $NPP(\theta(\cdot))$  sales process, we can use the following for  $Q(t)$ :

$$Q(t) = \frac{\int_0^{\infty} (F(t+u) - F(u))\theta(-u)du}{\int_0^{\infty} (1-F(s))\theta(-s)ds}. \quad (2.9)$$

It is easy to check that Equation 2.9 reduces to Equation 2.6 if  $\theta(t) = \theta$  for all values of  $t$ .

In the next section, we will need the moments of  $X(t)$ ,  $X^n(t)$ , and  $X^o(t)$  in the computation of the moments of  $R(t)$ . Since  $X^n(t)$  and  $X^o(t)$  are independent of each other, we compute the moments of  $X(t)$  as:

$$x(t) = x^n(t) + x^o(t), \text{ and} \quad (2.10)$$

$$x_2(t) = x_2^n(t) + x_2^o(t) + 2x^n(t)x^o(t). \quad (2.11)$$

## 2.4 Differential Equations for Moments of $R(t)$

We will consider two cases:  $X^o(t)$  is unknown during  $[0, T]$  (here we use the distribution discussed in Section 2.3.2), and  $X^o(t)$  is known in its entirety during  $[0, T]$ . We cover the former case here and the latter case in Section 3.3. In the results that follow, we will need expressions for  $E[\Delta_h S(t)]$  and  $E[\Delta_h D(t)]$ , where  $h$  is small. Since  $\{S(t), t \geq 0\}$  is an  $NPP(\theta(\cdot))$ , we know that

$$E[\Delta_h S(t)] = E \left[ \int_{u=t}^{t+h} \theta(u) du \right] = \theta(t)h + o(h).$$

The stochastic process  $\{D(t), t \geq 0\}$  is a random sum of random variables. Let  $N(t)$  represent the number of claims from time 0 to  $t$ . For a given sample path of  $\{X(t), t \geq 0\}$ ,  $\{N(t), t \geq 0\}$  is an  $NPP(\lambda X(\cdot))$ . The repair costs are i.i.d. with common mean  $E[D]$  and second moment  $E[D^2]$ . Therefore,

$$\begin{aligned} E[\Delta_h D(t)] &= E[\Delta_h N(t)]E[D] = E \left[ \int_{u=t}^{t+h} \lambda X(u) du \right] E[D] \\ &= \lambda E[X(t)h]E[D] + o(h) = \lambda x(t)E[D]h + o(h). \end{aligned}$$

Similarly,

$$\begin{aligned}
\text{Var}(\Delta_h D(t)) &= E[\Delta_h N(t)]\text{Var}(D) + E^2[D]\text{Var}(\Delta_h N(t)) \\
&= \lambda x(t)h [E[D^2] - E^2[D]] + \lambda x(t)E^2[D]h + o(h) \\
&= \lambda x(t)E[D^2]h + o(h).
\end{aligned}$$

We next introduce notation for item failure rates. Consider an arbitrary item that was sold in  $[0, t]$ . Let  $U$  be its time of sale. Then,  $U$  has cdf

$$P(U \leq u) = \frac{\Theta(u)}{\Theta(t)}, \quad 0 \leq u \leq t,$$

$$\text{where } \Theta(t) = \int_0^t \theta(s) ds.$$

Let  $W$  represent the warranty period random variable. Then, the probability that the item is under warranty at time  $t$  is  $P(U + W > t)$ . Given that it is under warranty at time  $t$ , the probability that its warranty expires in  $[t, t + \delta]$  is given by

$$h^n(t)\delta = \frac{f_{U+W}(t)}{1 - F_{U+W}(t)}\delta + o(\delta). \quad (2.12)$$

Since the warranty periods are i.i.d. and the sales process is an *NPP*, we see that the items behave independently of each other. Hence, if  $X^n(t) = i$ , the probability that a single item fails in  $[t, t + \delta]$  is  $ih^n(t)\delta + o(\delta)$ . We do a similar analysis for the items under warranty at time 0. We assume that the remaining lifetimes are unknown but are independent of each other. The probability that an item is still under warranty at time  $t$  is  $1 - Q(t)$ . The probability that its warranty expires in  $[t, t + \delta]$  is given by

$$h^o(t)\delta = \frac{-Q'(t)}{1 - Q(t)}\delta + o(\delta). \quad (2.13)$$

For convenience, we define

$$H^i(t) = \int_{s=0}^t h^i(s) ds, \text{ for } i = n, o.$$

We are now ready to compute the moments of  $R(t)$ .

**Theorem 2** *Let  $r(t) = E[R(t)]$ . Then,*

$$\frac{dr(t)}{dt} = \alpha r(t) + c\theta(t) - \lambda E[D]x(t), \quad (2.14)$$

*with initial condition  $r(0) = R_0$ .*

**Proof.** We look at the change in the reserve from time  $t$  to time  $t + h$ , where  $h$  is small.

We have

$$R(t + h) - R(t) = (e^{\alpha h} - 1)R(t) + c[\Delta_h S(t)] - [\Delta_h D(t)] + o(h).$$

Taking expectation on both sides, we get

$$\begin{aligned} r(t + h) - r(t) &= (e^{\alpha h} - 1)r(t) + c(E[\Delta_h S(t)]) - (E[\Delta_h D(t)]) + o(h), \\ &= (\alpha h + o(h))r(t) + c(\theta(t)h + o(h)) - (\lambda E[D]x(t)h + o(h)) \end{aligned}$$

Diving by  $h$  and taking the limit as  $h \rightarrow 0$  yields Equation 2.14. ■

Deriving the differential equations for  $E[R^n(t)]$  and  $E[R^o(t)]$  is similar to Theorem 2.

**Theorem 3** *Let  $r^n(t) = E[R^n(t)]$  and  $r^o(t) = E[R^o(t)]$ . Then,*

$$\begin{aligned} \frac{dr^n(t)}{dt} &= \alpha r^n(t) + c\theta(t) - \lambda E[D]x^n(t), \\ \frac{dr^o(t)}{dt} &= \alpha r^o(t) - \lambda E[D]x^o(t), \end{aligned}$$

with initial conditions  $r^n(0) = 0$  and  $r^o(0) = R_0$ .

**Proof.** The definitions of  $R^n(t)$  and  $R^o(t)$  in Section 2.2 yield:

$$\begin{aligned} R^n(t+h) - R^n(t) &= (e^{\alpha h} - 1)R^n(t) + c[\Delta_h S(t)] - \Delta_h D^n(t) + o(h), \\ R^o(t+h) - R^o(t) &= (e^{\alpha h} - 1)R^o(t) - \Delta_h D^o(t) + o(h). \end{aligned}$$

The  $c$  term does not appear in the equation for  $R^o(t+h)$  since the revenue for sales is only generated by the new items. We apply the same techniques used in Theorem 1 to complete the result. ■

To derive the differential equation for the second moment, we first prove two lemmas.

**Lemma 3** *Let  $v(t) = E[R^o(t)X^o(t)]$ . Then*

$$\frac{dv(t)}{dt} = (\alpha - h^o(t))v(t) - \lambda E[D]x_2^o(t), \quad (2.15)$$

with initial condition  $v(0) = X(0)R_0$ , and  $h^o(t)$  is as in Equation 2.13.

**Proof.** We again look at  $v(t+h) - v(t)$  and take limits as  $h \rightarrow 0$ .

$$\begin{aligned} v(t+h) &= E[R^o(t+h)X^o(t+h)] \\ &= E[(e^{\alpha h}R^o(t) - \Delta_h D^o(t))(X^o(t) + \Delta_h X^o(t)) + o(h)], \\ v(t+h) - v(t) &= e^{\alpha h}E[R^o(t)\Delta_h X^o(t)] + (e^{\alpha h} - 1)E[R^o(t)X^o(t)] - E[\Delta_h D^o(t)\Delta_h X^o(t)] - \\ &\quad E[\Delta_h D^o(t)X^o(t)] + o(h), \\ v(t+h) - v(t) &= (1 + \alpha h + o(h))E[R^o(t)\Delta_h X^o(t)] + (\alpha h + o(h))v(t) - \\ &\quad E[\Delta_h D^o(t)\Delta_h X^o(t)] - E[\Delta_h D^o(t)X^o(t)] + o(h). \end{aligned} \quad (2.16)$$

We can find  $E[R^o(t)\Delta_h X^o(t)]$ ,  $E[\Delta_h D^o(t)X^o(t)]$ , and  $E[\Delta_h D^o(t)\Delta_h X^o(t)]$  by conditioning on  $X^o(t)$ .

$$\begin{aligned}
E[R^o(t)\Delta_h X^o(t)] &= \sum_i E[R^o(t)\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i E[R^o(t)|X^o(t) = i]E[\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i -E[R^o(t)|X^o(t) = i]ih^o(t)P[X^o(t) = i]h + o(h) \\
&= -h^o(t) \sum_i E[R^o(t)i|X^o(t) = i]P[X^o(t) = i]h + o(h) \\
&= -h^o(t)v(t)h + o(h).
\end{aligned}$$

$$\begin{aligned}
E[\Delta_h D^o(t)X^o(t)] &= \sum_i E[\Delta_h D^o(t)X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i iE[\Delta_h D^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i i^2\lambda P[X^o(t) = i]E[D]h + o(h) \\
&= \lambda E[D]x_2^o(t)h + o(h).
\end{aligned}$$

$$\begin{aligned}
E[\Delta_h D^o(t)\Delta_h X^o(t)] &= \sum_i E[\Delta_h D^o(t)\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i E[\Delta_h D^o(t)|X^o(t) = i]E[\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i -i\lambda ih^o(t)E[D]P[X^o(t) = i]h^2 + o(h) \\
&= -\lambda E[D]x_2^o(t)h^o(t)h^2 + o(h) = o(h).
\end{aligned}$$

Plugging these three expressions into Equation 2.16, dividing by  $h$ , and taking the limit as  $h \rightarrow 0$  completes the result. ■

**Lemma 4** Let  $u(t) = E[R(t)X(t)]$ . Then

$$\begin{aligned} \frac{du(t)}{dt} &= (\alpha - h^n(t))u(t) + c\theta(t)(x(t) + 1) - \lambda E[D]x_2(t) + \theta(t)r(t) + \\ &\quad (h^n(t) - h^o(t)) * (r^n(t)x^o(t) + v(t)), \end{aligned} \quad (2.17)$$

with initial condition  $u(0) = X(0)R_0$ , and where  $v(t)$  satisfies Lemma 3,  $h^n(t)$  satisfies Equation 2.12,  $h^o(t)$  satisfies Equation 2.13, and  $r^n(t)$  satisfies Theorem 3.

**Proof.** We proceed as in Lemma 3. We have

$$\begin{aligned} u(t+h) &= E[R(t+h)X(t+h)] \\ &= E[(e^{\alpha h}R(t) + c\Delta_h S(t) - \Delta_h D(t))(X(t) + \Delta_h X(t)) + o(h)], \\ u(t+h) - u(t) &= E[(e^{\alpha h} - 1)R(t)X(t)] + cE[\Delta_h S(t)X(t)] - E[\Delta_h D(t)X(t)] + \\ &\quad E[e^{\alpha h}R(t)\Delta_h X(t)] + cE[\Delta_h S(t)\Delta_h X(t)] - E[\Delta_h D(t)\Delta_h X(t)] + o(h). \end{aligned} \quad (2.18)$$

We investigate each term on the right hand side of Equation 2.18 below:

(1)  $E[(e^{\alpha h} - 1)R(t)X(t)] = (\alpha h + o(h))u(t)$ .

(2)  $cE[\Delta_h S(t)X(t)] = cE[\Delta_h S(t)]E[X(t)] = c\theta(t)x(t)h + o(h)$  (The number of additional sales from  $t$  to  $t+h$  is independent of the number of items under warranty at time  $t$ ).

(3) We calculate  $E[\Delta_h D(t)X(t)]$  by conditioning on  $X(t)$ :

$$\begin{aligned} \sum_k E[\Delta_h D(t)X(t)|X(t) = k]P[X(t) = k] &= \sum_k kE[\Delta_h D(t)|X(t) = k]P[X(t) = k] \\ &= \sum_k \lambda k^2 P[X(t) = k]E[D]h + o(h) = \lambda E[D]x_2(t)h + o(h). \end{aligned}$$

(4)  $E[R(t)\Delta_h X(t)] = E[R(t)\Delta_h X^o(t)] + E[R(t)\Delta_h X^n(t)]$ .

We calculate  $E[R(t)\Delta_h X^o(t)]$  by conditioning on  $X^o(t)$ :

$$E[R(t)\Delta_h X^o(t)] = \sum_i E[R(t)\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i]$$

$$\begin{aligned}
&= \sum_i E[R(t)|X^o(t) = i]E[\Delta_h X^o(t)|X^o(t) = i]P[X^o(t) = i] \\
&= \sum_i -ih^o(t)E[R(t)|X^o(t) = i]P[X^o(t) = i]h + o(h) \\
&= -h^o(t) \sum_i E[R(t)i|X^o(t) = i]P[X^o(t) = i]h + o(h) \\
&= -h^o(t)E[R(t)X^o(t)]h + o(h) \\
&= -h^o(t)E[(R^n(t) + R^o(t))X^o(t)]h + o(h) \\
&= -h^o(t)(E[R^n(t)X^o(t)] + E[R^o(t)X^o(t)])h + o(h) \\
&= -h^o(t)(r^n(t)x^o(t) + v(t))h + o(h).
\end{aligned}$$

We calculate  $E[R(t)\Delta_h X^n(t)]$  by conditioning on  $X^n(t)$ :

$$\begin{aligned}
E[R(t)\Delta_h X^n(t)] &= \sum_i E[R(t)\Delta_h X^n(t)|X^n(t) = i]P[X^n(t) = i] \\
&= \sum_i E[R(t)|X^n(t) = i]E[\Delta_h X^n(t)|X^n(t) = i]P[X^n(t) = i] \\
&= \sum_i E[R(t)|X^n(t) = i](\theta(t)h - ih^n(t)h)P[X^n(t) = i] + o(h) \\
&= \theta(t)h \sum_i E[R(t)|X^n(t) = i]P[X^n(t) = i] \\
&\quad - h^n(t)h \sum_i E[R(t)i|X^n(t) = i]P[X^n(t) = i] + o(h) \\
&= \theta(t)r(t)h - h^n(t)E[R(t)X^n(t)]h + o(h) \\
&= \theta(t)r(t)h - h^n(t)(E[R(t)X(t)] - E[R(t)X^o(t)])h + o(h) \\
&= \theta(t)r(t)h - h^n(t)(u(t) - r^n(t)x^o(t) - v(t))h + o(h) \\
&= \theta(t)r(t)h - h^n(t)u(t)h + h^n(t)(r^n(t)x^o(t) + v(t))h + o(h).
\end{aligned}$$

(5) To calculate  $E[\Delta_h S(t)\Delta_h X(t)]$ , we must consider the dependence of  $S(t)$  and  $X(t)$ . If there is a sale in  $\Delta_h t$ , then both  $\Delta_h S(t)$  and  $\Delta_h X(t)$  are 1. This happens with probability  $\theta(t)h + o(h)$ . If there is an expiration, then  $\Delta_h X(t)$  is  $-1$  while  $\Delta_h S(t)$  is 0 (hence their

product is 0). Therefore,

$$cE[\Delta_h S(t)\Delta_h X(t)] = c\theta(t)h + o(h).$$

(6) We calculate  $E[\Delta_h D(t)\Delta_h X(t)]$  by conditioning on  $\Delta_h X(t)$ :

$$\begin{aligned} & \sum_k E[\Delta_h D(t)\Delta_h X(t)|\Delta_h X(t) = k]P[\Delta_h X(t) = k] \\ &= \sum_k kE[\Delta_h D(t)|\Delta_h X(t) = k]P[\Delta_h X(t) = k] \\ &= \sum_k \frac{\lambda k^2 E[D]}{2} P[\Delta_h X(t) = k]h + o(h) \\ &= \frac{\lambda E[D]}{2} h(\theta^2(t)h^2 + \theta(t)h) + o(h) = o(h). \end{aligned}$$

To complete the proof, we substitute the expressions found in (1)-(6) into Equation 2.18, divide by  $h$ , and take the limit as  $h \rightarrow 0$ . ■

We are now ready to provide the differential equation for the second moment of  $R(t)$ .

