Modeling and Analysis of a Variety of Exchangeable-Item Repair Systems with Spares

Thesis submitted in partial fulfillment of the requirements for the degree of "DOCTOR OF PHILOSOPHY"

by

Michael Dreyfuss

Submitted to the Senate of Ben-Gurion University of the Negev

July 22, 2015

Beer Sheva
Modeling and Analysis of a Variety of Exchangeable-Item Repair Systems with Spares

Thesis submitted in partial fulfillment of the requirements for the degree of "DOCTOR OF PHILOSOPHY"

by

Michael Dreyfuss

Submitted to the Senate of Ben-Gurion University of the Negev

Approved by the advisors _________________________________

Approved by the Dean of the Kreitman School of Advanced Graduate Studies

July 22, 2015

Beer Sheva
This work was carried out under the supervision of

Prof. Morton Posner

Prof. E. Korach

In the Department of Industrial Engineering and Management

Faculty of Engineering
Abstract

The goal of this research is to analyze a variety of models in exchangeable item repair systems with general repair distribution and ample servers. The goal in each system is to find the number of spares which optimizes some objective function. The research explores the behavior of two different systems when several classes of items or components are assumed. The first system assumes that a customer brings a product (consisting of several classes) which fails only via one of the items (ERSOF) and the second system assumes that the product can fail in more than one item (ERSMF). This thesis explores the behavior of several optimization models for the first system. In one class of model we optimize (i.e. max or min) a service criterion such as the fillrate, the average waiting time, the average number of customers in the system, or the probability that a customer waits longer than a specified time, under one or several linear constraints. In a second model class, we optimize (minimize) the amount of money invested for spares under one or several service-criteria constraints. In a third class of model we optimize several goals under several linear constraints, dubbed goal programming in today's literature. We provide a full solution of an example, and demonstrate different algorithms to solve the models. Sensitivity analysis is added to assist in exploring the behavior of the model. For the second system, an analytic formula is developed for the waiting time distribution for a system containing two item classes. This assumption can be relaxed easily in further research. Extensions to bulk arrivals, scrapping, multi-echelon systems are also analyzed and can be introduced easily into the basic model. As part of the validation and also as a new general technique, an integer programming approach is used to show how to solve all of our smaller problems. Finally an item-based approach is developed to find the distribution of the number of customers in the system, which was already presented incorrectly by Hausman and Cheung.
Acknowledgments

First, I wish to thank Prof. Posner for his help, for his advice and especially for his friendliness and readiness to help me during this research. I also wish to thank Prof. Korach for his help and guidance during my studies at Ben Gurion University. Third, I wanted to thank Dr. Dvir Shabtay and the other referees for their helpful comments which helped me to improve this thesis. Finally, I want to thank my wife, who supported me with love and understanding during this research. Without her, I could not have finished this research.
# Table of Contents

Abstract ........................................................................................................................................ i
Acknowledgments ..................................................................................................................... ii
Table of Contents ....................................................................................................................... iii
List of Figures ............................................................................................................................ vii
List of Tables ............................................................................................................................. xi
Part I ............................................................................................................................................... 1
1. Introduction ............................................................................................................................. 1
   1.1. Introduction ....................................................................................................................... 1
   1.2. Literature review ............................................................................................................... 3
   1.3. Objectives of current research ............................................................................................ 6
2. The Basic Model ....................................................................................................................... 9
   2.1. The exchangeable-item repair system ............................................................................... 9
   2.2. The basic analytical model .............................................................................................. 11
      2.2.1. The waiting time distribution ....................................................................................... 12
   2.3. The software package developed for the basic model ..................................................... 14
3. The Exchangeable-item Repair System with Several Item Classes .................................... 17
   3.0. Introduction ....................................................................................................................... 17
   3.1. The model for an exchangeable-item repair system when failures are from a single class (ERSOF) ................................................................................................................. 17
   3.2. Service measures .............................................................................................................. 19
   3.3. Defining the optimization model ...................................................................................... 23
   3.4. Optimizing the ERSOF model ......................................................................................... 24
      3.4.1. The fillrate of the system ............................................................................................ 24
      3.4.2. The average waiting time of a customer in the system ............................................ 26
   3.5. Algorithms: ....................................................................................................................... 30
   3.6. Validation of the results of the algorithms ....................................................................... 33
   3.7. Post-optimization ............................................................................................................. 35
   3.8. Numerical example ........................................................................................................... 35
3.8.1. Sensitivity analysis about the budget allocation, M................................................. 39
3.8.2. Sensitivity analysis on arrival rate ................................................................. 42
3.9. Model validation ......................................................................................... 43

4. Modeling an Exchangeable-Item Repair System with more than one Failure Class (ERSMF) ...................................................................................... 47

4.0. Introduction .................................................................................................. 47
4.1. Service measures ........................................................................................ 49
  4.1.1. The Waiting Time Distribution ................................................................. 50
  4.1.2. The Average Waiting Time ...................................................................... 51
  4.1.3. The Fillrate .............................................................................................. 52
  4.1.4. The number of customers in system ....................................................... 53
  4.1.5. The average number of customers in the system .................................. 53

Part II ............................................................................................................. 54

5. Optimizing Additional Service Criteria....................................................... 55

5.0. Introduction .................................................................................................. 55
5.1. Solving the fillrate problem using dynamic programming ....................... 55
5.2. An approximate Dynamic Programming algorithm for large scale problems .. 57
5.3. Developing another algorithm for the fillrate ........................................... 58
5.4. The total number of customers in the system ......................................... 61
5.5. The probability of waiting less than x ....................................................... 62

6. Different ERSOF-Models ........................................................................... 63

6.0. Introduction .................................................................................................. 63
6.1. The Average Waiting Time ........................................................................ 63
  6.1.1. Validation of the algorithm .................................................................... 67
6.2. The fillrate ................................................................................................... 68
6.3. Connecting the constraint and the objective function ............................ 71
  6.3.1. The fixed time penalty in an ERSOF-model ......................................... 71
  6.3.1.1. The fixed time penalty in an ERSOF-model for a given period .......... 72
  6.3.1.2. The fixed time penalty in an ERSOF-model for an infinite period ....... 73
6.3.2. The fixed penalty for any not immediately satisfied customer in an ERSOF model 74
6.3.2.1. The fixed penalty for any not immediately satisfied customer in an ERSOF model for a given period $\tau$ .......................................................... 74
6.3.2.2. The fixed penalty for any not immediately satisfied customer in an ERSOF model for an unbounded interval .......................................................... 75
6.3.3. The fixed penalty for delay longer than time $x$ in an ERSOF model .......... 77
6.3.3.1. The fixed penalty for delay longer than time $x$ in an ERSOF model for a given fixed time period .......................................................... 77
6.3.3.2. The fixed penalty for delay longer than time $x$ in an ERSOF model for over an unbounded interval .......................................................... 77

7. Multiple Goals ................................................................. 79
7.0. Introduction........................................................................ 79
7.1. Combining two objectives .................................................. 79
7.2. Missing the goals ................................................................. 83
7.3. Goal programming ................................................................. 83

8. Integer Programming .......................................................... 86
8.0. Introduction........................................................................ 86
8.1. The average waiting time..................................................... 86
8.2. The fillrate......................................................................... 89
8.3. The probability of waiting time $x$ ........................................ 92
8.3.1. Sensitivity analysis .............................................................. 93
8.4. The probability of waiting more than $x$ for waiting customers .......... 95

9. Bulk systems ........................................................................ 98
9.0. Introduction........................................................................ 98
9.1. An ERSOF - system where every customer brings exactly 2 failed items of the same class.............................................................................. 98
9.2. An ERSOF-system where each customer brings exactly $b$ failed items of the same class which failed................................................................. 104
9.3. An ERSOF-system where each customer brings a random number $B$ of failed items of the same class................................................................. 109

10. Developing ERSMF-models.................................................. 114
10.0. Introduction...................................................................... 114
10.1. An ERSMF-model where all the customers are of type 3 .............................. 114
10.2. An ERSMF-model, where the systems contains CT1, CT2 and CT3 ......... 128
    10.2.1. When the CT3 is arriving ........................................................................ 128
    10.2.2. When the CT1 is arriving ........................................................................ 134
    10.2.3. When CT2 is arriving ................................................................................ 136
    10.2.4. The Waiting Time Distribution of a random customer .......................... 136
    10.2.5. Service Measures ..................................................................................... 138

11. Model Extensions ............................................................................................. 144
    11.0. Several model extensions ........................................................................... 144
    11.1. Scrapping in the basic model ...................................................................... 144
    11.2. Multi-echelon systems ............................................................................... 145
    11.3. Ample Servers ............................................................................................. 146

12. Summary and further research ...................................................................... 148

Notation summary ................................................................................................. 151

Appendix A: Research to solve the EROSF model using Normal Approximation ................................................................................................................................. 153
Appendix B: Using the software developed for the basic model ................. 155
Appendix C: The error of Hausman and Cheung ............................................ 159
Appendix D: Simulating the numerical result of an ERSOF-model .. 160
Appendix E: Proof of the waiting time distribution ........................................... 163
Appendix F: Proof of the Waiting Time Distribution of a ERSMF-model ................................................................................................................................. 164
Appendix G: Developing formula (6.10) .............................................................. 167
Appendix H: Proof of the waiting time distribution of a ERSMF-model ................................................................................................................................. 168
Appendix I: Check mechanism for the ERMF-formula ................................. 170

References .............................................................................................................. 173
List of Figures

Figure 1.1: Lifecycle of an item ................................................................. 2
Figure 1.2: Division of the Literature review of Repair Facilities .................... 5
Figure 2.1: Lifecycle of an item ....................................................................... 9
Figure 2.2: Load Data Dialog to get the data from the user. .............................. 14
Figure 2.3: Result Matrix of Optimality .......................................................... 15
Figure 2.4: Interface of the software ............................................................... 15
Figure 3.1: Model of ERSOF ......................................................................... 18
Figure 3.2: Example of the behavior of the fillrate ............................................. 26
Figure 3.3: An example of the average Waiting Time of a basic system .............. 28
Figure 3.4: Converting graph of Figure 3.3 to piece-wise linear ......................... 29
Figure 3.5: Input and Output data of the software ............................................ 31
Figure 3.6: Results of the second algorithm ...................................................... 32
Figure 3.7: The Cumulative waiting time for the numerical example .................. 37
Figure 3.8: The density function of the waiting time in the system .................... 38
Figure 3.9: The distribution of the number of customers in the system ............... 38
Figure 3.10: The average waiting time for the different budgets ....................... 39
Figure 3.11: The fillrate of the optimized system when adding increasing the budget ..................................................................................................................... 39
Figure 3.12: An example of the waiting time distribution when the budget changes 40
Figure 3.13: All different waiting time distribution when the budget changes ...... 40
Figure 3.14: The effect of the budget on the waiting time distribution ................ 41
Figure 3.15: The effect of the arrival rate on the average waiting time and the fillrate ............................................................... 42

Figure 3.16: The effect of the arrival rate on the cumulative waiting time distribution ............................................................... 42

Figure 3.17: The output of the simulation for the number of customers in the system ............................................................................. 44

Figure 3.18: Comparison of the outputs of the simulation and the analysis ............................................................... 44

Figure 3.19: The waiting time distribution of the simulation ............................................................................. 45

Figure 3.20: Comparison of the waiting time density function of the simulation and analysis ............................................................................. 45

Figure 4.1: The repair facility with several item classes and several failures ............................................................... 47

Figure 5.1: Graph which shows the results given in Table 5.1 ............................................................................. 57

Figure 5.2: The Optimal Fillrate when calculated with different steps ............................................................................. 58

Figure 5.3: The Fillrate of a sub-system ............................................................................. 59

Figure 5.4: Output for the fillrate when 1000 dollars are available ............................................................................. 60

Figure 7.1: An example of the average waiting time versus budget allocation ............................................................................. 66

Figure 7.2: An example of budget allocation versus the average waiting time ............................................................................. 66

Figure 7.3: Example of the interface of the first program with 1000 dollars ............................................................................. 67

Figure 7.4: Example of the interface of the second program with the AWT given ............................................................................. 67

Figure 7.5: Example of the interface of the first program with 500 dollars ............................................................................. 68

Figure 7.6: Example of the interface of the second program with the AWT given ............................................................................. 68

Figure 7.7: Graph of the fillrate versus dollars of ss_i ............................................................................. 70

Figure 7.8: Money versus the fillrate of ss_i ............................................................................. 70

Figure 7.9: Total cost of ss_i ............................................................................. 75

Figure 8.1: Example of the behavior of the fillrate ............................................................................. 80
Figure 8.2: Graph of the second part of the equation .................................................. 81
Figure 9.1: Optimal spares vector $\bar{n}$ for the Integer Problem .................................. 88
Figure 9.2: Optimal vector $\bar{n}$ for the Linear Problem ............................................... 88
Figure 9.3: Optimal spares vector for the fillrate using integer programming .......... 90
Figure 9.4: Optimal spares vector for the fillrate using linear programming. ........ 90
Figure 9.5: Graph of the overall average waiting time...................................................... 90
Figure 9.6: Graph of the overall fillrate......................................................................... 91
Figure 9.7: Graph of the overall fillrate with smaller intervals. ................................. 92
Figure 9.8: Optimal spares vector for the probability of no waiting ....................... 93
Figure 9.9: The probability of waiting less than x with different budget allocation. 94
Figure 9.10: Waiting Time Distribution based on different budget allocation. ....... 94
Figure 9.11: Output of the conditional probability of waiting more than x. ............ 96
Figure 9.12: The conditioned probability of waiting more than x. ............................. 96
Figure 9.13: Sensitivity analysis of different budgets for a given x. ............................ 96
Figure 10.1: Customer arriving in $(0,t)$ and leaving after $t+x$ .................................. 100
Figure 10.2: Customer arriving in $(t,t+x)$ and having his item repaired before $t+x$.
......................................................................................................................................... 102
Figure 11.1: Graph of customer arriving in du (before t) and item of class 1 not
repaired and item of class 2 not repaired before t+x....................................................... 117
Figure 11.2: Customer arriving in du (0,t) and having the item of class 2 repaired
before $t+x$ and the item of class 1 repaired after $t+x$..................................................... 118
Figure 11.3: Customer arriving in (0,t) and having his item of class 1 repaired
before $t+x$ and his item of class 2 repaired after $t+x$ ...................................................... 119
Figure 11.4: Customer arriving in (0,t) and having his item of class 1 repaired
before $t+x$ and his item of class 2 repaired before $t+x$.................................................. 120
Figure 11.5: Customer arriving in du (after t) and having his item of class 1 repaired before t+x and his item of class 2 repaired before t+x .......................................................... 121

Figure 11.6: Customer arriving in du (after t) and his item of class 1 repaired and his item of class 2 not repaired before t+x. .................................................................................. 121

Figure 11.7: Graph of customer arriving in du(after t) and item of class 2 repaired and item of class 1 not repaired before t+x. .......................................................... 122

Figure 11.8: Customer arriving in (0,t) and having his item of class 1 and his item of class 2 repaired after t+x .................................................................................. 123
List of Tables

Table 3.1: Spreadsheet of a numerical example.......................................................... 34
Table 3.2: Four different approaches to solve the problem........................................ 34
Table 3.3: Data of the system with 13 classes of items.............................................. 36
Table 3.4: Service-level measures for the numerical example..................................... 37
Table 3.5: The results of the software and of the simulation....................................... 43
Table 4.1: Example of the customer types with two item classes in the system......... 48
Table 5.1: Time (in seconds) for different amounts of classes and budget.............. 57
Table 9.1: Example of the Spreadsheet.................................................................... 87
Table 9.2: Table of the average waiting time for each class i and for each value j... 88
Table 9.3: Values of the fillrates for each ss_i used in the integer programming...... 89
Table 11.1: Example of the customer types with two item classes in the system.... 114
Part I
1. Introduction

1.1. Introduction

An exchangeable-item repair system is a system to which customers bring a failed product, consisting of a number of items, for repair, and receive serviceable ones in return which are then reinstalled into the product. Items are considered exchangeable in the sense that customers are ready to take any serviceable item of the same kind they brought to the system. The system operates repair facilities in which the failed items are repaired, if not scrapped, and turned into serviceable ones. New items can be added during work and removed based on the needs of the system. The arriving customers join a queue—the customer queue—to wait for serviceable items, if at their arrival the system does not have available items on the shelves (stock). The failed item is sent to a repair facility where it will be scrapped or repaired. After repair, the item will be stocked on the shelves for further use. This stock contains new items, which are called spares (at the beginning of running the system and added items during the processing of the system) and repaired items (during work). A customer will get an item from this stock whenever there are items in the stock. After getting such a serviceable item which is then inserted into the product, the customer leaves the system. Obviously, when the system contains spares, the customer is served better and quicker. Based on this system, the following model was depicted in Figure 1.1:
This basic model contains **no scrapped items, and no new spares**. This means that during work no new spares can be added. At the beginning there are exactly \( n \) spares on the shelf. Hence, the number of items in the system will always be at least \( n \), because every customer brings exactly one item and ultimately receives one, but it does not have to be the one he brought to the system. Thus, if there are customers waiting in the queue, this means that there are no good items on the shelf, and therefore, the number of items in the repair facility equals the number of customers waiting plus the number of spares. If there are no customers waiting, then \( n \) items will be either in the repair facility or on the shelf. Hence,

\[
\text{Total number of items in the system} = n + \text{Number of customers waiting in the queue.} \tag{1.1}
\]

The repair facility contains an ample number of servers. This means that the repair time of an item is independent of the number of items in the repair facility. The customer waiting time will be developed later as a function of the number of spares in the system.
The repair time, measured from the instant of arrival, has a distribution $G(\cdot)$ with mean $\frac{1}{\mu}$. Conditioning on the number of spares $n$, and assuming equilibrium conditions prevail, we let $W_n$ be the time a customer waits until he gets a serviceable item, and $X_n$ the corresponding number of customers in the system. Then, $P(W_n < t)$ is the probability that a customer waits less than time $t$ ($t > 0$), and $P(X_n = r)$ is the probability that there are $r$ customers in the system in steady state, $r = 0, 1, 2, \ldots$. To complete the model, we will assume that customers arrive independently, with the time between arrivals distributed exponentially with mean $\frac{1}{\lambda}$. Thus, customer arrivals follow a homogenous Poisson process with rate $\lambda$. A system such as that described must offer ways to measure its efficiency via some service level. It can be measured by focusing on the customer or on the system. In most cases they are equivalent. An example of a "customer service-level" is "the probability that a customer waits less than half an hour is more than 95%". Another example for the service level of the system is the average number of customers waiting in the system.

1.2. Literature review

When repair facilities are examined, several aspects are considered. Different systems can be divided into different classes which share common properties. The first division to make is the property of spares. We can deal with systems with spares or without spares. In either case, exchangeability of the failed items is assumed. This means that a customer who arrives at a repair location bringing a failed product which is composed of several items, does not mind getting other items as long as they are serviceable and can be included in his product. This class of system is common in today’s repair systems. Our research goal is to analyze systems with spares (exchangeable-item repair systems), which are more complex and which deal with a variety of different assumptions and approaches.

When an exchangeable-item repair system is examined, it can be that the items are not repaired at this repair facility but at another location, so that these failed items must be sent further (two or multi-echelon systems). For example, in the army, centralized
depots are often used to replenish local depots to minimize the number of spares for the weapons systems (Sherbrooke [26],[27],[28] and Albright [2],[3]). Multi-echelon systems have received considerable attention in the literature because of their importance for industry or the army. Several topics such as batching (Lee H and K. Moinzadeh [20]), different shipment times (Graves [11],[12],[13]) and priority shipment (Dada [9]) cover many different possibilities of multi-echelon systems with some possible variations.

We will first consider systems where items are repaired at one location or sent further but with negligible additional delay (single-echelon system). To create models of single-echelon systems, several assumptions are made such as:

- Assuming general repair time distribution of an item or some special one (exponential: Posner [6], Daduna [10] and others).
- Assuming that arrivals of customers bringing products follow a Poisson stream or not.
- Assuming that the system contains a small number of servers (finite: Berg & Posner[6], Daduna [10]) or a large amount (ample, Berg & Posner [5])
- Assuming that the items of the product can be scrapped or they all must be repaired.
- Assuming that the customers wait for service and do not leave, or they leave if they anticipate that they may have to wait too long.
- Assuming steady-state behaviour or not.
- Assuming dynamic repair systems which include scrapping and buying new spares or not.
- Assuming fixed shelf-lives when items are perishable (Perry and Posner [24]) or not.
- Assuming that the product which the customer brings includes one item or several.
- Assuming that the product which the customer brings includes several items but the product failed because ONE of the items failed or several items can fail.
In the literature, many of these and other assumptions are made in order to create models for special cases. A general model including ALL the different assumptions is still not feasible due to its extreme complexity as well as the complexity of the mathematics which will be required to solve these models. Special cases, which include several of these assumptions, can be found in the literature. Figure 1.2 depicts a diagram of the different research fields of different models of Repair Facilities. We concentrated our research on the M/G/∞ class of models. These classes of models have in common that customer arrivals follow a Poisson stream, the repair times of items follow any general distribution, and the system includes an infinite number of servers.

Sherbrooke [21-23] developed models optimizing the availability of the products and the number of products waiting for repair in an multi-echelon system. The waiting time of the product was not included, nor even mentioned. Posner and Berg [5] developed
the mathematical formulas for the waiting time, which were apparently not used. They all developed a model for products consisting a single item class, which may also include arrivals in bulk (the product contains more than one of the same item class).

Hausman and Cheung [16] tried to solve a more complex model. They developed a model for a product with multiple failures types; this means that a customer arrives to a system with a product which can fail in several ways and waits until his product is repaired. We found that their formulas were not correct and this is shown in Appendix C. Another interesting model presented by Jing-Sheng Song [30] is not a repair system but an ordering system with spares where the customer can choose different items. In fact, the models of repair and ordering deal with similar service measures and problems. Solving repair systems can affect ordering systems and vice-versa. Thus, a deep and broad knowledge of different systems is required for a researcher starting to develop new models. Recent researchers have focused their work on multi-echelon systems adding different assumptions regarding service criteria [33], [8], limited capacities [17] and more [31].

1.3. Objectives of current research

As explained in the literature review, a model for an exchangeable item repair system with spares which deals with only ONE item class already exists. But what happens if the product contains more than one item, which is the case in almost all products? There are no answers for that problem; Hausman and Cheung did approach this problem and tried to solve it, albeit unsuccessfulty.

A product can fail through exactly one item failure or through multiple item failures. Therefore, in this work we will concentrate on exchangeable item repair systems with spares where each product brought to the repair system includes more than one item. Spares are added to the system to improve the service level. Due to the fact that the system has an exchangeability property, which means that customer do not mind getting another item as long as it serviceable, adding spares to the system will automatically improve the availability of items to replace the failed ones.

The assumptions for all the models in this research are:
Assumption 1: The repair times for all items follow a general distribution function.

Assumption 2: The arrivals of the customers bringing products follow a Poisson stream.

Assumption 3: The system contains an ample number of servers.

Assumption 4: The items of the product are repaired and not scrapped.

Assumption 5: The customers wait for service and do not leave.

Most models assume that the repair time is distributed exponentially to ease mathematical analysis. Here, the repair time can follow any type of distribution function. Another assumption we make is that the customer arrivals are independent. This means that they do not arrive in groups, at given times, but they arrive at random. This is called in literature a Poisson Process. When customers arrive to the system, they enter a customer queue and wait until they get service. They will wait and might leave the system before satisfaction. Dealing with problems where customers leave the system before satisfaction is beyond the current research scope. We initially assume that the repair facility contains ample servers. All these assumptions create a small field of classes of models which can be explored. In this reduced field, we found two articles which fit into these assumptions. Hausman and Cheung [16] and Berg and Posner [5].

Step by step, we will analyze and relax assumptions so that the reader can easily follow the analysis of the different models. In Part I, we will analyze more standard problems, whereas in Part II (from Chapter 5), we will develop more complex models. Thus, in Chapter 2, we will present the basic model developed by Berg and Posner [5] and different aspects which can help us in the further research. In Chapter 3, we will present our first system: The ERSOF-system. This is a system to which customers bring a product which failed only in ONE item. We will define two models: One where the objective function is to maximize the fillrate and one where the objective function is to minimize the average waiting time under a budgetary constraint on expenditures for spares. Unfortunately, using standard mathematical techniques the model with the fillrate can not be solved. In Chapter 3, we will look at a specific bicriteria type of problem – we will maximize the service level subject to a budget limitation. After analyzing one
ERSOF-model, we will look at the ERSMF-model. An ERSMF-model is a model of a system to which customers brings products which failed via more than one item. In Chapter 4, we will present the system and its service criteria. Due to the fact that this model cannot be solved using standard techniques, we will leave this model to Chapter 10 to be analyzed and solved. This concludes Part I, which presents basic ideas and simple models of the research.

In Part II of the research which starts at Chapter 5, we will introduce more complicated models, develop new analytic formulas, and present new techniques for solving general problems. In short, the research becomes more complex but also more interesting for the reader. In Chapter 5, we will optimize a model with the fillrate as the service levels criteria and also the number of customers in system as another criterion. In Chapter 6, we will analyze a second bicriteria type of problems – we want to minimize the total investment subject to a service level. Instead of a non-linear objective function with linear constraints, we look at models where the objective function is linear and the constraints are non-linear. In Chapter 7 we will focus on multiple goals. We will define a third bicriteria type of problems – we want to minimize an aggregate objective function with no constraints; this means that a manager can define multiple goals, which is a novelty in this research. Another novelty is the technique to solve problems by integer programming to be presented in Chapter 8. Throughout, we should not forget that one of the minor goals of this research is to correct the formula of Hausman and Cheung. For this purpose, we need to develop the waiting time distribution for bulk arrivals of an ERSOF system (in Chapter 9), and the waiting time distribution for the ERSMF system (in Chapter 10). We will present a technique to correct the formula of Hausman and Cheung. Finally, we will introduce new features in the model such as scrapping, and analyze the impact of multi-echelon systems and the ample servers assumption. This is what is awaiting you, dear reader, in the following chapters. This research combines new techniques, nice examples, mathematical developments and new formulas with support whenever possible, using sophisticated software which was developed to illustrate numerical examples and validate the algorithms.
2. The Basic Model

2.1. The exchangeable-item repair system

An exchangeable-item repair system is a system to which customers bring a failed product consisting of items for repair, and receive serviceable ones in return which are reinstalled into the product. Items are considered exchangeable in the sense that customers are ready to take any serviceable item of the same kind they brought to the system. The system operates repair facilities in which failed items are repaired, if not scrapped, and turn them into serviceable ones. New items may be added during work or removed based on the needs of the system.

An arriving customer joins a queue (the customer queue) to wait for a serviceable item; if, upon arrival, the system does not have any available items on the shelf, he will have to wait. But if there are good items on the shelf he takes one and prepares to leave.

![Diagram of the lifecycle of an item](image)

**Figure 2.1: Lifecycle of an item**

The failed item is sent to a repair facility where it will be scrapped or repaired. After
repair, the item will be placed onto the shelf for further use. This shelf contains only new items, which are called spares (at the beginning of running the system and added items during the processing of the system) and repaired items (during work). A repaired item is considered as good as new. Obviously, when the system contains spares, the customer is served quicker. Based on this system, the following model was created:

This model (the basic model) contains **no scrapped items, and no new spares**. That means that during work no new spares can be added. At the beginning there are exact \( n \) spares on the shelf. The number of items in the system will therefore always be at least \( n \), because every customer brings exactly one item and eventually gets one, which need not to be the one he brought to the system. Thus, if there are customers waiting in the queue, this means that there are no items on the shelf and therefore, the number of items in the repair facility is equal to the number of customers waiting plus the number of spares. If there are no customers waiting, then \( n \) items will be either in the repair facility or on the shelf.

**Total number of items in the system** = \( n + \text{Number of customers waiting in the queue} \). \hspace{1cm} (2.1)

The repair facility contains an ample number of servers. This means that the time an item requires to complete repair follows a general distribution \( G(. \) ) and is independent of the number of items in the repair facility. The customer waiting time will be developed later as a function of the number of spares in the system.

**Service level criteria:**

**Fillrate – \( FR_n \)**
The fillrate is the probability that a customer bringing a product to a system stocked with \( n \) spares, gets the failed item in his product replaced straight away. This means that the customer is not required to wait for satisfaction. The more spares there are, the higher the probability that a customer gets his replacement item directly.

**Average Queue size – \( L_n \)**
The average queue size is the average number of customers waiting for a serviceable item in a system with n spares.

*Average Waiting Time - $\bar{W}_n$*

The average waiting time is the average time a customer must wait until he receives a serviceable item in a system with n spares.

*The Probability of waiting more than x - $P(W_n > x)$*

The average waiting time may not be the main criteria. Another important criterion is the probability that a customer will wait longer than x minutes. The manager will not be happy when, say 10% of the customers wait more than 20 minutes although the average waiting time is 10 seconds. In most cases, the average waiting time is a good indication how the systems works, but not how the customer is served. Thus, another measure for the system is the probability that a random customer will wait more than some specified x.

*The Probability of waiting more than x under the condition that the customer waits - $P(W_n > x | W_n > 0)$*

What happens when 90% of the customers do not wait and only 1% of the customers wait more then 20 minutes? The manager of the repair facility will think that his facility works fine. But when looking more specifically at all customers who do wait, we may see that more than 50% wait longer than 20 minutes. Therefore, the probability of waiting more than x may not be adequate in certain cases. Thus, we introduce the probability of waiting longer than x given that the customer must wait ($P(W_n > x | W_n > 0)$).

### 2.2. The basic analytical model

The basic analytical model was developed by Berg and Posner [5] and is summarized below.
2.2.1. The waiting time distribution

In an exchangeable-item FIFO system with n spares that satisfies the assumptions of the basic model (Assumptions 1-5, Page 8), we define $W_n$ as the steady-state waiting time of an arriving customer, with density $f(t \mid n)$ and c.d.f $F(t \mid n)$. Thus,

$$P(W_n \leq t) = F_n(t) = F(t \mid n) = P(Y_1(t) - Y_2(t) \leq n - 1) + G(t)P(Y_1(t) - Y_2(t) = n), \quad 0 \leq t < \infty$$

(2.2)

where $Y_1(t)$ and $Y_2(t)$ are independent generic Poisson random variables, and $G(t)$ is the cumulative repair time distribution. Here, $Y_1(t)$ and $Y_2(t)$ are Poisson with parameters $\lambda_1(t)$ and $\lambda_2(t)$, respectively, where

$$\lambda_1(t) = \lambda \int_u^\infty G(u)du$$

(2.3)

and

$$\lambda_2(t) = \lambda \int_0^t G(u)du$$

(2.4)

$F(t \mid n)$ is the probability that the customer will get an item by time $t$ given $n$ spares. Therefore, with probability $1-F(t \mid n)$, he must wait longer than $t$. For an explanation, please see Berg and Posner [5].

2.2.2 The average waiting time and the average number backlogged

The average waiting time is the average time a customer waits until he gets a serviceable. Thus,

$$\bar{W}_n = \int_{t=0}^\infty tf(t \mid n)dt$$
\[ W_n^2 = \int_{t=0}^{\infty} t^2 f(t \mid n) dt \]

\[ \int_{t=0}^{\infty} (1 - F(t \mid n)) dt \]

Let \( X_n \) be the steady state number of customers in the system, and \( P(X_n = r) \) \( r = 0, 1, 2, \ldots \) the probability that there are \( r \) customers in the system. Then, \( P(X_n = r) \) is equal to the probability that there \( r + n \) items in the system (2.1), since every customer brings one failed item and there are \( n \) spares in the system. Let \( Y_n \) be the number of items in the system. Thus, the probability that there are \( Y_n \) items in the system is Poisson with parameter \( \frac{\lambda}{\mu} \) (since the model for items is M/G/\( \infty \)), and

\[ P(Y_n = y) = \frac{(\lambda / \mu)^y}{y!} e^{-\lambda / \mu} \quad \text{for} \quad y = 0, 1, 2, \ldots \]

By (2.1), \( X_n + n = Y_n \).

At any time the number of customers waiting and the number of spares \( n \) will be equal to the total number of items in the system (in stock and in repair).

Thus,

\[ P(X_n = r) = P(Y_n = r + n) = \frac{(\lambda / \mu)^{r+n}}{(r+n)!} e^{-\lambda / \mu} , \quad r = 1, 2, 3, \ldots \]

and

\[ P(X_n = 0) = 1 - \sum_{r=1}^{\infty} P(X_n = r) \]

where \( n \) is the number of spares.

The average number of customers in the system is by definition:
\[ L_n = \sum_{r=1}^{\infty} r \cdot P(X_n = r), \]  

and the variance of the number of customer in the system is

\[ V(X_n) = \sum_{r=1}^{\infty} r^2 \cdot P(X_n = r) - (L_n)^2. \]  

These two values are important for the service level criteria. Little’s formula connects \( \bar{W}_n \) and \( L_n \) by

\[ L_n = \lambda \cdot \bar{W}_n \]  

2.3. The software package developed for the basic model

A flexible software package was developed to support further research into exchangeable-item repair systems. Since the functions (2.2-2.5) are generally difficult to calculate because of the large number of operations, the software package assisted our understanding of how those functions behave. In addition, sensitivity analysis was added to complete our understanding of the basic model. As an example, to calculate the waiting time distribution, we need data about the repair time and the arrival rate. The software enables the user to choose different repair time distributions, such as Normal, Gamma and Exponential distribution. (Fig 2.2)

Figure 2.2: Load Data Dialog to get the data from the user.
The software then gives a user-friendly interface to handle given data and to support decisions, such as the Result Matrix (Fig 2.3), which also provides detailed sensitivity analysis of the mean repair time and the arrival rate of the customers. Figure 2.3 shows

### Results of Sensitivity Analysis

<table>
<thead>
<tr>
<th>Failure Rate</th>
<th>Optimal Number</th>
<th>Mean Repair Time [hours]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>0.75</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>0.9</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>1.1</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4.5</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>5.5</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>7.5</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>

The Optimal Number

Figure 2.3: Result Matrix of Optimality

Figure 2.4: Interface of the software the result matrix. On the left side, there are different arrival rates and on the top different mean repair times. In the example shown, the mean repair time was 1 hour and the failure
rate 5/hour. In this example a value was chosen for the fillrate. The result matrix shows the number of spares to buy to achieve this condition. You observe that the optimal value is 10. The figure shows the sensitivity matrix which supports the decision maker. Another screen of the software shows additional material, such as graphs of the waiting time distribution, the repair time and additional service levels (Figure 2.4). New software features were also added such as buttons to easily change the total number of spares(N+,N-) and service level constraints to make the software even more user friendly.

This software package was not modified to include all new results of the research, but it served as a basic tool in understanding the given structure of the basic model and in analyzing more complex models.
3. The Exchangeable-item Repair System with Several Item Classes

3.0. Introduction

An exchangeable-item repair system with several classes of failure is a system to which customers bring one product which has failed. However, the cause of the failure may be via any one of several items within the product. In a modular sense, the product may be composed of a number of component modules (items), and a failure in any component results in the failure of the product. When the customer arrives with a failed product to the system, the product is analyzed and the failed items are sent for repair. For better service (the customer would like to wait less), the system keeps a number of spares from each component class. Each class of item has a stock of items and a repair facility where the items are repaired. In fact, these stocks and repair facilities need not be physically separated. After repair, the items join the stock. The customer leaves the system when his product is repaired.

In this following chapter, we will deal with a variety of models. One of the service measures (the average waiting time) will serve as the objective function to be optimized under a budgetary constraint. Then, we will introduce the fillrate as the objective function and show why we cannot use KKT-methods. The ERSOF-model with the fillrate as objective function will be analyzed in the second part. Finally, we will introduce the average waiting time as a service-level and solve the model, including post-optimality, through a numerical example.

3.1. The model for an exchangeable - item repair system when failures are from a single class (ERSOF)

In this model, we make the assumption that only one component module (item) can fail at a time, and therefore, the following model was developed. Customers arrive to the system with their product at the main desk where the failed item class is identified and then routed for repair. We suppose there are I classes of failure possible and the overall arrival rate \( \lambda \) is partitioned into rates \( \lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_I \) corresponding to the individual failure
rates. We will define \( a_i \) as the proportion of class-\( i \) failures \((i=1,2,\ldots,I)\), so that \( \lambda_i = \lambda \) and \( \sum_{i=1}^{I} a_i = 1 \). Thus, since the product arrival stream is Poisson with rate \( \lambda \), the class-\( i \) item failures form a Poisson stream of rate \( \lambda a_i \). These are “sent” to a sub- ample server repair facility dealing only with class-\( i \) failures, which is stocked with spares at level \( n_i \). We call this subsystem, subsystem \( i \), or \( ss_i \). Again, because of the ample server assumption, all repairs are carried out independently, and we can view the overall facility as partitioned into \( I \) subsystems. This is depicted in Figure 3.1.

---

**Figure 3.1:** The EROSF-model
Here, we can imagine the customer accompanying his specific class-i failure into ss$_i$ and experience there precisely what the basic model dictates. If we assume that desk time (failure identification) and reinstall times are negligible, then the basic model gives the required subsystem performance measures in terms of the subsystem parameters. Thus, for ss$_i$, (i = 1,2,...I) we have repair (or reproduction) time distribution function $G_d(\cdot)$ with mean $1/\mu_i$. We will designate the waiting time of a customer at ss$_i$ by r.v $T_{i,n_i}$, (i = 1,2,...I) and the overall time in the system by $W_{\tilde{n}}$, $\tilde{n} = (n_1,...,n_I)$. We will designate the number of customers at specific ss$_i$ by r.v $X_{i,n_i}$, and the overall number of customers in the whole system by $X_{\tilde{n}}$. We now write $F_i(t \mid n_i) = F_{i,n_i}(t) = P(T_{i,n_i} \leq t)$ and this is obtained by appropriate use of the basic model. Thus, the first two moments of the waiting time the number of customers in ss$_i$ are given by:

$$\overline{T}_{i,n_i} = \int_{t=0}^{\infty} (1 - F_i(t \mid n_i)) dt = \int_{t=0}^{\infty} (1 - F_{i,n_i}(t)) dt$$

(3.1)

$$\overline{T}_{i,n_i}^2 = 2 \int_{t=0}^{\infty} t (1 - F_i(t \mid n_i)) dt = 2 \int_{t=0}^{\infty} t (1 - F_{i,n_i}(t)) dt$$

(3.2)

$$L_{i,n_i} = \sum_{r=1}^{\infty} r * P(X_{i,n_i} = r) ,$$

(3.3)

for i = 1,2,...I.

3.2. Service measures

In the sequel, service measures are introduced to describe the performance of the given system. Each sub-system i contain $n_i$ spares and we write $\tilde{n} = (n_1,n_2,n_3,...n_I)$.

3.2.1. The waiting time distribution

The waiting time distribution of a random customer arriving at the system is given by:

$$P(W \leq t \mid \tilde{n}) \equiv F(t \mid \tilde{n})$$

$$= F_{\tilde{n}}(t)$$
\[ W = \sum_{i=1}^{I} a_i P(T_{i,n_i} \leq t) \]
\[ = \sum_{i=1}^{I} a_i F_{i,n_i}(t) \]  
(3.4)

3.2.2. The waiting time moments

The average waiting time \( \overline{W}_{\tilde{n}} \) of a random customer coming to the system characterized by spares vector \( \tilde{n} \), is

\[ \overline{W}_{\tilde{n}} = \int_{t=0}^{\infty} \sum_{i=1}^{I} a_i P(T_{i,n_i} > t) dt \]
\[ = \sum_{i=1}^{I} a_i \int_{t=0}^{\infty} P(T_{i,n_i} > t) dt \]
\[ = \sum_{i=1}^{I} a_i \overline{T_{i,n_i}}. \]  
(3.5)

where \( \overline{T_{i,n_i}} \) is the average waiting time of a customer bringing item class \( i \) to \( ss_i \).

\[ E(W_{\tilde{n}}^2) = \sum_{i=1}^{I} a_i \overline{T_{i,n_i}^2} \]  
(3.6)

where \( \overline{T_{i,n_i}^2} \) is the second moment of a customer bringing item class \( i \) to \( ss_i \), and

\[ V(W_{\tilde{n}}) = E(W_{\tilde{n}}^2) - \overline{W_{\tilde{n}}}^2 \]  
(3.7)

3.2.3. The Fillrate

For a system characterized by spares vector \( \tilde{n} \), we define \( FR_{\tilde{n}} \) as the probability that a random customer obtains immediate satisfaction:

\[ FR_{\tilde{n}} = \sum_{i=1}^{I} a_i FR_{i,n_i} \]

where \( FR_{i,n_i} \) is the fillrate corresponding to \( ss_i \).

Thus, from (2.2)
\[ FR_{i,n_i} = F_i(0 \mid n_i) = P(Y_{i1}(0) - Y_{i2}(0) \leq n_i - 1) \]
\[ = \sum_{r=0}^{\infty} P(Y_{i2}(0) = r) P(Y_{i1}(0) \leq n_i - 1 + r) \text{ for } i=1,2,\ldots1 \]
since \( \lambda_{i2}(0) = 0 \).

Thus,
\[ P(Y_{i2}(0) = r) = \begin{cases} 1, & r = 0 \\ 0, & \text{else,} \end{cases} \]

and since \( \lambda_{i1}(0) = \frac{\lambda_i}{\mu_i} \), we have
\[ FR_{i,n_i} = P(Y_{i1}(0) \leq n_i - 1) \]
\[ = \sum_{r=0}^{n_i-1} \frac{(\lambda_i / \mu_i)^r}{r!} e^{-\left(\lambda_i / \mu_i\right)} \]
(3.8)

### 3.2.4. The waiting time \( W^c \) under the condition that the customer waits

Often, a manager of a system doesn’t want to take decisions based on the waiting time distribution, but rather on the waiting times of customer who wait. Let \( T_{i,n_i}^c \) be the waiting time at ss\(_i\) for a customer who waits.

Therefore, for \( t \geq 0 \)
\[ P(T_{i,n_i}^c \leq t) = \frac{P(T_{i,n_i} < t)}{1 - P(T_{i,n_i} = 0)} \text{ for } i = 1, \ldots, I. \]
(3.9)

Thus, the overall “conditioned” waiting time distribution is given by
\[ P(W_i^c \leq t) = \sum_{i=1}^{I} a_i \frac{P(T_{i,n_i} \leq t)}{1 - FR_{i,n_i}}. \]
(3.10)

By using the above equations, the moments can also be easily calculated. We leave this to the reader.
3.2.5. The average number of customers in the system

The average number of customers in the system is the sum of the average numbers of customers waiting for repair of all component classes. \( L_{i,n_i} \) is the average number of customers in \( s_i \) and is given by

\[
L_{i,n_i} = \sum_{r=1}^{\infty} r \cdot P(X_{i,n_i} = r)
\]

(3.11)

where

\[
P(X_{i,n_i} = r) = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^{r+n_i}}{(r+n_i)!} e^{-\left(\frac{\lambda_i}{\mu_i}\right)}, \text{ for } r=1,2,3,\ldots,
\]

(3.12)

\[
P(X_{i,n_i} = 0) = 1 - \sum_{r=1}^{\infty} P(X_{i,n_i} = r).
\]

Thus, \( L_{\bar{n_i}} \), the total expected number of customers in the system, is given by

\[
L_{\bar{n_i}} = \sum_{i=1}^{l} L_{i,n_i}.
\]

(3.13)

3.2.6. The number of customers in the system

Another important service level corresponds to the distribution of the number of customers in the system. Let \( Q \) be the total number of customers in the system. When there are two sub-systems we get that

\[
P(Q = r \mid \bar{n_i}) = \sum_{i_2=0}^{r} P(X_1 = i_2 \mid n_1) P(X_2 = r - i_2 \mid n_2)
\]

When there are three sub-systems,

\[
P(Q = r \mid n) = \sum_{i_3=0}^{r} \left[ \sum_{i_2=0}^{i_3} P(X_2 = i_2 \mid n_2) P(X_1 = r - i_3 - i_2 \mid n_1) \right].
\]

With more than three sub-systems, we may likely assume a normal approximation, and thus, \( Q \sim N(\sum_{i=1}^{l} L_{i,n_i}, \sum_{i=1}^{l} V(X_i \mid n_i)) \).

(3.14)
3.3. Defining the optimization model

After introducing a variety of service level measures, we now want to optimize the solution to the model. A natural goal is to provide the maximum service level possible to a customer who arrives at the system. For this purpose, we will assume that a budgetary allocation of M dollars is at our disposal. We make the assumption that M is used only to buy spares. Initially there are no spares in the system and we purchase $n_i$ spares for $s_{si}$ at unit cost $c_i$ (without discounts). In the following chapters, we will use “max a service-level”, which means that we want to optimize a given service measure. In certain cases, such as the average waiting time, this notion will be inverted, because here we would want to minimize the average waiting time. Thus, we use max as a general terminology.

The problem then becomes

$$\max_{\bar{a}} (\text{Service level})$$

s.t.:

$$\sum_{i=1}^{I} c_i n_i \leq M .$$

$$n_i = 0, 1, 2, 3, \ldots \text{ for } i = 1, 2, \ldots I.$$ 

where the service level depends on $F_i(t \mid n_i), i = 1, 2, \ldots I$, given by

$$F_{i,n_i}(t) = P(Y_{i1}(t) - Y_{i2}(t) \leq n_i - 1) + G_i(t) P(Y_{i1}(t) - Y_{i2}(t) = n_i),$$

where $Y_{i1}(t)$ and $Y_{i2}(t)$ are generic Poisson random variables with parameters $\lambda_{i1}(t)$ and $\lambda_{i2}(t)$, and

$$\dot{\lambda}_{i1}(t) = \lambda_i \int_{u=t}^{\infty} G_i(u) du$$

$$\dot{\lambda}_{i2}(t) = \lambda_i \int_{u=0}^{t} G_i(u) du$$
3.4. Optimizing the ERSOF model

As explained in section 3.2, there are several service level performance criteria which can serve as the objective function for the model. We will apply two of them in the following section. The others will be explored in further research as a basis for new developments. The first is the fillrate of the system expressed as the average overall sub systems, and the second is the average waiting time of a random customer also expressed as the average over all subsystems.

3.4.1. The fillrate of the system

We wish to maximize the fillrate FR. Hence,

$$\max FR = \sum_{i=1}^{I} a_i * FR_i(n_i),$$

(3.19)

where $FR_i(n_i)$ is the fillrate of ss_i with n_i spares. If FR is either a concave or a convex function then the Kuhn Tucker [19] conditions are both necessary and sufficient conditions to achieve an optimal solution for the continuous version of this model. The sum of concave functions is also a concave function. Thus, if each $FR_i(n_i)$ is concave for every $n_i = 0,1,2,…$, then FR will also be concave.

Lemma 3.1: FR is a not a concave function.

As mentioned above, each $FR_i$ must be concave so that FR is concave. Thus, if at least one $FR_i$ is not concave, FR is not automatically a concave function. By definition, the second difference of any function must be less than zero for the function to be concave. Since $FR_i$ is not a continuous function, the first difference is calculated as follows:

$$FR_i'(n_i) = FR_i(n_i + 1) - FR_i(n_i),$$

(3.20)

where by (3.8),

$$FR_i(n_i) = P(T_{i,n_i} = 0) = P(Y_{i1}(0) - Y_{i2}(0) \leq n_i - 1) + G_i(0)P(Y_{i1}(0) - Y_{i2}(0) = n_i)$$

(3.21)

for each $i = 1,2,…,I$. 

Thus,

\[ FR'_i(n_i) = FR'_i(n_i + 1) - FR'_i(n_i) \]

\[ = P(Y_{ii}(0) \leq n_i) - P(Y_{ii}(0) \leq n_i - 1) \]

\[ = P(Y_{ii}(0) = n_i) \]

\[ = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^n e^{-\frac{\lambda_i}{\mu_i}}}{n_i!} \]

(3.22)

which is a Poisson distribution with parameter \((\frac{\lambda_i}{\mu_i})\).

The second difference is calculated as follows:

\[ FR''_i(n_i) = FR'_i(n_i + 1) - 2FR'_i(n_i) + FR'_i(n_i) \]

\[ = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^{n+1} e^{-\frac{\lambda_i}{\mu_i}}}{(n_i + 1)!} - \frac{\left(\frac{\lambda_i}{\mu_i}\right)^n e^{-\frac{\lambda_i}{\mu_i}}}{n_i!} \]

\[ = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^n e^{-\frac{\lambda_i}{\mu_i}}}{n_i!} \left(\frac{\lambda_i}{\mu_i} - 1\right) \]

(3.23)

The function is concave as long as it satisfies \(FR'_i(n_i) < 0\) for all \(n_i\). Thus,

\[ FR'_i(n_i) = \frac{\left(\frac{\lambda_i}{\mu_i}\right)^n e^{-\frac{\lambda_i}{\mu_i}}}{n_i!} \left(\frac{\lambda_i}{\mu_i} - 1\right) < 0 \]

Since \(\frac{\left(\frac{\lambda_i}{\mu_i}\right)^n e^{-\frac{\lambda_i}{\mu_i}}}{n_i!} > 0\), we have

\[ \left(\frac{\lambda_i}{\mu_i} - 1\right) < 0 \]

or \(n_i > (\frac{\lambda_i}{\mu_i} - 1)\).

(3.24)

This means that for \(n_i > (\frac{\lambda_i}{\mu_i}) - 1\), the function is concave and for \(n_i \leq (\frac{\lambda_i}{\mu_i}) - 1\), the function is convex. Consequently, every \(FR_i\) has an inflection point and therefore, it has a concave part and a convex part.

QED.

Figure 3.2 shows the fillrate of a system when \(n\) changes, as generated by the software. Indeed, it can be seen that the first part of the function is convex and the second part concave.
Thus, the Kuhn-Tucker method cannot be used. An alternative method used to solve this problem is dynamic programming and will be addressed in chapter 5.

### 3.4.2. The average waiting time of a customer in the system

The second service criterion we examine is the average waiting time of a customer in the system.

\[
\min_n \bar{W}_n = \sum_{i=1}^{I} a_i \ast \bar{T}_i(n_i),
\]

where \( \bar{T}_i(n_i) \) is the average waiting time of sub-system i with \( n_i \) spares. Since \( \bar{T}_i(n_i) = L_i(n_i)/\lambda_i \), we have

\[
\bar{W}_n = \sum_{i=1}^{I} \lambda_i \ast \bar{T}_i(n_i) = \sum_{i=1}^{I} L_i(n_i)/\lambda = \sum_{i=1}^{I} L_i(n_i)/\lambda.
\]

If \( \bar{W}_n \) is a convex function then the Kuhn-Tucker conditions are both necessary and sufficient conditions to achieve an optimal solution for the continuous model. Thus, if each \( \bar{T}_i(n_i) \) is convex for every \( n_i = 0,1,2,... \) then \( \bar{W}_n \) will also be convex.
Lemma 3.2: $\overline{W}_n$ is a convex function.

If every $L_i(n_i)$ for every $i = 1,... I$ is convex $\overline{W}_n$ is also a convex function. To check this, we will calculate the second difference. The first difference of $L_i(n_i)$ is calculated as follows:

$$L_i(n_i) = L_i(n_i + 1) - L_i(n_i)$$

$$= \sum_{r=1}^{\infty} r^* \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r+n_i)!} e^{-(\lambda_i / \mu_i)} - \sum_{r=1}^{\infty} r^* \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r+n_i)!} e^{-(\lambda_i / \mu_i)}$$

$$= e^{-(\lambda_i / \mu_i)} \left[ - \sum_{r=1}^{\infty} r^* \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r+n_i)!} + \sum_{k=2}^{\infty} (k-1)^* \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} \right]$$

$$= e^{-(\lambda_i / \mu_i)} \left[ - \sum_{r=1}^{\infty} r^* \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r+n_i)!} + \sum_{k=2}^{\infty} k^* \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} - \sum_{k=2}^{\infty} (\lambda_i / \mu_i)^{k+n_i} \right]$$

$$= e^{-(\lambda_i / \mu_i)} \left[ - \frac{(\lambda_i / \mu_i)^{1+n_i}}{(1+n_i)!} - \sum_{k=2}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} \right]$$

$$= -\left[ \sum_{k=1}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right]$$

(3.25)

The second difference of $L_i(n_i)$ is then calculated as follows:

$$L''_i(n_i) = L'_i(n_i + 1) - L'_i(n_i)$$

$$= \left[ - \sum_{k=1}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right] + \left[ \sum_{k=1}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right]$$

$$= \left[ - \sum_{k=1}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right] + \left[ \sum_{k=1}^{\infty} \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right]$$

$$= \frac{(\lambda_i / \mu_i)^{1+n_i}}{(1+n_i)!} e^{-(\lambda_i / \mu_i)} > 0,$$

which is clearly true for every $n_i$. Therefore, every $L_i(n_i)$ is a convex function and thus $\overline{W}_n$ is also a convex function.

Thus, solving the following model for his continuous version can be done by applying the KKT conditions:
\[
\min _{n} \left( \bar{W}_n = \sum_{i=1}^{I} a_i \bar{T}_{i,n_i} \right)
\]  

(3.26)

s.t.: 
\[
\sum_{i=1}^{I} c_i n_i \leq M .
\]
\[
n_i = 0, 1, 2, 3, \ldots \text{ for } i=1, 2, \ldots I.
\]

This optimization model is an integer Knapsack problem, which can be solved using dynamic programming. But, when costs are low and the budget \( M \) high, which means that the model will have a large number of states, dynamic programming becomes impractical. Therefore, using the convexity property of the function, we decided to make the objective function continuous and piece-wise linear by making all \( \bar{T}_{i,n_i} \) for \( i=0,1,\ldots I \) piece-wise linear. This approach is meaningful due the fact that from an analytical point of view, getting a result such as \( n_i = 4.5 \), may mean that one year we choose 4, and the next year, 5. Figure 3.3 and Figure 3.4 demonstrate how we make the discrete function piece-wise linear.

![Figure 3.3: An example of the average Waiting Time of a basic system](image-url)
Figure 3.4: Converting the graph in Figure 3.3 to a piece-wise linear curve

The mathematical definition of the added “lines” is as follows: \( x_i = n_i + \Delta \), where \( 0 \leq \Delta \leq 1 \), which means that \( n_i \) is the integer component of \( x_i \). 

\[
\bar{T}_i(x_i) = \mathcal{T}_i(n_i + \Delta) = \mathcal{T}_i(n_i) + \Delta(\bar{T}_i(n_i + \Delta) - \bar{T}_i(n_i)) \quad \text{where} \quad 0 \leq \Delta \leq 1, \quad \text{for all} \quad i = 1, \ldots, I, \quad \text{and} \quad x_i \quad \text{is the continuous variable equivalent of the integer variable} \quad n_i. 
\]

By making the objective function into a continuous function, the constraint will become binding, and thus,

\[
\sum_{i=1}^{I} c_i x_i = M. 
\]

To solve this model, a Lagrange multiplier \( \theta \) was added.

\[
\bar{W}_n^* = \min_{\bar{a}} \left( \sum_{i=1}^{I} a_i * \bar{T}_i(x_i) \right) + \theta \left( M - \sum_{i=1}^{I} c_i x_i \right) 
\]

\[
\frac{\partial \bar{W}_n^*}{\partial x_i} = a_i * \bar{T}_i(x_i) - \theta(c_i) = 0. 
\]

\[
\frac{\partial \bar{W}_n^*}{\partial \theta} = M - \sum_{i=1}^{I} c_i x_i = 0. 
\]

Thus,

\[
\theta = \frac{a_i * \bar{T}_i(x_i)}{c_i}, \quad i = 1, 2, \ldots, I. \tag{3.27} 
\]
In Appendix A, we tried to solve a two-item class problem analytically, using the normal approximation for the Poisson term. But the results are not practical, as can be seen in that appendix. Therefore, we developed two approximate algorithms to solve the problem.

3.5. Algorithms:

Algorithm 1: The first approximate algorithm starts at 0. This means that at the beginning no items are chosen and no money spent. Then, based on the optimality condition we have \( \theta_i = \frac{a_i \cdot \bar{T}_i(x_i)}{c_i} \), for \( i=1,\ldots,I \) (3.28)

Then, \( \theta_r = \min \theta_i \) is chosen, which means that we now have the best service level per dollar, and \( r \) is the resulting item class selected and added. These steps are implemented until the whole budget is depleted.

The algorithm is as follows:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>( x_1, x_2, x_3, \ldots, x_I = 0 ), where ( x_i ) is the number of spares of class ( i ), and ( m = M ), where ( m ) is the money left for spares.</td>
</tr>
<tr>
<td>b.</td>
<td>Calculate ( \theta_i = \frac{a_i \cdot \bar{T}_i(x_i)}{c_i} ) for ( i=0,1,2,\ldots,I ).</td>
</tr>
<tr>
<td>c.</td>
<td>Choose ( r ) satisfying ( \theta_r = \min \theta_i ), where ( r ) is the resulting item class.</td>
</tr>
<tr>
<td>d.</td>
<td>If ( m \leq c_r ), then ( x_r = x_r + (m/c_r) ). Finish.</td>
</tr>
<tr>
<td>e.</td>
<td>Otherwise, set ( m = m - c_r ), ( x_r = x_r + 1 ), Go to step b.</td>
</tr>
</tbody>
</table>

The software which was developed for this algorithm creates a queue of items to choose from, ordered from the smallest \( \theta_i \) to the largest. Each time the first item is taken out of the queue (it has the smallest current \( \theta_i \)), the queue is reordered by \( \theta_i \). In this way, the program avoids calculating all \( \theta_i \) at each iteration. This program also found combinations up to 20 items, when the budget is high and the unit costs are small, within a second, which is negligible.
Figure 3.5: Input and Output data of the software

| results          | 1.5, 6, 6;  
|                 | 1.5, 8;    
|                 | 2.5, 20;  
|                 | 0.5, 5, 10; 
|                 | 1.2, 6, 8; 
|                 | 2.5, 20;  
|                 | 1.5, 6, 6; 
|                 | 1.5, 8;  
|                 | 2.5, 20;  
|                 | 0.5, 5, 10; 
|                 | 1.2, 6, 8; 
|                 | 2.5, 20;  
|                 | 2.5, 20; 
|                 | 1000;     
| 11              | 6 6 2 8 6 6 2.4 8 6 6 | 0.369532 |

Figure 3.5 shows the input and the output of the software for a random example of 13 item classes. The input data is ordered into three columns: mean repair time, arrival rate, and price per unit. After the input data, the total budget allocation is listed (1000). The last line is the output: The \( n_i \) number gives the optimal number of spares \( n_i \) for \( i=1,2,\ldots,13 \). Finally, the last number is the average waiting time of a random customer arriving at the system. It can be seen that all values of the results are integer except one.