**Theorem 4** *Let  $r_2(t) = E[R^2(t)]$  and  $r(t), u(t), v(t)$ , and  $r^n(t)$  be defined as before. Then*

$$\frac{dr_2(t)}{dt} = 2\alpha r_2(t) + c^2\theta(t) + \lambda E[D^2]x(t) + 2c\theta(t)r(t) - 2\lambda E[D]u(t), \quad (2.19)$$

where  $r_2(0) = R_0^2$ .

**Proof.** We proceed as in Lemma 3. We have

$$R(t+h) = e^{\alpha h}R(t) + c\Delta_h S(t) - \Delta_h D(t) + o(h).$$

Squaring both sides and rearranging terms, we get

$$R^2(t+h) - R^2(t) = (e^{2\alpha h} - 1)R^2(t) + c^2(\Delta_h S(t))^2 + (\Delta_h D(t))^2 + 2ce^{\alpha h}R(t)\Delta_h S(t) -$$

$$2e^{\alpha h}R(t)\Delta_h D(t) - 2c\Delta_h S(t)\Delta_h D(t) + o(h).$$

Taking expectation, we obtain

$$\begin{aligned} r_2(t+h) - r_2(t) &= (e^{2\alpha h} - 1)r_2(t) + c^2 E[\Delta_h S(t)]^2 + E[\Delta_h D(t)]^2 + 2ce^{\alpha h} E[R(t)\Delta_h S(t)] - \\ & 2e^{\alpha h} E[R(t)\Delta_h D(t)] - 2cE[\Delta_h S(t)\Delta_h D(t)] + o(h). \end{aligned} \quad (2.20)$$

We investigate each term of the right hand side of Equation 2.20 below:

- (1)  $(e^{2\alpha h} - 1)r_2(t) = (2\alpha h + o(h))r_2(t)$ .
- (2)  $c^2 E[\Delta_h S(t)]^2 = c^2 (\theta(t)h + \theta^2(t)h^2) + o(h) = c^2\theta(t)h + o(h)$ .
- (3) The mean and variance of  $\Delta_h D(t)$  was computed prior to Lemma 1. We have

$$\begin{aligned} E[\Delta_h D(t)]^2 &= Var(\Delta_h D(t)) + E^2[\Delta_h D(t)] \\ &= \lambda E[D^2]x(t)h + (\lambda E[D]x(t)h)^2 + o(h) \\ &= \lambda E[D^2]x(t)h + o(h). \end{aligned}$$

- (4)  $R(t)$  is independent of  $\Delta_h S(t)$  since future sales do not impact the current reserve level. Therefore,  $E[R(t)\Delta_h S(t)] = E[R(t)]E[\Delta_h S(t)] = \theta(t)r(t)h + o(h)$ .

- (5) We calculate  $E[R(t)\Delta_h D(t)]$  by conditioning on  $X(t)$ :

$$\begin{aligned} & \sum_k E[R(t)\Delta_h D(t)|X(t) = k]P[X(t) = k] \\ &= \sum_k E[R(t)|X(t) = k]E[\Delta_h D(t)|X(t) = k]P[X(t) = k] \\ &= \sum_k E[R(t)|X(t) = k]\lambda k P[X(t) = k]E[D]h + o(h) \\ &= \lambda E[R(t)X(t)]E[D]h + o(h) = \lambda E[D]u(t)h + o(h). \end{aligned}$$

(6) We calculate  $E[\Delta_h S(t)\Delta_h D(t)]$  by conditioning on  $X(t)$ :

$$\begin{aligned}
&= \sum_k E[\Delta_h S(t)\Delta_h D(t)|X(t) = k]P[X(t) = k] \\
&= \sum_k E[\Delta_h S(t)|X(t) = k]E[\Delta_h D(t)|X(t) = k]P[X(t) = k] \\
&= \sum_k \theta(t)h * \lambda E[D]kP[X(t) = k]h + o(h) \\
&= \theta(t)\lambda E[D]x(t)h^2 + o(h) = o(h).
\end{aligned}$$

To complete the proof, we substitute the expressions found in (1)-(6) into Equation 2.20, divide by  $h$ , and take the limit as  $h \rightarrow 0$ . ■

Theorem 4 provides a system of equations for the first and second moments of  $R(t)$ . We can solve this linear system analytically by solving the equations in the following order:  $r(t), r^a(t), v(t), u(t), r^2(t)$ . This is because each differential equation only uses functions of  $t$  that are either known or previously solved in the system – this is known as a triangular system. We can also use a software package, such as MATLAB, to solve the system numerically. Clearly, we can use the solution of the system to find the variance by applying the formula

$$Var(R(t)) = r_2(t) - r^2(t).$$

In the next section, we present the general solution to this system and some examples for simple warranty distributions.

## 2.5 Solution for Moments of $R(t)$

We now provide the solution to the differential equations derived in Section 2.4. For a complete solution, it is necessary to know the functions  $x(t), x_2(t), x^n(t), x^o(t), h^o(t), h^n(t), H^o(t),$  and  $H^n(t)$ . These expressions, defined in Equations 2.1 – 2.13 of Section 2.3, depend only on the warranty distribution  $F(\cdot)$  and the given sales rate  $\theta(\cdot)$ .

**Theorem 5** Let  $r(t), r^n(t), v(t), u(t)$ , and  $r_2(t)$  be defined as in Section 2.4. Then, we have

$$r(t) = R_0 e^{\alpha t} + e^{\alpha t} \int_{s=0}^t e^{-\alpha s} (c\theta(s) - \lambda E[D]x(s)) ds, \quad (2.21)$$

$$r^n(t) = e^{\alpha t} \int_{s=0}^t e^{-\alpha s} (c\theta(s) - \lambda E[D]x^n(s)) ds,$$

$$v(t) = X(0)R_0 e^{\alpha t - H^o(t)} - \lambda e^{\alpha t - H^o(t)} E[D] \int_{s=0}^t x_2^o(s) e^{-\alpha s + H^o(s)} ds,$$

$$u(t) = X(0)R_0 e^{\alpha t - H^n(t)} + e^{\alpha t - H^n(t)} \int_{s=0}^t e^{-\alpha s + H^n(s)} [(h^n(s) - h^o(s)) (r^n(s)x^o(s) + v(s)) + c\theta(s)(x(s) + 1) - \lambda x_2(s)E[D] + \theta(s)r(s)] ds,$$

$$r_2(t) = R_0^2 e^{2\alpha t} + e^{2\alpha t} \int_{s=0}^t e^{-2\alpha s} [c^2\theta(s) + \lambda x_2(s)E[D^2] + 2c\theta(s)r(s) - 2\lambda E[D]u(s)] ds.$$

**Proof.** We apply the techniques to solve linear differential equations for each of  $r(t)$ ,  $r^n(t)$ ,  $v(t)$ ,  $u(t)$ , and  $r_2(t)$ . For the sake of brevity, we omit the details. ■

### 2.5.1 Example: Constant Warranty Period and Constant Sales Rate

We provide the solution for the first and second moments of  $R(t)$  for the example of a constant warranty period  $w$  and a constant sales rate function  $\theta(t) = \theta$  for all  $t \geq 0$ . The second moment  $r_2(t)$  is quite complex, so we instead provide the variance of  $R(t)$ .

First, we provide the moments of  $X^n(t)$  and  $X^o(t)$ , and the expressions for  $F(t)$ ,  $h^n(t)$ , and  $h^o(t)$ . We assume that the remaining warranty periods of the items sold prior to time 0 is unknown. We apply the result from Section 2.3.2 to determine the distribution of  $X^n(t)$ . We have

$$F(t) = \begin{cases} 0, & 0 \leq t < w \\ 1, & t \geq w \end{cases}, \text{ and}$$

$$\int_0^t \theta(t-u)[1-F(u)] du = \theta \min(t, w).$$

Applying the formulas for the first and second moments of  $X^n(t)$  and  $X^o(t)$  yields

$$x^n(t) = \theta \min(t, w),$$

$$x_2^n(t) = \theta \min(t, w) + \theta^2 \min(t^2, w^2),$$

$$x^o(t) = \begin{cases} X(0) \left(\frac{w-t}{w}\right) & 0 \leq t < w \\ 0 & t \geq w \end{cases},$$

$$x_2^o(t) = \begin{cases} X(0) \left(\frac{(w-t)t}{w^2}\right) + \left(X(0) \left(\frac{w-t}{w}\right)\right)^2 & 0 \leq t < w \\ 0 & t \geq w \end{cases}.$$

Using the expressions for  $h^n(t)$  and  $h^o(t)$  defined in 2.12 and 2.13, we obtain

$$h^n(t) = \begin{cases} 0, & t < w \\ \frac{1}{w} & t \geq w \end{cases}, \text{ and}$$

$$h^o(t) = \begin{cases} \frac{1}{w-t}, & t < w \\ 0 & t \geq w \end{cases}.$$

We consider two cases.

**Case 1.**  $0 \leq t \leq w$ . Here, we have

$$r(t) = A_0 + A_1 t + A_2 e^{\alpha t}, \text{ and}$$

$$\text{Var}(R(t)) = C_0 + C_1 t + C_2 t^2 + C_3 e^{\alpha t} + C_4 t e^{\alpha t} + C_5 e^{2\alpha t},$$

where

$$\begin{aligned}
A_0 &= \frac{1}{\alpha^2} \left( \lambda E[D]\theta + \lambda E[D]X(0)\alpha - c\theta\alpha - \frac{\lambda E[D]X(0)}{w} \right), \\
A_1 &= \frac{\lambda E[D]}{\alpha w} (\theta w - X(0)), \\
A_2 &= -\frac{1}{\alpha^2} \left( \lambda E[D]\theta + \lambda E[D]X(0)\alpha - \frac{\lambda E[D]X(0)}{w} - c\theta\alpha - R_0\alpha^2 \right), \\
C_0 &= \frac{1}{4\alpha^4 w^2} (6\lambda^2\theta E^2[D]\alpha w^2 - 4\lambda^2 X(0)E^2[D] - 4\lambda c\theta E[D]\alpha^2 w^2 + 6\lambda^2 X(0)E^2[D]\alpha w \\
&\quad - \lambda\theta E[D^2]\alpha^2 w^2 + \lambda X(0)E[D^2]\alpha^2 w - 2\lambda X(0)E[D^2]\alpha^3 w^2 - 2c^2\theta\alpha^3 w^2), \\
C_1 &= \frac{1}{2\alpha^3 w^2} (\lambda X(0)E[D^2]\alpha^2 w - 4\lambda^2 X(0)E^2[D] + 2\lambda^2\theta E^2[D]\alpha w^2 \\
&\quad + 2\lambda^2 X(0)E^2[D]\alpha w - \lambda\theta E[D^2]\alpha^2 w^2), \\
C_2 &= -\frac{\lambda^2 X(0)E^2[D]}{\alpha^2 w^2}, \\
C_3 &= \frac{2E[D]}{\alpha^4 w^2} (\lambda c\theta\alpha^2 w^2 + \lambda^2 X(0)E[D] - \lambda^2\theta E[D]\alpha w^2 - \lambda^2 X(0)E[D]\alpha w), \\
C_4 &= \frac{2\lambda^2 X(0)E^2[D]}{\alpha^3 w^2}, \text{ and} \\
C_5 &= \frac{1}{4\alpha^4 w^2} (\lambda\theta E[D^2]\alpha^2 w^2 + 2c^2\theta\alpha^3 w^2 + 2\lambda X(0)E[D^2]\alpha^3 w^2 + 2\lambda^2\theta E^2[D]\alpha w^2 \\
&\quad - \lambda X(0)E[D^2]\alpha^2 w - 4\lambda^2 X(0)E^2[D] + 2\lambda^2 X(0)E^2[D]\alpha w - 4\lambda c\theta E[D]\alpha^2 w^2).
\end{aligned}$$

**Case 2.**  $t > w$ . In this case, there are no longer any old items under warranty. Therefore, we have

$$\begin{aligned}
r(t) &= A_0 + A_1(t - w) + A_2 e^{\alpha(t-w)} + r(w), \text{ and} \\
Var(R(t)) &= C_0 + C_1(t - w) + C_2 e^{\alpha(t-w)} + C_3 e^{2\alpha(t-w)} + Var(R(w)),
\end{aligned}$$

where

$$\begin{aligned}
A_0 &= \frac{\theta}{\alpha^2} (\lambda E[D] - c\alpha), \\
A_1 &= \frac{\theta \lambda E[D]\alpha}{\alpha^2},
\end{aligned}$$

$$\begin{aligned}
A_2 &= \frac{1}{\alpha^2} (c\theta\alpha + R_0\alpha^2 - \lambda\theta E[D]), \\
C_0 &= \frac{1}{4\alpha^3} (6\lambda^2\theta E^2[D] - 4\lambda c\theta E[D]\alpha - 2c^2\theta\alpha^2 - \lambda\theta E[D^2]\alpha), \\
C_1 &= \frac{1}{2\alpha^2} (2\lambda^2\theta E^2[D] - \lambda\theta E[D^2]\alpha), \\
C_2 &= \frac{2}{\alpha^3} (\lambda c\theta E[D]\alpha - \lambda^2\theta E^2[D]), \text{ and} \\
C_3 &= \frac{1}{4\alpha^3} (\lambda\theta E[D^2]\alpha + 2c^2\theta\alpha^2 - 4\lambda c\theta E[D]\alpha + 2\lambda^2 E[D^2]\theta).
\end{aligned}$$

## 2.6 Deciding the Values of $c$ and $R_0$

The manufacturer must decide on the values of  $c$  and  $R_0$  at the beginning of a new period, with the goal of satisfying all of the warranty costs in the period while remaining above some target  $B > 0$  with some prespecified probability  $1 - \beta$ . Of course, setting artificially high values of  $c$  and  $R_0$  will achieve this goal, but this will tie up excess money that may be used for other business interests. In this section, we suggest criteria that the manufacturer may use as a basis for its decision.

### 2.6.1 Distribution of $R(t)$

We first point out that the variance of  $R(t)$  is independent of the initial reserve level,  $R_0$ . Let  $S_i$  be the time of the  $i$ th sale, and let  $Y_j$  be the time of the  $j$ th failure (with repair cost  $D_i$ ). Then, the reserve at time  $t$  can be computed as:

$$R(t) = R_0 e^{\alpha t} + c \sum_{S_i \leq t} e^{\alpha(t-S_i)} - \sum_{Y_i \leq t} D_i e^{\alpha(t-Y_i)}. \quad (2.22)$$

Note that the last 2 terms of Equation 2.22 are the random components of  $R(t)$  and do not contain  $R_0$ . Therefore, the variance is independent of  $R_0$  and only depends on the variable  $c$ ; we use this fact in deriving a heuristic to determine the values of  $c$  and  $R_0$  in the next section.

In general, the distribution of  $R(t)$  is difficult to determine. However, if the sales process is a non-homogeneous Poisson Process, the distribution is asymptotic Normal as the number of sales becomes large. A proof of this can be obtained as a minor extension of the result given by Ja [17], which is based on [18]. Therefore, we use the Normal distribution as an approximation. In our simulations, the values of  $R(t)$  seem to follow a Normal distribution, especially for large  $t$ . We show an example in the next section.

Let  $\Phi(\cdot)$  be the standard Normal cumulative distribution function (zero mean and variance one) and let  $z_\beta$  be the number such that  $\Phi(z_\beta) = 1 - \beta$ . Therefore, at a given point of  $t$ , approximately  $100(1 - \beta)\%$  of sample paths of  $R(t)$  will remain above  $r(t) - z_\beta\sqrt{\text{Var}(R(t))}$ . The two examples in Figure 2.2 show examples of sample paths of  $R(t)$  and the  $100(1 - 2\beta)\%$  confidence bands at each value of  $t$ . In each graph, the jagged line represents a typical sample path of  $R(t)$ . The smooth central line is the plot of  $r(t)$ , while the outer lines are the plots of  $r(t) \pm z_\beta\sqrt{\text{Var}(R(t))}$ . In the left graph,  $\min_{t \in [0, T]} r(t) - z_\beta\sqrt{\text{Var}(R(t))}$  occurs at time  $T$ , while the minimum in the right graph occurs within  $(0, T)$ .