**Algorithm 2:** The second heuristic algorithm is based on the idea that at any stage, we can improve the objective function while still satisfying the constraint. We do this by "moving" money from one class to another. For example, if the unit cost of one item of the first class is 6 dollar and of the second 8, then we choose to move one dollar from the first class to the second and thus \( n_1 = n_1 - 1/6 \) and \( n_2 = n_2 + 1/8 \). Due to the fact that we are dealing with piece-wise linear functions, we can add and remove fractions from the number of spares in the system. We proceed this way until we reach a local optimum. In the case of the average waiting time, we know that there is only one local optimum and this will consequently also be the global one. The software developed for this purpose chooses one out of the three different options to start. There could be many more.

- To divide the budget \( M \) equally among the items. This means that
  \[
  x_i = \frac{(M/I)/c_i=M/(I*c_i)}{I} \text{ for all } I \text{ different class of items.}
  \]  

(3.29)
• All items have the same number of spares at the beginning. This means,

\[ x_1 = x_2 = \ldots = x_I = \frac{M}{\sum_{i=1}^{I} c_i} \]  \tag{3.30}

• The most expensive item class gets all the items at the beginning.

\[ c_k = \max(c_i), x_k = \frac{M}{c_k} \]  \tag{3.31}

The exact algorithm looks like:

a. \( x_1, x_2, x_3, \ldots, x_I \) are initialized as mentioned (3.29), (3.30), (3.31) by one of the three options, where \( x_i \) is the amount of spares of class \( i \) chosen.

b. Calculate \( \theta_i = \frac{a_i * \overline{T}_i(x_i)}{c_i} \), for \( i = 1, \ldots, I \).

c. Choose \( l \) and \( h \) satisfying \( \theta_l = \min(\theta_i) \) and the \( \theta_h = \max(\theta_i) \), where \( l \) and \( h \) are the item class indices chosen.

d. If \( \theta_l = \theta_h \), finish.

e. If not, calculate \( c_l = (\lfloor x_l + 1 \rfloor - x_l)c_i \) and \( c_h = (x_h - \lfloor x_h - 1 \rfloor)c_h \).

f. Choose \( r \) so that \( c_r = \min(c_l, c_h) \)

g. \( x_i = x_i + c_r / c_i \) and \( x_h = x_h - c_r / c_h \). Go to b.

<table>
<thead>
<tr>
<th>results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5, 6, 6;</td>
</tr>
<tr>
<td>1, 5, 8;</td>
</tr>
<tr>
<td>2, 5, 20;</td>
</tr>
<tr>
<td>0.5, 5, 10;</td>
</tr>
<tr>
<td>1.2, 6, 8;</td>
</tr>
<tr>
<td>2, 5, 20;</td>
</tr>
<tr>
<td>1.5, 8;</td>
</tr>
<tr>
<td>2, 5, 20;</td>
</tr>
<tr>
<td>0.5, 5, 10;</td>
</tr>
<tr>
<td>1.2, 6, 8;</td>
</tr>
<tr>
<td>2, 5, 20;</td>
</tr>
<tr>
<td>1000;</td>
</tr>
<tr>
<td>11 6 6 2 8 6 11 6 6 2.4 8 6 6 0.369582</td>
</tr>
</tbody>
</table>

Figure 3.6: Results of the algorithm 2
The example shows that both algorithms give identical results for the given example. The first algorithm is used when no initial solution is available. It can also be shown that in large systems this algorithm is efficient. The second algorithm is used when an initial solution is available and needs to be changed; e.g.; when the prices or variables of one or two sub-systems change.

3.6. Validation of the results of the algorithms

It is difficult to validate the results of the algorithm because the correct answer is not always known. The problem is a Knapsack problem. Since the Knapsack problem is NP-Hard, it is hard to compare the heuristic solution to the optimal one due to large computational requirements. In fact, there could be \( k = M/\min(c_1, c_2, c_3, c_I) \) spares for some class, resulting in \( kI \) different solutions, where \( k \) may be very large. In spite of this fact, a spreadsheet was created to validate the existing program. Using Solver of the Excel program we solved the given problem (Table 3.1) also and found an exact solution. This is not always guaranteed. But in Chapter 8, we will use Integer Programming to validate all the results. Table 3.2 depicts four approaches to solving this particular problem. The first row shows the results using the Solver to solve it as an integer problem. The second row shows the results using the software developed for this problem using non-integer values. The third row takes the result of row two and transforms it into integer values as we will show. The fourth row shows the results using an LP software. However, in large systems with large budgets, the LP and the Knapsack algorithms use more time resources due to their complexity. From Table 3.2, we can see that the Software yielded a correct solution. In big problems, time is a constraint so that good quick algorithms are required. Therefore, we use the Software developed as a good and efficient heuristic algorithm.

Consider a system with 13 classes, not necessary distinct classes, where the repair times are distributed normally with given means and standard deviations. Obviously, the product fails via a single class. In addition, the arrival rate of item \( i \) to \( S_i \) and the price of item \( i \) are also given in Table 3.1. The mean repair time and the standard deviation were chosen so that the probability of a negative repair time is negligible. The last column shows the number of spares to buy, determined by the software, using a given budget.
<table>
<thead>
<tr>
<th>Item class</th>
<th>1/μ</th>
<th>Λ</th>
<th>Price</th>
<th>λ/μ</th>
<th>money spent</th>
<th>rest of n_i</th>
<th>int n_i</th>
<th>chosen n_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.5</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>60</td>
<td>0</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>48</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>0.5</td>
<td>5</td>
<td>10</td>
<td>2.5</td>
<td>20</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1.2</td>
<td>6</td>
<td>8</td>
<td>7.2</td>
<td>64</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>1.5</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>66</td>
<td>0</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>48</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
<td>5</td>
<td>10</td>
<td>2.5</td>
<td>30</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>1.2</td>
<td>6</td>
<td>8</td>
<td>7.2</td>
<td>64</td>
<td>0</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>5</td>
<td>20</td>
<td>10</td>
<td>120</td>
<td>0</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

| budget     | 1000 |

Arrival rate at the system: 69

Objective Function: 0.369876 Min

Table 3.1: Spreadsheet of a numerical example

<table>
<thead>
<tr>
<th>Knapsack</th>
<th>10</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>8</th>
<th>6</th>
<th>11</th>
<th>6</th>
<th>6</th>
<th>3</th>
<th>8</th>
<th>6</th>
<th>6</th>
<th>0.369876</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2.4</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>0.369582</td>
</tr>
<tr>
<td>Softw. K</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>0.372227</td>
</tr>
<tr>
<td>LP</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2.2</td>
<td>8</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>6</td>
<td>2.2</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>0.369582</td>
</tr>
</tbody>
</table>

Table 3.2: Four different approaches to solve the problem
3.7. Post-optimization

After optimization, it is now possible to answer the following questions:

1. What is the fillrate of a customer arriving to the system?
2. What is the probability that a customer waits less than x for “his” product
3. What is the average waiting time of a customer in the system?
4. What is the average number of customer in the system?
5. What is the probability that there are more than a given number of customers in the system?
6. What is the probability a customer has to wait less than x, if it is known that he has to wait?

These and more questions can be answered, when \( \hat{n} \) is given. Several of these questions cannot be answered by Sherbrooke and Albright ([20-23], [2-3]) which dealt with different multi-item systems because they did not use knowledge of \( F_n(t) \) (the waiting time distribution). This is the main novelty of our work; it provides the capability to answer a large variety of different service questions. Section 3.8 provides answers to all the above questions through a worked out example.

3.8. Numerical example

For this section, a completely new software was created, which was based on the previous software. To check the new software, we also determined the fillrate, the average waiting time and the average queue size analytically. Thus by (3.8),

\[
FR_i = P(Y_{ni}(0) \leq n_i - 1)
\]

\[
= \begin{cases} 
    \sum_{r=0}^{n_i-1} \frac{(\lambda_i / \mu_i)^r}{r!} e^{-\lambda_i / \mu_i}, & n_i \geq 1 \\
    0, & n_i = 0
\end{cases}
\]

Therefore, by (2.7) and (2.8),

\[
L_i = \sum_{r=1}^\infty \frac{r \lambda_i / \mu_i}{(r + n_i)!} (r + n_i) e^{-\lambda_i / \mu_i}
\]
\[
\sum_{r=1}^{\infty} \frac{(r + n_i) (\lambda_i / \mu_i)^{r+n_i}}{(r + n_i)!} e^{-\lambda_i/\mu_i} = -n_i \sum_{r=1}^{\infty} \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r + n_i)!} e^{-\lambda_i/\mu_i}
\]

\[
= (\lambda_i / \mu_i) \sum_{r=n_i}^{\infty} \frac{(\lambda_i / \mu_i)^{r+n_i-1}}{(r + n_i - 1)!} e^{-\lambda_i/\mu_i} - n_i \sum_{r=1}^{\infty} \frac{(\lambda_i / \mu_i)^{r+n_i}}{(r + n_i)!} e^{-\lambda_i/\mu_i}
\]

\[
= (\lambda_i / \mu_i) \sum_{r=n_i}^{\infty} \frac{(\lambda_i / \mu_i)^{r}}{r!} e^{-\lambda_i/\mu_i} - n_i \sum_{r=n_i+1}^{\infty} \frac{(\lambda_i / \mu_i)^{r}}{r!} e^{-\lambda_i/\mu_i}
\]

Hence, by Little's rule

\[
\bar{W}_i = \frac{L_i}{\lambda_i} = \frac{1}{\mu_i} \sum_{r=n_i}^{\infty} \frac{(\lambda_i / \mu_i)^{r}}{r!} e^{-\lambda_i/\mu_i} - \frac{n_i}{\lambda_i} \sum_{r=n_i+1}^{\infty} \frac{(\lambda_i / \mu_i)^{r}}{r!} e^{-\lambda_i/\mu_i}
\]

for \(i=1,2,\ldots,I\).

We will examine the example system in Table 3.3 below, and answer the questions mentioned in the previous section.

Example:

<table>
<thead>
<tr>
<th>NR. Item</th>
<th>Repair Distribution</th>
<th>Mean repair time</th>
<th>Stdev</th>
<th>Arrival Rate</th>
<th>Price/Item</th>
<th>Number_spares</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NORMAL</td>
<td>1.5</td>
<td>0.2</td>
<td>6</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>NORMAL</td>
<td>1</td>
<td>0.2</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>NORMAL</td>
<td>2</td>
<td>0.2</td>
<td>5</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>NORMAL</td>
<td>0.5</td>
<td>0.1</td>
<td>5</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>NORMAL</td>
<td>1.2</td>
<td>0.2</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>NORMAL</td>
<td>2</td>
<td>0.2</td>
<td>5</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>NORMAL</td>
<td>1.5</td>
<td>0.2</td>
<td>6</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>8</td>
<td>NORMAL</td>
<td>1</td>
<td>0.2</td>
<td>5</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>NORMAL</td>
<td>2</td>
<td>0.2</td>
<td>5</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>NORMAL</td>
<td>0.5</td>
<td>0.1</td>
<td>5</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>NORMAL</td>
<td>1.2</td>
<td>0.2</td>
<td>6</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>NORMAL</td>
<td>2</td>
<td>0.2</td>
<td>5</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>NORMAL</td>
<td>2</td>
<td>0.2</td>
<td>5</td>
<td>20</td>
<td>6</td>
</tr>
</tbody>
</table>

Total budget 1000 dollars.

Table 3.3: Data of the system with 13 classes of items.

Using the software developed for this research, we found the system measures in Table 3.4. We didn't analyze the impact of the standard deviation on the results, and in fact, this could be part of a further research. All the results are part of the software which produces
the graphs in Figure 3.7-3.11 to illustrate the system behavior. Table 3.4 summarizes all service measurements. Figure 3.7 and Figure 3.8 show how the waiting time is distributed. Figure 3.8 shows that the density may have multiple local maxima and consequently is unpredictable.

<table>
<thead>
<tr>
<th>Service Measurement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>The average fillrate of a customer</td>
<td>0.37694</td>
</tr>
<tr>
<td>The average number of customers in the system</td>
<td>25.6836</td>
</tr>
<tr>
<td>The average waiting time of a customer</td>
<td>0.372227</td>
</tr>
<tr>
<td>The “conditioned” average waiting time</td>
<td>0.5974</td>
</tr>
<tr>
<td>The standard deviation of the waiting time.</td>
<td>0.476353</td>
</tr>
<tr>
<td>Coefficient of variation: Standard Deviation/mean</td>
<td>1.26972</td>
</tr>
</tbody>
</table>

Table 3.4: Service-level measures for the numerical example

Figure 3.7: The Cumulative waiting time distribution for the numerical example
Another important aspect of the system is the number of customers in the system. From Chapter 2, we know the distribution of the number of customers in each sub-system. The total number of customers in the system is the sum of the numbers of items in each sub-system; its distribution is given as the sum of Poisson terms, and reasonably follows a Normal distribution with mean 25.68 and the variance 56.56. Figure 3.9 shows the graph of the number of customers in the system.

Figure 3.8: The density function of the waiting time in the system

Figure 3.9: The distribution of the number of customers in the system
3.8.1. Sensitivity analysis about the budget allocation, M

After analyzing the service-level measures of the system, we want to know how sensitive the system is to the budget. Therefore, sensitivity analysis was added to show the behavior of the different service levels as a function of the budget. Figure 3.10 shows the Average Waiting Time for different budgets M.

![Figure 3.10: The average waiting time for the different budgets](image)

Another interesting service level measure is the fillrate depicted in Figure 3.11 for varying M. The graphs 3.10 and 3.11 show that there are examples where a higher budget decreases the average waiting time but does NOT increase the fillrate (400-700$). Thus, the fillrate and the average waiting time are not necessarily connected and must be dealt

![Figure 3.11: The fillrate of the optimized system when adding increasing the budget](image)
with separately. This means that to optimize the fillrate, we require other methods such as Dynamic Programming. The fact that a higher budget will not directly improve a given service level can also be seen in Figure 3.12, where the probability of waiting time of a random customer is less than 0.7, and similarly in Figure 3.13 for other probabilities.

Figure 3.12: An example of the waiting time distribution when the budget changes

Figure 3.13: Different waiting time distributions when the budget changes
The same data can be shown in another way and we can see the effect of the budget on the waiting time distribution. Figure 3.14 shows the data from the perspective of the budget.

![Figure 3.14](image)

Figure 3.14: The effect of the budget on the waiting time distribution

The non-smooth character of the graphed functions is somewhat surprising and will be subsequently investigated.
3.8.2. Sensitivity analysis on arrival rate

After checking how the solution is sensitive to the budget factor, we now check the sensitivity of the solution to the arrival rate factor. Thus, we consider 30%, 20%, 10% lower and 10%, 20% and 30% higher on the arrival rates. Figure 3.15 shows the effect of the arrival rate on the average waiting time and the fillrate, assuming the numbers of spares in Table 3.3.

![Graph showing the effect of arrival rate on average waiting time and fillrate](image)

Figure 3.15: The effect of the arrival rate on the average waiting time and the fillrate

![Graph showing the cumulative waiting time distribution](image)

Figure 3.16: The effect of the arrival rate on the cumulative waiting time distribution
3.9. Model validation

Appendix D shows the simulation using ARENA of the ERSOF-model and the results are summarized in Table 3.5 below:

<table>
<thead>
<tr>
<th>Sub-System</th>
<th>Simulation</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\bar{T}_i$</td>
<td>$L_i$</td>
</tr>
<tr>
<td>1</td>
<td>0.07389</td>
<td>0.43916</td>
</tr>
<tr>
<td>2</td>
<td>0.09611</td>
<td>0.47545</td>
</tr>
<tr>
<td>3</td>
<td>0.82377</td>
<td>4.1432</td>
</tr>
<tr>
<td>4</td>
<td>0.16797</td>
<td>0.82045</td>
</tr>
<tr>
<td>5</td>
<td>0.11183</td>
<td>0.65272</td>
</tr>
<tr>
<td>6</td>
<td>0.82421</td>
<td>4.1289</td>
</tr>
<tr>
<td>7</td>
<td>0.08843</td>
<td>0.53323</td>
</tr>
<tr>
<td>8</td>
<td>0.1031</td>
<td>0.51522</td>
</tr>
<tr>
<td>9</td>
<td>0.80016</td>
<td>3.9193</td>
</tr>
<tr>
<td>10</td>
<td>0.17006</td>
<td>0.8302</td>
</tr>
<tr>
<td>11</td>
<td>0.11366</td>
<td>0.67335</td>
</tr>
<tr>
<td>12</td>
<td>0.81488</td>
<td>4.0434</td>
</tr>
<tr>
<td>13</td>
<td>0.8017</td>
<td>3.9451</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bar{W}$</td>
<td>0.36806</td>
</tr>
<tr>
<td>$L$</td>
<td>25.119</td>
</tr>
</tbody>
</table>

Table 3.5: The results of the software and of the simulation

The table compares the average waiting time $\bar{T}_i$ and average number of customers $L_i$ of sub-system i, and of the whole system. The simulation also validates our assumption that the number of customers is distributed normally. Although the values are not exact, we can say that the differences come from the short-running simulation. The intention of the simulation was to get an idea if we are "on the way" or not, and to have a tool to get measures whenever needed. Figure 3.17 and Figure 3.18 give the output of the simulation and shows that the number of customers is distributed normally. Figure 3.17 with smaller cell sizes depicts the number of customer distribution density and its cumulative. Taking the data from Figure 3.17 and comparing to the solution from analysis, we get Figure 3.18.
Figure 3.17: The output of the simulation for the number of customers in the system.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>From To</td>
<td>Cell Cumul.</td>
<td>Cell Cumul.</td>
</tr>
<tr>
<td>1 -Infinity</td>
<td>5 0.6790 0.6790</td>
<td>0.0006790 0.0006790</td>
</tr>
<tr>
<td>2 5-10</td>
<td>11.39 12.07</td>
<td>0.01139 0.01207</td>
</tr>
<tr>
<td>3 10-15</td>
<td>51.27 63.34</td>
<td>0.05127 0.06334</td>
</tr>
<tr>
<td>4 15-20</td>
<td>156.9 222.2</td>
<td>0.1569 0.2222</td>
</tr>
<tr>
<td>5 20-25</td>
<td>261.1 483.4</td>
<td>0.2611 0.4834</td>
</tr>
<tr>
<td>6 25-30</td>
<td>257.3 740.7</td>
<td>0.2573 0.7407</td>
</tr>
<tr>
<td>7 30-35</td>
<td>160.6 991.2</td>
<td>0.1606 0.9912</td>
</tr>
<tr>
<td>8 35-40</td>
<td>67.61 968.8</td>
<td>0.06761 0.9688</td>
</tr>
<tr>
<td>9 40-45</td>
<td>21.57 990.4</td>
<td>0.02157 0.9994</td>
</tr>
<tr>
<td>10 45-50</td>
<td>7.031 997.4</td>
<td>0.007031 0.9994</td>
</tr>
<tr>
<td>11 50-55</td>
<td>1.477 998.9</td>
<td>0.001477 0.9999</td>
</tr>
<tr>
<td>12 55-60</td>
<td>0.4419 999.3</td>
<td>0.0004419 0.9999</td>
</tr>
<tr>
<td>13 60-65</td>
<td>0.3675 999.7</td>
<td>0.0003675 0.9997</td>
</tr>
<tr>
<td>14 65-70</td>
<td>0.116 999.9</td>
<td>0.000116 0.9999</td>
</tr>
</tbody>
</table>

Figure 3.18: Comparison of the outputs of the simulation and the analysis.

Figure 3.18 shows that outputs of the analysis and of the simulation are clearly close and therefore the analysis can be deemed to provide a good result. After having compared the number of customers in the system, we want to compare the waiting time distribution of the analysis and of the simulation.
Figure 3.19: The waiting time distribution of the simulation

Figure 3.20 shows how close the simulation and the analysis are. Thus, the model should be considered to be validated.

Figure 3.20: Comparison of the waiting time density function of the simulation and analysis
To summarize, we can say that we can easily approximate all the service measures given a spares vector. In addition, we see that optimizing the fillrate doesn't automatically give an optimal solution for the average waiting and vice versa. Finally, the waiting time distribution cannot be known in advance; it depends on different factors, such as the spares vector, the repair time mean and distribution and arrival rates of the customers to each sub-system. At this point of the research, we leave the ERSOF-model and analyze another model which customers can bring products which failed in more than one class. Then, in Part II, we will deepen our understanding of ERSOF-models by analyzing a variety of different models. First, we will solve the fillrate by different methods, and then we will add constraints, objective functions and make the model "more interesting".
4. Modeling an Exchangeable-Item Repair System with more than one Failure Class (ERSMF)

4.0. Introduction

After having analyzed the ERSOF-model, we now move on to analysis of an ERSMF-model. But what is an ERSMF-model? As in the previous chapter, we also deal with an exchangeable-item repair system; but with several item classes. This is a system to which customers bring a product consisting of a variety of failed items for repair and get serviceable ones in return for all failed items. We suppose there are I classes of failure.

Figure 4.1: The repair facility with several item classes and several failures
possible and the overall arrival rate $\lambda$ is partitioned into rates $\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_I$ corresponding to the individual component failure rates. Thus, since the item arrival stream is Poisson with rate $\lambda$, the class-$i$ component failures form a Poisson stream with rate $\lambda a_i$. These are “sent” to a repairable ample server repair facility dealing only with class-$i$ failures, which is stocked with $n_i$ spares. We call this subsystem $i$, or again ss$_i$. A;sp, because of the ample server assumption, all repairs are carried out independently, and we can view the overall facility as partitioned into $I$ subsystems. This is depicted in Figure 4.1.

We assume that the system contains only two classes of items. The customers which arrive at the desk can be divided into three categories. One category (or customer type) brings only item class 1 (this is analogous to a product which failed only in item class 1), another category (customer type 2) brings one item of class 2 and the third category, or customer type 3, brings one of each (Table 4.1).

<table>
<thead>
<tr>
<th>Category</th>
<th>Item class 1</th>
<th>Item class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer type 1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Customer type 2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Customer type 3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.1: Example of the customer types with two item classes in the system

We denote the three customer types by $\bar{j}_1 = (1,0)$, $\bar{j}_2 = (0,1)$, $\bar{j}_3 = (1,1)$, respectively, and let $J$ be the set of all 3 combinations. Let $\Lambda$ be a vector r.v. of all different customers arriving to the system($\Lambda \in J = \{\bar{j}_1, \bar{j}_2, \bar{j}_3\}$). Therefore, the probability of specific combination $\bar{j}_s$ (specific customer type) to arrive to the system is:

$$P(\Lambda = \bar{j}_1) = \frac{a_1(1-a_2)}{1-a_1a_2}, \quad P(\Lambda = \bar{j}_2) = \frac{a_2(1-a_1)}{1-a_1a_2}$$

and $P(\Lambda = \bar{j}_3) = \frac{a_1a_2}{1-a_1a_2}$.

Generalizing, the customer can bring every variety of items which includes all possible combinations of items $\bar{j}_s \in J$. The product which the customer brings to the system may contain combinations of all $I$ different items of the system. Thus, the number of combinations $J$ will be $S = 2^I - 1$. Using historical data, we will calculate the probability of a customer arriving to the system bringing a specific product with specific
failures. Let \( \Lambda \) be a vector r.v. of all different customers arriving to the system 
\( \Lambda \in J = \{j_1, j_2, \ldots, j_s, \ldots, j_S \} \), where a specific customer brings a product which can be 
described by vector \( \vec{j}_s = (j_{s,1}, \ldots, j_{s, s}) \), where the probability of a failed item \( i \) to be in \( \vec{j}_s \) is 
\( a_i \). Thus, \( P(j_{s,i}) = \begin{cases} a_i & j_{s,i} = 1 \\ 1-a_i & j_{s,i} = 0 \end{cases} \). Therefore, the probability of specific combination 
\( \vec{j}_s \) (specific customer type) arriving to the system is:

\[
P(\Lambda = \vec{j}_s) = \prod_{i=1}^{I}[a_i-(1-j_{s,i})(2a_i-1)]
\]

\[
= \prod_{i=1}^{I}[2a_i j_{s,i} - a_i + 1 - j_{s,i}]
\]

\[
= \prod_{i=1}^{I}[a_i(2j_{s,i} - 1) + 1 - j_{s,i}] \quad (4.1)
\]

where \( s \) is the assigned customer type number between 1 and \( 2^l - 1 \). For example, if \( s = 5 \), then the binary code is 1,0,1,0,0,… . This means that customer type 5 brings a product 
which failed in item type 1 and in item type 3. Thus, the probability of a customer type 5 
to arrive at the system is \( a_1a_2a_3a_4a_5a_6\ldots a_{I} \)

### 4.1. Service measures

Before we continue to build an optimization model, we first analyze how to obtain 
measures for such a model. We write \( F_{i,n_i}(t) \equiv F_i(t | n_i) = P(T_i \leq t | n_i) \) obtaining 
these by appropriate use of the basic model. Thus, the average waiting time in sub-system \( i \) is 
given by:

\[
\overline{T}_{i,n_i} = \int_{0}^{\infty}(1 - F_{i,n_i}(t))dt \quad (4.2)
\]

We assumed that it is clear to the reader what it means "to wait". But in the next section, 
we will show that waiting time can have different meanings.
4.1.1. The Waiting Time Distribution

4.1.1.1. The Waiting Time of a specific customer type

The waiting time of a specific customer can be defined in two different ways:

The first satisfaction time $W_{F, j, \bar{n}}$

This is the time that a specific customer bringing $\bar{j}$ to a system with spares vector $\bar{n}$ must wait until he gets at least one of those items. This is also called the first response time. How long must the customer wait until he gets something from the repair facility. This is done for a specific customer type.

The last satisfaction time $W_{L, j, \bar{n}}$

This is the time that a specific customer bringing $\bar{j}$ to a system with spares vector $\bar{n}$ must wait until he gets ALL items repaired or replaced. This is generally what we define as the waiting time. We do not care if the customer received some of the items immediately, but rather we do care when he can actually leave the system.

4.1.1.2. The waiting time of a random customer

The waiting time of a random customer can be defined in two different ways:

The first satisfaction time $W_{F, \bar{n}}$

This waiting time is the time that a random customer arriving to a system with spares vector $\bar{n}$ must wait until he gets at least one item. Thus,

$$P(W_{F, \bar{n}} \leq t) = \sum_{s=1}^{\bar{s}} P(W_{F, j_s, \bar{n}} \leq t) P(\Lambda = j_s). \quad (4.3)$$

The last satisfaction time $W_{L, \bar{n}}$

The waiting time is the probability that a random customer arriving to a system with spares vector $\bar{n}$ must wait until he gets all items and can then leave. Thus,
\[ P(W_{L,\vec{n}} \leq t) = \sum_{s=1}^{S} P(W_{L,s,\vec{n}} \leq t) P(\Lambda = \vec{j}_s). \] (4.4)

In fact, we are talking in theoretic terms. We still don't know the probabilities \( P(W_{F,j_s,\vec{n}} \leq t) \) and \( P(W_{L,j_s,\vec{n}} \leq t) \) so that we can derive general formulas for a random customer. This part of the puzzle will be developed in Chapter 10, but at the moment we assume that they are known in developing our service measures.

### 4.1.2. The Average Waiting Time

#### 4.1.2.1. The average waiting time of a specific customer

The average waiting time of a specific customer can be defined in two different ways:

**The expected first satisfaction time** \( \overline{W_{F,j_s,\vec{n}}} \)

This is the expected time a specific customer bringing \( \vec{j}_s \) to a system with spares vector \( \vec{n} \) must wait until he gets the first item. Thus,

\[ \overline{W_{F,j_s,\vec{n}}} = \int_{t=0}^{\infty} P(W_{F,j_s,\vec{n}} > t)dt \] (4.5)

**The expected last satisfaction time** \( \overline{W_{L,j_s,\vec{n}}} \)

This is the expected time a specific customer bringing \( \vec{j}_s \) to a system with spares vector \( \vec{n} \) must wait until he gets all items. So,

\[ \overline{W_{L,j_s,\vec{n}}} = \int_{t=0}^{\infty} P(W_{L,j_s,\vec{n}} > t)dt \] (4.6)

#### 4.1.2.2. The average waiting time of a random customer

The average waiting time of a random customer can be defined in two different ways:

**The expected first satisfaction time** \( \overline{W_{F,\vec{n}}} \)
This is the expected time a random customer must wait until he gets his first item. Thus,

\[ \overline{W_{F,\bar{n}}} = \sum_{j_s \in J} \overline{W_{F,J_s,\bar{n}}} P(\Lambda = \overline{j_s}). \]  \hspace{1cm} (4.7)

The expected last satisfaction time \( \overline{W_{L,\bar{n}}} \)

This is the expected time a random customer must wait until he gets the last item. So,

\[ \overline{W_{L,\bar{n}}} = \sum_{s=1}^{S} \overline{W_{L,J_s,\bar{n}}} P(\Lambda = \overline{j_s}). \]  \hspace{1cm} (4.8)

4.1.3. The Fillrate

4.1.3.1. The Fillrate of a specific customer

The fillrate of a specific customer can be defined in two different ways:

The Fillrate based on time to first satisfaction \( FR_{F,j_s,\bar{n}} \)

This fillrate is the probability that a specific customer bringing \( \overline{j_s} \) to a system with spares vector \( \bar{n} \) receives at least one item without any delay.