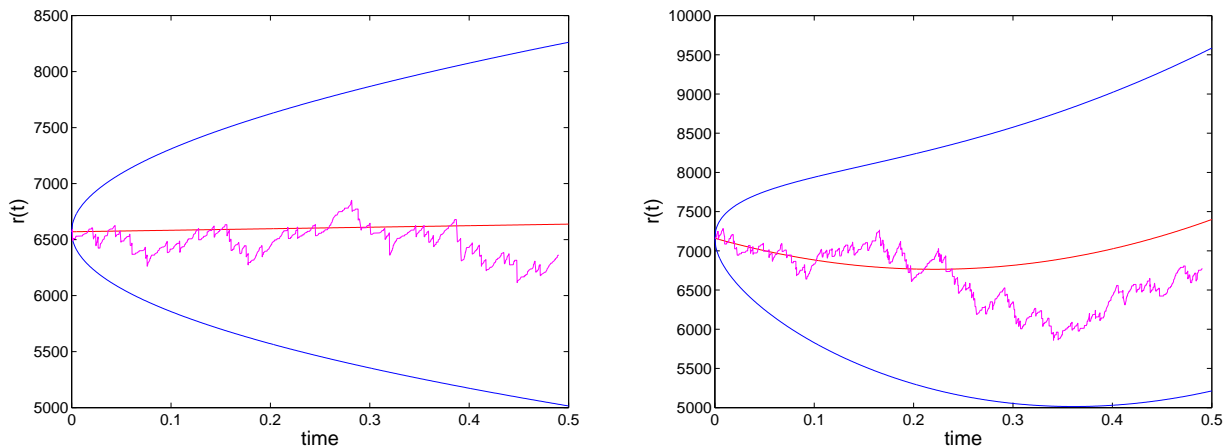


Figure 2.2: Examples of Confidence Bands for  $R(t)$

## 2.6.2 Heuristic for Deciding $c$ and $R_0$

The manufacturer has great flexibility in choosing the values of  $c$  and  $R_0$ . The original problem is to satisfy all of the warranty claims up to time  $T$  and remain above a target  $B$  with a given probability. That is, we wish to choose  $c$  and  $R_0$  so that

$$\psi(c, R_0) = 1 - P\left(\min_{t \in [0, T]} R(t) \geq B\right) \quad (2.23)$$

is bounded above by a prespecified probability. This problem of calculating a ruin probability is very complicated (see [3]). Since we cannot evaluate Equation 2.23 exactly, we develop an approximation using the result that the distribution of  $R(t)$  is approximately Normal. That is,

$$\frac{R(t) - r(t)}{\sqrt{\text{Var}(R(t))}} \approx N(0, 1),$$

as the number of sales increases to infinity. From this we see that

$$r(t) - z_\beta \sqrt{\text{Var}(R(t))} \geq B \implies P(R(t) \geq B) \geq 1 - \beta. \quad (2.24)$$

Now suppose that values  $c$  and  $R_0$  are chosen to satisfy

$$\min_{t \in [0, T]} \left( r(t) - z_\beta \sqrt{\text{Var}(R(t))} \right) = B. \quad (2.25)$$

Then, from 2.24 we see that this choice of  $(c, R_0)$  implies that

$$\min_{t \in [0, T]} P(R(t) \geq B) = 1 - \beta.$$

However, the ruin probability  $\psi(c, R_0)$  is greater than  $\beta$ , since

$$\psi(c, R_0) = 1 - P\left(\min_{t \in [0, T]} R(t) \geq B\right) \geq 1 - \min_{t \in [0, T]} P(R(t) \geq B) = \beta.$$

Thus  $\beta$  provides a lower bound on the ruin probability  $\psi(c, R_0)$ . Intuitively, the quantities  $\beta$  and  $\psi(c, R_0)$  appear to be related. We use simulation to help uncover a possible relationship between the two quantities.

Therefore, we estimate a parameter  $q_\beta$  so that

$$\min_{t \in [0, T]} \left( r(t) - q_\beta \sqrt{\text{Var}(R(t))} \right) = B \quad (2.26)$$

$$\implies P \left( \min_{t \in [0, T]} R(t) > B \right) \approx 1 - \beta.$$

Clearly we cannot guarantee that such a  $q_\beta$  will work in all possible situations, but we believe such an estimate will be instructive to a manufacturer.

Since there are many values of  $c$  and  $R_0$  which will satisfy Equation 2.25, we offer a heuristic to select one set. The heuristic assumes that we have an additional condition to satisfy: at time  $T$ , the expected reserve level is  $R_0 e^{\alpha T}$  (to account for accumulated interest). Therefore, we choose  $c$  so that

$$r(T) = R_0 e^{\alpha T}.$$

From Equation 2.21, we see that this is equivalent to solving the equation

$$e^{\alpha T} \int_{s=0}^T e^{-\alpha s} (c\theta(s) - \lambda E[D]x(s)) ds = 0. \quad (2.27)$$

This equation does not contain  $R_0$ . Rearranging Equation 2.27 to isolate the variable  $c$  yields

$$c = \frac{\lambda E[D] \int_{s=0}^T e^{-\alpha s} x(s) ds}{\int_{s=0}^T e^{-\alpha s} \theta(s) ds}. \quad (2.28)$$

We use this value of  $c$  in solving Equation 2.25. Since the left hand side of Equation 2.25 is a monotone increasing function of  $R_0$ , there is a unique value of  $R_0$  that satisfies the equation.

We used simulation to estimate  $q_\beta$  for different values of  $\beta$ . For ten different sets of parameter values and distributions, we ran 5000 trials and recorded the minimum value of  $R(t)$  (and the occurrence time  $t^*$ ) over  $[0, T]$  for each trial. We first recorded the time,  $t_B$ , when  $r(t) - z_\beta \sqrt{\text{Var}(R(t))}$  reached its minimum value over  $[0, T]$  for the given parameters. Then, for each simulated trial, we record the minimum value of the reserve level  $R(t^*)$  and the time of occurrence  $t^*$ . We select  $q$  so that the simulated reserve process  $R(t)$  lies above  $r(t) - q \sqrt{\text{Var}(R(t))}$  for all  $t \in [0, T]$ . The quantity  $q$  is given by the following formula:

$$q = \frac{r(t_B) - R(t^*)}{\sqrt{\text{Var}(R(t_B))}}.$$

We then computed the  $100(1 - \beta)$ th percentile of  $q$  for each parameter set  $i$  (call this  $q_{i\beta}$ ). Our suggested value of  $q_\beta$  for each value of  $\beta$  is  $\max_i q_{i\beta}$ . In our experience, the worst cases (large  $q_{i\beta}$ ) occurred when the minimum of  $r(t) - z_\beta \sqrt{\text{Var}(R(t))}$  did not occur at time  $T$ . However, the dispersion in  $q_{i\beta}$  for the different parameter sets was not large enough to consider different cases. For reference, we give the values of  $z_\beta$  to compare with our suggested values of  $q_\beta$ . We also give the range and standard deviation of  $q_{i\beta}$  to note the relative error in the estimate. The results are summarized in Table 2.1.

$\beta$	$z_\beta$	$q_\beta$	$\max_i q_{i\beta} - \min_i q_{i\beta}$	$\sqrt{\text{Var}(q_{i\beta})}$
.10	1.282	1.842	0.441	0.146
.05	1.645	2.197	0.414	0.152
.025	1.960	2.594	0.501	0.168
.01	2.326	3.059	0.492	0.174
.005	2.576	3.349	0.591	0.205
.001	3.090	4.163	0.814	0.275

Table 2.1: Suggested Values for  $q$

The values for  $q_{i\beta}$  in each of the simulation trials are not vastly different from  $z_\beta$ . For  $\beta > .01$ , the value of  $\max q_{i\beta} - \min q_{i\beta}$  was less than 0.5 and the standard deviation of  $q_{i\beta}$  was less than 0.2. From our experience, the value of  $q_{i\beta}$  is most affected by the variation in the repair costs and the initial number of items under warranty. While these estimates for  $q_\beta$  will not work for all problems, we observed that they were fairly robust for our simulations. We applied the heuristic with these values of  $q_\beta$  for other parameter sets and always had  $\{R(t) > B \forall t\}$  in at least  $100(1 - \beta)\%$  of the trials.

This heuristic has many advantages: it is easy to compute, provides stability to the expected reserve level from period to period, yields a unique answer, and performs well under simulation.

## 2.7 Numerical Computations

We dedicate this section to a numerical example. Consider a non-renewable, free-replacement warranty with constant period 1 year. All the items are independent and identical, with mean 0.1 failures per year and each replacement cost to the manufacturer is fixed at \$100. The sales process is a Poisson process with mean 1000/year. The interest rate on the account is 6% compounded continuously, and we consider a period  $T$  of one-half year. Some products that might have this structure are electronic devices (such as calculators) or small appliances (such as toasters or microwaves). More complex products, such as computers, have similar properties where repairs are "good as new".

Let  $E[C_W(\alpha)]$  be the expected total warranty cost discounted to present value for a single item. One option for the manufacturer is to contribute this amount to the reserve after each sale. We compute  $E[C_W(\alpha)]$  by the following formula (see Section 4.2 of [5]):

$$E[C_W(\alpha)] = E[D] \int_0^w e^{\alpha t} dM(t), \quad (2.29)$$

where  $M(\cdot)$  is the ordinary renewal function associated with the product failure distribution (here  $M(t) = \lambda t$ ). Substituting the numbers of the example yields  $E[C_W(\alpha)] = 9.71$ . We use this as a comparison for the recommended value of  $c$  obtained through the heuristic.

First, we illustrate the effect of  $X(0)$  on the mean of the reserve. Figure 2.3 plots  $r(t)$  for  $X(0) = \{500, 1000, 1500, 2000\}$ . Note that the expected number of items under warranty in steady state is given by  $\theta w = 1000$ . In each case, we set the reserve contribution from each sale to  $E[C_W(\alpha)] = 9.71$ .

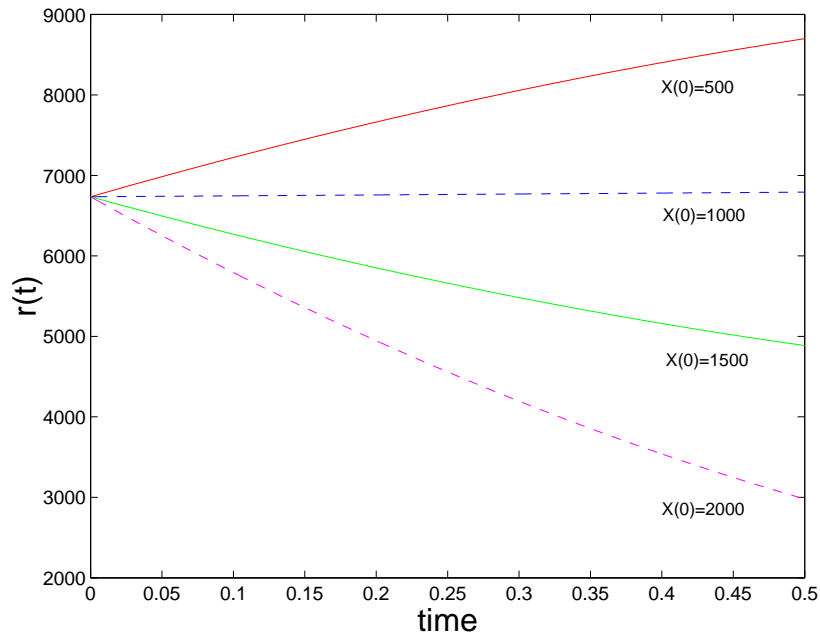


Figure 2.3: Expected Reserve for Various Values of  $X(0)$

This plot clearly shows that letting  $c = E[C_W(\alpha)]$  is very effective if  $X(0) \approx \theta w$ . However, if these two values are far apart, another value of  $c$  is recommended. In the instance when  $X(0) = 2000$ , it requires a value of  $c = 17.51$  to keep the expected reserve level at the end of the period equal to  $R_0 e^{\alpha T}$ . Conversely, a value of  $c = 6.24$  achieves the same goal when  $X(0) = 500$ .

Next, we illustrate the heuristic to determine  $c$  and  $R_0$  when there are 1500 items under warranty at time 0 and the remaining warranty periods of these items are unknown.

Suppose that the manufacturer must keep the reserve level above  $B = 5000$  for the entire period  $[0, T]$  with 95% probability. Plugging the parameters into Equation 2.28, we get  $c = 13.756$ . We see from Table 2.1 that  $q_{.05} = 2.197$ . We solve Equation 2.26 for  $R_0$ , obtaining a value of  $R_0 = 6734.8$ .

We ran 5000 simulated trials with the above parameters to illustrate the effectiveness of the heuristic and check for normality of  $R(t)$ . For reference this set of parameters is different from the 10 used to determine  $q$ . Of the 5000 trials, 223 (4.46%) of them fell below the target  $B = 5000$  at some point during the period  $[0, T]$ . This is slightly less than the target of 5%. We recorded the values of  $R(t)$  of each trial at time points  $t = 0.125, 0.25, 0.375$ , and  $0.5$ . We compare the average and standard deviation of the simulated trials with the theoretical values calculated in Section 2.5. Also, we give the  $p$ -value of the chi-squared test for a Normal distribution. We summarize the results in Table 2.2.

		Time $t$			
		0.125	0.25	0.375	0.5
Mean	Sim.	6658.2	6662.9	6757.9	6928.9
	Theor.	6668.6	6680.3	6770.5	6939.8
Standard Deviation	Sim.	453.5	641.8	775.9	886.7
	Theor.	454.8	636.7	772.1	882.9
$p$ -value of $\chi^2$ -test		$1.24 \cdot 10^{-7}$	0.137	0.259	0.720

Table 2.2: Simulation Results

As expected, the simulated mean and standard deviation at each time point match up with the theoretical mean and variance. Clearly, the  $p$ -value of the  $\chi^2$ -test increases with increasing  $t$ . Although normality is clearly violated at  $t = 0.125$ , it is accepted at  $t \geq 0.25$ . Since the distribution is asymptotic Normal with respect to increasing sales, we can be fairly confident that a large sales process will have an approximately Normal distribution. Our example has a modest sales rate of 1000/yr; we see more intensive sales processes lead to a Normal distribution much quicker.

# Chapter 3

## Warranty Reserve: Extensions

In this chapter we consider three separate extensions to our basic warranty reserve model:

- The reserve contribution after the  $j$ th sale is  $C_j$ , a random variable. (Section 3.1)
- There are multiple products for the manufacturer which use the same reserve fund. For each product  $i$ , the sales rate is  $\theta_i(\cdot)$ , the warranty period has cdf  $F_i(\cdot)$ , and the reserve contribution after each sale of product  $i$  is a constant  $c_i$ . (Section 3.2)
- The entire sample path of  $X^o(t)$  is known during  $[0, T]$ . This case arises if the manufacturer has access to the time of sale and the length of the warranty period of each item under warranty. (Section 3.3)

### 3.1 Random contribution to the reserve after each sale

Suppose that the contribution to the reserve after the  $i$ th sale is a random variable. We assume that the successive contributions  $\{C_i, i \geq 1\}$  are i.i.d and independent of the reserve level and the sales process. A possible reason for having a random contribution amount independent of the other relevant stochastic processes is when the product costs

a different amount in different distribution areas of the manufacturer. This assumption is more critical in application areas such as insurance models or pension funds. We can compute the moments of  $R(t)$  under this new assumption as in Section 2.4. Not surprisingly, the only effect on the system of differential equations is a change from  $c$  to  $E[C]$  and from  $c^2$  to  $E[C^2]$ .

**Theorem 6** *Let the reserve contributions after each sale be i.i.d random variables with common mean  $E[C]$  and second moment  $E[C^2]$ . Suppose they are independent of the reserve level and the sales process. Then,*

$$\begin{aligned}\frac{dr(t)}{dt} &= \alpha r(t) + E[C]\theta(t) - \lambda E[D]x(t), \\ \frac{dr^n(t)}{dt} &= \alpha r^n(t) + E[C]\theta(t) - \lambda E[D]x^n(t), \\ \frac{dv(t)}{dt} &= (\alpha - h^o(t))v(t) - \lambda E[D]x_2^o(t), \\ \frac{du(t)}{dt} &= (\alpha - h^n(t))u(t) + E[C]\theta(t)(x(t) + 1) - \lambda E[D]x_2(t) + \theta(t)r(t) \\ &\quad + (h^n(t) - h^o(t)) * (r^n(t)x^o(t) + v(t)), \\ \frac{dr_2(t)}{dt} &= 2\alpha r_2(t) + E[C^2]\theta(t) + \lambda E[D^2]x(t) + 2E[C]\theta(t)r(t) - 2\lambda E[D]u(t).\end{aligned}$$

**Proof.** We recalculate each term in the derivations of the equations where the term  $c$  originally occurred. We use the assumption that the contribution is independent of the reserve level and the sales process to write  $E[C_i M(t)] = E[C]E[M(t)]$  and  $E[C_i^2 M(t)] = E[C^2]E[M(t)]$ , where  $M(t)$  is a stochastic process independent of  $C_i$ . This occurs twice in the derivations of  $\frac{dr(t)}{dt}$ ,  $\frac{dr^n(t)}{dt}$ , and  $\frac{du(t)}{dt}$ , and three times in  $\frac{dr_2(t)}{dt}$ . ■

## 3.2 Multiple products using a single reserve

Suppose that a manufacturer manages warranties for  $k$  products which may be  $k$  different products, the same product with  $k$  different warranty policies, or a combination of the

two. It may be desirable for the manufacturer to have a single warranty reserve for all of these products. We can apply our previous results to handle this situation. For each product  $i$ , we assume that the sales process is an  $NPP(\theta_i(\cdot))$ , the failure rate is exponential with rate  $\lambda_i$ , the contribution to the reserve after each sale is  $c_i$ , and each of these are independent of the other products. We track the total reserve contribution from each of the products and denote this as  $R_i(t)$ . The mean and variance of  $R_i(t)$  is computed from the results in section 2.4.

Since each of the products are independent of each other, the total expected reserve  $E[R(t)]$  can be computed as a sum of the independent expected reserves of each product and the total variance  $Var(R(t))$  is the sum of the individual variances of each product:

$$E[R(t)] = E \left[ \sum_{i=1}^k R_i(t) \right] = \sum_{i=1}^k E[R_i(t)],$$

$$\sqrt{Var(R(t))} = \sqrt{\sum_{i=1}^k Var(R_i(t))}.$$

It is clearly beneficial to maintain a combined account rather than  $k$  separate ones, due to pooling of the risks involved. Mathematically, this is because the variances combine additively, so the standard deviation of the combined reserve is less than the sum of the standard deviation of  $k$  individual reserves.

### 3.3 $X^o(t)$ is known during $[0, T]$

The case where  $X^o(t)$  is known is much easier than when it is unknown. We still divide  $X(t)$  as

$$X(t) = X^o(t) + X^n(t),$$

with the change that  $X^o(t)$  is a deterministic quantity.

**Theorem 7** Let  $r(t) = E[R(t)]$ ,  $r^n(t) = E[R^n(t)]$ ,  $y(t) = E[R^n(t)X^n(t)]$ , and  $r_2(t) = E[R^2(t)]$ . Assume that  $\{X^o(t), 0 \leq t \leq T\}$  is known. Then

$$\begin{aligned}\frac{dr(t)}{dt} &= \alpha r(t) + c\theta(t) - \lambda E[D]x(t), \\ \frac{dr^n(t)}{dt} &= \alpha r^n(t) + c\theta(t) - \lambda E[D]x^n(t), \\ \frac{dy(t)}{dt} &= (\alpha - h^n(t))y(t) + c\theta(t)(1 + x^n(t)) + \theta(t)r^n(t) - \lambda E[D]x_2^n(t),\end{aligned}\quad (3.1)$$

$$\begin{aligned}\frac{dr_2(t)}{dt} &= 2\alpha r_2(t) + c^2\theta(t) + \lambda E[D^2]x(t) + 2c\theta(t)r(t) \\ &\quad - 2\lambda E[D][X^o(t)r(t) + r^o(t)x^n(t) + y(t)],\end{aligned}\quad (3.2)$$

where  $r(0) = R_0$ ,  $r^n(0) = 0$ ,  $u(0) = 0$ , and  $r_2(0) = R_0^2$ .

**Proof.** We apply the same proof technique as in Section 2.4. Most of the calculations are exactly the same; we will mention and omit these and just provide the changes.

The calculations for  $\frac{dr(t)}{dt}$  and  $\frac{dr^n(t)}{dt}$  are the same as in Theorems 2 and 3. Since  $\frac{dy(t)}{dt}$  is a new quantity, we provide the calculations.

$$\begin{aligned}y(t+h) &= E[R^n(t+h)X^n(t+h)] \\ &= E[(e^{\alpha h}R^n(t) + c\Delta_h S(t) - \Delta_h D(t))(X^n(t) + \Delta_h X^n(t)) + o(h)] \\ y(t+h) - y(t) &= E[(e^{\alpha h} - 1)R^n(t)X^n(t)] + cE[\Delta_h S(t)X^n(t)] - E[\Delta_h D^n(t)X^n(t)] + \\ &\quad E[e^{\alpha h}R^n(t)\Delta_h X^n(t)] + cE[\Delta_h S(t)\Delta_h X^n(t)] - E[\Delta_h D^n(t)\Delta_h X^n(t)] + o(h)\end{aligned}\quad (3.3)$$

We investigate each term on the right hand side of Equation 3.3. The expressions for  $E[(e^{\alpha h} - 1)R^n(t)X^n(t)]$ ,  $cE[\Delta_h S(t)X^n(t)]$ ,  $E[\Delta_h S(t)\Delta_h X^n(t)]$ , and  $E[\Delta_h D^n(t)\Delta_h X^n(t)]$  are computed very similarly to their counterparts from Lemma 4. The computations yield

$$E[(e^{\alpha h} - 1)R^n(t)X^n(t)] = (\alpha h + o(h))y(t),$$

$$\begin{aligned}
cE[\Delta_h S(t)X^n(t)] &= c\theta(t)x^n(t)h + o(h), \\
E[\Delta_h S(t)\Delta_h X^n(t)] &= c\theta(t)h + o(h), \text{ and} \\
E[\Delta_h D^n(t)\Delta_h X^n(t)] &= o(h)
\end{aligned}$$

We compute the other two quantities below:

(1) We calculate  $E[\Delta_h D^n(t)X^n(t)]$  by conditioning on  $X^n(t)$ :

$$\begin{aligned}
&\sum_k E[\Delta_h D^n(t)X^n(t)|X^n(t) = k]P[X^n(t) = k] = \sum_k kE[\Delta_h D^n(t)|X^n(t) = k]P[X^n(t) = k] \\
&= \sum_k \lambda k^2 P[X^n(t) = k]E[D]h + o(h) = \lambda E[D]x_2^n(t)h + o(h).
\end{aligned}$$

(2) We calculate  $E[R^n(t)\Delta_h X^n(t)]$  by conditioning on  $X^n(t)$ :

$$\begin{aligned}
E[R^n(t)\Delta_h X^n(t)] &= \sum_i E[R^n(t)\Delta_h X^n(t)|X^n(t) = i]P[X^n(t) = i] \\
&= \sum_i E[R^n(t)|X^n(t) = i]E[\Delta_h X^n(t)|X^n(t) = i]P[X^n(t) = i] \\
&= \theta(t)r^n(t)h - \sum_i ih^n(t)E[R^n(t)|X^n(t) = i]P[X^n(t) = i] + o(h) \\
&= \theta(t)r^n(t)h - h^n(t)E[R^n(t)X^n(t)]h + o(h) \\
&= \theta(t)r^n(t)h - h^n(t)y(t)h + o(h).
\end{aligned}$$

To obtain Equation 3.1, we substitute the expressions found in (1)-(6) into Equation 3.3, divide by  $h$ , and take the limit as  $h \rightarrow 0$ .

It remains to derive Equation 3.2. We proceed in the usual fashion, beginning with Equation 2.20 derived in Section 2.4. The only change from the derivation in Theorem 4 is in  $E[R(t)\Delta_h D(t)]$ ; we provide that below:

$$\begin{aligned}
E[R(t)\Delta_h D(t)] &= \sum_k E[R(t)\Delta_h D(t)|X(t) = k]P[X(t) = k] \\
&= \sum_k E[R(t)|X(t) = k]E[\Delta_h D(t)|X(t) = k]P[X(t) = k]
\end{aligned}$$

$$\begin{aligned}
&= \sum_k E[R(t)|X(t) = k] \lambda k P[X(t) = k] E[D] h + o(h) \\
&= \lambda E[R(t)X(t)] E[D] h + o(h) \\
&= \lambda E[D] [X^o(t)r(t) + r^o(t)x^n(t) + E[R^n(t)X^n(t)]] + o(h) \\
&= \lambda E[D] [X^o(t)r(t) + r^o(t)x^n(t) + y(t)] h + o(h).
\end{aligned}$$

Making this substitution, we obtain Equation 3.2, completing the proof. ■

A manufacturer might use both assumptions regarding  $X^o(t)$ . For example, if they obtain a new computer package that tracks every customer's warranty expiration date during  $[0, T]$ , they may use the original assumption in the period  $[0, T]$  but use their additional information in the period  $[T, 2T]$ . We leave as future work the case where the purchase times of the items sold before time 0 are known, but the remaining warranty periods are not.

# Chapter 4

## Outsourcing Prioritized Warranty Repairs

### 4.1 Overview

We now discuss the problem of outsourcing warranty repairs when items have priority in service. Consider a manufacturer that has a contract with  $V$  repair vendors for a fixed fee per repair. Specifically, vendor  $j$  charges the manufacturer  $c_j$  dollars for each repair made under warranty, independent of the type of repair and the priority of the item. The contract does not specify a minimum or maximum number of repairs. Usually this contract situation arises when the manufacturer provides the replacement parts and the vendor just executes the repairs. Another scenario is that the vendor charges the expected cost per repair over the duration of the contract. We only consider a closed population model, i.e., we assume the number of items under warranty at any given time is constant.

For most manufacturers, it is important to return some repaired items faster than others. One example is a company that makes large purchases with the manufacturer. They will expect a faster repair turnaround time than an individual customer. Occasion-

ally, the manufacturer will include a maximum repair turnaround time in the purchase contracts with a large purchaser. Also, the manufacturer might offer a choice of warranties that specify the repair turnaround time. For example, the standard product warranty guarantees a one week repair turnaround time but can be upgraded by the customer to a two day repair turnaround time. The length of these turnaround times require that some of the repairs take priority over others. We treat the case where each item belongs to one of  $m$  priority classes.

The manufacturer must assign each item to one of the  $V$  repair vendors at time 0. This is *static allocation*; the items are all assigned at a single time point. One example might be the sale of small electronic appliances. Typically, the warranty card inside the product packaging will have a phone number to contact for warranty repairs. When the manufacturer outsources the warranty repairs, this phone number might be a direct line to the repair vendor. In this case, the manufacturer assigns the items to a vendor at production time. Another possible assignment method is *dynamic allocation*; the manufacturer assigns an item to a repair vendor at the time of failure. This proves to be a very difficult problem, even when priorities are not considered (see [27]). We do not address this method here.

## 4.2 Notation and Assumptions

We give some notation for the repair outsourcing problem. There are  $m$  priority types, where type  $j$  has priority in service over all types  $i > j$ . The number of items of priority type  $i$  ( $i = 1, \dots, m$ ), is  $K_i$ , a constant. Note that

$$\sum_{i=1}^m K_i = K.$$

The  $K$  items must be allocated among the  $V$  vendors.

We assume that the lifetimes of items are i.i.d. random variable with mean  $1/\lambda$  (independent of the priority type), the  $j$ th vendor employs  $s_j$  servers to repair items, and the repair times are i.i.d. exponential random variables with parameter  $\mu_j$ . We point out that Bunday and Scraton [8] provide an invariance result for a  $G/M/r$  interference model that the steady state probabilities depend only on the mean failure time  $1/\lambda$  and not the failure distribution  $G(\cdot)$ . Therefore, since we are only concerned with long-run average cost, we need not assume that the lifetimes of items are exponentially distributed. We will assume that all information is perfect, meaning that all vendors know the item failure rate  $\lambda$ , and the manufacturer knows the service rate  $\mu_j$  and the number of repair people,  $s_j$ , at each vendor. While a priority type  $i$  item is in service (or waiting for service) at vendor  $j$ , it costs the manufacturer  $h_{ij}$  dollars per unit time. This can be interpreted as a goodwill cost and is designed to prevent long delays in service. An example is the cost of a loaner while a type  $i$  item is in service. We assume that the holding cost is increasing for increasing priority type for each vendor. That is,

$$h_{1j} > h_{2j} > \dots > h_{mj}, j = 1, \dots, V.$$

This agrees with the intuition that items are given priority in service because of higher holding costs. Typically the holding cost for each priority type is independent of the vendor.

The overall goal of the manufacturer is to minimize their expected long-run average warranty cost. Given the above information, the manufacturer must decide on the optimal allocation matrix  $(x_{ij})_{\substack{i=1,\dots,m \\ j=1,\dots,V}}$ , where  $x_{ij}$  represents the number of priority class  $i$  items assigned to vendor  $j$ . We assume that this allocation, made at the beginning of the product life cycle, remains in effect during the entire contract period.

### 4.3 Problem Formulation

Based on the assumed item failure and service distributions of the previous section, we can model vendor  $j$  as an  $M/M/s_j/\cdot/x_j$  finite population queue with  $m$  priority classes, where  $x_j = (x_{1j}, \dots, x_{mj})$ . The priority structure gives priority class  $i$  a preemptive resume priority over classes  $j > i$ . This means that when a type  $i$  item enters the service queue and a type  $j > i$  is in service, the service of the type  $j$  item is preempted and is resumed after all higher priority items are serviced. Preemptive resume priority allows for easy calculation of the expected queue lengths at each vendor.