The Fillrate based on time to receive all items \( FR_{L,j_s,\bar{n}} \)

This fillrate is the probability that a specific customer bringing \( \overline{j_s} \) to a system with spares vector \( \bar{n} \) will not have wait at all when he arrives at the system.

4.1.3.2. The Fillrate of a random customer

The Fillrate based on time to first satisfaction \( FR_{F,\bar{n}} \)

The fillrate is the probability that a random customer who arrives at the system gets at least one item straight away:
\[ FR_{F,\tilde{n}} = P(W_{F,\tilde{n}} = 0) = \sum_{j_s \in J} P(W_{F,j_s,\tilde{n}} = 0 | \Lambda = \tilde{j}_s)P(\Lambda = \tilde{j}_s). \quad (4.9) \]

**The Fillrate based on time to receive all items** \( FR_{L,\tilde{n}} \)

This fillrate is the probability that a random customer will not have to wait at all when he arrives at the system. Thus,

\[ FR_{L,\tilde{n}} = P(W_{L,\tilde{n}} = 0) = \sum_{j_s \in J} P(W_{L,j_s,\tilde{n}} = 0 | \Lambda = \tilde{j}_s)P(\Lambda = \tilde{j}_s). \quad (4.10) \]

### 4.1.4. The number of customers in system

The number of customers in the system cannot be calculated directly. This will be done in Part II where we analyze the ERSMF-model in greater detail. There, we will be able to show a way to correct the error of Hausman and Cheung.

### 4.1.5. The average number of customers in the system

The average number of customers in the system can be calculated in two ways. First, through the number of customers distribution, or second, through Little’s formula, when the average waiting time is known. Unfortunately, at this stage of the research, neither can be done, and therefore this is postponed until Chapter 10.

In this section, we showed that an ERSMF-model can have many different measures for the service level. Our goal is to define an optimization model, but there is no analytic formula for any of these service measures at this stage. Therefore, we need to wait for Part II of this research to determine these.
Part II
5. Optimizing Additional Service Criteria

5.0. Introduction

In Chapter 3, we analyzed a model where the average waiting time served as the objective function and the constraint was the available budget for spares. We obtained an approximate solution and a way to calculate several service measures. But what happens if the manager does not want to minimize the average waiting time? What happens if, for example, he wants to maximize the fillrate? What can he do? Is the fact that the fillrate contains a convex part as well as a concave part an obstacle for optimization? In section 5.1, we will use a method of solving non linear, non-continuous objective functions - - dynamic programming - - to maximize the fillrate. In section 5.2, we will present ways to improve the dynamic program to get more quickly to an approximation of the optimal solution, and in section 5.3, we will present a better algorithm for the fillrate maximization. Then, we will focus the research on optimizing two other service- measures - - the total number of customers in the system and the probability of waiting more than time x - - to be presented in sections 5.4 and 5.5, respectively.

5.1. Solving the fillrate problem using dynamic programming

In section 3.4, we assumed that there is no analytical solution to the problem of the fillrate. To recall, we were given an ER Sof-system, where the objective function is not the average waiting time, but the fillrate. In this case, we wanted to maximize the percentage of customers who get service without waiting. That is, we wanted to solve the following model:

\[
\text{max } FR_x = \frac{1}{\lambda} \sum_{i=1}^{L} \lambda_i \cdot FR_i(n_i)
\]

s.t.:

\[
\sum_{i=1}^{L} c_i n_i \leq M
\]
\[ n_i \in \eta, \quad \eta = \{0, 1, 2, \ldots\}, \]
\[ i \in \xi, \quad \xi = \{1, 2, \ldots, I\}. \]

Here, we make an assumption that wherever we need to make the \( n_i \)'s continuous to simplify the presentation, then we do so. Our standard example presented in section 3.8 contains 13 classes (I) and a budget of \( M = 1000 \) dollars. Therefore, due to the fact that the prices of items of all classes are in units of dollars, we developed the following dynamic programming model. \( FRT(k, i) \) is the total fillrate for \( k \) dollars when distributed optimally among a specific set of \( i \) classes. Using the idea of dynamic programming, we use the recursive function, \( FR_{k-j, i} \), the fillrate of sub-system \( i \) with \( k-j \) dollars allocated optimally among its classes. Thus, \( FRT(0, 0) = 0 \) and

\[ FRT(k, i) = \max_j \left( FRT(j, i - 1) + \frac{\lambda_i FR_{k-j, i}}{\lambda} \right) \text{ for } j=1, \ldots, k, \text{ and } k=1, \ldots, M, \ i=1, \ldots, I. \]

We start with \( k=1 \) and do recursion it until we spend \( M \) dollars. At each stage, we optimize by adding another class. For every item, we need \( \Theta(M^2) \) instructions and therefore the complexity of the algorithm is \( \Theta(IM^2) \). Thus, the problem \( FRT(M, I) \) is solved using an exact pseudo-polynomial algorithm. The following table shows an example of the runtime of the software on a standard Pentium 4 computer. Table 5.1 and Figure 5.1 illustrate that the more classes there are in the system and the higher the budget, the time to calculate the optimal solution exhibits polynomial growth, as expected.

<table>
<thead>
<tr>
<th>Dollars</th>
<th>100</th>
<th>1000</th>
<th>2000</th>
<th>2500</th>
<th>5000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>37</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>48</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>13</td>
<td>75</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>16</td>
<td>94</td>
</tr>
<tr>
<td>200</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>7</td>
<td>26</td>
<td>149</td>
</tr>
</tbody>
</table>
Table 5.1: Time (in seconds) for different amounts of classes and budget

<table>
<thead>
<tr>
<th>Classes</th>
<th>$10000</th>
<th>$25000</th>
<th>$50000</th>
<th>$100000</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec</td>
<td>0</td>
<td>20</td>
<td>60</td>
<td>160</td>
</tr>
</tbody>
</table>

Figure 5.1: Graph which shows the results given in Table 5.1.

Figure 5.1 shows that, as expected, the time grows with increasing budget or classes. 200 classes with 10000 dollars is a medium problem which takes more than 2 minutes of running time on our PC. However, the actual algorithm is clearly inadequate to solve very large problems.

5.2. An approximate dynamic programming algorithm for large scale problems

The algorithm developed in the previous problem gives an exact solution to the problem. Indeed, for large size problems, approximations are good enough as long as the problem is solved in a reasonable time. Therefore, the following algorithm was developed, which is applied to large size problems. Instead of using steps of 1 dollar, we will take steps of $s$ dollars, where $s$ varies from 10 to 200 or 1000, depending on the size problem. A complete analysis of the algorithm was not done, but an example to illustrate the efficacy of the algorithm is still useful to show the power of the tool. As explained above, all the examples were taken from the same sample, so we don’t have enough data.
to get definitive results; but in any case, the following example demonstrates the quality of our example. The problem includes 1000 dollars and the steps were by 1 dollar.

Figure 5.2: The Optimal Fillrate when calculated with different steps

Figure 5.2 shows that larger steps do not lead the algorithm to miss the optimal solution. Only when steps approach 100, which is 10% of the total amount, is the optimal solution missed. In any case, due to the fact that a better algorithm will be developed, the continuation of the analysis of using dynamic programming will be abandoned.

5.3. Developing another algorithm for the fillrate

The idea of developing another algorithm was conceived in several stages. In the first step, we concluded that it is impossible that there be no algorithm which does not take into account the shape of the function. This presumption brought us to consider a
new algorithm. As mentioned above, the fillrate of the system is defined as the weighted sum of the fillrate of each sub-system. Each sub-system behaves like Figure 5.3.

![Figure 5.3: The Fillrate of a sub-system](image)

In fact, the optimal algorithm tries to use the steepest slope, where fillrate versus dollars is the highest. In a function which is only convex, like the average waiting time, this can be done easily. Therefore, the initial solution starts at the inflection point where the slope is the greatest and tries to find better solutions in two ways. The first way is to find a local optimum by exchanging items. This means that if the algorithm sees that adding an item to class j and taking off one item from class k, improves the fillrate, we will keep on exchanging until there is no further option for exchanging items. (In fact, this is done, using dollars instead of items, due to the fact that here we made the items “continuous”). In a second step, we try to find other local optima, by taking off several items of class j and replacing them by other items.

**Algorithm 3:**

This algorithm starts at the inflection point for each class. Then by interchanging items of different classes, we improve the fillrate and get the local optimum. This is an "area". By finding all other areas and finding their local optima, we can find the global optimum.
a. $x_1, x_2, x_3, \ldots, x_I$ are initialized as mentioned (3.24).

b. Calculate $\theta_i = \frac{a_i \cdot FR_i(x_i)}{c_i}$, for $i = 1, \ldots, I$.

c. Set $\theta_l = \min_i \theta_i$ and the $\theta_h = \max_i \theta_i$, where $l$ and $h$ are the indices of the item classes which were chosen.

d. If $\theta_l = \theta_h$, we have a local optimum; go to h.

e. If not, calculate $c_l = (\lfloor x_i + 1 \rfloor - x_i)c_i$ and $c_h = (x_h - \lfloor x_h - 1 \rfloor)c_h$.

f. Choose the smaller value of $c_l$ and $c_h$: $c_r = \min(c_l, c_h)$

g. $x_i = x_i + c_r / c_i$ and $x_h = x_h - c_r / c_h$. Go to b.

h. $\theta_i = \sum_{j=1}^{s} \frac{a_j \cdot FR_i(x_j - j)}{s c_i}$, for $s = 1, \ldots, x_i$ and $i = 1, \ldots, I$.

i. Set $\theta_l = \min_i \theta_i$ and $\theta_h = \max_i \theta_i$, where $l$ and $h$ are the indices of the item classes which were chosen.

j. If $\theta_l < \theta_h$, $x_i = x_i - j$, go to b.

k. Compare all local optima to find the global optimum.

Figure 5.4 shows the output of the software which implements the algorithm.

```
1.step
results-------------------
nitems:13
fillrate: 0.836065
time0
amount1000
```

Figure 5.4: Output for the fillrate when 1000 dollars are available
Figure 5.4 shows the output of the software developed for that purpose. It shows also that the solution is instant. Interesting to note that the result can be cross-checked with the previous solution using dynamic programming.

5.4. The total number of customers in the system

After having solved the model with the fillrate as the objective function, we now solve a model with the total number of customers in the system as objective function. Consider an EROF-system in which every customer brings a product which failed in exactly one component. Let \( \bar{n} = (n_1, n_2, \ldots, n_I) \) be the spares vector of the system, \( L_i(n_i) \) the expected number of customers in \( s_i \) having \( n_i \) spares, and \( L_{\bar{n}} \) the total number in the system. Thus, we have

\[
\min L_{\bar{n}} = \sum_{i=1}^{I} L_i(n_i) \tag{5.2}
\]

s.t.:

\[
\sum_{i=1}^{I} c_i n_i \leq M ,
\]

\[
n_i \in \eta, \quad \eta = \{0,1,2,\ldots\}
\]

\[
i \in \xi, \quad \xi = \{1,2,\ldots,I\}.
\]

By Little’s formula, we know that \( L_i(n_i) = \lambda_i T_i(n_i) \), so that \( \min L_{\bar{n}} = \sum_{i=1}^{I} \lambda_i \overline{T_i(n_i)} \), and thus, the model to optimize is now

\[
\min L_{\bar{n}} = \sum_{i=1}^{I} \lambda_i \overline{T_i(n_i)} \tag{5.3}
\]

s.t.:

\[
\sum_{i=1}^{I} c_i n_i \leq M ,
\]

\[
n_i \in \eta, \quad \eta = \{0,1,2,\ldots\}
\]

\[
i \in \xi, \quad \xi = \{1,2,\ldots,I\}.
\]

the same model as (3.26).
5.5. The probability of waiting less than \( x \).

The service levels discussed until now did not focus on the customer waiting time in the system. In this ERSOF model we want to maximize the probability \( F_n(x) \) that a random customer waits no longer than time \( x \). Thus we have

\[
\max z = F_n(x) = \sum_{i=1}^{I} \frac{\lambda_i}{\lambda} F_i(x | n_i)
\]

s.t.:

\[
\sum_{i=1}^{I} c_i n_i \leq M.
\]

\( n_i \in \eta, \quad \eta = \{0,1,2,\ldots\} \)

\( i \in \xi, \quad \xi = \{1,2,\ldots,I\} \).

This problem will be solved later, in section 8.3, using integer programming.

To summarize, we can say that we developed an efficient algorithm for the fillrate. We showed how to solve the model with the fillrate as the objective function. Another service measure we want to use in our optimization model is the probability of waiting less than \( x \), but with standard analytical methods we cannot solve it. But we will solve them later in Chapter 8 where we develop a method to solve any problem using integer programming. But for the moment we leave the objective functions and relax another constraint of our optimization model which is the assumption that we have exactly one constraint.
6. Different ER Sof-Models

6.0. Introduction

In Chapter 5, we considered an ER Sof-model in which we wanted to maximize a service criterion with a budget constraint. An ER Sof-model is a model of a system to which customers bring a product containing different components. But the product can fail only via one of its components. In this chapter, we will discuss models in which we wish to minimize the budget with a constraint on the service criterion (such as the average waiting time or the fillrate).

6.1. The Average Waiting Time

We now consider an ER Sof-model in which we want to minimize the budget required under the constraint of a desired average waiting time. We have theoretically an infinite budget at our disposal, but like in any industry, we don’t want to spend more than necessary, and therefore, we want to invest it “well”, and buy items from classes which have a major impact on the average waiting time. In fact, this model is in some sense the dual of the previous model discussed in section 3.4. The model with budget as constraint looks like:

$$\min z_1 = \sum_{i=1}^{l} \frac{\lambda_i \tilde{T}_{l,r}}{\lambda}$$

(6.1)

s.t.: $$\sum_{i=1}^{l} c_i n_i \leq M$$.

(6.2)

In Chapter 3, we discussed the model (6.1) where we minimized the average waiting time under the constraint of available money (6.2). There, we designated this as the first model: its dual, under the average waiting time constraint, as the second:
\[
\min z_2 = \sum_{i=1}^{I} c_i n_i \\
\text{s.t.: } \sum_{i=1}^{I} \frac{\lambda_i T_{i,n_i}}{\lambda} \leq SL
\]  

(6.3) \hspace{1cm} (6.4)

where SL is the imposed limit required for the average waiting time of the system.

The following theorems will prove that the models are quite similar in the sense that the vector \( \bar{n} \) is the same when inputting a pair of values. For example, if with M dollars we can achieve a service level of SL, then we also have to pay M if a limit of SL is required for the average waiting time. This might seem obvious, but it needs to be proven that the optimal vector \( \bar{n} \) will be the same in both cases. That is why we talk about the dual problem. We look at the exact same problem from a different perspective. Later, this will help us to validate our analyses by solving both. The idea behind Theorem 1 is that a given pair \((M, SL)\) is interchangeable with the pair \((SL, M)\) in the sense that by defining the first value as a constraint (M for the first model, SL for the second model), the second value will be the result of the objective function (SL for the first model, M for the second). This idea leads to a graph where the x-axis is the budget and the y-axis the service level. (see Figure 6.1). More than that, the vector \( \bar{n}^* \) will be the same in both models.

**Theorem 1:**

The optimal solution \((\bar{n}^*, z_1^*, M)\) with parameter M for the first model is also the optimal solution \((\bar{n}^*, z_2^*, SL)\) with parameter SL for the second model, where \(z_1^* = SL\), \(z_2^* = M\). That is, \((\bar{n}^*, z_1^* = SL, M) = (\bar{n}^*, z_2^* = M, SL)\).

**Proof:** By contradiction.

Suppose that vector \( \bar{n}^* = (n_1^*, ..., n_I^*) \) is the optimal solution to the first model. This means that the vector satisfies (due to the assumption that \( n_i \) is continuous) the equation

\[
\sum_{i=1}^{I} c_i n_i = M \text{ and minimizes } z_1 = \sum_{i=1}^{I} \frac{\lambda_i T_{i,n_i}}{\lambda} \text{ with value SL. This same vector satisfies}
\]

64
\[ \sum_{i=1}^{I} \frac{\lambda_i T_{i,n_i}}{\lambda} = SL \], but we will assume that it doesn’t minimize \( z_2 = \sum_{i=1}^{I} c_i n_i \). Because of the assumption that (6.3) is not optimized, we can do better in the sense that we can exchange at least one item of class \( i \) by another item of class \( j \) to get \( \bar{n}^{**} \), still satisfying the constraint. Thus, \( z_2^* = \sum_{i=1}^{I} c_i n_i^* \) and \( z_2^{**} = \sum_{i=1}^{I} c_i n_i^{**} \), such that \( z_2^{**} < z_2^* \), where the constraint \( \sum_{i=1}^{I} \frac{\lambda_i T_{i,n_i}}{\lambda} = SL \) is in both cases satisfied. But, due to the fact that \( z_2^* = M \), we have \( z_2^{**} = M_1 < M \). However, the first model is optimized, so trying to minimize the objective function of the first model with only \( z_2^{**} \) dollars, we will get a service level \( z_1^* \) which is smaller than SL. But by definition, \( z_1^* = SL \); thus, we have a contradiction. Hence, for every specific budget, we will get a specific service level and vice versa.

Q.E.D.

Let us now proceed to solve the second model. Standardize the problem by calculating the Langrangian \( \theta \) as follows:

\[ L_2 = \sum_{i=1}^{I} c_i n_i + \theta \left( \sum_{i=1}^{I} \frac{\lambda_i T_{i,n_i}}{\lambda} - SL \right) \]

whence

\[ \frac{\partial L_2}{\partial n_i} = c_i + \theta \frac{\lambda_i T_{i,n_i}}{\lambda} = 0, \quad i = 1, \ldots, I \]

and hence

\[ \theta = -\frac{\lambda c_i}{\lambda_i T_{i,n_i}}, \quad i \in \xi. \quad (6.5) \]

The two models are in fact the same, but are viewed through a different perspective. Figure 6.1 and Figure 6.2 show these different perspectives. If we calculate the differences (slope) of each graph, we will get (3.27) and (6.5), which represents the different graphs. That is why we talk about different perspectives of the same model. In fact, the graph of the average waiting time for each sub-system looks like Figure 6.1.
whereas the graph of the budget allocation looks like Figure 6.2.

Figure 6.1 An example of the average waiting time versus budget allocation

Figure 6.2: An example of budget allocation versus the average waiting time
Both graphs have the same properties such as convexity, but with different slopes. Thus, we can use the algorithm developed in section 3.5 with a slightly different $\theta$ to minimize the money objective function.

### 6.1.1. Validation of the algorithm

In this section, we present an example of the algorithm of 6.1. For that purpose, we developed two programs. The first program was developed to solve the first model, and the second for the second model. Obviously, we expect to get the solution of the dual model, which was presented in the previous chapter. The input to the first program is the budget allocated and the output will be the optimal spares vector and the average waiting time. In the second program we input the average waiting time required and output the dollars that the system would cost and the corresponding optimal spares vector for the items to buy. We present both examples and assess the results.

1. Example:

   ![Image of the first program's output](image1.png)

   **Figure 6.3:** Example of the interface of the first program with 1000 dollars

   ![Image of the second program's output](image2.png)

   **Figure 6.4:** Example of the interface of the second program with the AWT given

In the first program, the software ran with an input of 1000 dollars, which means that we have at disposal 1000 dollars to spend on spares. The number of spares for each sub-system starting from left to right is 11 for the first sub-system, 6 for the second and so on.
The Average Waiting Time of this system containing these numbers of spares is 0.369582. For the second run with the input 0.369582, which is the maximum allowable average waiting time permitted, we wanted to know what budget is required. The result obtained was 1000 dollar, as expected. Although, these are two different programs, the resulting vector is the same, as expected.

2.Example:

![Figure 6.5: Example of the interface of the first program with 500 dollars](image)

![Figure 6.6: Example of the interface of the second program with the AWT given](image)

Overall, these programs were run on 50 different examples and the results were ALL virtually identical in both programs. The examples above show small irrelevant differences because of heavy mathematical calculations with resultant rounding operations performed by the software.

6.2. The fillrate

We first analyzed the average waiting time minimized with the constraint of the budget and its inverse, where we wanted to minimize the money spent with the constraint of a given average waiting time. Now we want to focus our research on another service-level, which was analyzed several times during this research, namely the Fillrate. Recall
that the fillrate is the probability of a random customer obtaining service immediately without waiting. In section 3.4.1, we discussed the first model below, which maximizes the average fillrate:

\[
\text{Max } z_1 = \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i FR_{i,n_i} \tag{6.6}
\]

\[
s.t.: \sum_{i=1}^{I} c_i n_i \leq M \tag{6.7}
\]

If we look at the inverse of this model, then, as in the previous case with the average waiting time, we get the following model where ValueFR is the required fillrate of the system:

\[
\text{Min } z_2 = \sum_{i=1}^{I} c_i n_i \tag{6.8}
\]

\[
s.t.: \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i FR_{i,n_i} \geq \text{ValueFR} \tag{6.9}
\]

In this model, we want to minimize the expenditures on the spares to achieve a given system fillrate. \(z_1\) is in terms of "probability", whereas \(z_2\) is in terms of budget.

In the previous section we showed that the two models form a pair, where one value is expressed in the form of budget allocation and the second in the form of service level. In this section, we will show that the same idea can also be applied to the average fillrate of the system. In the previous section, the graphs of the average waiting time versus the budget allocation, and the budget allocation versus the average waiting time were both convex. To illustrate the behavior of both models, the fillrate of subsystem i is graphed in Figure 6.7.
A similar algorithm to the one developed in section 6.1 can be developed finding the optimal vector \( \bar{n}^* \). Our starting point is also the inflection point, where we get the most “fillrate” for the money in the sense that an incremental improvement of the fillrate is the
cheapest. We did not pursue this line of research, but we are confident that the models are also interchangeable as with the average waiting time in the sense that the solution of one model is also adequate as a dual for the other.

6.3. Connecting the constraint and the objective function

Until now, we analyzed ERSOF models with a variety of constraints, including available funds. But in today’s business world, a planner of a repair facility need not have a specific sum of money at his disposal, but rather, he can assign a penalty for a low level of service. In this class of models, we minimize the total cost of the system, including the price of the items as well as an implied “Shortage-Cost” incurred for low quality service.

6.3.1. The fixed time penalty in an ERSOF-model

A customer arrives to the system, enters the queue if he does not get full satisfaction, (where full satisfaction means that he gets all his failed components which failed replaced), and leaves the system after receiving full satisfaction. But what is the motivation of the manager of the system to buy spares, to improve the system? The motivation is internal or external. Internal means that the manager feels that he cannot let customers wait indefinitely, so he buys spares. External means that the company has signed a contract obligating to pay a certain amount of money for prolonged delay of satisfaction. In our case this refers to the waiting time. If the repair facility is operated by the company, the product which failed is not usable and incurs other loss of profits. Thus, in all these cases, it can be estimated how much it is worth to the system to provide quicker service. In this model, we assume that it is worth $b$ dollars for every unit of time a customer waits. That’s why the manager must decide how many spares to buy. In the following sections, we will deal with two different models. One, where we assume that the required policy for buying spares is for a given period, such as one year or 10 years, or where the spares are eventually scrapped and not worth anything at the end of the period. The other model will deal with an infinite period, taking into account the fact that product failures may also occur over a long time, by using interest rate and present value considerations.
6.3.1.1. The fixed time penalty in an ERSOF-model for a given period

When a customer arrives at the system, he enters the queue and will subsequently leave the system after satisfaction. His waiting time in $s_{si}$ will incur a penalty of $b_i$ per unit time for the system. Thus, for a period $\tau$, an average of $\lambda \tau$ customers will arrive at the system. Each customer who arrives at sub-system $i$, $s_{si}$, will have to wait $\bar{T}_i$ on average and incur an average cost of $b_i \bar{T}_i$ dollars due to his waiting. Thus, the overall problem reduces to

$$\min TC = \left( \sum_{i=1}^{I} b_i \bar{T}_{i,n_i} * \lambda_i T \right) + \left( \sum_{i=1}^{I} c_i n_i \right)$$

where $\sum_{i=1}^{I} c_i n_i$ is the amount spent for the spares. Treating the $n_i$ for the present as continuous variables,

$$\frac{\partial TC}{\partial n_i} = b_i \bar{T}_{i,n_i} * \lambda_i T + c_i = b_i \frac{L_{i,n_i}}{\lambda_i} * \lambda_i T + c_i = b_i T L_{i,n_i} + c_i = 0, \quad i \in \eta.$$

From (3.25) we have

$$L_{i,n_i} = -\sum_{k=1}^{\infty} \left[ \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right],$$

so that

$$\sum_{k=1}^{\infty} \left[ \frac{(\lambda_i / \mu_i)^{k+n_i}}{(k+n_i)!} e^{-(\lambda_i / \mu_i)} \right] = \frac{c_i}{Tb_i},$$

or,

$$\sum_{k=n_i+1}^{\infty} \left[ \frac{(\lambda_i / \mu_i)^{i}}{k!} e^{-(\lambda_i / \mu_i)} \right] = \frac{c_i}{Tb_i}.$$

Thus,

$$\sum_{k=0}^{n_i} \left[ \frac{(\lambda_i / \mu_i)^{k}}{k!} e^{-(\lambda_i / \mu_i)} \right] = 1 - \frac{c_i}{Tb_i}.$$
But we know that the normal distribution is a good approximation for the Poisson distribution when \( \lambda_i / \mu_i \gg 1 \). In this case, we can use instead \( X \sim N(\lambda_i / \mu_i, \sqrt{\lambda_i / \mu_i}^2) \). Thus, \( P(X \leq n_i) = 1 - \frac{c_i}{\theta_i} \).

6.3.1.2. The fixed time penalty in an ERSOF-model for an infinite period

When a customer arrives at the system, he enters the queue and leaves the system after satisfaction. His waiting time will incur a penalty for the system: For each \( s_i \), a customer who waits in the queue will cost \( b_i \bar{T}_i \) dollars on average. On average, the first customer arrives at time \( \frac{1}{\lambda_i} \), the second at \( \frac{2}{\lambda_i} \), and the \( k \)-th customer at time \( \frac{k}{\lambda_i} \). Unlike the previous problem, the policy we are looking for is for \( \tau \to \infty \). If the interest rate is \( \alpha \), the present value of \( b_i \) dollars at time \( \frac{k}{\lambda_i} \) is \( \frac{b_i}{(1 + \frac{\alpha}{\lambda_i})^k} \). Thus, the total average cost of buying the spares and all penalties for all customers who will arrive to the system is:

\[
TC = \left( \sum_{i=1}^{I} \sum_{k=1}^{\infty} \frac{b_i}{(1 + \frac{\alpha}{\lambda_i})^k} \frac{\bar{T}_{i,n_i}}{\lambda_i} \right) + \left( \sum_{i=1}^{I} c_i n_i \right)
\]

Hence,

\[
TC = \left( \sum_{i=1}^{I} b_i \frac{\bar{T}_{i,n_i}}{\lambda_i} \right) + \left( \sum_{i=1}^{I} c_i n_i \right)
\]

(For the development of (6.10) see Appendix G)

Thus, minimizing \( TC \),

\[
\frac{\partial TC}{\partial n_i} = \frac{\lambda_i b_i}{\alpha} \bar{T}_{i,n_i} \frac{1}{\lambda_i} + c_i = \frac{b_i}{\alpha} \bar{T}_{i,n_i} + c_i = 0,
\]

So that

\[
-L_i = \frac{\alpha c_i}{b_i} = \sum_{k=n_i+1}^{\infty} \left( \frac{(\lambda_i / \mu_i)^k}{k!} e^{-\lambda_i / \mu_i} \right).
\]

But we know that the normal distribution is a good approximation for the Poisson distribution for \( \lambda_i / \mu_i \gg 1 \). Thus, \( X \sim N(\lambda_i / \mu_i, \sqrt{\lambda_i / \mu_i}^2) \), and
\[ P(X \leq n_i) = 1 - \frac{\alpha c_i}{b_i} > 0, \text{ if and only if } \frac{\alpha c_i}{b_i} < 1. \text{ Consequently, } \alpha < \frac{b_i}{c_i}. \]

If \( b_i < \alpha c_i \) this means that the penalty is so low that it is not worth buying any spares.