The long-run average warranty cost to the manufacturer consists of both repair costs and holding costs. First, we compute the long-run average repair cost. Let

$$X_{kj} = \sum_{i=1}^k x_{ij},$$

i.e.  $X_{kj}$  is the total number of items of priorities  $1, \dots, k$  assigned to server  $j$ . Let  $L_j(x)$  be the expected queue length of items (of all priorities) at vendor  $j$  when  $x = X_{mj}$  items are allocated to it. This can be computed by using the birth and death process analysis for an  $M/M/s_j/\cdot/x$  queue (performed in Section 4.7). The expected number of properly functioning items at any particular time is

$$x - L_j(x).$$

Since each functioning item has failure rate  $\lambda$ , the expected number of arrivals per unit time to the  $j^{\text{th}}$  vendor is

$$\lambda(x - L_j(x)).$$

Each such arrival costs the manufacturer  $c_j$  dollars. Hence, the long-run average repair cost is given by

$$\sum_{j=1}^V \lambda c_j (x - L_j(x)). \quad (4.1)$$

Next we compute the long-run average holding cost. Since the holding cost depends on the priority class, we will need an expression for the expected queue length of each priority class. We argue as follows: priority class 1 items only see other type 1 items in the queue since they preempt service for all other items. Therefore, the expected number of type 1 items in the queue at vendor  $j$  is  $L_j(x_{1j})$ . Type 2 items see both type 1 and type 2 items in the queue. The expected number of type 1 and type 2 items in the queue at vendor  $j$  is  $L_j(x_{1j} + x_{2j})$ ; therefore, the expected number of type 2 items in the queue at vendor  $j$  is

$$L_j(X_{2j}) - L_j(X_{1j}).$$

Similar reasoning yields the expected queue length for type  $i$  items at vendor  $j$  as

$$L_j(X_{ij}) - L_j(X_{i-1,j}).$$

Each type  $i$  item in the queue at vendor  $j$  costs the manufacturer  $h_{ij}$  dollars per unit time. Therefore, the long-run average holding cost is given by

$$\begin{aligned} & h_{1j}L_j(X_{1j}) + h_{2j}[L_j(X_{2j}) - L_j(X_{1j})] + \dots + h_{mj}[L_j(X_{mj}) - L_j(X_{m-1,j})] \\ &= \sum_{i=1}^{m-1} [(h_{ij} - h_{i+1,j})L_j(X_{ij})] + h_{mj}L_j(X_{mj}). \end{aligned} \quad (4.2)$$

We combine the expressions for repair cost (4.1) and holding cost (4.2) to express the total long-run cost rate of all items assigned to vendor  $j$ ,  $f_j(x_{1j}, x_{2j}, \dots, x_{mj})$ , as follows:

$$\begin{aligned} & \lambda c_j [X_{mj} - L_j(X_{mj})] + \sum_{i=1}^{m-1} [(h_{ij} - h_{i+1,j}) L_j(X_{ij})] + h_{mj} L_j(X_{mj}) \\ &= \sum_{i=1}^{m-1} [(h_{ij} - h_{i+1,j}) L_j(X_{ij})] + \lambda c_j X_{mj} + (h_{mj} - \lambda c_j) L_j(X_{mj}). \end{aligned} \quad (4.3)$$

The manufacturer wishes to minimize the total long-run average cost by outsourcing all warranted items to the vendors. Therefore, the manufacturer wishes to solve the following optimization problem:

$$\begin{aligned} P_m : \quad & \min \sum_{j=1}^V f_j(x_{1j}, x_{2j}, \dots, x_{mj}) \\ & \text{s.t.} \quad \sum_{j=1}^V x_{ij} = K_i, \quad i = 1, \dots, m \\ & \quad \quad x_{ij} \geq 0, \text{ and integer; } \quad i = 1, \dots, m, j = 1, \dots, V \end{aligned}$$

where  $f_j(x_{1j}, x_{2j}, \dots, x_{mj})$  is given by Equation 4.3.

When  $m = 1$ , the optimization problem defined above reduces to a standard resource allocation problem with a separable objective, studied by Gross [15], Ibariki and Katoh [16], Bretthauer and Shetty [6], and Opp et al. [28]. We provide these results in Section 4.4. However, if  $m > 1$  the problem is substantially more complicated. Ibariki and Katoh [16] mention a dynamic programming procedure to solve the problem, but it is essentially the same as enumerating all possible solutions. In Section 4.6, we exploit the structure of the objective to develop an algorithm to solve the problem with multiple priority classes.

Computing  $L(x)$  is the key to evaluating  $f_j(x_{1j}, x_{2j}, \dots, x_{mj})$ . Dowdy et al. [10] provides a result which states that the expected queue length  $L(x)$  of a  $M/M/s_j/\cdot/x$  finite population is convex in  $x$ , the size of the finite population. If there is only one server,

this reduces to the convexity of the Erlang Loss Function, which was given by Messerli [26] and Jagers and Van Doorn [20]. They provide a stable technique for computing  $L(x)$  by an efficient recursion formula. If there are multiple servers at a vendor, we use mean-value analysis in closed-queueing system to compute  $L(x)$ . Both of these computational techniques were described by Opp [27]. We summarize these results in the Section 4.7.

## 4.4 Single Priority Class

We begin by describing the solution method for the single priority class case. Since there is only one priority class, we omit it from the notation and write  $x_{1j} = x_j$  and  $h_{1j} = h_j$ . The problem described in the previous section reduces to

$$\begin{aligned}
 P_1 : \quad & \min \sum_{j=1}^V f_j(x_j) \\
 & \text{st } \sum_{j=1}^V x_j = K \\
 & x_j \geq 0 \text{ and integer}
 \end{aligned}$$

where the objective is now separable (i.e. it is a sum of functions of a single variable each):

$$f_j(x_j) = \lambda c_j x_j + (h_j - \lambda c_j) L_j(x_j).$$

Since  $L_j(x_j)$  is a convex function in  $x_j$  (see Dowdy, et al. [10]),  $f_j(x_j)$  is convex if  $h_j - \lambda c_j \geq 0$  and concave if  $h_j - \lambda c_j < 0$ . If all  $f_j(x_j)$  are all convex, this problem is the separable convex resource allocation problem. We will use the following notation

$$\delta f(x) = f(x) - f(x - 1), \quad x \geq 1.$$

Gross [15] first proved that the following greedy algorithm ( $G_1$ ) finds the optimal allocation to problem  $P_1$ , where  $K$  is a positive integer.

**Algorithm  $G_1$ :**

- **Step 0:** Set  $x_j = 0$  for all  $j = 1, \dots, V$ .
- **Step 1:** Choose a vendor  $k$  such that  $k \in \arg \min_{j=1, \dots, V} \delta f_j(x_{1j} + 1)$ .
- **Step 1a:** Increment  $x_k$  by 1.
- **Step 2:** If  $\sum_{j=1}^V x_j = K$ , stop; else, go to Step 1.

This algorithm selects an allocation vector  $\mathbf{x}^i = (x_1^i, x_2^i, \dots, x_V^i)$  at each stage  $i$ . We provide an alternate proof that  $\mathbf{x}^i$  is the optimal allocation vector for all values of  $i$ .

**Theorem 8** *Suppose that  $f_j(x)$  are convex functions and  $f_j(0) = 0$ . Then, the allocation vector  $\mathbf{x}^K$  selected by the greedy algorithm is an optimal solution to the optimization problem  $P_1$ .*

**Proof.** We proceed by induction on  $k$ , the number of items allocated. For the case  $k = 1$ , the problem  $P_1$  reduces to finding  $\min_{j=1, \dots, V} f_j(1)$ . Therefore,  $\mathbf{x}^1$  is an optimal solution vector since

$$\min_{j=1, \dots, V} f_j(1) = \min_{j=1, \dots, V} \delta f_j(1).$$

As an induction hypothesis, assume that the allocation  $\mathbf{x}^k$  produced by the greedy algorithm is an optimal solution to  $P_1$  with  $K = k$ . The greedy algorithm next picks an integer  $n(k)$  such that

$$n(k) \in \arg \min_{j=1, \dots, V} (\delta f_j(x_j^k + 1)),$$

and then sets

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{e}_{n(k)},$$

where  $\mathbf{e}_{n(k)}$  is a  $V$ -vector with a 1 in component  $n(k)$  and 0 in all of the remaining components.

Let  $S^n = \{(u_1^n, u_2^n, \dots, u_V^n) : \sum u_i^n = n\}$  be the set of all possible allocation vectors of  $n$  items. Consider an allocation vector  $\mathbf{u}^{k+1} \in S^{k+1}$ . We claim that

$$\sum_{j=1}^V f_j(u_j^{k+1}) \geq \sum_{j=1}^V f_j(x_j^{k+1}).$$

By the induction hypothesis,

$$\delta f_{n(k)}(x_{n(k)}^{k+1}) \uparrow \text{ in } k \text{ (in the weak sense)}. \quad (4.4)$$

Consider  $1 \leq x \leq x_j^{k+1}$  for a specific vendor  $j$ . There exists an  $m \leq k$  such that the allocation vector  $\mathbf{x}^m$  produced by the greedy algorithm has  $n(m) = j$ , i.e.,

$$x_j^{m+1} = x \text{ and } x_j^m = x - 1.$$

Therefore, we have

$$\delta f_j(x) = \delta f_j(x_j^{m+1}) = \delta f_{n(m)}(x_{n(m)}^{m+1}) \leq \delta f_{n(k)}(x_{n(k)}^{k+1}), \quad 1 \leq x \leq x_j^{k+1}. \quad (4.5)$$

The inequality above follows from 4.4. Also note that the convexity of  $f_j$  gives

$$\delta f_j(x) \geq \delta f_j(x_j^{k+1} + 1) \geq \delta f_{n(k)}(x_{n(k)}^{k+1}), \quad x > x_j^{k+1}. \quad (4.6)$$

Let  $A = \{j : u_j^{k+1} > x_j^{k+1}\}$  and  $B = \{j : u_j^{k+1} < x_j^{k+1}\}$ . We have:

$$\begin{aligned} & \sum_{j=1}^V [f_j(u_j^{k+1}) - f_j(x_j^{k+1})] \\ &= \sum_{j \in A} [f_j(u_j^{k+1}) - f_j(x_j^{k+1})] + \sum_{j \in B} [f_j(u_j^{k+1}) - f_j(x_j^{k+1})] \\ &= \sum_{j \in A} \sum_{x=x_j^{k+1}+1}^{u_j^{k+1}} \delta f_j(x) - \sum_{j \in B} \sum_{x=u_j^{k+1}+1}^{x_j^{k+1}} \delta f_j(x) \end{aligned}$$

$$\begin{aligned}
&\geq \sum_{j \in A} \sum_{x=x_j^{k+1}+1}^{u_j^{k+1}} \delta f_{n(k)}(x_{n(k)}^{k+1}) - \sum_{j \in B} \sum_{x=u_j^{k+1}+1}^{x_j^{k+1}} \delta f_{n(k)}(x_{n(k)}^{k+1}) \\
&= \delta f_{n(k)}(x_{n(k)}^{k+1}) \left[ \sum_{j \in A} \sum_{x=x_j^{k+1}+1}^{u_j^{k+1}} 1 - \sum_{j \in B} \sum_{x=u_j^{k+1}+1}^{x_j^{k+1}} 1 \right] \\
&= 0.
\end{aligned}$$

The inequality in the fourth line comes from 4.5 and 4.6. The last equality is true because

$$\sum_{j=1}^V u_j^{k+1} = \sum_{j=1}^V x_j^{k+1} = k + 1.$$

This completes the proof. ■

Ibaraki and Katoh [16] provide many algorithms to produce the optimal solution; some of them run in polynomial time. The complexity of the greedy algorithm is  $O(V + K \log V)$ , since the minimization step takes  $O(\log V)$  time. It is fairly efficient; we ran the algorithm for  $K = 10000$  and  $V = 5$ , it took about 3 seconds on a standard machine to run to optimality.

If the functions  $f_j(x)$  are all concave, the optimum occurs at an extreme point. The optimal solution can be found by picking the vendor  $j$  where  $f_j(K)$  is minimum, and allocating all items to vendor  $j$ . Opp, et al. [28] also discusses the case where the functions  $f_j(x)$  are mixed convex and concave; we omit that discussion in this dissertation for the sake of brevity.

To solve the optimization problem  $P_m$  for  $m$  priority classes, we reformulate the problem as a convex minimum cost network flow problem. In the next section, we provide some background information on minimum cost network flow problems.

## 4.5 Minimum Cost Network Flow Problems

We first provide the background on minimum cost network flow problems with linear costs and discuss the convex cost problem in Section 4.5.1. Let  $G = (N, A)$  be a directed network with a cost  $c_{ij}$  and capacity  $u_{ij}$  on each arc  $(i, j) \in A$ . We associate each node with a number  $b(i)$ . If  $b(i) > 0$ , then  $b(i)$  indicates the supply at node  $i$ , while if  $b(i) < 0$ , then  $-b(i)$  indicates the demand at node  $i$ . A value of  $b(i) = 0$  implies that node  $i$  is purely a transshipment node. We assume that  $\sum_i b(i) = 0$ , i.e. there is just enough supply to satisfy the demand exactly. The flow from node  $i$  to node  $j$  along arc  $(i, j)$  is denoted by  $x_{ij}$ . The objective is to find the flow  $\mathbf{x} = (x_{ij}, (i, j) \in A)$  that satisfies all of the demand at minimum cost. The problem can be stated as a linear integer program as follows:

$$\begin{aligned} & \text{Minimize} && \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{subject to:} &&& \sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = b(i) \quad \forall i \in N \\ &&& 0 \leq x_{ij} \leq u_{ij} \text{ and integer, } \quad \forall (i, j) \in A \end{aligned} \tag{4.7}$$

For a given flow  $\mathbf{x}$ , the residual network  $G(\mathbf{x})$  of a graph  $G$  plays an important role in network algorithms. To obtain  $G(\mathbf{x})$ , we replace each arc  $(i, j) \in A$  by two arcs,  $(i, j)$  and  $(j, i)$ . The forward arc  $(i, j)$  has cost  $c_{ij}$  with residual capacity  $r_{ij} = u_{ij} - x_{ij}$  and the backward arc  $(j, i)$  has cost  $-c_{ij}$  and residual capacity  $r_{ji} = x_{ij}$ .

Ahuja, Magnanti, and Orlin [1] provide many algorithms to solve minimum cost network flow problems. We will focus on the successive shortest path algorithm found in Section 9.7 of that book, provided below for ready reference. The algorithm uses node

potentials  $\pi(i)$  for  $i \in N$ . It also uses the imbalance of a node  $e(i)$ , defined by

$$e(i) = b(i) + \sum_{k:(k,i) \in A} x_{ki} - \sum_{j:(i,j) \in A} x_{ij}, \quad \forall i \in N.$$

If  $e(i) > 0$ , we call node  $i$  an excess node. Similarly, if  $e(i) < 0$ , node  $i$  is a deficit node and if  $e(i) = 0$ , the node is balanced.

**Successive Shortest Path Algorithm:**

initialize  $\mathbf{x} := \mathbf{0}$ ,  $\pi := \mathbf{0}$ ,  $E := \{i \mid e(i) > 0\}$ ,  $D := \{i \mid e(i) < 0\}$ ;

while  $|E| > 0$  do

- select a node  $k \in E$  and a node  $l \in D$ ;
- determine the shortest path distances  $d(j)$  from node  $k$  to all other nodes in the residual network  $G(\mathbf{x})$  with respect to the reduced costs  $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j)$ ;
- denote the shortest path from  $k$  to  $l$  by  $P$ ;
- update  $\pi := \pi - \mathbf{d}$ ;
- augment  $\delta = \min \left[ e(k), -e(l), \min_{(i,j) \in P} \{r_{ij}\} \right]$  units of flow along the path  $P$ ;
- update  $\mathbf{x}$ ,  $G(\mathbf{x})$ ,  $E$ ,  $D$  and the reduced costs;

end;

Proof of optimality of the algorithm is given on page 323 of [1]. The node potentials  $\pi(i)$  for each node are tracked to show that the reduced cost optimality conditions are satisfied at each step of the algorithm. Therefore, once a feasible flow is obtained, that flow is an optimal solution. However, in the actual implementation of the algorithm, we need not track the node potentials. We justify this as follows: during each iteration of the algorithm, we find the shortest path from an excess node to a deficit node with respect to the reduced costs in the residual network  $G(\mathbf{x})$ . The reduced costs are defined

in terms of the node potentials by

$$c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j), \quad \forall (i, j) \in A. \quad (4.8)$$

For given excess node  $k$  and deficit node  $l$ , the cost of a path  $P$  from  $k$  to  $l$  in the residual network is the sum of its constituent arcs. Hence

$$c^\pi(P) = c(P) - \pi(k) + \pi(l), \quad \forall \text{ paths } P.$$

Since the shortest path from  $k$  to  $l$  has distance  $\min_P c^\pi(P)$ , the constants  $\pi(k)$  and  $\pi(l)$  are ignored. Therefore, we can find the shortest path from  $k$  to  $l$  in the residual network  $G(\mathbf{x})$  from the original arc costs and do not need to track the node potentials  $\pi$  or the reduced costs  $c_{ij}^\pi$ . Hence, the above algorithm can be simplified to:

**Successive Shortest Path Algorithm:**

initialize  $\mathbf{x} := \mathbf{0}$ ,  $E := \{i \mid e(i) > 0\}$ ,  $D := \{i \mid e(i) < 0\}$ ;

while  $|E| > 0$  do

- select a node  $k \in E$  and a node  $l \in D$ ;
- determine the shortest path distances  $d(j)$  from node  $k$  to all other nodes in the residual network  $G(\mathbf{x})$  with respect to the costs  $c_{ij}$ ;
- denote the shortest path from  $k$  to  $l$  by  $P$ ;
- augment  $\delta = \min \left[ e(k), -e(l), \min_{(i,j) \in P} \{r_{ij}\} \right]$  units of flow along the path  $P$ ;
- update  $\mathbf{x}$ ,  $G(\mathbf{x})$ ,  $E$ , and  $D$ ;

end;

Let  $n$  be the number of nodes and  $m$  be the number of arcs. This algorithm terminates in at most  $nU$  iterations, where  $U$  is the largest supply of any node. Each iteration requires solving a shortest path problem on  $n$  nodes,  $m$  arcs. Let  $S(n, m, C)$

denote the time to solve a shortest path problem, where  $C$  is the maximum arc cost. Therefore, the complexity of the successive shortest path algorithm is  $O(nUS(n, m, nC))$ , which is pseudopolynomial in the input size since it is polynomial in  $n$ ,  $m$ , and  $U$ . (Note that  $nC$  is used rather than  $C$  in the expression, since the costs in residual network are bounded by  $nC$ .)

Ahuja, Magnanti, and Orlin provide the capacity scaling algorithm to solve the minimum cost network flow problem in Section 10.2 of [1], which is polynomial in the input size. The main idea is to push flow in sufficiently large quantities to reduce the number of augmentations required in the successive shortest path algorithm. We find that the successive shortest path algorithm performs quite well on our networks and use it to solve our problems.

### 4.5.1 Convex Network Problems

In this section we consider the convex cost case: it costs  $c_{ij}(x)$  to send a flow of  $x$  units along the arc  $(i, j)$ . The aim is to find the flow that solves the following optimization problem:

$$\begin{aligned}
 & \text{Minimize} && \sum_{j=1}^V c_{ij}(x_{ij}) \\
 & \text{subject to:} && \sum_{j:(i,j) \in A} x_{ij} - \sum_{k:(k,i) \in A} x_{ki} = b(i) \quad \forall i \in N \\
 & && 0 \leq x_{ij} \leq u_{ij} \text{ and integer, } \quad \forall (i, j) \in A
 \end{aligned} \tag{4.9}$$

This case can be handled similarly to the linear cost case, with a slight modification of the network. We achieve this transformation by breaking the convex cost function  $c_{ij}(x)$  into a piecewise linear function consisting of  $u_{ij}$  pieces, with the breakpoints occurring at each integer point between 1 and  $u_{ij}$ . Replace each arc  $(i, j)$  in the network  $G$  by  $u_{ij}$  parallel arcs. The  $k$ th arc  $(i, j)^k$  has capacity 1 and cost  $c_{ij}^k = c_{ij}(k) - c_{ij}(k - 1)$

( $k = 1, \dots, u_{ij}$ ). Let  $y_{ij}^k$  be the flow on arc  $(i, j)^k$ , hence

$$x_{ij} = \sum_{k=1}^{u_{ij}} y_{ij}^k.$$

Therefore, we can formulate the convex minimum cost network flow problem as the following integer linear program:

$$\begin{aligned} & \text{Minimize} && \sum_{(i,j) \in A} \sum_{p=1}^{u_{ij}} c_{ij}^p y_{ij}^p \\ \text{subject to:} &&& \sum_{j:(i,j) \in A} \sum_{p=1}^{u_{ij}} y_{ij}^p - \sum_{k:(k,i) \in A} \sum_{p=1}^{u_{ki}} y_{ki}^p = b(i) \quad \forall i \in N \\ &&& 0 \leq y_{ij}^p \leq 1 \text{ and integer, } \quad \forall (i, j) \in A, p = 1, \dots, u_{ij}. \end{aligned}$$

Since this formulation has only linear costs, we can use the successive shortest path algorithm to solve it. In practice, we do not need to physically replace each arc by  $u_{ij}$  parallel arcs. The convexity of the cost function implies that the arc costs  $c_{ij}^k$  are increasing in  $k$ . When a unit of flow is sent from  $i$  to  $j$  along a shortest path, the flow will go on the arc of lowest cost. If the current amount of flow from  $i$  to  $j$  is  $x_{ij}$ , the cost of sending one additional unit of flow from  $i$  to  $j$  is

$$c_{ij}(x_{ij} + 1) - c_{ij}(x_{ij}).$$

Therefore, we can use the successive shortest path algorithm to solve the convex cost problem by just updating the cost on arc  $(i, j)$  at each iteration as  $c_{ij}(x_{ij} + 1) - c_{ij}(x_{ij})$ . We point out that a similar method as the capacity scaling algorithm will solve the convex problem in polynomial time, given in Section 14.5 of [1].

## 4.6 Network Flow Formulation

We next show that the optimization problem  $P_m$  can be reformulated as a network flow problem. We begin with the single priority case.

### 4.6.1 Single Priority Class

In section 4.4, we discussed the single priority case and showed that the greedy algorithm  $G_1$  provides an optimal solution. Here we reformulate the problem as a convex minimum cost network flow problem. Consider the network shown in Figure 4.1 below. The source

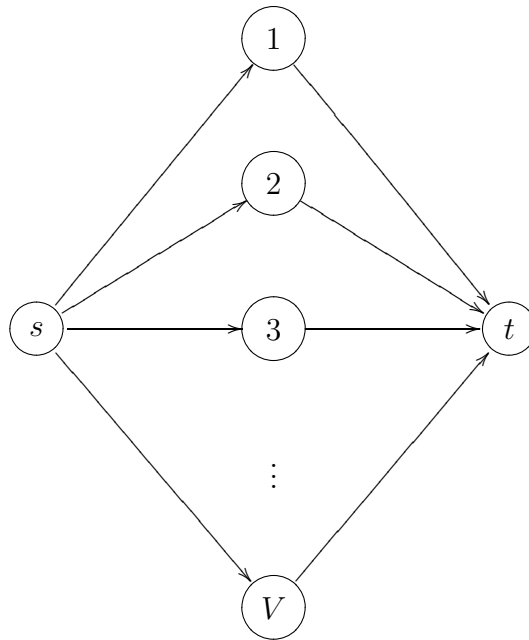


Figure 4.1: Network Model of Single Priority Problem

node  $s$  has a supply of  $K$  units and the sink node  $t$  has a demand of  $K$  units, i.e.  $b(s) = K$ ,  $b(t) = -K$ , and  $b(j) = 0$  ( $j = 1, \dots, V$ ). Each of the arcs  $(s, j)$  ( $j = 1, \dots, V$ ) has capacity  $u_{sj} = K$  and cost  $c_{sj}(x) = 0 \forall x$ . Each arc  $(j, t)$  ( $j = 1, \dots, V$ ) has capacity  $K$  and cost function  $c_{jt}(x) = f_j(x_j)$ , where  $x_j$  is the flow along arc  $(j, t)$  (which equals the flow along  $(s, t)$ ) and represents the number of items assigned to server  $j$ . The integer

linear program developed in 4.9 reduces to:

$$\begin{aligned} \min \quad & \sum_{j=1}^V f_j(x_j) \\ \text{st} \quad & \sum_{j=1}^V x_j = K \\ & 0 \leq x_j \leq K \text{ and integer, } j = 1, \dots, V \end{aligned}$$

Therefore, the network formulation of the problem is equivalent to the original single-priority problem  $P_1$ . We apply the successive shortest path algorithm to solve this minimum cost network flow problem for this network. The algorithm significantly simplifies since there is only 1 deficit node and 1 excess node and the shortest path step is very easy. At each iteration there are only  $V$  paths from  $s$  to  $t$  (the residual network has no negative cycles). The cost to send one additional unit of flow through node  $j$  when there is already a flow of  $x_j$  is  $\delta f_j(x_j + 1)$ . Hence the algorithm given in section 4.5 to solve the network problem reduces to:

**Successive Shortest Path Algorithm for 1-priority:**

initialize  $\mathbf{x} := \mathbf{0}$ ;

while  $\sum_{j=1}^V x_j < K$  do

- compute  $\min_{j=1, \dots, V} \delta f_j(x_j + 1)$
- increment  $x_k$  by 1, where  $k \in \arg \min_{j=1, \dots, V} \delta f_j(x_j + 1)$ ;

end

This is exactly the same as the  $\mathbf{G}_1$  algorithm given in Section 4.4, and therefore we have another proof of correctness of the algorithm. As stated in Section 4.4, the complexity of this algorithm is  $O(V + K \log V)$ .

## 4.6.2 Two Priority Classes

We next turn to the two priority problem. The network model for this problem is provided in Figure 4.2 below. The source node  $s_1$  has a supply of  $b(s_1) = K_1$  units, node  $s_2$  has

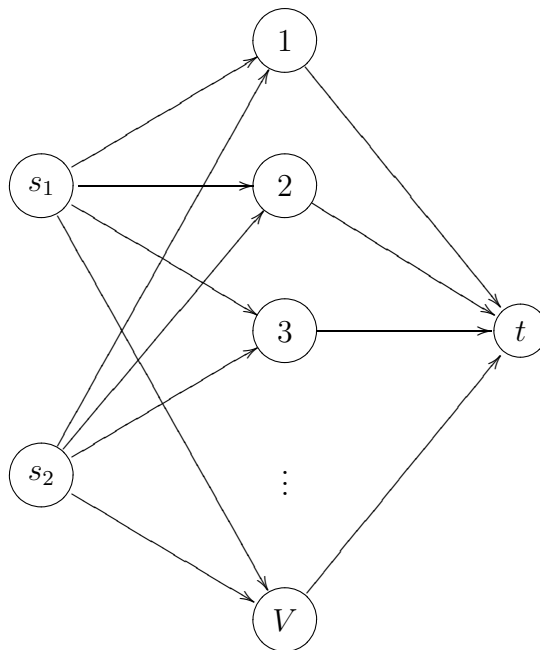


Figure 4.2: Network Model of Two-priority Problem

a supply of  $b(s_2) = K_2$  units, and the sink node  $t$  has a demand of  $K_1 + K_2$  units (i.e.  $b(t) = -(K_1 + K_2)$ ). All other nodes  $j$  are transshipment nodes and have  $b(j) = 0$  ( $j = 1, \dots, V$ ). Let  $x_{ij}$  be the flow from node  $s_i$  to node  $j$  ( $i = 1, 2$ ), which represents the number of type  $i$  customers assigned to server  $j$ . The arcs have the following capacities, flows, and costs ( $j = 1, \dots, V$ ):

Arc	Capacity	Flow	Cost
$(s_1, j)$	$K_1$	$x_{1j}$	$f_{1j}(x_{1j})$
$(s_2, j)$	$K_2$	$x_{2j}$	0
$(j, t)$	$K_1 + K_2$	$x_{1j} + x_{2j}$	$f_{2j}(x_{1j} + x_{2j})$

Table 4.1: Arc Properties for Two-priority Network

The minimum cost network flow problem defined in 4.9 reduces to:

$$\begin{aligned}
& \text{Minimize} && \sum_{j=1}^V [f_{1j}(x_{1j}) + f_{2j}(x_{1j} + x_{2j})] \\
& \text{subject to:} && \sum_{j=1}^V x_{1j} = K_1, \quad \sum_{j=1}^V x_{2j} = K_2, \\
& && 0 \leq x_{ij} \leq K_i \text{ and integer, } \quad i = 1, 2, \quad j = 1, \dots, V.
\end{aligned}$$

This is equivalent to the optimization problem  $P_2$ ; hence we can apply the successive shortest path algorithm to this network shown in Figure 4.2 to solve the two-priority problem. Like the single-priority case, the network structure for the two-priority problem simplifies the successive shortest path algorithm significantly. Now there are two excess nodes; without loss of generality we start with node  $s_2$ , pushing  $K_2$  units of flow to the sink  $t$ . Since node  $s_1$  is not accessible from node  $s_2$ , we can omit  $s_1$  and all of the arcs incident with it from the network. Thus we determine  $x_{2j}$  ( $j = 1, \dots, V$ ) by the  $G_1$  algorithm on this reduced network. Next we return to the original network and push the  $K_1$  units of flow from the source  $s_1$  to the sink  $t$ . Here there are  $V + 1$  paths in the residual network  $G(\mathbf{x})$  to consider. There are  $V$  direct paths to  $t$  through node  $j$  ( $j = 1, \dots, V$ ). The cost of these paths are  $\delta f_{1j}(x_{1j} + 1) + \delta f_{2j}(x_{1j} + x_{2j} + 1)$ . Another possible shortest path to  $t$  goes through  $s_2$ . The shortest path from  $s_1$  to  $s_2$  is through a node  $k$  such that

$$k \in \arg \min_{x_{2j} > 0} \delta f_{1j}(x_{1j} + 1)$$

Note that the cost of pushing flow from  $j$  to  $s_2$  is 0 if  $x_{2j} > 0$ . From  $s_2$  to the sink  $t$ , the shortest path is determined by choosing a vendor  $m$  such that

$$m \in \arg \min_j \delta f_{2j}(x_{1j} + x_{2j} + 1).$$

Augmenting a unit of flow along this path requires that we increase  $x_{1k}$  and  $x_{2m}$  by 1 unit and decrease  $x_{2k}$  by 1 unit. Therefore, we have the following algorithm to solve the two-priority problem:

**Successive Shortest Path Algorithm for 2-priority (Algorithm  $\mathbf{G}_2$ ):**

initialize  $\mathbf{x} := \mathbf{0}$ ;

while  $\sum_{j=1}^V x_{2j} < K_2$  do

- compute  $\min_{j=1, \dots, V} \delta f_{2j}(x_{2j} + 1)$
- increment  $x_{2k}$  by 1, where  $k \in \arg \min_{j=1, \dots, V} \delta f_j(x_{2j} + 1)$ ;

end

while  $\sum_{j=1}^V x_{1j} < K_1$  do

- compute  $d_j = \delta f_{1j}(x_j + 1) + \delta f_{2j}(x_{1j} + x_{2j} + 1)$ ,  $j = 1, \dots, V$  and  $d_{V+1} = \min_{x_{2j} > 0} \delta f_{1j}(x_{1j} + 1) + \min_k \delta f_{2k}(x_{1k} + x_{2k} + 1)$ .
- let  $q \in \arg \min_{j=1, \dots, V+1} d_j$ .  
If  $q \in \{1, \dots, V\}$ , increment  $x_{1q}$  by 1.  
If  $q = V + 1$ , let  $k \in \arg \min_{x_{2j} > 0} \delta f_{1j}(x_{1j} + 1)$  and  $p \in \arg \min_j \delta f_{2j}(x_{1j} + x_{2j} + 1)$ .  
Increment  $x_{1k}$  and  $x_{2p}$  by 1 and decrement  $x_{2k}$  by 1 unit.

end

This algorithm has the advantage of being very easy to code. It is very similar to the greedy algorithm  $\mathbf{G}_1$  since it adds one item at a time, with the possibility of rearranging the lower priority items previously assigned. The complexity of the algorithm is  $O(V + K \log V)$ .