6.3.2. The fixed penalty for any not immediately satisfied customer in an ERSOF model

In this model, every customer bringing a product which failed in component \( i \), and who arrives at the system and does not get immediate satisfaction will get \( b_i \) dollars.

6.3.2.1. The fixed penalty for any not immediately satisfied customer in an ERSOF model for a given period \( \tau \)

In this model, we want to minimize the total average cost. The fillrate provides the probability that a customer to \( s_s \) is immediately satisfied. Thus, the average number of not immediately satisfied customers during period \( \tau \) is \( \sum_{i=1}^{l}(1- FR_{i,n_i})\lambda_i \tau \). The total cost for the whole system over \((0, \tau)\) will be

\[ TC = \sum_{i=1}^{l}(1- FR_{i,n_i})\lambda_i \tau b_i + \sum_{i=1}^{l}c_i n_i. \]

Thus,

\[ \frac{\partial TC}{\partial n_i} = -\lambda_i \lambda_i (FR_{i,n_i})^' + c_i = 0. \]

From (3.22)

\[ FR_{i,n_i} = \frac{(\lambda_i / \mu_i)^{n_i} e^{-(\lambda_i / \mu_i)}}{(n_i)!} = \frac{c_i}{\lambda_i \mu_i}. \]

If \( X_i \) is Poisson with parameter \( \lambda_i / \mu_i \), and \( n_i \) is dealt as continuous then

\[ P(X_i = n_i) = \frac{c_i}{\lambda_i \mu_i}. \quad (6.11) \]

From Normal tables we can find the values for \( n_i \).
But where is the optimal value of the fillrate? From (6.11) we conclude that the optimal value for each $s_{si}$ can be calculated independently, and so, we will analyze each $s_{si}$ separately.

<table>
<thead>
<tr>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>cl*ni</td>
</tr>
</tbody>
</table>

**Figure 6.9: Total cost of $s_{si}$**

Figure 6.9 shows a typical example of the graph of $s_{si}$. In our example, we see two minima, one at 0 and one “in the middle”. The graph is the sum of a straight line and the fillrate which is divided into a convex and concave part. Thus, the function is also divided into a convex and a concave part. Thus, there are at most 2 local minima.

Here, $TC_{i,n_i}$ is the total cost when $n_i$ spares are bought for $s_{si}$. Thus, $TC_{i,0} = b_i \lambda_i \tau$.

The answer to our problem will be $\min(TC_{i,n_i}, TC_{i,0})$ with $n_i$ calculated by (6.11). Hence,

$$TC_{i,n_i} = \min(b_i \lambda_i \tau, b_i \lambda_i \tau (1 - FR_{i,n_i}) + c_i n_i)$$

$$= b_i \lambda_i \tau + \min(0, -b_i \lambda_i \tau FR_{i,n_i} + c_i n_i) \quad , i \in \eta$$

6.3.2.2. The fixed penalty for any not immediately satisfied customer in an ERSOF model for an unbounded interval

When a customer arrives at the system, he enters the queue of the class which failed in his product and leaves the system after satisfaction. If he does not get immediate
satisfaction, the system incurs a penalty. If the interest rate is $\alpha$, then the present value cost of any customer to $ss_i$ is

$$PV = \frac{b_i \lambda_i (1 - FR_{i,n_i})}{\alpha}.$$ 

The probability of not getting instant satisfaction is $(1 - FR_{i,n_i})$. Thus, the total average discounted cost for the system, $TAC$, is given by

$$TAC = \sum_{i=1}^{I} \frac{b_i \lambda_i}{\alpha} (1 - FR_{i,n_i})^2 + \sum_{i=1}^{I} c_i n_i.$$ 

Hence,

$$\frac{\partial TAC}{\partial n_i} = -2 \frac{b_i \lambda_i}{\alpha} (1 - FR_{i,n_i})(FR_{i,n_i})^0 + c_i = 0,$$

and

$$(1 - FR_{i,n_i})(FR_{i,n_i})^0 = \frac{\alpha c_i}{2b_i \lambda_i}.$$ 

Hence,

$$\frac{\partial (1 - FR_{i,n_i})^2}{\partial n_i} = \frac{\alpha c_i}{2b_i \lambda_i}.$$ 

Thus,

$$(1 - FR_{i,n_i})^2 \bigg|_{n_i}^{n_i} = -\frac{\alpha c_i}{b_i \lambda_i} \bigg|_{n_i}^{n_i}$$

or

$$(1 - FR_{i,n_i})^2 - 1 = -\frac{\alpha c_i}{b_i \lambda_i} n_i,$$

which leads to

$$\frac{1 - (1 - FR_{i,n_i})^2}{\alpha c_i} \left(\frac{b_i \lambda_i}{\alpha c_i}\right) = n_i, \text{ for } i \in \xi.$$ 

$FR_{i,n_i}$ is a monotonic increasing function between 0 and 1. Hence, $1 - (1 - FR_{i,n_i})^2$ is also a monotonic increasing function. Thus, to solve this kind of equation, we can use any one dimensional search method.
We set \( n_i \equiv x \) and \( \frac{[1 - (1 - FR_{i,n_i})^2]}{\alpha c_i} \equiv h(x) \)

Now we choose an initial \( x_0 \) and calculate \( h(x_j) = x_{j+1} \) to get \( x_1 \). We continue until \( x_j \) converges.

### 6.3.3. The fixed penalty for delay longer than time \( x \) in an ERSOF model

As explained in the introduction to this chapter, the manager wants to know how many items to buy for the whole system. In this model, the motivation for him to buy items is the fact that every customer with item \( i \) who arrives at the system and has to wait longer than time \( x \) will cost \( b_i \) as compensation for waiting "too long".

#### 6.3.3.1. The fixed penalty for delay longer than time \( x \) in an ERSOF model for a given fixed time period

The total average cost of the system for time period \((0, \tau)\) will be

\[
TAC = \sum_{i=1}^{I} (1 - F_{i,n_i}(t)) \lambda_i \tau b_i + \sum_{i=1}^{I} c_i n_i .
\]

Hence to find the optimal spares vector, we derive this objective function.

\[
\frac{\partial TAC}{\partial n_i} = 
[(1 - F_{i,n_i+1}(t)) \lambda_i \tau b_i + c_i (n_i + 1)] - 
[(1 - F_{n_i}(t)) \lambda_i \tau b_i + c_i n_i]
\]

for \( i = 1, ..., I \).

To find the optimal, we will check where \( [(F_{i,n_i}(t) - F_{i,n_i+1}(t)) \lambda_i \tau b_i + c_i n_i] \geq 0 \) for the first time.

#### 6.3.3.2. The fixed penalty for delay longer than time \( x \) in an ERSOF model for over an unbounded interval

As with the previous models, we now deal with an infinite time model, where the values of future penalties are calculated and aggregated in the form of the present value. Therefore, the total average discounted cost is:

\[
TADC = \sum_{i=1}^{I} \frac{b_i \lambda_i}{\alpha (1 - F_{i,n_i}(t))^2} + \sum_{i=1}^{I} c_i n_i ,
\]

from which
\[
\frac{\partial TC}{\partial n_i} = -\frac{2b_i\lambda_i}{\alpha} (1 - F_{i,n_i}(t))(F_{i,n_i}(t))' + c_i = 0.
\]

Thus,

\[
(1 - F_{i,n_i}(t))(F_{i,n_i}(t))' = \frac{\alpha c_i}{2b_i\lambda_i}
\]

and

\[
\frac{\partial (1 - F_{i,n_i}(t))^2}{-2\partial n_i} = \frac{\alpha c_i}{2b_i\lambda_i}.
\]

Hence,

\[
(1 - F_{i,n_i}(t))^2\bigg|_0^n = -\frac{\alpha c_i}{b_i\lambda_i} n_i
\]

\[
(1 - F_{i,n_i}(t))^2 - (1 - F_{i,0}(t))^2 = -\frac{\alpha c_i}{b_i\lambda_i} n_i.
\]

Since \( F_{i,0}(t) = G_i(t) \). Therefore, we can write \( (1 - F_{i,n_i}(t))^2 - (1 - G_i(t))^2 = -\frac{\alpha c_i}{b_i\lambda_i} n_i \) as

\[
-\frac{b_i\lambda_i}{\alpha c_i} \left[(1 - F_{i,n_i}(t))^2 - (1 - G_i(t))^2\right] = n_i \quad i \in \mathcal{X},
\]

and use a numerical algorithm to solve it.

We set \( n_i = x \) and \( \frac{b_i\lambda_i}{\alpha c_i} \left[(1 - G_i(t))^2 - (1 - F_{i,n_i}(t))^2\right] \equiv h(x) \).

\( F_{i,n_i} \) is a monotonic increasing function between 0 and 1. Thus,

\[
\left[(1 - G_i(t))^2 - (1 - F_{i,n_i}(t))^2\right]
\]

is also a monotonic increasing function. Then to solve

\( h(x) = x \), we choose an initial \( x_0 \) and calculate \( h(x_j) = x_{j+1} \) until \( x_j \) converges.
7. Multiple Goals

7.0. Introduction

After all these developments, we will now proceed to multiple goals. Until now, we had one or more constraints and only one objective function, either the average waiting time or the fillrate or the probability of waiting a certain time. In section 7.1, we combine two objective functions to be one new objective function. In section 7.2, we see what happens if a goal is prescribed and the system is penalized for missing the goal, and in section 7.3, we proceed to Goal Programming.

7.1. Combining two objectives

What happens if the manager of a company wants to maximize the fillrate and minimize the average waiting time as well? Thus, he would get the following model:

\[
\begin{align*}
\text{Max} \left( z_1 = \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i FR_{i,n_i} \right) & \quad (7.1) \\
\text{Min} \left( z_2 = \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i \overline{T_{i,n_i}} \right) & \quad (7.2)
\end{align*}
\]

\text{s.t.:}

\[\sum_{i=1}^{I} c_i n_i \leq M \]

\[n_i \geq 0 \text{ for all } i=1,\ldots,I\]

One possibility for dealing with this kind of problem is to create a single objective function, incorporating relative weights of the objectives, assuming of course, that they are known or that they can be estimated. In this case, we would have

\[
\begin{align*}
\text{Max}(z = a_1 \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i FR_{i,n_i} - a_2 \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i \overline{T_{i,n_i}}) & \quad (7.3) \\
\text{s.t.:}
\end{align*}
\]

\[\sum_{i=1}^{I} c_i n_i \leq M, \]
\( n_i \geq 0 \) for all \( i=1,\ldots,I \)

where \( a_1 \) and \( a_2 \) \((a_1 + a_2 = 1)\) are the relative weights of the fillrate and the average waiting time respectively.

To get the optimal solution, we form the Lagrangian

\[
L(\bar{n}, \theta) = \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i FR_{i,n_i} - \frac{a_2}{a_1} \frac{1}{\lambda} \sum_{i=1}^{I} \lambda_i T_{i,n_i} + \theta(M - \sum_{i=1}^{I} c_i n_i) \tag{7.4}
\]

and derive:

\[
\frac{\partial L(\bar{n}, \theta)}{\partial \theta} = (M - \sum_{i=1}^{I} c_i n_i) = 0 \tag{7.5}
\]

\[
\frac{\partial L(\bar{n}, \theta)}{\partial n_i} = \frac{1}{\lambda} \lambda_i FR_{i,n_i} - \frac{a_2}{a_1} \lambda_i T_{i,n_i} = 0. \tag{7.6}
\]

Before continuing the analysis, we look at the properties of the two parts of the function.

The first part, the fillrate \((7.1)\) has one inflection point. The second part \((7.2)\), the average waiting time has none. The first part will behave like Figure 7.1 and the second part like Figure 7.2. The fillrate has a convex part and a concave part and the average waiting time has only a concave part. Therefore the sum of both functions will have a concave part and possibly a convex part.

![Fillrate of subsystem i](image)

**Figure 7.1: Example of the behavior of the fillrate**
Figure 7.2: Graph of the second part of the equation

Thus, from (7.6) we have

\[
\frac{\partial^2 L(\tilde{n}, \theta)}{\partial n_i^2} = \frac{1}{\lambda} \lambda_i FR_{i,n_i} - \frac{a_1}{a_1} \frac{1}{\lambda} \lambda_i T_{i,n_i} = 0 ,
\]

which reduces to

\[
a_1 \frac{\lambda_i}{\lambda} \left( \frac{\lambda_i}{\mu} \right)^n e^{-\left( \frac{\lambda_i}{\mu} \right)} \left( \frac{\lambda_i}{n_i + 1} - 1 \right) - a_2 \frac{1}{\lambda} \left( \frac{\lambda_i}{n_i + 1} \right)^{n+1} e^{-\left( \frac{\lambda_i}{n_i + 1} \right)} = 0 .
\]

Thus,

\[
a_1 \frac{\lambda_i}{\lambda} \left( \frac{\lambda_i}{\mu} \right)^n \left( \frac{\lambda_i}{n_i + 1} - 1 \right) = a_2 \frac{1}{\lambda} \left( \frac{\lambda_i}{n_i + 1} \right)^{n+1} \]

\[
\frac{\lambda_i}{n_i!} \left( \frac{\lambda_i}{n_i + 1} - 1 \right) = a_2 \frac{\left( \frac{\lambda_i}{\mu} \right)^{n+1}}{a_1 (n_i + 1)!}
\]
\[
\lambda_i \frac{1}{n_i!} \left( \frac{\lambda_i}{\mu_i} - 1 \right) = a_2 \frac{\left( \frac{\lambda_i}{\mu_i} \right)}{a_1 (n_i + 1)!}
\]
\[
(n_i + 1) \left( \frac{\lambda_i}{n_i + 1} - 1 \right) = a_2 \frac{1}{a_1 \mu_i}
\]
\[
\frac{\lambda_i}{\mu_i} = n_i + 1 + \frac{a_2}{a_1 \mu_i}
\]
\[
n_i = \frac{\lambda_i}{\mu_i} - 1 - \frac{a_2}{a_1 \mu_i}.
\]

If \(a_2=0\), meaning that we are dealing with the original model without Average Waiting Time, we get the inflection point to be at \(n_i = \frac{\lambda_i}{\mu_i} - 1\), which we have already found in section 3.3. If \(a_2\) is large, then the weight for the Average Waiting Time is so big that the fillrate is not important at all, in which case the function is strictly concave. But for which values of \(a_2\) will the function be concave and have no inflection point?

\[
n_i = \frac{\lambda_i}{\mu_i} - 1 - \frac{a_2}{a_1 \mu_i} \leq 0
\]
\[
\lambda_i - \mu_i - \frac{a_2}{a_1} \leq 0
\]
\[
\frac{a_2}{a_1} \geq \lambda_i - \mu_i.
\]

If \(\frac{a_2}{a_1} \geq \lambda_i - \mu_i\), then there will be no inflection point and we can deal with the function as if it were a concave function. In section 3.4.2, we already dealt with this problem. If \(\frac{a_2}{a_1} < \lambda_i - \mu_i\) then there is exactly ONE inflection point and we can deal with the function as if it were similar to the fillrate. The average waiting time does not play a major role except in “moving” the inflection point in the direction of the origin. Then, the algorithm developed in section 5.3 can be applied.
7.2. Missing the goals

Another way to look at multiple goals is to weigh the missing of goals presented by the decision maker. The decision maker presents a value $z_1$ as a goal for the fillrate and $z_2$ for the average waiting time. In fact, we want to minimize the differences of the service levels from the given levels $z_1$ and $z_2$. $a_1$ is the penalty for missing the desired fillrate and is the amount of dollars per percentage of fillrate we missed, and $a_2$ is the penalty of missing the average waiting time and is the amount of dollars per unit time missed. Thus, the objective function will be expressed in terms of dollars which we want to minimize.

$$
Min \left[ z = a_1 \left( z_1 - \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i FR_{i,n_i} \right) + a_2 \left( \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i \overline{T}_{i,n_i} - z_2 \right) \right] 
$$

s.t.:

$$
\sum_{i=1}^{l} c_i n_i \leq M.
$$

The objective function can be lightly transformed, without changing the solution, to

$$
Min(z = -a_1 \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i FR_{i,n_i} + a_2 \left( \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i \overline{T}_{i,n_i} \right))
$$

If we transform the problem into a maximum, we will get

$$
Max(z = a_1 \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i FR_{i,n_i} - a_2 \left( \frac{1}{\lambda} \sum_{i=1}^{l} \lambda_i \overline{T}_{i,n_i} \right))
$$

In fact, although we thought that we have here a new problem, we ended up with exactly the previous problem. In this sense, we solved another problem indirectly.

7.3. Goal programming

This section will deal with several goals where the goals are not weighted but ordered by their priorities. We want to maximize two, three or four service criteria.

1: max(service – criterion1)

2: max(service – criterion2)

s.t.: budget allocation

Let’s analyze this problem using an example. Let’s say we want first to maximize the fillrate and then minimize the average waiting time. But from 3.4.1, we know that
maximizing the fillrate will give us ONE vector $\vec{n}$. Using the techniques of goal programming, we must optimize different systems sequentially. But after the first step, we already have a vector $\vec{n}$, which cannot be changed without changing the values of the service criterion. Thus, due to the fact that the service criteria are linked, we can only solve the following systems:

1: $service - criterion1 \geq SL_1$
2: $service - criterion2 \geq SL_2$  \hspace{1cm} (7.9)
3: $service - criterion3 \geq SL_3$

$s.t.: budget allocation$

which means that the manager wants to achieve the service levels of certain values.

For example, we transform the first inequality into an equality by adding slack and surplus variables, $s_i^-, s_i^+$, respectively.

\[
service - criterion1 + s_i^- - s_i^+ = SL_i.
\]

In our case, because the left-side of the equality is bigger than the right side, $s_i^- = 0$ is desirable. Because, we want the service-criterion1 to be as big as possible, but at least SL1, we transform the problem into a goal programming problem and get:

\[
\min z = -s_i^- \text{ or } \max s_i^- \\
\text{s.t.:} \\
\hspace{1cm} service - criterion1 + s_i^- - s_i^+ = SL_i \\
\hspace{1cm} service - criterion1 + s_2^+ - s_2^- = SL_2 \hspace{1cm} (7.10)
\]

budget allocation

$s_i^+, s_i^- \geq 0$

If $s_i^- = 0$, then it is possible to get a model which satisfies at least this objective function but eventually also others. If $s_i^- > 0$, it is impossible to find a solution which satisfies this objective function and therefore we must change the model by changing the values SL1 or by reducing objective functions. Now, we want to optimize the model by using service-criterion2. Thus, we solve the following system:

\[
\min z = s_2^+ \\
\text{s.t.: budget allocation}
\]
\begin{align*}
\text{service - criterion 1} & \quad s_1^- - s_1^+ = SL_1 \\
\text{service - criterion 2} & \quad s_2^- + s_2^+ = SL_2 \\
\end{align*}
(7.11)

\begin{align*}
& s_1^- = 0 \\
& s_1^-, s_1^+ \geq 0 \\
& s_2^-, s_2^+ \geq 0
\end{align*}

And so on. If we can satisfy all the objectives, we found the solution to the whole model. If not, we need to rethink what should be the value for SL_1. By choosing \( SL_1, SL_2 \) intelligently, we can achieve certain goals on different service-criteria.
8. Integer Programming

8.0. Introduction

In the previous chapter, we often conveniently assumed that an integer variable can be viewed as piece-wise linear, and can therefore be dealt as if it were continuous. But what is the “price” of this assumption? To what extent are we missing the optimal solution? This chapter will deal with these questions. But how can we solve such an integer program?

8.1. The average waiting time

In section 3.4.2, we presented the model wherein we wanted to minimize the average waiting time under a budget constraint. So first, let us define the integer model:

$$\min z = \sum_{i} \frac{\lambda_i T_{i,n_i}}{\lambda}$$

s.t.: $\sum c_i n_i \leq M$

$$n_i \in \mathbb{Z}, i \in \eta.$$

In this case, $n_i$ is an integer variable of the model we want to optimize. But how can we solve it? In section 3.5, we made these variables piece-wise linear to get a solution. Now, we want to use another technique to get a linear model.

We define new variables as follows:

We say $n_{ij} = 1$, if $n_i=j$, and $n_{ij} = 0$ otherwise. Clearly $\sum_{j=0}^{\infty} n_{ij} = 1, i \in \eta$, so that for each $I$ one and only one of the $n_{ij}$ is 1. We define $AW(n_{ij}) = \frac{\lambda_i T_{i,j}}{\lambda}$, where $j$ is the number of spares to be bought. Hence, we can now write the objective function as

$$\min z = \sum_{i=1}^{I} \sum_{j=0}^{M} AW(n_{ij}) n_{ij},$$

and model (8.1) can be reconfigured as follows:
\[
\min z = \sum_{i=1}^{I} \sum_{j=0}^\infty AW(n_{ij}) \cdot n_{ij}
\]  
(8.2)

s.t.: 
\[
\sum_{j=0}^\infty n_{ij} = 1, \text{ for all } i \in \eta,
\]
\[
\sum_{i=1}^{I} \sum_{j=0}^\infty c_{ij} n_{ij} \cdot j \leq M,
\]

where \( n_{ij} = 0,1 \). \quad j=0,1,\ldots,\infty.

Table 8.1 shows an example of the variables \( n_{ij} \). Every column describes the item classes (from 1,\ldots,13) and every row the number of items bought for that class. For example, \( n_{42}=1 \) which means that for class 4, we will buy 2 items.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.1: Example of the Spreadsheet

We then calculate AW(\( n_{ij} \)) for \( i=1,\ldots,13, j=0,\ldots,\infty \). Table 8.2 shows AW(\( n_{ij} \)) which will be used to calculate the objective function.
The solution yields the following vector which is highlighted.

<table>
<thead>
<tr>
<th>class</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr of items</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1.333353902</td>
<td>0.801348</td>
<td>1.800009</td>
<td>0.316417</td>
<td>1.033458</td>
<td>1.800009</td>
<td>1.3333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.16times032</td>
<td>0.69433</td>
<td>1.600109</td>
<td>0.173876</td>
<td>0.967811</td>
<td>1.600109</td>
<td>1.1688</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00126496</td>
<td>0.43436</td>
<td>1.400663</td>
<td>0.082639</td>
<td>0.70539</td>
<td>1.400663</td>
<td>1.0012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.838139032</td>
<td>0.087369</td>
<td>1.20273</td>
<td>0.034154</td>
<td>0.55671</td>
<td>1.20273</td>
<td>0.8381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.68029972</td>
<td>0.175467</td>
<td>1.008681</td>
<td>0.01239</td>
<td>0.409962</td>
<td>1.008681</td>
<td>0.6802</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.633240559</td>
<td>0.09866</td>
<td>0.821998</td>
<td>0.003986</td>
<td>0.289279</td>
<td>0.821998</td>
<td>0.6332</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.401382606</td>
<td>0.051096</td>
<td>0.648026</td>
<td>0.001148</td>
<td>0.192671</td>
<td>0.648026</td>
<td>0.4014</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.203560285</td>
<td>0.02422</td>
<td>0.49207</td>
<td>0.000299</td>
<td>0.120626</td>
<td>0.49207</td>
<td>0.2036</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.197633456</td>
<td>0.010803</td>
<td>0.358634</td>
<td>7.09E-05</td>
<td>0.071273</td>
<td>0.358634</td>
<td>0.1976</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.128868167</td>
<td>0.04438</td>
<td>0.25022</td>
<td>1.54E-05</td>
<td>0.039548</td>
<td>0.25022</td>
<td>0.1288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.07996622</td>
<td>0.001698</td>
<td>0.168828</td>
<td>3.1E-06</td>
<td>0.020661</td>
<td>0.168828</td>
<td>0.0799</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.047034388</td>
<td>0.006088</td>
<td>0.106183</td>
<td>5.77E-07</td>
<td>0.010176</td>
<td>0.106183</td>
<td>0.0470</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.026329865</td>
<td>0.000204</td>
<td>0.064495</td>
<td>1E-07</td>
<td>0.004734</td>
<td>0.064495</td>
<td>0.0263</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.014201395</td>
<td>6.45E-05</td>
<td>0.03787</td>
<td>1.63E-08</td>
<td>0.002084</td>
<td>0.03787</td>
<td>0.0142</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.00711034</td>
<td>1.92E-05</td>
<td>0.020696</td>
<td>2.5E-09</td>
<td>0.000869</td>
<td>0.020696</td>
<td>0.0071</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.00343773</td>
<td>5.43E-06</td>
<td>0.010948</td>
<td>3.61E-10</td>
<td>0.000345</td>
<td>0.010948</td>
<td>0.0034</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.2: Table of the average waiting time for each class i and for each value j.

To compare, we present the result of the linear problem from section 3.8.

<table>
<thead>
<tr>
<th>class</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>nr of items</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1.333353902</td>
<td>0.801348</td>
<td>1.800009</td>
<td>0.316417</td>
<td>1.033458</td>
<td>1.800009</td>
<td>1.3333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.16times032</td>
<td>0.69433</td>
<td>1.600109</td>
<td>0.173876</td>
<td>0.967811</td>
<td>1.600109</td>
<td>1.1688</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00126496</td>
<td>0.43436</td>
<td>1.400663</td>
<td>0.082639</td>
<td>0.70539</td>
<td>1.400663</td>
<td>1.0012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.838139032</td>
<td>0.087369</td>
<td>1.20273</td>
<td>0.034154</td>
<td>0.55671</td>
<td>1.20273</td>
<td>0.8381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.68029972</td>
<td>0.175467</td>
<td>1.008681</td>
<td>0.01239</td>
<td>0.409962</td>
<td>1.008681</td>
<td>0.6802</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.633240559</td>
<td>0.09866</td>
<td>0.821998</td>
<td>0.003986</td>
<td>0.289279</td>
<td>0.821998</td>
<td>0.6332</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.401382606</td>
<td>0.051096</td>
<td>0.648026</td>
<td>0.001148</td>
<td>0.192671</td>
<td>0.648026</td>
<td>0.4014</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.203560285</td>
<td>0.02422</td>
<td>0.49207</td>
<td>0.000299</td>
<td>0.120626</td>
<td>0.49207</td>
<td>0.2036</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.197633456</td>
<td>0.010803</td>
<td>0.358634</td>
<td>7.09E-05</td>
<td>0.071273</td>
<td>0.358634</td>
<td>0.1976</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.128868167</td>
<td>0.04438</td>
<td>0.25022</td>
<td>1.54E-05</td>
<td>0.039548</td>
<td>0.25022</td>
<td>0.1288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.07996622</td>
<td>0.001698</td>
<td>0.168828</td>
<td>3.1E-06</td>
<td>0.020661</td>
<td>0.168828</td>
<td>0.0799</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.047034388</td>
<td>0.006088</td>
<td>0.106183</td>
<td>5.77E-07</td>
<td>0.010176</td>
<td>0.106183</td>
<td>0.0470</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.026329865</td>
<td>0.000204</td>
<td>0.064495</td>
<td>1E-07</td>
<td>0.004734</td>
<td>0.064495</td>
<td>0.0263</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.014201395</td>
<td>6.45E-05</td>
<td>0.03787</td>
<td>1.63E-08</td>
<td>0.002084</td>
<td>0.03787</td>
<td>0.0142</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.00711034</td>
<td>1.92E-05</td>
<td>0.020696</td>
<td>2.5E-09</td>
<td>0.000869</td>
<td>0.020696</td>
<td>0.0071</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.00343773</td>
<td>5.43E-06</td>
<td>0.010948</td>
<td>3.61E-10</td>
<td>0.000345</td>
<td>0.010948</td>
<td>0.0034</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.1: Optimal spare vector $\mathbf{n}$ for the Integer Problem

Figure 8.2: Optimal vector $\mathbf{n}$ for the Linear Problem

From Figure 8.2, we obtain the optimal vector for the linear problem, and then compare it with the IP results in Figure 8.1. The spreadsheet approach not only validates the linear solution, but gives us also the possibility to solve problems which could not be analyzed until today, such as the probability of waiting more than a specified time.
In both cases, the input was 1000 dollars. We can see that there are no essential differences between the results. Due to restrictions of our software, it is impractical to solve large size problems. Thus, whenever smaller problem have to be solved, we can do so via spreadsheets, but for larger problems, we must develop other techniques. In conclusion, we see that we can choose linear programming instead of integer programming. Although, this is not a full proof, it certainly seems plausible.

8.2. The fillrate

The same idea of integer programming was applied to the fillrate. In section 3.4.1, we calculated the fillrate which leads to the following table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.1021093939</td>
<td>0.0333333333</td>
<td>0.4018424242</td>
<td>0.0000000000</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td>0.4040404040</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.6496666667</td>
<td>0.2020202020</td>
<td>1.4364364364</td>
<td>0.0000000000</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td>1.4364364364</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.4261904762</td>
<td>0.6226226226</td>
<td>2.7190476190</td>
<td>0.0138888889</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td>2.7190476190</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>5.9895043818</td>
<td>3.9545454545</td>
<td>5.5252525253</td>
<td>0.5000000000</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td>5.5252525253</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.3: Values of the fillrates for each ss_i used in the integer programming.

\[
\begin{align*}
\max \quad & z = \sum_{i=1}^{I} \sum_{j=0}^{\infty} FR(n_j) n_j \\
\text{s.t.:} & \sum_{j=0}^{\infty} n_j = 1, \text{ for all } i \in \eta, \\
& \sum_{i=1}^{I} \sum_{j=0}^{\infty} c_i n_j j \leq M, \\
\end{align*}
\]  

(8.3)

where \( n_{ij} = 0,1 \).