### 4.6.3 Multiple Priority Classes

We now address the  $m$  priority problem. The network model for this problem is provided in Figure 4.3. This network has  $m$  source nodes, labeled  $s_1, s_2, \dots, s_m$ , with node  $s_i$

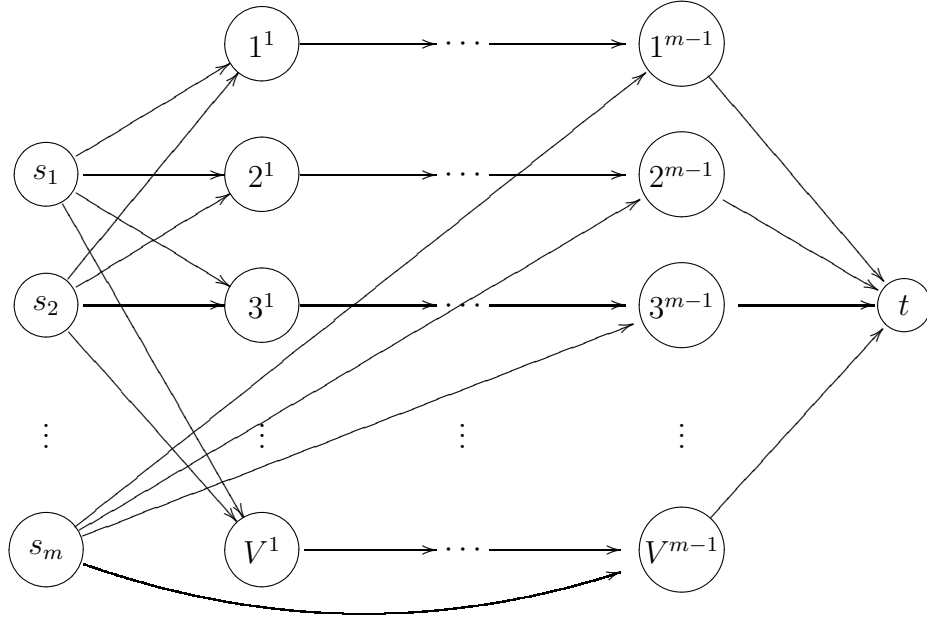


Figure 4.3: Network Model of  $m$  Priority Problem

having supply  $K_i$  (i.e.  $b(s_i) = K_i$  for  $i = 1, \dots, m$ ). There is a single sink node  $t$  that has demand  $\sum_{i=1}^m K_i$ , i.e.  $b(t) = -\sum_{i=1}^m K_i$ . There are  $m-1$  nodes for each vendor  $j$  (labeled  $j^1, \dots, j^{m-1}$ ) such that  $b(j^i) = 0$ . The node  $s_1$  is connected with nodes  $j^1$  ( $j = 1, \dots, V$ ). Similarly,  $s_i$  is connected with nodes  $j^{i-1}$  ( $i = 2, \dots, m, j = 1, \dots, V$ ). The node  $j^i$  is connected with node  $j^{i+1}$  ( $i = 1, \dots, m-2, j = 1, \dots, V$ ). Let  $x_{1j}$  be the flow from node  $s_1$  to node  $j^1$  and  $x_{ij}$  be the flow from node  $s_i$  to node  $j^{i-1}$  ( $i = 2, \dots, m$ ). The flow  $x_{ij}$  gives the number of type  $i$  items assigned to server  $j$ . We continue using the notation that  $\sum_{i=1}^p x_{ij} = X_{pj}$ . Due to the flow balance constraints, the flow on arc  $(j^i, j^{i+1})$  is  $X_{i+1,j}$  and the flow on arc  $(j^{m-1}, t)$  is  $X_{mj}$ . The arcs capacities, flows, and costs are summarized in Table 4.2 ( $j = 1, \dots, V$  in each row):

Arc	Capacity	Flow	Cost
$(s_1, j^1)$	$K_1$	$x_{1j}$	$f_{1j}(x_{1j})$
$(s_i, j^{i-1})$	$K_i$	$x_{ij}$	0 $(i = 2, \dots, m)$
$(j^i, j^{i+1})$	$\sum_{p=1}^{i+1} K_p$	$X_{i+1,j}$	$f_{i+1,j}(X_{i+1,j})$ $(i = 1, \dots, m-1)$
$(j^{m-1}, t)$	$\sum_{p=1}^m K_p$	$X_{mj}$	$f_{mj}(X_{mj})$

Table 4.2: Arc Properties for  $m$ -priority Network

We now give the minimum cost network flow formulation defined in 4.9 for this network:

$$\begin{aligned}
\min \quad & \sum_{j=1}^V f_j(x_{1j}, x_{2j}, \dots, x_{mj}) \\
\text{s.t.} \quad & \sum_{j=1}^V x_{ij} = K_i, \quad i = 1, \dots, m \\
& 0 \leq x_{ij} \leq K_i, \text{ and integer; } \quad i = 1, \dots, m, j = 1, \dots, V
\end{aligned}$$

where  $f_j(x_{1j}, x_{2j}, \dots, x_{mj}) = \sum_{p=1}^m f_{pj}(X_{pj})$ . Note that this is the same formulation that we developed in Section 4.3. Hence, we can apply the successive shortest path algorithm to this network to solve the  $m$ -priority problem. The network  $G$  has  $m + 1 + (m - 1)V$  nodes and  $(2m - 1)V$  arcs. We turn to one of the well-established shortest path algorithms to find the shortest path at each iteration of the algorithm. We start pushing flow from  $s_m$  to  $t$  in the reduced network  $G_m$ , ignoring the other source nodes and the arcs incident with it. In general, we refer to the reduced network that omits the source nodes  $s_1$  through  $s_{i-1}$  and their incident arcs by  $G_i$ . Next, we push flow from  $s_{m-1}$  to  $t$  in the reduced network  $G_{m-1}$ . This procedure continues until we push the flow from  $s_1$  to  $t$  over the full network in Figure 4.3. We outline the algorithm to solve the  $m$ -priority problem below:

### Successive Shortest Path Algorithm for m-priority (Algorithm $\mathbf{G}_m$ ):

initialize  $\mathbf{x} := \mathbf{0}$ ;

for  $i = m$  to 1 while  $\sum_{j=1}^V x_{ij} < K_i$  do

- determine the shortest path  $P$  from node  $s_i$  to node  $t$  in the residual network  $G_i(x)$  constructed from the reduced network  $G_i$ ;
- augment 1 unit of flow along the path  $P$ ;
- update  $\mathbf{x}$  and  $G_i(\mathbf{x})$ ;

end while loop;

end for loop;

The residual graph  $G_i(x)$  has at most  $N = m + 1 + (m - 1)V$  nodes and at most  $A = 2(2m - 1)V$  edges. We use the label correcting algorithm to solve the shortest path problem at each iteration; it has complexity  $O(NA)$ , where  $N$  is the number of nodes and  $A$  is the number of arcs. Therefore, this algorithm has complexity  $O(m^2KV^2)$  (since from  $NA \approx 4m^2V^2$ ). Typical parameters values are  $m \in \{1, \dots, 4\}$  and  $V \in \{1, \dots, 10\}$ . With these parameters, the algorithm is quite efficient. For an example with  $m = 4$ ,  $V = 6$ , and  $K = 10000$ , the algorithm took about 15 seconds on a standard machine.

## 4.7 Computational Issues

In this section, we discuss some of the computational issues related to the outsourcing problem. First, we compute the expected queue length in a  $M/M/s_j/\cdot/N$  finite population queue. The easiest case is when  $s_j = 1 \forall j$ . We derive the steady-state probabilities for exponential lifetimes and use the result of Bunday and Scranton [8] that these are the same steady state probabilities when item lifetimes are not exponentially distributed.

The state space is  $\{0, 1, \dots, N\}$  with transition rates

$$\begin{aligned}\lambda_i &= (N - i)\lambda, & 0 \leq i \leq N \\ \mu_i &= \mu, & 1 \leq i \leq N.\end{aligned}$$

Let  $p_i$  be the long-run probability that the queue has  $i$  items, including the item in service. Solving the balance equations for  $p_i$ , we get the recurrence relation

$$p_{i+1} = \frac{N - i}{\rho} p_i \quad 0 \leq i \leq N - 1, \quad (4.10)$$

where  $\rho = \mu/\lambda$ . The expected queue length  $L(N)$  can be computed by

$$N - L(N) = \sum_{i=0}^N (N - i)p_i.$$

We use Equation 4.10 to compute  $L(N)$ . We have

$$\begin{aligned}N - L(N) &= \sum_{i=0}^N \frac{N - i}{\rho^i (N - i)!} \bigg/ \sum_{j=0}^N \frac{1}{\rho^j (N - j)!} \\ &= \rho \left( \sum_{i=0}^N \frac{1}{\rho^i (N - i)!} - \frac{1}{N!} \right) \bigg/ \sum_{j=0}^N \frac{1}{\rho^j (N - j)!} \\ &= \rho - \frac{\rho/N!}{\sum_{j=0}^N 1/(\rho^j (N - j)!)} \\ &= \rho - \rho \frac{\rho^N/N!}{\sum_{j=0}^N \rho^j/j!} = \rho - \rho B(\rho, N),\end{aligned}$$

where

$$B(\rho, N) = \frac{\rho^N/N!}{\sum_{j=0}^N \rho^j/j!}$$

is the Erlang Loss function (see [19]). Therefore, we use

$$L(N) = N - \rho + \rho B(\rho, N) \quad (4.11)$$

to compute the expected queue length. When  $N$  is large, directly calculating  $B(\rho, N)$  is difficult. Therefore, we use the following recursion to compute  $B(\rho, N)$ :

$$B(\rho, N) = \frac{\rho B(\rho, N-1)/N}{1 + \rho B(\rho, N-1)/N}, \quad N \geq 1, \quad (4.12)$$

where  $B(\rho, 0) = 1$ .

Opp, et al. [28] dedicates a section to discussing calculation issues when there are multiple servers at each vendor; we omit it for the sake of brevity. They show that a queue with  $s_j$  servers working at rate  $\mu_j$  can be well approximated by a single server queue working at rate  $s_j\mu_j$ . This approximation works best when the servers are rarely idle, which is common in practice. If the exact expected queue length is needed when  $s_j > 1$  for some  $j$ , we can use an analysis of a two station closed queueing network, given in Section 2.7 of Gross and Harris [14]. Opp [27] gives a recursive method of computing the expected queue length in this case. We provide those results here for ready reference.

Consider a two-station closed queueing network, station 1 an infinite server station with service rate  $\lambda$  and station 2 is a  $s$  server station with service rate  $\mu$  per server. There are  $N$  items circulating continuously in the network between stations 1 and 2 (assume that  $N \geq s$ ). We calculate the expected number of customers at station 2. Let  $\rho = s\mu/\lambda$ . Then, the probability that all  $s$  servers at station 2 are busy, and hence the probability that an incoming item is blocked, is given by

$$P_B(N) = \frac{\sum_{i=0}^{N-s} \rho^i / i!}{\sum_{i=0}^{N-s} \rho^i / i! + A(N)\rho^{N-s}/(N-s)!}, \quad (4.13)$$

where

$$A(N) = \sum_{i=1}^s \frac{\rho^i s! / (s-i)!}{s^i (N-s+i)! / (N-s)!}.$$

Note that  $A(N)$  has only  $s$  terms, so we need not compute it recursively. Clearly  $P_B(N) = 0$  for  $N < s$ . From Equation 4.13, we get the following expression for  $P_B(N)$  in terms of  $A(N)$  and  $B(\rho, N)$ :

$$P_B(N) = \frac{1}{1 + A(N)B(\rho, N-s)}.$$

Recall that  $B(\rho, N-s)$  can be computed recursively from Equation 4.12.

Let  $W_q(N)$  be the mean waiting time in the queue at station 2 with  $N$  customers circulating in the network. Similarly, define  $L_q(N)$  as the expected number of customers waiting in the queue at station 2 and  $L(N)$  as the expected number of customers in the queue at station 2. The following recursion holds (initialize  $L_q(0) = 0$ ):

$$\begin{aligned} W_q(N) &= \frac{P_B(N-1) + L_q(N-1)}{s\mu}, \\ \Lambda(N) &= \frac{N}{1/\lambda + 1/\mu + W_q(N)}, \\ L_q(N) &= \Lambda(N)W_q(N), \text{ and} \\ L(N) &= L_q(N) + \Lambda(N)/\mu. \end{aligned}$$

## 4.8 An Example

We provide an example to illustrate the  $\mathbf{G}_m$  algorithm. Consider a manufacturer that has warranted items of  $m = 4$  different priority types. The number of items of each priority type is  $K_1 = 150$ ,  $K_2 = 250$ ,  $K_3 = 200$ , and  $K_4 = 400$  (for a total of 1000 items). Each item has a common failure rate of  $\lambda = 1.5$  failures per year. There are  $V = 6$  vendors at his disposal. Typically, each repairperson at a vendor is busy with repairs during the majority of the time. Hence, we use the result stated in the previous section that  $s$  repairpeople working at rate  $\mu$  is very similar to a single repairperson working at

rate  $s\mu$ . For this reason, we assume that there is a single repairperson working at rate  $s_j\mu_j$  at each vendor  $j$ . The vendors' properties are summarized in Table 4.3.

Vendor $j$	$s_j\mu_j$	$c_j$	$(h_{1j}, h_{2j}, h_{3j}, h_{4j})$
1	80	15	(500, 350, 300, 175)
2	62	19	(500, 400, 250, 175)
3	70	18	(500, 350, 300, 160)
4	50	15	(500, 400, 250, 160)
5	45	14	(500, 400, 300, 175)
6	25	9	(500, 350, 300, 175)

Table 4.3: Costs and Service Rates for Each Vendor

Note that for every vendor  $j$ ,  $h_{ij} > h_{i+1,j}$  ( $i = 1, \dots, 3$ ) and  $h_{4j} > \lambda c_j$ , so the convexity properties for the  $f_{ij}$  functions are satisfied. We apply the  $\mathbf{G}_m$  algorithm on these parameters. The allocation matrix below is the solution, where the row indicates the priority type and the column indicates the vendor. For example, the number of type 3 items assigned to server 4 is 80.

$$X = \begin{pmatrix} 39 & 34 & 31 & 24 & 21 & 1 \\ 62 & 33 & 56 & 30 & 33 & 36 \\ 0 & 120 & 0 & 80 & 0 & 0 \\ 0 & 0 & 300 & 95 & 5 & 0 \end{pmatrix}$$

The expected yearly cost of this allocation is \$146,012.42.

## 4.9 Cost Benefits of the Multi-Priority Approach

In this section, we discuss the benefits of giving priority in service to items that have higher holding costs. We first illustrate the benefits on the example in the previous section. Suppose there was no priority structure in place, but the holding costs are still given as in Section 4.8. Consider a specific vendor  $j$ . If  $N$  items are randomly assigned to

vendor  $j$  ignoring priority levels, the expected number of type  $i$  items assigned to server  $j$  is:

$$\frac{K_i}{\sum_{i=1}^m K_i} N.$$

If there is no priority structure, the expected waiting time will be independent of the priority type. Therefore, the expected holding cost per item  $\hat{h}_j$  assigned to server  $j$  is

$$\hat{h}_j = \left( \frac{K_1 N}{\sum_{i=1}^m K_i} h_{1j} + \dots + \frac{K_V N}{\sum_{i=1}^m K_i} h_{Vj} \right) / N = \frac{\sum_{i=1}^m K_i h_{ij}}{\sum_{i=1}^m K_i}. \quad (4.14)$$

We apply the formula in 4.14 to get the following average holding costs for the vendors in the Example in Section 4.8:

Vendor $j$	1	2	3	4	5	6
$\hat{h}_j$	292.5	295	286.5	289	305	292.5

Table 4.4: Average Holding Costs for Vendors

Now we solve the single priority problem with the holding costs from Table 4.4, and repair costs and service rates from Table 4.3 on  $K = 1000$  items. The optimal allocation vector is  $(106, 83, 637, 73, 61, 40)$  and the long-run average yearly cost is \$197,520.56. This represents a 35.3% increase over the cost with the priority structure in place. The reason for the increase is that the higher priority items are in service for a longer portion of time and hence create a higher overall holding cost.

We next turn to an example where there are only two priority classes and their holding costs differ by only 10%. We consider 5 vendors and 10,000 total items, with  $K_1 = 2500$  and  $K_2 = 7500$ , so the majority of the items are low priority items. The items fail at rate 2 per year. The 5 vendors have the following properties:

Vendor $j$	1	2	3	4	5
$c_j$	20	18	23	16	25
$s_j\mu_j$	1000	200	400	600	700
$h_{1j}$	220	203.5	198	231	209
$h_{2j}$	200	185	180	210	190
$\hat{h}_j$	205	189.62	184.5	215.25	194.75

Table 4.5: Vendor Properties for Similar Cost Example

When the type 1 items are not given priority in service, the allocation vector is  $(1063, 238, 7293, 644, 762)$  and the total long-run average cost is \$1,374,210 per year. On the other hand, when the type 1 items are given priority in service we apply the  $\mathbf{G}_m$  algorithm, which produces the following allocation matrix:

$$X = \begin{pmatrix} 890 & 152 & 335 & 512 & 611 \\ 173 & 86 & 6958 & 132 & 151 \end{pmatrix}.$$

In this case, the long-run average cost is \$1,342,645 per year. The cost savings using the priority structure is 2.35%. Note that in this example the total number of items allocated to each vendor is the same regardless of the priority structure used. There are other examples where the total allocation does depend on the priority structure (e.g. the previous example). Clearly the priority structure is more beneficial when high priority items dramatically affect the holding cost. However, even in the case of similar holding costs among the different priority types, the priority structure can reduce cost by a small amount with relatively little effort.

## 4.10 Selecting the Values of $K_i$

Suppose that the manufacturer has  $K$  items to sell. The manufacturer has the option of offering up to  $m$  different warranty contracts. The contracts dictate the priority level, i.e. the most expensive contract will guarantee first priority in service, the second most

expensive contract guarantees second priority in service, etc. The manufacturer charges the customer  $r_i$  dollars for each item to have a type  $i$  priority in service ( $i = 1, \dots, m$ ). We assume that  $r_m = 0$  for the standard lowest priority warranty, and  $r_i > r_{i+1}$  for  $i = 1, \dots, m - 1$ . The values of  $r_i$  are fixed and are determined by the competition and the manufacturer cannot control these. We assume that the manufacturer can sell any number of items under any warranty contract. That is, he will sell all of the items regardless of the warranty contract(s) he offers. Typically, these assumptions are satisfied if the manufacturer is a small player in a large market.

Now, the manufacturer must decide on the values of  $K_i$  ( $i = 1, \dots, m$ ) in addition to the optimal allocation matrix. The optimization problem is:

$$\begin{aligned}
 \min \quad & \sum_{j=1}^V f_j(x_{1j}, x_{2j}, \dots, x_{mj}) \\
 \text{s.t.} \quad & \sum_{i=1}^m K_i = K, \\
 & \sum_{j=1}^V x_{ij} = K_i, \quad i = 1, \dots, m \\
 & x_{ij}, K_i \geq 0, \text{ and integer; } \quad i = 1, \dots, m, j = 1, \dots, V
 \end{aligned}$$

where  $f_j(x_{1j}, x_{2j}, \dots, x_{mj}) = \sum_{p=1}^m (f_{pj}(X_{pj}) - r_p K_p)$ . We can model the problem as a minimum cost network flow problem, similar to the problem discussed in Section 4.6.3 with the addition of a single node  $S$  that is connected to each of the nodes  $s_i$ ,  $i = 1, \dots, m$ . The node  $S$  is the single source node with  $b(S) = K$ . The nodes  $s_i$  ( $i = 1, \dots, m$ ) are now transshipment nodes, i.e.  $b(s_i) = 0$ . The arcs  $(S, s_i)$  have capacity  $K$  and cost  $-r_i$ . The rest of the network is the same as described in Section 4.6.3. We provide the network diagram in Figure 4.4.

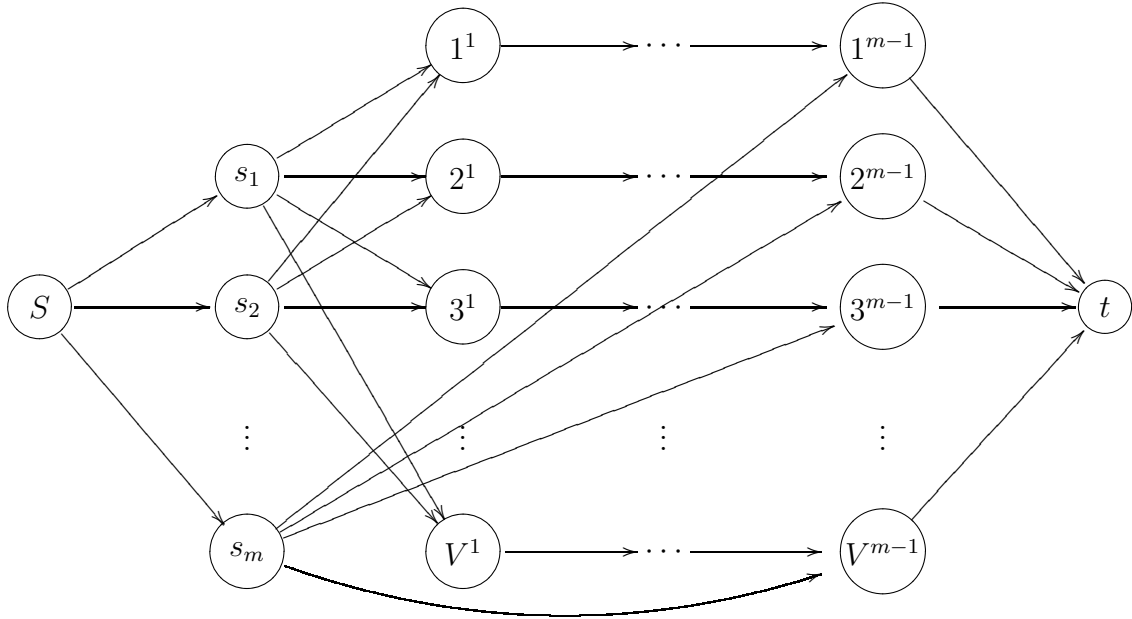


Figure 4.4: Network Model of Reward Problem

For ready reference, we provide the entire set of arc properties in Table 4.6:

Arc	Capacity	Flow	Cost	
$(S, s_i)$	$K$	$K_i$	$-r_i$	$(i = 1, \dots, m)$
$(s_1, j^1)$	$K$	$x_{1j}$	$f_{1j}(x_{1j})$	
$(s_i, j^{i-1})$	$K_i$	$x_{ij}$	$0$	$(i = 2, \dots, m)$
$(j^i, j^{i+1})$	$\sum_{p=1}^{i+1} K_p$	$X_{i+1,j}$	$f_{i+1,j}(X_{i+1,j})$	$(i = 1, \dots, m-1)$
$(j^{m-1}, t)$	$\sum_{p=1}^m K_p$	$X_{mj}$	$f_{mj}(X_{mj})$	

Table 4.6: Arc Properties for the Reward Network

We apply the successive shortest path algorithm to solve this problem by continually augmenting one unit of flow along the shortest path from  $S$  to  $t$ . Note that there are still no negative cycles in the graph. We illustrate the problem with an example.

**Example 4.1** Consider the example of Section 4.8, with  $r_i = (15, 10, 5, 0)$ . Here are the parameters of the problem:

Vendor $j$	$s_j \mu_j$	$c_j$	$(h_{1j}, h_{2j}, h_{3j}, h_{4j})$
1	80	15	(500, 350, 300, 175)
2	62	19	(500, 400, 250, 175)
3	70	18	(500, 350, 300, 160)
4	50	15	(500, 400, 250, 160)
5	45	14	(500, 400, 300, 175)
6	25	9	(500, 350, 300, 175)

Table 4.7: Costs and Service Rates for Each Vendor

Recall that the original values of  $K_i$  were (150, 250, 200, 400) and the expected yearly cost was \$146,012.42. If we use these values of  $K_i$  with the above values of  $r_i$ , the expected yearly cost is now \$140,262.40 after deducting the additional policy costs that the manufacturer charges. However, using the successive shortest path algorithm gives us an optimal allocation of

$$X = \begin{pmatrix} 34 & 25 & 27 & 17 & 16 & 0 \\ 13 & 0 & 10 & 0 & 0 & 6 \\ 0 & 11 & 0 & 6 & 0 & 0 \\ 52 & 42 & 573 & 95 & 43 & 30 \end{pmatrix},$$

with  $K = (119, 29, 17, 835)$ . The cost of allocation is \$112,326.61, a 20% reduction in cost over the previous solution. In this example, it is optimal for the manufacturer to offer all four warranty types. However, this may not occur in every example. ■

# Chapter 5

## Conclusions and Future Work

In this dissertation, we investigated two problems in warranty cost analysis. First, we addressed the problem of modeling a warranty reserve where contributions are made after each sale (where the sales process is a nonhomogeneous Poisson process). We derived the mean and variance of the reserve for all time points  $t$  and offered a heuristic for determining good values of  $c$ , the contribution amount after each sale, and  $R_0$ , the initial reserve level. We also provided extensions of the model by allowing a random contribution amount and multiple products from the same manufacturer. Next, we addressed the problem of outsourcing warranty repairs when items have priority in service. We provided the known result for the single priority case and used a network flow model to solve the multi-priority problem. We also provided a numerical comparison of the single-priority problem versus the multi-priority problem.

There are many extensions of our basic warranty reserve model for future research. One simple extension is to allow a fluctuating interest rate. We mention several other extensions below:

- **Bulk arrivals:** The sales occur in batches with a common probability distribution. The successive batch arrivals are independent of each other, but the warranty periods within each batch may be dependent.
- **Ruin Probability:** A common problem in risk theory is to compute the probability of ruin prior to time  $t$ . That is, we want to analytically compute the probability, in terms of the problem parameters, that the reserve account drops below a given amount  $B > 0$  prior to time  $t$ . Asmussen [3] provides an extensive reference on ruin probabilities. However, the specific case of premiums depending on the number of items or policies is not addressed.
- **Reserve-dependent Contributions:** We allow the reserve contribution to depend on the reserve level and the number of items under warranty. There are two possible approaches: to determine a contribution function  $c(R(t), X(t))$  prior to time 0 that can be applied for the entire fiscal period, or to dynamically determine the contribution amount at time  $t$ .
- **Criteria for Selecting  $c$  and  $R_0$ :** In this dissertation, we selected  $c$  and  $R_0$  so the reserve level remains above a target  $B$  with approximate  $100(1 - \beta)\%$  probability. We can consider other objectives for the manufacturer, such as minimizing a penalty function. That is, if the reserve falls below the target  $B$ , the manufacturer incurs sanctions as a function of the time below the target and the amount below the target.
- **Other Applications:** We believe adaptations of our reserve management model can be applied to pension funds and insurance accounts. Currently, the literature on these subjects considers a constant premium input and a static number of customers. For pension funds, we would track the workers currently eligible for pension and the workers eligible for pension in the future. The contribution amount can be a function of both of these, while the subtraction amount will depend on the

former. The insurance model would consider many classes of customers (as our model allows), but may have different failure distributions.

The second problem we addressed in this dissertation is the static allocation of prioritized warranty repairs. Some extensions of the static allocation model include:

- **Different Priority Structures:** We can consider other priority structures, such as non-preemptive priority. In this case, the computation of the expected queue lengths is more complicated and we lose the special structure of the objective.
- **Enforce Maximum Waiting Times:** If the contracts specify a maximum repair turnaround time for the different priority types, we should incorporate this into the model.
- **Simulation:** We can use simulation to determine if our allocation strategy performs well under different failure and service distributions.
- **Game Theoretic Problems:** In our research, we assumed that the manufacturer pays a fixed fee per repair to each vendor. Often these contracts are negotiated between the manufacturer and the vendor and may depend on the number of items allocated to that vendor.
- **Dynamic Allocation:** Another possible allocation strategy is dynamic allocation, i.e. the items are allocated only after a claim is made. Opp [27] uses index policies to address the single priority case. We are interested in how index policies can be used to estimate the optimal solution in the multi-priority case.

# Bibliography

- [1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993.
- [2] H. N. Amato and E. E. Anderson. Determination of warranty reserves: an extension. *Management Science*, 22(12):1391–1394, 1976.
- [3] S. Asmussen. *Ruin Probabilities*. World Scientific, Singapore, 2000.
- [4] Y. Balcer and I. Sahin. Replacement costs under warranty: Cost moments and time variability. *Operations Research*, 34(4):554–559, 1986.
- [5] W. R. Blischke and D. N. P. Murthy. *Warranty Cost Analysis*. Marcel Dekker, New York, 1994.
- [6] K. M. Bretthauer and B. Shetty. The nonlinear resource allocation problem. *Operations Research*, 43(4):670–683, 1995.
- [7] K. M. Bretthauer and B. Shetty. The nonlinear knapsack problem - algorithms and approaches. *European Journal of Operations Research*, 138(3):459–472, 2002.
- [8] B. D. Bunday and R. E. Scranton. The  $G/M/r$  machine interference model. *European Journal of Operations Research*, 4:399–402, 1980.
- [9] H. Cramér. *Collective Risk Theory*. Esselte Reklam, Stockholm, 1955.
- [10] L. W. Dowdy, D. L. Eager, K. D. Gordon, and L. V. Saxton. Throughput concavity and response time convexity. *Information Processing Letters*, 19:209–212, 1984.
- [11] S. G. Eick, W. A. Massey, and W. Whitt. The physics of the  $M_t/G/\infty$  queue. *Operations Research*, 41(4):731–742, 1993.
- [12] J. Eliashberg, N. D. Singpurwalla, and S. P. Wilson. Calculating the warranty reserve for a time and usage indexed warranty. *Management Science*, 43(7):966–975, 1997.
- [13] T. S. Glickman and P. D. Berger. Optimal price and protection period decisions for a product under warranty. *Management Science*, 22:1381–1390, 1976.
- [14] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley, second edition, 1985.
- [15] O. Gross. A class of discrete type minimization problems. Technical Report RM-1644, RAND Corporation, 1956.
- [16] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, Cambridge, MA, 1988.

- [17] S. S. Ja. *Computation of warranty reserves for non-stationary sales processes*. PhD thesis, University of North Carolina at Chapel Hill, 1998.
- [18] S. S. Ja, V. G. Kulkarni, A. Mitra, and J. F. Patankar. Warranty reserves for non-stationary sales processes. *Naval Research Logistics*, 49(5):499–513, 2002.
- [19] D. L. Jagerman. Some properties of the erlang loss function. *The Bell System Technical Journal*, 53(5):525–551, 1974.
- [20] A. A. Jagers and E. A. V. Doorn. On the continued Erlang loss function. *Operations Research Letters*, 5:43–46, 1986.
- [21] A. Y. Khintchine. *Mathematical Models in the Theory of Queueing (in Russian)*. Trudy Mat Inst. Steklov 49, 1955. English translation by Charles Griffin and Co., London, 1960.
- [22] S. A. Klugman, H. H. Panjer, and G. E. Willmot. *Loss Models: From Data to Decisions*. Wiley, 1998.
- [23] J. W. Mamer. Cost analysis of pro rate and free-replacement warranties. *Naval Research Logistics Quarterly*, 29:345–356, 1982.
- [24] J. W. Mamer. Discounted and per unit costs of product warranty. *Management Science*, 33(7):916–930, 1987.
- [25] W. W. Menke. Determination of warranty reserves. *Management Science*, 15(10):542–549, 1969.
- [26] E. J. Messerli. Proof of a convexity property of the Erlang B formula. *Bell System Technical Journal*, 51:951–953, 1972.
- [27] M. Opp. *Outsourcing Warranty Repair Services*. PhD thesis, University of North Carolina at Chapel Hill, 2003.
- [28] M. Opp, I. Adan, V. G. Kulkarni, and J. M. Swaminathan. Outsourcing warranty repairs: Static alloation. Technical Report 03-01, University of North Carolina at Chapel Hill, 2003. Submitted to Management Science.
- [29] C. Palm. Intensity variations in telephone traffic (in german). *Ericcson Technics*, 44:1–189, 1943. English translation by North-Holland, Amsterdam, 1988.
- [30] T. Rolski, H. Schmidli, V. Schmidt, and J. Teugels. *Stochastic Processes for Insurance and Finance*. Wiley, Chichester, England, 1999.
- [31] L. Takács. *Introduction to the Theory of Queues*. Oxford University Press, New York, 1962.
- [32] C. S. Tapiero and M. J. Posner. Warranty reserving. *Naval Research Logistics*, 35:473–479, 1988.

- [33] U.S. Federal Trade Commission Improvement Act (1975). 88 Stat 2183, Washington, DC, 101–112.
- [34] A. Zaporozhets. A short proof of optimality of the bottom up algorithm for discrete resource allocation problems. *Operations Research Letters*, 21(2):81–85, 1997.