The optimal spares vector is found after solving the problem by setting \( n_i = \sum_{j=0}^{\infty} j^* n_{ij} \).

It was interesting to see that although the fillrate has both convex and concave parts, the integer program finds the optimal solution directly.
Figure 8.3: Optimal spares vector for the fillrate using integer programming

Figure 8.3 depicts an example of the given system. As mentioned, although there are both convex and concave parts, which means that there are multiple local maxima, the Spreadsheet Solver found the optimal solution. In this case, the linear program already presented in chapter 3 found the same solution but with small differences.

<table>
<thead>
<tr>
<th>class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>repair time rate</td>
<td>1.5</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>1.2</td>
<td>2</td>
<td>1.5</td>
<td>1</td>
<td>2</td>
<td>0.5</td>
<td>1.2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>arrival rate</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>cost</td>
<td>6</td>
<td>8</td>
<td>20</td>
<td>10</td>
<td>8</td>
<td>20</td>
<td>6</td>
<td>8</td>
<td>20</td>
<td>10</td>
<td>8</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>optimal solution</td>
<td>8</td>
<td>9</td>
<td>5</td>
<td>13</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>4</td>
<td>0</td>
<td>9</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>optimal fillrate</td>
<td>0.8353</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.4: Optimal spares vector for the fillrate using linear programming.

Figure 8.5: Graph of the overall average waiting time
Until now, we had almost no idea how the overall average waiting time or the overall fillrate behaves when increasing or decreasing the budget. We decided to analyze this further. Thus, we solved the above models for several budget allocations yielding the following graphs, Figure 8.5 and Figure 8.6. The graph of Figure 8.5 is what we expected. The average waiting time of each \( ss_i \) behaves somewhat exponentially. This explains why we therefore expected the overall average waiting time also to look hyperexponential, Figure 8.6.

![Graph of the overall fillrate](image)

**Figure 8.6: Graph of the overall fillrate**

Figure 8.6 shows the curve of the overall fillrate of the system. At the beginning, we expected inflection points, but after analyzing the behavior of the system, we came to the conclusion that the variety of different \( ss_i \) affects the overall fillrate so that it does not contain any inflection point. From this view, it looks a smooth curve, but when choosing smaller intervals (see Figure 8.7), the fact that the variables are integer causes the graph not to be smooth. Also, the fact that there are convex and concave parts also affects the graph in this way.
8.3. The probability of waiting time $x$

In Chapter 4, we discussed the model where the service-level objective function is to minimize the probability of waiting more than $x$. Here,

$$\min z = \sum_{i=1}^{I} \sum_{j=0}^{\infty} P(W(n_{ij}) > x) \cdot n_{ij}$$

s.t.:

$$\sum_{j=0}^{\infty} n_{ij} = 1, \text{ for all } i \in \eta,$$

$$\sum_{i=1}^{I} \sum_{j=0}^{\infty} c_{ij} n_{ij} \cdot j \leq M,$$

where $n_{ij} = 0, 1$.

Until now, we had no technique to solve this system. But, using IP, we can solve the model. In all the models, we used Visual Basic to calculate the formulas for this model using the spreadsheet solver. First, we calculated $P(W_n = 0)$. Of course, this is the fillrate and obviously we got the same result as shown in Figure 8.8. The Spreadsheet

Figure 8.7: Graph of the overall fillrate with smaller intervals.
presented can be changed for any \( x \) to find the optimal vector of waiting \( x \). We chose \( x=0 \) to validate the solution. When \( x=0 \), in fact, we deal with the fillrate, which we solved in section 8.2.

![Spreadsheet with data](image)

**Figure 8.8:** Optimal spares vector for the probability of no waiting

In the next section, we use the Spreadsheet to run different examples to get some idea of the shapes of the functions. The shapes are taken from a random example.

### 8.3.1. Sensitivity analysis

In this section, we want to analyze the sensitivity of the model: For this purpose, we chose several budget amounts, and ran the model with different \( x \). We present the outputs in the following figures. Figure 8.9 show the influence of budget allocation on the waiting time distribution. On the one hand, when no budget is available, the waiting time distribution depends strongly on the repair time distribution. On the other hand, the bigger the budget, the higher the probability of not waiting.
Figure 8.9: The probability of waiting less than $x$ with different budget allocation.

Figure 8.10: Waiting Time Distribution based on different budget allocation.
As a second step, we wanted to explore the influence of budget allocation on the waiting time distribution. Obviously, the more money we invest the less we will wait. The bigger the length of time we agree to wait (0.8), the higher is the probability to be satisfied within that interval.

8.4. The probability of waiting more than x for waiting customers

One of the most important service criteria is the probability of waiting more than x under the condition that the customer waits. In a service facility, it can happen that only 20% must wait at all, but they inevitably have to wait very long. This is not desirable for a manager, who seeks to optimize his system performance. He wants the customers who must wait to wait as short as possible. Thus,

\[
\min z = \sum_{i=1}^{L} \sum_{j=0}^{\infty} P(W > x \mid W > 0) \cdot n_{ij}
\]

(8.5)

s.t.:

\[
\sum_{j=0}^{\infty} n_{ij} = 1, \text{ for all } i \in \eta,
\]

\[
\sum_{j=0}^{\infty} c_{ij} \cdot n_{ij} \leq M,
\]

where \( n_{ij} = 0,1 \) and where

\[
P(W > x \mid W > 0) = \frac{1}{\lambda_i} \sum_{i=1}^{L} \frac{1 - F_i(x)}{1 - F_i(0)}.
\]

This is not a linear function at all and certainly a most complicated function to optimize. But with our method of integer programming, it becomes very simple. The following graphs show an example with sensitivity analysis.
Figure 8.11: Output of the probability of waiting more than x for waiting customers.

Figure 8.11 shows the output of the probability that the customer waits more than 0.5 under the condition that he must wait, when 1000 dollars are available for spares. Figure 8.12 depicts the waiting time distribution for waiting customer, whereas Figure 8.13 the probability of waiting more than 0.3 for different budget amounts.

![Graph showing the conditioned probability of waiting more than x.](image-url)
In any case, the introduction of integer programming opened up a new method of solving problems. We explored a new service level, the probability of waiting more than a certain time $x$. In fact, all published articles about spares deal exclusively with the average waiting time or with the fillrate, but none with the waiting time distribution. This part represents an essential contribution to the research in this field. An additional model to be solved by integer programming includes discounting. When buying multiple spares there are often discounts which influence the decisions of a manager. In many models this assumption cannot be introduced, but the approach of integer programming enables us to even solve models including discounting.
9. **Bulk systems**

9.0. **Introduction**

In the preceding chapters, we dealt with systems where a customer brings a product which failed in a single class. Due to the fact that every item class contains exactly one item, we could easily develop formulas and methods to solve the model for this system. But what happens if the product contains, say two, of the same class and both fail. How does that influence modeling of the system? Hausman and Cheung [16] presented a customer-based approach, but failed to develop correct formulas for the number in system. In this chapter, we will develop an item-based approach, and find correct solutions for the waiting time distribution and for the number of customers in system. Berg and Posner [5] briefly dealt with the idea of bulk arrival systems where a customer may bring more than one item. They provided a solution for a transient model where the number of items brought is also a random variable. In section 9.1, we first discuss an ERSOF-model where every customer brings a product which failed in exactly two items of the same class. Recall that an ERSOF-model is a model involving only 1 class failure at the same time. In 9.2, we generalize the model to where every customer brings a product which failed in exactly $b$ items of the same class, where $b$ is a constant. Finally, in 9.3, we generalize the model to deal more fully with random bulk arrivals.

9.1. **An ERSOF - system where every customer brings exactly 2 failed items of the same class**

A customer arrives at time $t$ with 2 items of class $i$ requiring repair and waits until he gets full satisfaction and leaves the system. $W_i$ is the waiting time of the customer who arrives at time $t$. $\{W_i < x\}$ is the event that a tagged customer who arrives at time $t$ waits less than $x$. This customer who arrives at time $t$ will wait less than $x$ if by time $t+x$, all the customers who had brought 2 items of class $i$ before him have received satisfaction and at least 2 items are left. Thus,
\[ \{ W_{t,x} \leq x \} \iff \{ \text{by time } t+x, \text{ our tagged customer reaches the front of the queue and there are at least 2 items on the shelf from class } i \} \].

To serve all customers in front of our tagged customer as well as the tagged customer himself, we need the total number of items arriving at the shelf (from spares or from repair) during (0,t+x) to exceed the number of items required to satisfy all the customers who arrived in time (0,t), including our tagged customer. Thus,

\[ \{ W_{t,x} \leq x \} \iff \{ \text{no. items arrived on shelf in } (0,t+x) \geq 2 + \text{no. items from customers who arrived in } (0,t) \} \.

We will define the following random variables:

- \( Z(t,x) \) is the number of items repaired by time t+x from our tagged customer who arrived at t. Clearly, \( Z(t,x) \) is \( \text{bin}[n = 2, p = G(x)] \). Thus,
  
  \[
  P(Z(t,x) = 0) = [1 - G(x)]^2 \\
  P(Z(t,x) = 1) = 2G(x)[1 - G(x)] \\
  P(Z(t,x) = 2) = G(x)^2.
  \]

The number of items arrived to the shelf is equal to the number of spares plus the number of items repaired by time t+x. Let

\[ S(t+x) = \text{the number of items repaired in } (0,t+x) \text{ from all customers apart from our tagged customer}. \]

Then,

\[ \{ W_{t,x} \leq x \} \iff \{ \text{Number items arriving to the shelf in } (0,t+x) \geq 2 + \text{number items brought by other customers in } (0,t) \} \iff \{ n + S(t+x) + Z(t,x) \geq 2 + 2N(t) \} \] \hspace{1cm} (9.1)

In order to determine the distribution of \( S(t+x) \), we first define the following random variables:

- \( N_1(t,x) \) is the number of items which arrived before time t, and complete repair after time t+x.
- \( N_2(t,x) \) is the number of items which arrived after time t, and complete repair before time t+x.
- \( S(t+x) \) : The number of items repaired in time \( (0,t+x) \) is the sum of the number of items repaired from customers who came before time t, plus those from the customers who arrived after time t. Therefore,
\[ S(t + x) = (2N(t) - N_1(t, x)) + N_2(t, x). \]  

(9.2)

Let’s look at \( N_1(t, x) \).

![Graph](image)

**Figure 9.1:** Customer arriving in \((0,t)\) and leaving after \(t+x\)

The probability of an item arriving in \(du\) to be repaired by time \(t+x\) is \( G(t+x-u) \). Thus, the probability of an item arriving in \(du\) completing repair after \(t+x\) is \( G(t+x-u) \). The probability of arrival in \(du\) is \( du/t \). Thus, the probability of one item of a random customer arriving before \(t\) and having his item repaired after \(t+x\) is \( \int_0^t \frac{1}{t} G(t + x - u) du \), which can be written as

\[ p_1(x,t) = \int_x^{x+t} \frac{1}{t} G(v) dv, \]  

(9.3)

with \( q_1(t,x) = 1 - p_1(t,x) \).

\( X_k(t) \) is the number of items of customer \(k\) (unordered), who arrived in \((0,t)\), which are not repaired by time \(t+x\). \( X_k(t) \in \{0,1,2\} \). Thus, the number of items repaired after \(t+x\), which arrived before \(t\) is the sum of \(N(t)\) identical random variables \(X_k(t)\), where every \(X_k(t)\sim\text{binomial}(2,p_1(t,x))\) and \(N(t) \sim \text{Poisson}(\lambda t)\). Thus,

\[ N_1(t,x) = X_1 + \ldots + X_{N(t)} = \sum_{k=1}^{N(t)} X_k(t). \]

We will use generating functions to determine the distribution of \( N_1(t,x) \).

The generating function of a binomial \((2,p_1)\) variable is \((p_1 z + q_1)^2\) and of the Poisson is \(e^{\lambda(t-z)}\). Thus, the generating function of \( N_1(t,x) \) is
\[ E\left\{ N_1(t, x) \right\} = E\left[ E\left[ z^{X_1 + \ldots + X_{N(t)}} \mid N(t) \right] \right] = E\left[ E\left[ z^{X_i} \right]^{N(t)} \right] \]

But what happens when \( t \) goes to infinity?

We have \( p_1(t, x) = \int_x^{\infty} \frac{1}{t} G(v) dv \). Clearly,

\[ \lim_{t \to \infty} p_1(t, x) = 0. \text{ But } p_1 = \lim_{t \to \infty} p_1(t, x) = \lim_{t \to \infty} \int_x^{\infty} G(v) dv = \int_x^{\infty} G(v) dv. \quad (9.4) \]

Thus,

\[ \lim_{t \to \infty} e^{\lambda \left[ p_1(z + q_1) \right]^{t-1}} = \lim_{t \to \infty} e^{2 \lambda p_1 (1-z) \left[ p_1(1-z) - 2 \right]} = e^{\left[ \int_x^{\infty} G(v) dv \right] (1-z) [1-2]} = e. \quad (9.5) \]

But this is the generating function of a Poisson variable with parameter \( 2 \lambda p_1 \). Thus, when \( t \) approaches \( \infty \), \( N_1(x) \) is Poisson with parameter \( 2 \lambda p_1 \).

That is,

\[ P(N_1(x) = r) = \frac{(2 \lambda p_1 x)^r}{k!} e^{-2 \lambda p_1 x}, \quad r = 0, 1, 2, \ldots \]

Thus,

\[ E\left[ \left. E\left[ z^{X_i} \right]^{N(t)} \right\} \right] = E\left\{ p_1 z + q_1 \right\}^{2N(t)} = e^{2 \lambda \left[ p_1 z + q_1 \right]^{t-1}} \] \quad (9.6)
Now consider $N_2(t,x)$:

![Figure 9.2: Customer arriving in (t,t+x) and having his item repaired before t+x.](image)

Let $Y_k$ be the number of items of customer $k$ (unordered) that we brought in $(t,t+x)$, and which are then repaired by time $t+x$. $Y_k \in \{0,1,2\}$. The probability of an item that arrived in $du = (t,t+x)$ is repaired by time $t+x$ is $G(t+x-u)$. Thus, the probability that any one item from random customer $i$ arriving after $t$ and then having his item repaired in $(t,t+x)$ is

$$
\int_t^{t+x} \frac{1}{x} G(t+x-u) du,
$$

which can be written as

$$p_2 \equiv p_2(x) = \int_0^x \frac{1}{x} G(v) dv, \quad (9.7)
$$

with $q_2(x) = 1 - p_2(x)$.

Thus, $N_2(t,x)$ is the sum of $N(x)$ random variable $Y_k$, where every $Y_k \sim bin(2, p_2)$ and $N(x) \sim Poisson(\lambda x)$, so that

$$N_2(t,x) = Y_1 + Y_2 + \ldots + Y_{N(x)} = \sum_{k=1}^{N(x)} Y_k$$

Due to the fact that the distribution of the number of $k$ arriving customers in $(t,t+x)$ is the same as in $(0,x)$ for a homogenous Poisson Process, we can redefine $N_2(t,x)$ as equivalent to $N_2(x)$, and
\[ P(N_2(t, x) = r) = P(N_2(x) = r) = \sum_{k=0}^{\infty} \binom{2k}{r} p_2^r q_2^{2k-r} \frac{(\lambda x)^k}{k!} e^{-\lambda x}, \quad r = 0, 1, \ldots, \infty. \]

Now, returning to our original problem, we have the equivalence
\[ \{W_i < x\} \leftrightarrow \{2N(t) - S_1(t + x) - Z(t, x) \leq n - 2\} = \{N_i(t, x) - N_2(t, x) \leq n + Z(t, x) - 2\} \]

For simplicity, we introduce new random variables
\[ D(t, x) = N_i(t, x) - N_2(t, x) \]
\[ D(x) = N_i(x) - N_2(x) \]

Proposition:

In an exchangeable-item FIFO system, the nonstationary delay distribution of a customer that arrives at time \( t \) with exactly two items of class \( i \) is given by

\[
F_{x,t}(x) = P(W_{i,t} \leq x) = P(D(t, x) \leq n_i - 2) + P(D(t, x) = n_i - 1) \left[ 2G(x) - G(x)^2 \right] + P(D(t, x) = n_i) \left[ G(x)^2 \right] \quad \text{(9.8)}
\]

Thus, the delay distribution of a customer that arrives at time \( t \) with exactly two items is given by

\[
F_{x,t}(x) = P(W_t \leq x) = \sum_{i=1}^{I} \frac{\lambda_i}{\lambda} P(W_{i,t} \leq x) = \sum_{i=1}^{I} \frac{\lambda_i}{\lambda} \left[ P(D(t, x) \leq n_i - 2) + P(D(t, x) = n_i - 1) \left[ 2G(x) - G(x)^2 \right] + P(D(t, x) = n_i) \left[ G(x)^2 \right] \right]^1 \quad \text{(9.9)}
\]

\[
\sum_{r=0}^{\infty} \sum_{k=0}^{\infty} \binom{2k}{r} p_2^r q_2^{2k-r} \frac{(\lambda x)^k}{k!} e^{-\lambda x} = \sum_{k=0}^{2k} \binom{2k}{r} p_2^r q_2^{2k-r} \frac{(\lambda x)^k}{k!} e^{-\lambda x}
\]

\[
= \sum_{k=0}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x} \sum_{r=0}^{2k} \binom{2k}{r} p_2^r q_2^{2k-r} = \sum_{k=0}^{\infty} \frac{(\lambda x)^k}{k!} e^{-\lambda x} = 1
\]
The stationary delay distribution of a customer is

\[
F(x) = P(W \leq x) = \sum_{i=1}^{l} \frac{1}{\lambda_i} \left[ P(D(x) \leq n_i - 2) + P(D(x) = n_i - 1) \left[ 2G(x) - G(x)^2 \right] \right] + P(D(x) = n_i) \left[ G(x)^2 \right] \]  

(9.10)

where

\[
P(D(x) = r) = P(N_1(x) - N_2(x) = r) = \sum_{k=-\infty}^{\infty} P(N_2(x) = k) P(N_1(x) = k + r), -\infty \leq r \leq \infty.
\]

(For the proof see Appendix E.)

Obviously, we are not interested in such a restricted bulk system, in which customers brings precisely two items of the same class, but we want a more general model, wherein a customer brings \( b \) items of the same class. In section 9.2, we will analyze a system where customer brings exactly \( b \) items and in section 9.3, we generalize this to a system where the customer brings a random number of items.

### 9.2. An ERSOF-system where each customer brings exactly \( b \) failed items of the same class which failed

After having analyzed the 2 case, we now generalize to a bulk size of \( b \). A customer arrives at time \( t \) and waits until he gets satisfaction and leaves the system with all failed items replaced. \( W_i \) is the waiting time of the customer who arrives at time \( t \). \( \{W_i < x\} \) is the event that a tagged customer who arrives at time \( t \) waits less than \( x \). We will assume that the customer brought \( b \) items of class \( i \), so that we can deal with this problem separately. At the end, we will generalize the model so that the customer can bring \( b \) of any class.

A customer who arrives at time \( t \) will wait less than \( x \) if by time \( t+x \), all customers who arrived before him have gotten satisfaction and at least \( b \) items are left.  

\[
\{W_{t,i} \leq x\} \iff \{ \text{by time } t+x, \text{ our tagged customer reaches the front of the queue and there at least } b \text{ items on the shelf of class } i \}.
\]

To serve all the customers in front of our tagged customer as well our tagged customer, we need the total number of items arriving at the shelf (from spares or from repair) during
(0,t+x) to exceed the number of items required to satisfy all the customers who arrived in time (0,t) as well as our tagged customer. Thus,

\[ W_{t,x} \leq x \iff \{ \text{Number items arrived on shelf in } (0,t+x) \geq b + \text{Number items for customers who arrived in } (0,t) \} \]

We will define the following random variables:

\( Z(t,x) \) is the number of items repaired by time t+x from our tagged customer who arrived at t. Thus,

\[ P(Z(t,x) = k) = \binom{b}{k} G(x)[1 - G(x)]^{b-k}, \quad k = 0, 1, 2, \ldots, b. \]

The number of items that arrived to the shelf is equal to the number of spares plus the number of items repaired by time t+x.

\[ S(t+x) = \text{the number of repaired items by time t+x from all other customers}. \]

\[ \{ \text{items arrived on shelf in } (0,t+x) \geq b + \text{items for customer in time } (0,t) \} = \{ n_t + S(t+x) + Z(t,x) \geq b + bN(t) \}. \]

Hence

\[ \{ W_{t,x} \leq x \} = \{ n_t + S(t+x) + Z(t,x) \geq b + bN(t) \}. \]

In order to determine \( S(t+x) \), we first define the following random variables:

\( N_1(t,x) \) is the number of items which arrived before time t and complete repair after time t+x, and

\( N_2(t,x) \) is the number of items which arrived after time t and are complete repair before time t+x.

The number of items repaired by time t+x is the sum of the number of items repaired from customers who came before time t, plus those from customers who arrived after time t:

\[ S(t+x) = (bN(t) - N_1(t,x)) + N_2(t,x) \]

Let’s first look at \( N_1(t,x) \). From the previous section, we already know that

\[ N_1(t,x) = X_1 + \ldots + X_{N(t)} \]

and thus

\[ X_i \sim \text{binomial}(bN(t), p_1) \]

\[ N(t) \sim \text{Poisson}(\lambda t) \]
p_1 was already calculated in (9.4).

We will use generating function to calculate N_1(t,x). The generating function of a binomial(b,p_1) variable is (p_1z + q_1)^b and of the Poisson e^{\lambda t(z-1)}. Thus, the generating function of N_1(t,x) is

\[ E\left\{ z^{N_1(t,x)} \right\} = E\left\{ z^{X_1 + \ldots + X_{N(t)}} \mid N(t) \right\} \]
\[ = E\left\{ z^{X_{N(t)}} \right\} \]
\[ = E\left\{ p_1z + q_1 \right\}^{bN(t)} \]
\[ = e^{\lambda t((p_1z + q_1)^b - 1)} \]

But what happens when t goes to infinity?

From the previous section, we know that \( \lim_{t \to \infty} p_1(t,x) = 0 \), and \( \lim_{t \to \infty} p_1t = \int_G(v)dv \).

Now,

\[ \lim_{t \to \infty} [\lambda t((p_1z + q_1)^b - 1)] \]
\[ = \lim_{t \to \infty} [\lambda t((p_1(z-1) + 1)^b - 1)] \]
\[ = \lim_{t \to \infty} \left[ \lambda t\left( \sum_{k=0}^{b} \binom{b}{k} [p_1(z-1)]^k \right) - 1 \right] \]
\[ = \lim_{t \to \infty} \left[ \lambda t\left( \sum_{k=1}^{b} \binom{b}{k} [p_1(z-1)]^k \right) \right] \]
\[ = \left[ \sum_{k=1}^{b} \binom{b}{k} \lim_{t \to \infty} \lambda tp_1^k(z-1)^k \right] \]

But \( \lim_{t \to \infty} (p_1(t,x))^i = \begin{cases} \int_G(v)dv & i = 1 \\ 0 & i \neq 1 \end{cases} \)

since \( \lim_{t \to \infty} p_1(t,x) = 0 \).

Hence, we get

\[ E\left\{ z^{N_1(x)} \right\} = \left[ \sum_{k=1}^{b} \binom{b}{k} \lim_{t \to \infty} \lambda tp_1^k(z-1)^k \right] = b \left[ \lambda \int_G(v)dv(z-1) \right] \]

Thus, N_1(x) is distributed Poisson with parameter \( \lambda b \int_G(v)dv \), so that
\[ P(N_1(x) = k) = \frac{(bp_1\lambda x)^k}{k!} e^{-bp_1\lambda x}, \quad k=0,1,\ldots,\infty, \text{ since } \int_0^\infty G(v)dv = p_1x \]

Now, let us look at \( N_2(t,x) \).
\( N_2(t,x) \) is the number of items arriving after \( t \) and repaired before \( t+x \). \( V_k \) be the number of items repaired before \( t+x \) from some customer \( k \) (unordered). In the interval \((t,t+x)\) there are \( N(x) \) customers arriving. Thus,
\[ N_2(t,x) = V_1 + \ldots + V_{N(x)} \]

We will use generating functions to determine the distribution of \( N_2(t,x) \).

But what happens when \( t \) goes to infinite?

We know that \( \lim_{t \to \infty} p_1(t,x) = 0 \), and \( \lim_{t \to \infty} p_1 t = \int_0^x G(v)dv \).

Now,
\[ \lim_{t \to \infty} \lambda t \left( (p_1(z) + q_1)^b - 1 \right) \]
\[ = \lim_{t \to \infty} \lambda t \left( (p_1(z-1) + 1)^b - 1 \right) \]
\[ = \lim_{t \to \infty} \left[ \lambda t \left( \sum_{k=0}^b \binom{b}{k} [p_1(z-1)]^k \right) - 1 \right] \]
\[ = \lim_{t \to \infty} \left[ \lambda t \left( \sum_{k=1}^b \binom{b}{k} [p_1(z-1)]^k \right) \right] \]
\[ = \left[ \sum_{k=1}^b \binom{b}{k} \lim_{t \to \infty} \lambda t p_1^k (z-1)^k \right] \]

But \( \lim_{t \to \infty} t (p_1(t,x)) = \begin{cases} \int_0^x G(v)dv & i = 1 \\ 0 & i \neq 1 \end{cases} \)

since \( \lim_{t \to \infty} p_1(t,x) = 0 \).

Hence, we get
Thus, $N_2(x)$ is distributed Poisson with parameter $\lambda b \int_0^z G(v)dv$, so that

$$P(N_2(x) = k) = \frac{(bp_x^k)^k}{k!} e^{-bp_x^k}, \ k=0,1,\ldots,\infty.$$ 

Now, let us develop the nonstationary and then the stationary distributions.

$$\{W_t \leq x\} = \{bN(t) - S_1(t + x) - Z(t, x) \leq n - b\}$$

$$=\{bN(t) - [bN(t) - N_1(t, x)] + N_2(t, x)] - Z(t, x) \leq n - b\}$$

$$=\{N_1(t, x) - N_2(t, x) \leq n + Z(t, x) - b\}$$

$$F_t(x) = P(W_t \leq x)$$

$$= \sum_{k=0}^b P(Z(t, x) = k)P(N_1(t, x) - N_2(t, x) \leq n - (b - k))$$

$$= \sum_{k=0}^b \binom{b}{k} G(x)^k [1 - G(x)]^{b-k} P(N_1(t, x) - N_2(t, x) \leq n - (b - k))$$

This is the waiting time distribution for any system.
Proposition:

In an exchangeable-item FIFO system with I classes, the nonstationary delay distribution of a customer that arrives at time \( t \) with \( b \) items of the same class is given by

\[
F_i(x) = P(W_i \leq x) = \sum_{i=1}^{I} \frac{\lambda_i}{\bar{\lambda}_i} \sum_{k=0}^{b} \left( \begin{array}{c} b \\ k \end{array} \right) G_i(x)^k \left[ 1 - G_i(x) \right]^{b-k} P(D_i(t,x) \leq n-(b-k))
\]

(9.11)

where \( D_i(t,x) = N_{i,2}(t,x) - N_{i,2}(t,x) \)

The stationary waiting time distribution of a random customer is

\[
F(x) = P(W \leq x) = \sum_{i=1}^{I} \frac{\lambda_i}{\bar{\lambda}_i} \sum_{k=0}^{b} \left( \begin{array}{c} b \\ k \end{array} \right) G_i(x)^k \left[ 1 - G_i(x) \right]^{b-k} P(D_i(x) \leq n-(b-k))
\]

(9.12)

In this section, we analyzed a system where every customer brings exactly \( b \) items. In the next section, we will generalize this model, so that a customer can bring a random number of items from the same class.

9.3. An ERSOF-system where each customer brings a random number \( B \) of failed items of the same class.

We now generalize to a bulk size of \( B \) where \( B \) is a random variable. Customer \( k \) (unordered) arrives at time \( t \) with \( B_k \) items requiring repair and waits until he gets satisfaction and leaves the system. \( W_i \) is the waiting time of the customer who arrives at time \( t \). \( \{W_i < x\} \) is the event that this tagged customer who arrives at time \( t \) waits less than \( x \). We will assume that customer \( k \) brought \( B_k \) items of class \( i \), so that we can deal with this problem separately. At the end, we will generalize to customers bringing \( B_k \) of any class. This customer who arrives at time \( t \) will wait less than \( x \) if by time \( t+x \), all the customers in front of him have gotten satisfaction and at least \( B_k \) items are left to service his demand for \( B_k \) items. The product contains \( Q \) items of the class we are dealing with and therefore, \( B_k \in \{1,...,Q\} \).
\[ \{ W_t < x \} = \{ \text{by time } t+x, \text{ our tagged customer reaches the front of the queue and there at least } B_i \text{ items on shelf of class } i \} \].

To serve all the customers in front of our tagged customer as well our customer, we need the total number of items arriving at the shelf (from spares or from repair) during (0,t+x) to exceed the number of items required to satisfy all the customers who arrived in time (0,t) and as well as our tagged customer. Due to the fact that our tagged customer is the \(N(t)+1^{\text{th}}\) customer,

\[ \{ W_t \leq x \} \iff \{ \text{items arrived on shelf in } (0,t+x) \geq B_{N(t)+1} + \text{Number of items for customers who arrived in time } (0,t) \} \].

We will define the following random variables:

- \(Z(t, x)\) is the number of items repaired by time \(t+x\) from our tagged customer.

Clearly, the distribution of \(Z(t, x) \mid B_{N(t)+1} = b\) is \(\text{bin}(n=b, p=G(x))\).

\[ P(Z(t, x) = k \mid B_{N(t)+1} = b) = \binom{b}{k} G(x)^k (1-G(x))^{b-k}, \quad k=0,1,...,b \]

The number of items arrived to the shelf in \((0,t+x)\) is equal to the number of spares plus the number of items repaired by time \(t+x\). Let \(S(t+x)\) be the number of items repaired in \((0,t+x)\).

\[ \{ \text{Number of items arrived on shelf in } (0,t+x) \geq B_i + \text{Number of items brought by customers in } (0,t) \} = \{ n + S(t+x) + Z(t, x) \geq B_{N(t)+1} + (B_i + ... + B_{N(t)}) \} \]

Thus, we get \( \{ W_t \leq x \} = \left\{ n + S(t+x) + Z(t, x) \geq \sum_{k=1}^{N(t)+1} B_k \right\} \)

In order to determine \(S(t+x)\), we first define the following random variables:

- \(N_1(t, x)\) is the number of items which arrived before time \(t\) and complete repair after time \(t+x\).

- \(N_2(t, x)\) is the number of items which arrived after time \(t\) and complete repair before time \(t+x\).
The number of items repaired by time \( t+x \) is the sum of the number of items repaired from customers who came before time \( t \), plus those from customers who arrived after time \( t \), excluding our tagged customer which we will deal with separately.

\[
S(t + x) = \sum_{k=1}^{N(t)} B_k - N_1(t, x) + N_2(t, x)
\]

Let’s look at \( N_1(t, x) \).

\( Y_k \) is the number of items of any customer \( k \), who arrived in \((0, t)\), which are not repaired by time \( t+x \). \( Y_k \in \{0, 1, 2, ..., B_k\} \). Thus, the number of items repaired after \( t+x \), which arrived before \( t \) is the sum of \( N(t) \) random variable \( Y_k \), where every \( Y_k \sim \text{binomial}(B_k, p_1(t, x)) \) and \( N(t) \sim \text{Poisson}(\lambda t) \). We can then write \( N_1(t, x) = Y_1 + \ldots + Y_{N(t)} \).

We will use generating functions to determine the distribution of \( N_1(t, x) \).

\[
E[z^{N_1(t,x)}] = E\left[E\left[z^{Y_1+...+Y_{N(t)}} \mid N(t)\right]\right]
\]
\[
= E\left[E\left[z^{Y} \left(N(t) \mid N(t)\right)\right]\right]
\]
\[
= e^{zt[E[z] - 1]}
\]
\[
= \exp(\lambda t [E((p_1 z + q_1)^B \mid B) - 1])
\]
\[
= \exp(\lambda t \sum_{b=0}^{\infty} (p_1 z + q_1)^b P(B = b) - 1)
\]
\[
= \exp(\lambda t \sum_{b=0}^{\infty} (p_1(z - 1) + 1)^b P(B = b) - 1)
\]
\[
= \exp(\lambda t [\mathcal{G}_B[p_1(z - 1) + 1] - 1])
\]

where \( \mathcal{G}_B(z) \) is the generating function of the random variable \( B \).

But what happens when \( t \) goes to infinity?

From the previous section, we know that \( \lim_{t \to \infty} p_i = 0 \), but \( \lim_{t \to \infty} p_i t = \int_x^{\infty} G(v)dv \).

Thus,
\[
\lim_{t \to \infty} (\lambda t \left( \sum_{b=0}^{\infty} (p_1(z-1) + 1)^b P(B = b) - 1 \right)) = \lim_{t \to \infty} \left( \sum_{b=0}^{\infty} \sum_{i=0}^{b} \binom{b}{i} p_1^i (z-1)^i P(B = b) - 1 \right) = \lim_{t \to \infty} \left( \sum_{b=0}^{\infty} \sum_{i=1}^{b} \binom{b}{i} p_1^i (z-1)^i P(B = b) \right)
\]

But \[
\lim_{t \to \infty} p_{1,i} = \begin{cases} \frac{\gamma}{x} G(v) dv & i = 1 \\ 0 & i \neq 1 \end{cases}
\]

Hence, we get

\[
\lim_{t \to \infty} (\lambda t \left( \sum_{b=0}^{\infty} \sum_{i=1}^{b} \binom{b}{i} p_1^i (z-1)^i P(B = b) \right)) = \left[ \sum_{b=0}^{\infty} \left( b \lambda p_1 (z-1) P(B = b) \right) \right]
\]

Thus, \( N_1(x) \) is distributed Poisson with parameter

\[
V = \left[ \sum_{b=0}^{\infty} (bP(B_i = b)) \right] \lambda \int_{x}^{\infty} G(v) dv = B \lambda \int_{x}^{\infty} G(v) dv.
\]

Thus, \( P(N_1(x) = k) = \frac{(V)^k}{k!} e^{-Vs} \).

Now, let us consider at \( N_2(t,x) \).

From the previous section, we know that \( N_2(x) \) is a compound Poisson Process with \( N(x) \) “customers”, where each customer brings \( B \) items. Thus,

\[
P(N_2(t,x) = r | B_i = b) = P(N_2(x) = r | B_i = b) = \sum_{k=0}^{b} \binom{bk}{r} p_2^r q_2^{bk-r} \frac{(\lambda x)^k}{k!} e^{-\lambda x}.
\]

Now, let us define the nonstationary and the stationary distributions.

\[
F_i(x) \equiv P(W_i \leq x | B = b) = P(n + ((B_1 + \ldots + B_{N(i)}) - N_1(t,x)) + N_2(t,x) + Z(t,x) \geq B_{N(i)+1} + (B_1 + \ldots + B_{N(i)}))
\]

\[
= P(N_1(t,x) - N_2(t,x) \leq n + Z(t,x) - B_{N(i)+1})
\]

\[
= \sum_{i=0}^{b} P(Z(t,x) = i) P(N_1(t,x) - N_2(t,x) \leq n - (B_{N(i)+1} - i))
\]

\[
= \sum_{b=0}^{\infty} \sum_{k=0}^{b} G(x)^k \left[ 1 - G(x) \right]^{b-k} P(N_1(t,x) - N_2(t,x) \leq n - (b - k)) P(B = b)
\]

112
To simplify, we define a new random variable:

\[ D(t, x) = N_1(t, x) - N_2(t, x) \]

Proposition:

In an exchangeable-item FIFO system, the nonstationary delay distribution of a customer that arrives at time \( t \) with \( B \) items is given by

\[
F_i(x) = P(W_i \leq x) = \sum_{b=0}^{\infty} \sum_{k=0}^{b} \binom{b}{k} G(x)^k \left[ 1 - G(x) \right]^{b-k} P(D(t, x) \leq n - (b - k)) P(B = b)
\]

The stationary waiting time distribution of a random customer is

\[
F(x) = P(W \leq x) = \sum_{b=0}^{\infty} \sum_{k=0}^{b} \binom{b}{k} G(x)^k \left[ 1 - G(x) \right]^{b-k} P(D(x) \leq n - (b - k)) P(B = b)
\]

Finally, we obtain to the waiting time distribution for each sub-system in our ERSOF-model, when bulk arrivals are allowed. These calculations can be included into the research above to make the model more realistic and more flexible. In Chapter 9, we analyzed bulk models, which means that every customer can bring a number of items of the same class. But what happens if the customer brings several items from different classes? This will be done in Chapter 10.
10. Developing ERSMF-models

10.0. Introduction

In the previous chapter, we analyzed an ERSOF-model with bulk arrivals. This means that every customer brings several items from the same class, which are then repaired and replaced. In this chapter, we want to analyze models where customers can bring items of different classes to be repaired and replaced. Here, we want to develop the language and basic formulas for an ERSMF-model. In chapter 4, we defined the basic language of the model. Thus, the analysis of these topics is progressively developed in steps over several sections. In this chapter, we will only deal with a model with 2 item classes. Clearly, these ideas can be generalized and this will be pursued in further research. The customers which arrive to the system can be divided into three categories. One category (or customer type) brings only item class 1 (this is analogous to a product which failed only in item class 1), another category (customer type 2) brings one item of class 2 and the third category, or customer type 3 (CT3), brings one of each (Table 10.1).

<table>
<thead>
<tr>
<th>Customer type</th>
<th>Item class 1</th>
<th>Item class 2</th>
<th>Arrival rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT1</td>
<td>1</td>
<td>0</td>
<td>$\lambda_{10}$</td>
</tr>
<tr>
<td>CT2</td>
<td>0</td>
<td>1</td>
<td>$\lambda_{01}$</td>
</tr>
<tr>
<td>CT3</td>
<td>1</td>
<td>1</td>
<td>$\lambda_{11}$</td>
</tr>
</tbody>
</table>

Table 10.1: Example of the customer types with two item classes in the system

In section 10.1, we will look at a system which contains only customer type 3 and in section 10.2, we consider a general system with customer type 1, 2 and 3.

10.1. An ERSMF-model where all the customers are of type 3

An ERSMF-model is a model where the customer arriving to the repair facility brings a product which failed in more than one item class. For simplicity, we will assume that every class contains exactly one item. Further to ease the analysis of the model we will assume that the product contains only two classes. A random customer arrives at
time \( t \) to this system with a product which failed through class 1 and through class 2. He waits until he gets full satisfaction, which means that both items are replaced and leaves the system. \( W_i \) is the waiting time of the customer who arrives at time \( t \). In fact, he enters a virtual queue for each item. Whenever he becomes completely satisfied by both subsystems, he leaves the system.

\[
\{ W_i \leq x \} \leftrightarrow \{ \text{by time } t+x, \text{ our tagged customer reaches the front of both queues and there is at least one item on each shelf} \}.
\]

To serve all the customers in front of our tagged customer as well as our tagged customer, we need the total number of items arriving at each shelf (from spares or from repair) during \((0,t+x)\) to exceed the number of items required to satisfy all the customers who arrived in time \((0,t)\), as well as our tagged customer. Thus,

\[
\{ W_i \leq x \} \leftrightarrow \begin{cases} \text{Number of items arrived on shelf 1 in } (0,t+x) \geq 1 + \text{Number of items from customers who arrived in } (0,t), & \text{and Number of items arrived on shelf 2 in } (0,t+x) \geq 1 \\ \text{+ Number of items from customers who arrived in } (0,t) \end{cases}
\]

(10.1)

We will define the following random variables:

\( Z(t, x) \) is a vector of items repaired by time \( t+x \) from our tagged customer who arrived at \( t \). Thus,

\[
Z(t, x) = \begin{cases} (0,0) & w.p. \ (1-G_1(x))*(1-G_2(x)) \\ (0,1) & w.p. \ (1-G_1(x))*G_2(x) \\ (1,0) & w.p. \ G_1(x)(1-G_2(x)) \\ (1,1) & w.p. \ G_1(x)*G_2(x) \end{cases}
\]

(10.2)

where \( G_1(x) \) is the cumulative repair time distribution of \( ss_1 \) and \( G_2(x) \) is the cumulative repair time distribution of \( ss_2 \).

The number of items arrived to the shelf is equal to the number of spares plus the number of items repaired by time \( t+x \). Let

\[
S_1(t+x) = \text{the number of repaired items of } ss_1 \text{ by time } t+x.
\]

\[
S_2(t+x) = \text{the number of repaired items of } ss_2 \text{ by time } t+x.
\]

From (10.1) we have \( \{ n_1+S_1(t+x)+Z(t,x) \geq 1 + N(t), n_2+S_2(t+x)+Z(t,x) \geq 1 + N(t) \} \).
We define the following random variables:

\( N_{1,11}(t,x) \) is the number of items which arrived in \((0,t)\) that complete repair in both \(ss_1\) and \(ss_2\) by time \(t+x\).

\( N_{1,10}(t,x) \) is the number of items which arrived in \((0,t)\) with 1 repaired in \(ss_1\) and the other not in \(ss_2\) by time \(t+x\).

\( N_{1,01}(t,x) \) is the number of items which arrived in \((0,t)\) with 1 repaired in \(ss_2\) and the other not completed repair in \(ss_1\) by time \(t+x\).

\( N_{1,00}(t,x) \) is the number of items which arrived in \((0,t)\) and none of the items is repaired by time \(t+x\).

Thus, \( N_{1,12}(t,x) + N_{1,10}(t,x) + N_{1,02}(t,x) + N_{1,00}(t,x) = N(t) \).

\( N_{2,11}(t,x) \) is the number of items which arrived in \((t,t+x)\), where the first is repaired in \(ss_1\) and the second in \(ss_2\) by time \(t+x\).

\( N_{2,10}(t,x) \) is the number of items, which arrived in \((t,t+x)\), where the first is repaired in \(ss_1\) and the second not in \(ss_2\) by time \(t+x\).

\( N_{2,01}(t,x) \) is the number of items, which arrived in \((t,t+x)\), where the first is not repaired in \(ss_1\) and the second repaired in \(ss_2\) by time \(t+x\).

\( N_{2,00}(t,x) \) is the number of items which arrived in \((t,t+x)\), where neither the first nor the second repaired in \(ss_2\) by time \(t+x\).

Thus, \( N_{2,12}(t,x) + N_{2,10}(t,x) + N_{2,02}(t,x) + N_{2,00}(t,x) = N(x) \).

The number of items repaired by time \(t+x\) is equal to the number of items which arrived before \(t\) and repaired before \(t+x\) plus the items which arrived after \(t\) and are repaired before \(t+x\). We will obtain the formulas for each class.

For class 1:

\[
S_1(t + x) = (N(t) - N_{1,00}(t,x) - N_{1,01}(t,x)) + N_{2,11}(t,x) + N_{2,10}(t,x)
\]

For class 2:

\[
S_2(t + x) = (N(t) - N_{1,00}(t,x) - N_{1,10}(t,x)) + N_{2,11}(t,x) + N_{2,01}(t,x)
\]
Thus,
\[ \{W_i \leq x\} \Longleftrightarrow \left\{ n_i + S_1(t + x) + Z(t, x) \geq 1 + N(t) \cap \left\{ n_2 + S_2(t + x) + Z(t, x) \geq 1 + N(t) \right\} \right\} \]

\[ = \left\{ n_1 + (N(t) - N_{1,00}(t, x) - N_{1,01}(t, x)) + N_{2,11}(t, x) + N_{2,10}(t, x) + Z(t, x) \geq 1 + N(t) \right\} \]
\[ = \left\{ n_2 + (N(t) - N_{1,00}(t, x) - N_{1,10}(t, x)) + N_{2,11}(t, x) + N_{2,01}(t, x) + Z(t, x) \geq 1 + N(t) \right\} \]
\[ = \left\{ n_1 - N_{1,00}(t, x) - N_{1,01}(t, x) + N_{2,11}(t, x) + N_{2,10}(t, x) + Z(t, x) \geq 1 \right\} \]
\[ = \left\{ n_2 - N_{1,00}(t, x) - N_{1,10}(t, x) + N_{2,11}(t, x) + N_{2,01}(t, x) + Z(t, x) \geq 1 \right\} \]
\[ = \left\{ n_1 + Z(t, x) - 1 \geq N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \right\} \]
\[ = \left\{ n_2 + Z(t, x) - 1 \geq N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \right\} \]
\[ = \left\{ N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \leq n_1 + Z(t, x) - 1 \right\} \]
\[ = \left\{ N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 + Z(t, x) - 1 \right\} \]

We consider each random variable in turn.

\[ N_{1,00}(t, x) : \]

![Graph of CT3 arriving in du (before t) and both items repaired after t+x](image)

Figure 10.1: Graph of CT3 arriving in du (before t) and both items repaired after t+x

A customer who arrived into du in (t,t+x) has a probability of \( G_1(t + x - u) G_2(t + x - u) \) of not having each of the items repaired until time t+x. Therefore, the probability of a random customer arriving and not having both his items repaired by time t+x is

\[ \int_0^t G_1(t + x - u) G_2(t + x - u) \frac{du}{t} \] or \[ \frac{1}{t} \int_x^{t+x} G_1(v) G_2(v) dv \]

117
Customers arrive as a Poisson process with rate $\lambda$. Thus, $N_{1,00}(t,x)$ is distributed Poisson with parameter $\lambda t \int_{x}^{t+x} G_1(v)G_2(v) \, dv$. When $t \to \infty$, we get parameter $\lambda \int_{x}^{\infty} G_1(v)G_2(v) \, dv$.

$N_{1,01}(t,x)$:

![Diagram](image)

**Figure 10.2: CT3 arriving in $du \in (0,t)$ and having the item of class 2 repaired before $t+x$ and the item of class 1 repaired after $t+x$**

A customer who arrived into $du \in (t,t+x)$ has a probability of $G_1(t+x-u)$ of not having his item of class 1 repaired and $G_2(t+x-u)$ of having repaired the item until time $t+x$. Therefore, the probability of a random customer arriving of having his item of class 2 repaired but not the item of class 1 by time $t+x$ is $\int_{0}^{t} G_1(t+x-u)G_2(t+x-u) \frac{du}{t}$, that is,

$$\frac{1}{t} \int_{x}^{t+x} G_1(v)G_2(v) \, dv.$$ Since customers arrive as a Poisson Process with rate $\lambda$, therefore $N_{1,02}(t,x)$ is distributed Poisson with parameter $\lambda \int_{x}^{t+x} G_1(v)G_2(v) \, dv$. When $t \to \infty$, we get parameter $\lambda \int_{x}^{\infty} G_1(v)G_2(v) \, dv$. 

118
Figure 10.3: CT3 arriving in (0,t) and having his item of class 1 repaired before t+x and his item of class 2 repaired after t+x

A customer who arrived into du ∈ (t,t+x) has a probability of $G_1(t + x - u)$ of having his item of class 1 repaired and $\overline{G}_2(t + x - u)$ of not having his item of class 1 repaired until time t+x. Therefore, the probability of a random customer arriving having his item of class 1 repaired and not having his item of class 2 repaired by time t+x is

$$\int_0^t G_1(t + x - u)\overline{G}_2(t + x - u)du$$

or

$$\int_x^{t+x} G_1(v)\overline{G}_2(v)dv.$$  

Since customers arrive as a Poisson Process with rate $\lambda$. Thus, $N_{1,10}(t,x)$ is distributed Poisson with parameter $\lambda \int_x^{t+x} G_1(v)\overline{G}_2(v)dv$. When $t \rightarrow \infty$, we get parameter $\lambda \int_x^\infty G_1(v)\overline{G}_2(v)dv$. 


Figure 10.4: CT3 arriving in (0,t) and having his item of class 1 repaired before t+x and his item of class 2 repaired before t+x

A customer who arrived into \(du \in (t,t+x)\) has a probability of \(G_1(t + x - u)\) of having his item of class 1 repaired and \(G_2(t + x - u)\) of having his item of class 2 repaired before time \(t+x\). Therefore, the probability of a random customer arriving and having his item of class 1 repaired and his item of class 2 repaired by time \(t+x\) is

\[
\int_0^t G_1(t + x - u)G_2(t + x - u) \frac{du}{t}, \text{ or } \int_x^{t+x} G_1(v)G_2(v) \frac{dv}{t}.
\]

Customers arrive as a Poisson Process with \(\lambda\). Thus, \(N_{1,11}(t,x)\) is distributed Poisson with parameter \(\lambda t \int_x^{t+x} G_1(v)G_2(v) \frac{dv}{t}\).

When \(t \to \infty\), we get \(\lambda \int_x^{\infty} G_1(v)G_2(v) dv\).

Now, let us analyze the second group of random variables.
$N_{2,11}(t,x)$:

![Figure 10.5: CT3 arriving in du (after t) and having his item of class 1 repaired before t+x and his item of class 2 repaired before t+x](image)

A customer who arrived into $du \in (t,t+x)$ has a probability of $G_1(t+x-u)$ of having his item of class 1 repaired and $G_2(t+x-u)$ of having his item of class 2 repaired by time t+x. Therefore, the probability of a random customer arriving and having both his items of class 1 repaired and his item of class 2 repaired by time t+x is

$$\int_{t}^{t+x} G_1(t+x-u)G_2(t+x-u)du,$$

or

$$\int_{0}^{x} G_1(v)G_2(v)dv.$$ 

The customers arrive as a Poisson process with rate $\lambda$. Thus, $N_{2,11}(t,x)$ is distributed Poisson with parameter $\lambda \int_{0}^{x} G_1(v)G_2(v)dv$.

$N_{2,10}(t,x)$:

![Figure 10.6: CT3 arriving in du (after t) and his item of class 1 repaired and his item of class 2 not repaired before t+x.](image)
A customer who arrived into \( du \in (t, t+x) \) has a probability of \( G_1(t + x - u) \) of having his item of class 1 repaired and \( \overline{G}_2(t + x - u) \) of not having his item of class 2 repaired by time \( t+x \). Therefore, the probability of a random customer arriving and having his item of class 1 repaired and not having his item of class 2 repaired by time \( t+x \) is

\[
\int_t^{t+x} G_1(t + x - u) \overline{G}_2(t + x - u) \frac{du}{t} \quad \text{or} \quad \int_0^x G_1(v) \overline{G}_2(v) \frac{dv}{t}.
\]

Customers arrive as a Poisson Process with rate \( \lambda \). Thus, \( N_{2,10}(t,x) \) is distributed Poisson with parameter \( \lambda \int_0^x G_1(v) \overline{G}_2(v)dv \).

\( N_{2,01}(t,x) \):

![Graph of CT3 arriving in du(after t) and item of class 2 repaired and item of class 1 not repaired before t+x.](image)

Figure 10.7: Graph of CT3 arriving in du(after t) and item of class 2 repaired and item of class 1 not repaired before t+x.

A customer who arrived in \( du \in (t, t+x) \) has a probability of \( \overline{G}_1(t + x - u) \) of not having his item of class 1 repaired and \( G_2(t + x - u) \) of having his item of class 2 repaired by time \( t+x \). Therefore, the probability of a random customer arriving and not having both his item of class 1 and his item of class 2 by time \( t+x \) is

\[
\int_t^{t+x} \overline{G}_1(t + x - u) G_2(t + x - u) \frac{du}{t}, \quad \text{or} \quad \int_0^x \overline{G}_1(v) G_2(v) \frac{dv}{t}.
\]

Customers arrive as a Poisson process with parameter \( \lambda \). Thus, \( N_{2,02}(t,x) \) is distributed Poisson with parameter

\[
\lambda \int_0^x \overline{G}_1(v) G_2(v)dv.
\]

\( N_{2,00}(t,x) \):
Figure 10.8: CT3 arriving in (0,t) and having his item of class 1 and his item of class 2 repaired after t+x

A customer who arrived in \( \text{du} \in (t,t+x) \) has a probability of \( G_1(t + x - u) \) of having his item of class 1 repaired and \( G_2(t + x - u) \) of having his item of class 2 repaired by time \( t+x \). Therefore, the probability of a random customer arriving and not having both his items repaired by time \( t+x \) is 

\[
\int_t^{t+x} \int_{t+u}^{x} G_1(t + x - v)G_2(t + x - u)\, dv \, du,
\]

or

\[
\int_0^x \int_0^{x-u} G_1(v)G_2(v+u)\, dv \, du.
\]

The customer arrive as a Poisson Process. Thus, \( N_{2,00}(t,x) \) is distributed Poisson with parameter \( \lambda \int_0^x G_1(v)G_2(v)\, dv \).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>( t \rightarrow \infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_{1,11}(t,x) )</td>
<td>( \lambda \int x )</td>
</tr>
<tr>
<td>( N_{1,10}(t,x) )</td>
<td>( \lambda \int G_1(v)G_2(v)dv )</td>
</tr>
<tr>
<td>( N_{1,01}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)G_2(v)dv )</td>
</tr>
<tr>
<td>( N_{1,00}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)\overline{G}_2(v)dv )</td>
</tr>
<tr>
<td>( N_{2,11}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)G_2(v)dv )</td>
</tr>
<tr>
<td>( N_{2,10}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)\overline{G}_2(v)dv )</td>
</tr>
<tr>
<td>( N_{2,01}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)G_2(v)dv )</td>
</tr>
<tr>
<td>( N_{2,00}(t,x) )</td>
<td>( \lambda \int \overline{G}_1(v)\overline{G}_2(v)dv )</td>
</tr>
<tr>
<td>( N(t) )</td>
<td>( \lambda t )</td>
</tr>
</tbody>
</table>
From (10.2) we get for $x \geq 0$

$$P(W_t \leq x) = \sum_{k=-\infty}^{n-1} \left[ P[N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k] \right]$$

$$+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P[N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k] \right]$$

$$+ G_1(x) P[N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = n_1]$$

$$+ G_2(x) P(n_1 + N_{2,10}(t,x) + N_{1,10}(t,x) - N_{1,01}(t,x) - N_{2,01}(t,x) = n_2)$$

$$+ P(n_1 + N_{1,10}(t,x) + N_{2,10}(t,x) - N_{1,01}(t,x) - N_{2,01}(t,x) \leq n_2 - 1)$$

(10.3)

where

$$P(N_{1,00}(x) + N_{1,02}(x) - N_{2,12}(x) - N_{2,10}(x) = r)$$

$$= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} P(N_{1,00}(x) = i) P(N_{1,02}(x) = k) P(N_{2,12}(x) = k) P(N_{2,10}(x) = i + j - k - r)$$

For the proof see Appendix E.

To simplify the above, we define the following random variables:

$$N_1(x) = N_{1,00}(x) + N_{1,01}(x)$$

$$N_2(x) = N_{2,11}(x) + N_{2,10}(x)$$

$$N_3(x) = N_{2,10}(x) + N_{1,10}(x)$$

$$N_4(x) = N_{1,01}(x) + N_{2,01}(x)$$

Since all $N_{i,j,k}$ are Poisson, therefore $N_j(x)$ ($j = 1, \ldots, 4$) are all Poisson. Their parameters are given below in the summary table.

$N_1(x)$ is the sum of two independent Poisson random variables. Thus, it is also distributed

Poisson with parameter $\lambda \int_{x}^{\infty} \overline{G}_1(v) \overline{G}_2(v) dv + \lambda \int_{x}^{\infty} \overline{G}_1(v) G_2(v) dv = \lambda \int_{x}^{\infty} \overline{G}_1(v) dv$
$N_2(x)$ is the sum of two independent Poisson random variables. Thus, it is also distributed Poisson with parameter \( \lambda \int_0^x G_1(v)G_2(v) dv + \lambda \int_0^x G_1(v)\overline{G}_2(v) dv = \lambda \int_0^x G_1(v) dv \)

$N_3(x)$ is the sum of two Poisson random variables. Thus, it is also distributed Poisson with parameter \( \lambda \int_0^x G_1(v)\overline{G}_2(v) dv + \lambda \int_0^x G_1(v)G_2(v) dv = \lambda \int_0^x G_1(v)G_2(v) dv \)

$N_4(x)$ is the sum of two Poisson random variables. Thus, it is also distributed Poisson with parameter \( \lambda \int_0^x G_1(v)G_2(v) dv + \lambda \int_0^x G_1(v)\overline{G}_2(v) dv = \lambda \int_0^x G_1(v)\overline{G}_2(v) dv \).

Summary Table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_1(x)$</td>
<td>$\lambda \int_x^\infty \overline{G}_1(v) dv$</td>
</tr>
<tr>
<td>$N_2(x)$</td>
<td>$\lambda \int_0^x G_1(v) dv$</td>
</tr>
<tr>
<td>$N_3(x)$</td>
<td>$\lambda \int_0^x G_1(v)\overline{G}_2(v) dv$</td>
</tr>
<tr>
<td>$N_4(x)$</td>
<td>$\lambda \int_0^x \overline{G}_1(v)G_2(v) dv$</td>
</tr>
</tbody>
</table>
Thus, from (10.3), we get for \( x \geq 0 \)

\[
P(W \leq x) = \sum_{k=-\infty}^{n-1} \left[ P(N_1(x) - N_2(x) = k) ight] P(N_1(x) - N_4(x) \leq n_2 - k - 1) 
+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_1(x) - N_2(x) = k) ight] P(N_3(x) - N_4(x) = n_2 - k) 
+ G_1(x) P(N_1(x) - N_2(x) = n_1) 
\]

\[+ G_2(x) P(N_3(x) - N_4(x) = n_2 - n_1) 
+ P(N_3(x) - N_4(x) \leq n_2 - n_1 - 1) \]  

(10.4)

We define the following random variables:

\[
D_1(x) = N_1(x) - N_2(x) 
\]

(10.5)

\[
D_2(x) = N_3(x) - N_4(x) 
\]

(10.6)

Thus, from (10.4), we get for \( x \geq 0 \)

\[
P(W \leq x) = \sum_{k=-\infty}^{n-1} \left[ P(D_1(x) = k) P(D_2(x) \leq n_2 - k - 1) \right] 
+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P(D_1(x) = k) P(D_2(x) = n_2 - k) \right] 
+ G_1(x) P(D_1(x) = n_1) \left[ G_2(x) P(D_2(x) = n_2 - n_1) 
+ P(D_2(x) \leq n_2 - n_1 - 1) \right] \]

(10.7)

\( D_1(x) \) has the mean \( \lambda \int_x^\infty G_1(v)dv - \lambda \int_0^x G_1(v)dv \)

\[
= \lambda \int_x^\infty (1 - G_1(v))dv + \lambda \int_0^x G_1(v)dv + \lambda 
\]

\[
= \lambda \left[ \int_0^x (1 - G_1(v))dv + \int_0^x dv \right] 
= \frac{\lambda}{\mu} + \lambda x. 
\]

If \( \frac{\lambda}{\mu} + \lambda x \gg 1 \), we can use the normal approximation \( N\left[ \frac{\lambda}{\mu} + \lambda x, \frac{\lambda}{\mu} + \lambda x \right] \).
\[ D_2(x) \text{ has mean } \lambda \int_0^\infty G_1(v)G_2(v)dv - \lambda \int_0^\infty G_1(v)G_2(v)dv \]

\[ = \lambda \int_0^\infty G_1(v)(1-G_2(v))dv = \lambda \int_0^\infty (1-G_1(v))(G_2(v))dv \]

\[ = \lambda \int_0^\infty G_1(v) - G_2(v)dv \]

\[ = \lambda \int_0^\infty 1 - G_2(v)dv - \lambda \int_0^\infty 1 - G_1(v)dv \]

\[ = \frac{\lambda}{\mu_1} - \frac{\lambda}{\mu_2}. \]

If \( \left| \frac{\lambda}{\mu_1} - \frac{\lambda}{\mu_2} \right| >> 1 \), we can use the normal approximation \( N \left[ \frac{\lambda}{\mu_1} - \frac{\lambda}{\mu_2}, \frac{\lambda}{\mu_1} + \frac{\lambda}{\mu_2} \right] \).

10.2. An ERSMF-model, where the systems contains CT1, CT2 and CT3

10.2.1. When the CT3 is arriving

When we are dealing with a system where the customer can bring up to two items, there are 3 different customer types. The first brings only one of class 1, the second brings only one of class 2, and the third brings 1 from each class. Hausman and Cheung [16] tried to solve this model, but were not able to do it successfully. In Appendix C we explained the fundamental errors in their development. In this section, we develop the waiting time for CT3 in the next section, the formula for CT1. To keep all the random variables in order, we will use superscripts. They will define to which group of random variable they belong. As an example, \( \lambda^1 \) is the arrival rate for customer type 3, \( \lambda^{01} \) the arrival rate for CT1 and \( \lambda^{01} \) the arrival rate of CT2. Thus,

\[ \{ W_i \leq x \} \leftrightarrow \{ \text{by time } t+x, \text{ our tagged reaches the front of both virtual queues and there is at least one item on each shelf} \}. \]

\[ \{ W_i \leq x \} = \{ \text{Number of items arrived onto shelf 1 in } (0,t+x) \geq 1 + \text{ items for customer in time } (0,t) \} \text{ and } \{ \text{ items arrived onto shelf 2 in } (0,t+x) \geq 1 + \text{ items for customer in time } (0,t) \} \]
We define the following random variables:

\[ Z^{11}(t,x) \] is a vector of items repaired by time \( t+x \) from our tagged customer.

\[
Z^{11}(t,x) = \begin{cases} 
(0,0) & \text{w.p.} \quad (1-G_1(x)) \cdot (1-G_2(x)) \\
(0,1) & \text{w.p.} \quad ((1-G_1(x)) \cdot G_2(x)) \\
(1,0) & \text{w.p.} \quad G_1(x) \cdot (1-G_2(x)) \\
(1,1) & \text{w.p.} \quad G_1(x) \cdot G_2(x) 
\end{cases}
\]

where \( G_1(x) \) is the cumulative repair time distribution of \( ss_1 \) and \( G_2(x) \) is the cumulative repair time distribution of \( ss_2 \).

The number of items arrived to the shelf in \((0, t+x)\) is equal to the number of spares plus the number of items repaired by time \( t+x \).

\[ S_1(t+x) = \text{the number of items repaired by time } t+x \text{ in } ss_1. \]

\[ S_2(t+x) = \text{the number of items repaired by time } t+x \text{ in } ss_2. \]

\( N^{10}(t,x) \) is the number of customer type 1 who arrived to the system by time \((t, t+x)\).

\( N^{01}(t,x) \) is the number of customer type 2 who arrived to the system by time \( t \) in \((t, t+x)\).

\( N^{11}(t,x) \) is the number of customer type 3 who arrived to the system by time \( t \) in \((t, t+x)\).

\( N^{11}_{1,11}(t,x) \) is the number of items which arrived in \((0, t)\) from CT3 and are repaired in \( ss_1 \) and \( ss_2 \) by time \( t+x \).

\( N^{11}_{1,10}(t,x) \) is the number of items which arrived in \((0, t)\) from CT3 with 1 repaired in \( ss_1 \) and the other not in \( ss_2 \) by time \( t+x \).

\( N^{11}_{1,01}(t,x) \) is the number of items which arrived in \((0, t)\) from CT3 with 1 repaired in \( ss_2 \) and the other not in \( ss_1 \) by time \( t+x \).

\( N^{11}_{1,00}(t,x) \) is the number of items which arrived in \((0, t)\) from CT3 with neither repaired by time \( t+x \).

Thus, \( N^{11}(0,t) = N^{11}_{1,11}(t,x) + N^{11}_{1,10}(t,x) + N^{11}_{1,01}(t,x) + N^{11}_{1,00}(t,x) \).

\( N^{11}_{2,11}(t,x) \) is the number of items which arrived from CT3 in \((t, t+x)\) and are repaired either in \( ss_1 \) and in \( ss_2 \) by time \( t+x \).
$N_{2,10}^{11}(t,x)$ is the number of items which arrived from CT3 in $(t,t+x)$ and are repaired in $ss_1$ but not in $ss_2$ by time $t+x$.

$N_{2,01}^{11}(t,x)$ is the number of items which arrived from CT3 in $(t,t+x)$ and are repaired in $ss_2$ but not in $ss_1$ by time $t+x$.

$N_{2,00}^{11}(t,x)$ is the number of items which arrived from CT3 in $(t,t+x)$ but are repaired neither in $ss_1$ nor in $ss_2$ by time $t+x$.

Thus,

\[
N_1^{11}(t,x) = N_{2,11}^{11}(t,x) + N_{2,10}^{11}(t,x) + N_{2,01}^{11}(t,x) + N_{2,00}^{11}(t,x).
\]

$N_1^{10}(t,x)$ is the number of items which arrived in $(0,t)$ from CT1 and are not repaired by time $t+x$.

$N_2^{01}(t,x)$ is the number of items which arrived in $(0,t)$ from CT2 and are not repaired by time $t+x$.

$N_2^{10}(t,x)$ is the number of items which arrived in $(t,t+x)$ from CT1 and are repaired by time $t+x$.

$N_2^{01}(t,x)$ is the number of items which arrived in $(t,t+x)$ from CT2 and are repaired by time $t+x$.

\[
\{ W_i \leq x \} \iff \{ \text{Number of items arrived onto shelf } 1 \text{ in } (0,t+x) \geq 1 + \text{items needed for customers in time(0,t), and Number of items arrived to shelf } 2 \text{ in } (0,t+x) \geq 1 + \text{items needed for customer in time(0,t)} \} = \{ n_1 + S_1(t+x) + Z(t,x) \geq 1 + N_{11}^{11}(t) + N_{10}^{10}(t), \]
\[
N_2^{11}(t,x) = \{ n_2 + S_2(t+x) + Z(t,x) \geq 1 + N_{21}^{11}(t) + N_{20}^{10}(t) \}.
\]

The number of repaired items by time $(t+x)$ is the total number of items which arrived before $t$ and are repaired before $(t+x)$, plus the number of items which arrived after $t$ and are repaired by time $(t+x)$.

Thus,

For class 1:
\[ S_1(t + x) = (N_{11}^{11}(t) - N_{1,00}^{11}(t, x) - N_{1,10}^{11}(t, x)) \\
+ N_{2,11}^{11}(t, x) + N_{2,10}^{11}(t, x) + N_1^{10}(t) - N_1^{10}(t, x) + N_2^{10}(t) \]

For class 2:
\[ S_2(t + x) = (N(t) - N_{1,00}^{11}(t, x) - N_{1,10}^{11}(t, x)) \\
+ N_{2,11}^{11}(t, x) + N_{2,01}^{11}(t, x) + + N_1^{01}(t) - N_1^{01}(t, x) + N_2^{01}(t, x) \]

Thus, for \( x \geq 0 \)

\[
P(W \leq x) = \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}^{11}(x) + N_{1,01}^{11}(x) - N_{2,11}^{11}(x) - N_{2,10}^{11}(x) + N_1^{10}(x) - N_2^{10}(x) = k) \right] \\
+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}^{11}(x) + N_{1,01}^{11}(x) - N_{2,11}^{11}(x) - N_{2,10}^{11}(x) + N_1^{10}(x) - N_2^{10}(x) = k) \right] \\
+ G_1(x) P(N_{1,00}^{11}(x) + N_{1,01}^{11}(x) - N_{2,11}^{11}(x) - N_{2,10}^{11}(x) + N_1^{10}(x) - N_2^{10}(x) = n_1) \\
+ G_2(x) P(N_{1,00}^{11}(x) + N_{1,01}^{11}(x) - N_{2,11}^{11}(x) - N_{2,10}^{11}(x) + N_1^{10}(x) - N_2^{10}(x) = n_2) \\
+ P(N_{1,00}^{11}(x) + N_{1,01}^{11}(x) - N_{2,11}^{11}(x) - N_{2,10}^{11}(x) + N_1^{01}(x) - N_2^{01}(x) + N_1^{10}(x) - N_2^{10}(x) \leq n_2 - 1) \]

(10.8)

For the Proof see Appendix H.

But still, the formulas are awkward and therefore to simplify, we introduce the following random variables:

\[ N_1(x) = N_{1,00}^{11}(x) + N_{1,01}^{11}(x) + N_1^{10}(x) \]
\[ N_2(x) = N_{2,11}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) \]
\[ N_3(x) = N_{1,10}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) + N_1^{01}(x) \]
\[ N_4(x) = N_{2,01}^{11}(x) + N_{1,01}^{11}(x) + N_1^{01}(x) + N_2^{10}(x) \]
Each $N_i(x)$ is the sum of a number of Poisson random variables and thus is itself Poisson.

Their means are summarized below:

$N_1(x)$ is Poisson with parameter $\int_x^\infty G_1(v)G_2(v)dv + \int_x^\infty \int G_1(v)G_2(v)dv$

+ $\int_x^\infty G_1(v)dv = (\lambda_1^{11} + \lambda_1^{10})\int G_1(v)dv$.

$N_2(x)$ is Poisson with parameter $\lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{10}\int G_1(v)dv$

= $(\lambda_1^{11} + \lambda_1^{10})\int G_1(v)dv$

$N_3(x)$ is Poisson with parameter $\lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{11}\int G_1(v)G_2(v)dv$

+ $\lambda_1^{10}\int G_1(v)dv + \lambda_1^{10}\int G_1(v)dv = \lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{10}\int G_1(v)dv + \lambda_1^{10}\int G_1(v)dv$

$N_4(x)$ is Poisson with parameter $\lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{10}\int G_1(v)dv$

+ $\lambda_1^{10}\int G_1(v)dv = \lambda_1^{11}\int G_1(v)G_2(v)dv + \lambda_1^{10}\int G_1(v)dv + \lambda_1^{10}\int G_1(v)dv$.

Thus,

$$P(W \leq x) = \sum_{k=0}^{n_2-1} P(N_1(x) - N_2(x) = k)$$

$$+ G_2(x)\sum_{k=0}^{n_1-1} P(N_1(x) - N_2(x) = k)$$

$$+ G_1(x)P(N_1(x) - N_2(x) = n_1)$$

$$+ G_3(x)P(N_1(x) - N_4(x) = n_2 - n_1)$$

$$+ P(N_1(x) - N_4(x) \leq n_2 - n_1 - 1)$$

(10.9)
But still the formula can be made look simpler by defining the following random variables:

\[
D_1(x) = N_1(x) - N_2(x), \quad D_1(x) \in [-\infty, \infty] \\
D_2(x) = N_3(x) - N_4(x) \quad D_2(x) \in [-\infty, \infty]
\]

(10.10) (10.11)

Thus,

\[
P(W \leq x) = \sum_{k=-\infty}^{n_2-1} \left[ P(D_1(x) = k) P(D_2(x) \leq n_2 - k - 1) \right]
+ G_2(x) \sum_{k=-\infty}^{n_2-1} \left[ P(D_1(x) = k) P(D_2(x) = n_2 - k) \right]
+ G_1(x) P(D_1(x) = n_1) \left[ \frac{G_2(x) P(D_2(x) = n_2 - n_1)}{P(D_2(x) \leq n_2 - n_1 - 1)} \right]
\]

(10.12)

\[
D_1(x) \text{ is the difference of two independent Poisson random variables each of which is the} \\
\text{sum of independent Poisson random variables. Thus, it has the mean}
\]

\[
(\lambda^{11} + \lambda^{10}) \int_0^\infty G_1(v) dv - (\lambda^{11} + \lambda^{10}) \int_0^\infty G_1(v) dv =
\]

\[
(\lambda^{11} + \lambda^{10}) \int_0^\infty G_1(v) dv + (\lambda^{11} + \lambda^{10}) \int_0^1 1 - G_1(v) - 1 dv = (\lambda^{11} + \lambda^{10}) \left[ \frac{1}{\mu} - x \right]
\]

\[
D_2(x) \text{ is the difference of two independent Poisson random variables each of which is the} \\
\text{sum of independent Poisson random variables. Thus, it is also distributed with mean}
\]

\[
\lambda \int_0^\infty G_1(v) G_2(v) dv + \lambda^{10} \int_0^\infty G_1(v) dv + \lambda^{11} \int_0^\infty G_2(v) dv - \lambda^{11} \int_0^\infty G_1(v) G_2(v) dv - \lambda^{10} \int_0^\infty G_2(v) dv -
\]

\[
\lambda^{10} \int_0^\infty G_1(v) dv
\]

\[
= \lambda^{11} \int_0^\infty G_1(v) [1 - G_2(v)] - [1 - G_1(v)] G_2(v) dv -
\]

\[
\lambda^{10} \int_0^\infty [1 - G_1(v) - 1 dv + \lambda^{10} \int_0^\infty G_2(v) dv + \lambda^{11} \int_0^\infty 1 - G_2(v) - 1 dv - \lambda^{10} \int_0^\infty G_1(v) dv
\]

133
\[
= \lambda^1 \int_0^\infty G_1(v) - 1 + G_2(v) dv - \lambda^0 \left[ \frac{1}{\mu_1} - x \right] + \lambda^0 \left[ \frac{1}{\mu_2} - x \right]
\]

\[
= \lambda^1 \left[ \frac{1}{\mu_2} - \frac{1}{\mu_1} \right] - \lambda^0 \left[ \frac{1}{\mu_1} - x \right] + \lambda^0 \left[ \frac{1}{\mu_2} - x \right].
\]

In Appendix I, we demonstrate a check mechanism for this formula.

10.2.2. When the CT1 is arriving

In the previous section, we dealt with a system which contains two class of items and where the customer is of CT3. But in the same system, there may be customers who bring only one item of class 1 or one of class 2. We will analyze a customer who brings only one of class 1. As in the previous section, we will start with the event of a random customer who arrived at time t and will wait less than x.

\[\{W_i \leq x\} = \{ \text{by time } t+x, \text{ our tagged customer reaches the front of the queue and there is at least one item on the shelf} \} .\]

\[\{W_i \leq x\} = \{ \text{Number of items arrived on shelf 1 in } (0,t+x) \geq 1 + \text{Number of items for all customer in } (0,t) \} .\]

We will define the following random variables:

\[Z(t,x) \text{ is the number of items repaired by time } t+x \text{ from our tagged customer.}\]

\[Z(t,x) = \begin{cases} 0 & G_1(x) \\ 1 & 1 - G_1(x) \end{cases}\]

The number of items arrived onto the shelf is equal to the number of spares plus the number of items repaired by time t+x.

\[S_i(t+x) = \text{the number of repaired items by time } t+x.\]

\[\{W_i \leq x\} = \{ \text{Number of items arrived onto shelf 1 in } (0,t+x) \geq 1 + \text{items for customer in time } (0,t) \} = \{ n_j + S_i(t+x) + Z(t,x) \geq 1 + N^{11}(t) + N^{10}(t) \} .\]

134
For class 1:
\[ S_1(t + x) = (N^{11}(t) - N_{1,00}^{11}(t, x) - N_{1,01}^{11}(t, x)) + N_{2,11}^{11}(t, x) + N_{2,10}^{11}(t, x) + N^{10}(t) - N_1^{10}(t, x) + N_2^{10}(t, x) \]

\[
\{ W_t \leq x \} = \left\{ n_1 + S_1(t + x) + Z(t, x) \geq 1 + N^{11}(t) + N^{10}(t) \right\}
\]
\[
= \left\{ n_1 + (N^{11}(t) - N_{1,00}^{11}(t, x) - N_{1,01}^{11}(t, x)) + N_{2,11}^{11}(t, x) + N_{2,10}^{11}(t, x) + N^{10}(t) - N_1^{10}(t, x) + N_2^{10}(t, x) \right\}
\]
\[
+ Z(t, x) \geq 1 + N^{11}(t) + N^{10}(t)
\]
\[
= \{ N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) - N_{2,11}^{11}(t, x) - N_{2,10}^{11}(t, x) + N_1^{10}(t, x) - N_2^{10}(t, x) \leq n_1 + Z(t, x) - 1 \}
\]

Thus,
\[
P(W_t \leq x) = G_1(x)P(N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) - N_{2,11}^{11}(t, x) - N_{2,10}^{11}(t, x) + N_1^{10}(t, x) - N_2^{10}(t, x) \leq n_1)
\]
\[
+ \overline{G}_1(x)P(N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) - N_{2,11}^{11}(t, x) - N_{2,10}^{11}(t, x) + N_1^{10}(t, x) - N_2^{10}(t, x) \leq n_1 - 1)
\]
\[
= G_1(x)P(N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) - N_{2,11}^{11}(t, x) - N_{2,10}^{11}(t, x) + N_1^{10}(t, x) - N_2^{10}(t, x) = n_1)
\]
\[
+ P(N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) - N_{2,11}^{11}(t, x) - N_{2,10}^{11}(t, x) + N_1^{10}(t, x) - N_2^{10}(t, x) \leq n_1 - 1)
\]

We will define the following random variables:
\[ M_1(t, x) = N_{1,00}^{11}(t, x) + N_{1,01}^{11}(t, x) + N_1^{10}(t, x) \]
\[ M_2(t, x) = N_{2,11}^{11}(t, x) + N_{2,10}^{11}(t, x) + N_2^{10}(t, x) \]

Thus, when \( t \to \infty \), we get
\[
P(W_t \leq x) = G_1(x)P(M_1(t, x) - M_2(t, x) = n_1)
\]
\[
+ P(M_1(t, x) - M_2(t, x) \leq n_1 - 1)
\]

\[ (10.13) \]

Obviously, this is the same formula as Berg and Posner [5]), where the random Poisson variables are adjusted to the model. \( M_1(t, x) \) is the sum of three Poisson random variables. Thus, it is also distributed Poisson with parameter
\[
\lambda^{11} \int_x^{\infty} G_1(v) G_2(v) dv + \lambda^{11} \int_x^{\infty} G_1(v) G_2(v) dv + \lambda^{10} \int_x^{\infty} \overline{G}_1(v) dv = (\lambda^{11} + \lambda^{10}) \int_x^{\infty} \overline{G}_1(v) dv .
\]

Similarly, \( M_2(t, x) \) is the sum of three Poisson random variables. and is also distributed
Poisson with parameter \( \lambda^1 \int_0^x G_1(v)G_2(v)dv + \lambda^1 \int_0^x G_1(v)\overline{G}_2(v)dv + \lambda^{10} \int_0^x G_1(v)dv = \\
(\lambda^1 + \lambda^{10}) \int_0^x G_1(v)dv. \\

Due to the fact that \( M_i \) are sums of Poisson random variables, we can deal with them as if they were distributed normal as follows:

\[
M_1 - M_2 \sim N \left[ \left( \lambda^{11} + \lambda^{10} \right) \left( \frac{1}{\mu_1} - x \right) \left( \lambda^{11} + \lambda^{10} \right) \left( \frac{1}{\mu_1} - x + 2 \int_0^x G_1(v)dv \right) \right]
\]

10.2.3. When CT2 is arriving.

Analogously, the waiting time of a customer bringing only the second item will be

\[
P(W \leq x) = G_2(x)P(M_3(t, x) - M_4(t, x) = n_2) + P(M_3(t, x) - M_4(t, x) \leq n_2 - 1)
\]

where \( M_3(t, x), M_4(t, x) \) are generic Poisson variables with parameters

\[
(\lambda^{11} + \lambda^{10}) \int_x^\infty \overline{G}_2(v)dv \quad \text{and} \quad (\lambda^{11} + \lambda^{10}) \int_0^x G_2(v)dv,
\]

respectively.

10.2.4. The waiting time distribution of a random customer

In this section, we analyze the waiting time distribution of a random customer. For this purpose, we define the following random variables:

\[
D_1(x) = N_1(x) - N_2(x) \\
D_2(x) = N_1(x) - N_4(x) \\
D_3(x) = M_1(x) - M_2(x) \\
D_4(x) = M_3(x) - M_4(x)
\]
D_1(x) has mean 

\[(\lambda^{11} + \lambda^{10}) \int_x^\infty G_1(v) \, dv - (\lambda^{11} + \lambda^{10}) \int_0^x G_1(v) \, dv =

(\lambda^{11} + \lambda^{10}) \left[ \int_x^\infty (1) \, dv - \int_0^x G_1(v) \, dv \right] = (\lambda^{11} + \lambda^{10}) \left[ \int_0^x (1 - G_1(v)) \, dv \right] = (\lambda^{11} + \lambda^{01}) \left[ \frac{1}{\mu_1} - x \right].\]

D_2(x) has mean 

\[\lambda^{11} \int_0^\infty G_1(v) G_2(v) \, dv + \lambda^{10} \int_0^x G_1(v) \, dv + \lambda^{01} \int_0^\infty G_2(v) \, dv - (\lambda^{11} + \lambda^{01}) \int_0^x G_1(v) G_2(v) \, dv + \lambda^{01} \int_0^x G_2(v) \, dv + \lambda^{10} \int_0^\infty G_1(v) \, dv =

\lambda^{11} \int_0^\infty G_1(v) G_2(v) \, dv + \lambda^{01} \int_0^\infty G_1(v) \, dv + \lambda^{10} \int_0^\infty G_2(v) \, dv - (\lambda^{11} + \lambda^{01}) \int_0^x G_1(v) G_2(v) \, dv =

\lambda^{11} \left[ \int_0^\infty kdv - \int_0^\infty G_2(v) \, dv \right] - \lambda^{10} \left[ \int_0^\infty kdv - \int_0^\infty G_1(v) \, dv \right] + \lambda^{01} \left[ \int_0^\infty kdv - \int_0^\infty G_2(v) \, dv \right].

= \lambda^{01} \left[ \frac{1}{\mu_2} - x \right] - \lambda^{10} \left[ \frac{1}{\mu_1} - x \right] + \lambda^{11} \left[ \frac{1}{\mu_1} - \frac{1}{\mu_2} \right].\]

D_3(x) has mean 

\[\lambda^{11} \int_0^\infty G_1(v) \, dv - (\lambda^{11} + \lambda^{10}) \int_0^x G_1(v) \, dv = (\lambda^{11} + \lambda^{01}) \left[ \frac{1}{\mu_1} - x \right].\]

D_4(x) has mean 

\[\lambda^{11} \int_0^\infty G_2(v) \, dv - (\lambda^{11} + \lambda^{10}) \int_0^x G_2(v) \, dv =

\lambda^{11} \int_0^\infty (1) \, dv - \int_0^x G_2(v) \, dv = (\lambda^{11} + \lambda^{01}) \left[ \int_0^x (1) \, dv - \int_0^x (1 - G_2(v)) \, dv \right] = (\lambda^{11} + \lambda^{01}) \left[ \frac{1}{\mu_1} - x \right].\]

Thus, the waiting time distribution of a random customer will be given by:
10.2.5. Service Measures

As in all the models, we want to calculate the different service measures for the given system.

**The Waiting Time Distribution**

In the previous section, we developed the formula for the Waiting Time Distribution. Thus, if we want to know what is the probability that a customer waits more than time \( x \), \( P(W > x) = 1 - P(W \leq x) = 1 - F(x) \).

**The Fillrate**

As in the ERSOF-models, one of the most useful service-measures is the fillrate. In this case, it is the probability of last satisfaction, which means that it is the probability that a customer gets immediate satisfaction upon arrival; i.e., for \( x = 0 \), so that
\[ G_1(0) = G_2(0) = 0 \]. Therefore,
\[
P(W = 0) = \frac{\lambda_{11}}{\lambda} \left[ \sum_{k=-\infty}^{n_2-1} [P(D_1(0) = k)P(D_2(0) \leq n_2 - k - 1)] \right] + \frac{\lambda_{01}}{\lambda} \left[ P(D_4(0) \leq n_2 - 1) \right] + \frac{\lambda_{01}}{\lambda} \left[ P(D_3(0) \leq n_1 - 1) \right] + \frac{\lambda_{10}}{\lambda} \left[ G_2(0)P(D_2(0) = n_2 - n_1) \right]
\]

We will now analyze what happens to the random variables \( D_i(0) \).
\( D_1(0) = N_1(0) - N_2(0) \) which are distributed

\[
N_1(0) \sim \text{Poisson} \left[ (\lambda^{11} + \lambda^{10}) \int_0^\infty G_1(v)dv \right] = \text{Poisson}(\lambda^{11} + \lambda^{10}) / \mu_1
\]

\( N_2(0) \sim \text{Poisson}(0) \)

Thus, \( D_1(0) \sim \text{Poisson}(\lambda^{11} + \lambda^{10}) / \mu_1 \).

\( D_2(0) = N_3(0) - N_4(0) \) for which

\[
N_3(0) \sim \text{Poisson}(\lambda^{11} \int_0^\infty G_1(v)G_2(v)dv + \lambda^{01} / \mu_2)
\]

\[
N_4(0) \sim \text{Poisson}(\lambda^{11} \int_0^\infty G_1(v)G_2(v)dv + \lambda^{10} / \mu_1)
\]

Thus,

\[
D_2(0) \text{ has mean } \lambda^{11} \int_0^\infty G_1(v)G_2(v)dv + \lambda^{01} / \mu_2 - \lambda^{11} \int_0^\infty G_1(v)G_2(v)dv - \lambda^{10} / \mu_1
\]

\[
= \lambda^{11} \int_0^\infty G_1(v)(1 - G_2(v))dv + \lambda^{01} / \mu_2 - \lambda^{11} \int_0^\infty (1 - G_1(v))G_2(v)dv - \lambda^{10} / \mu_1
\]

\[
= \lambda^{11} \int_0^\infty G_1(v) - G_2(v)dv + \lambda^{01} / \mu_2 - \lambda^{11} \int_0^\infty (1 - G_1(v))G_2(v)dv - \lambda^{10} / \mu_1
\]

\[
= \frac{\lambda^{11}}{\mu_2} - \frac{\lambda^{11}}{\mu_1} + \frac{\lambda^{01}}{\mu_2} - \frac{\lambda^{10}}{\mu_1}
\]

\[
= (\lambda^{11} + \lambda^{01}) \frac{1}{\mu_2} - (\lambda^{11} + \lambda^{10}) \frac{1}{\mu_1}
\]

\( D_3(0) = M_1(0) - M_2(0) \) for which

\( M_1(0) \sim \text{Poisson}( (\lambda^{11} + \lambda^{10}) / \mu_1 ) \)

\( M_2(0) \sim \text{Poisson}(0) \)

Thus, \( D_3(0) \sim \text{Poisson}( (\lambda^{11} + \lambda^{10}) / \mu_1 ) \).

\( D_4(0) = M_3(0) - M_4(0) \) for which

\( M_3(0) \sim \text{Poisson}( (\lambda^{11} + \lambda^{01}) / \mu_2 ) \)

\( M_4(0) \sim \text{Poisson}(0) \)

Thus, \( D_4(0) \sim \text{Poisson}( (\lambda^{11} + \lambda^{01}) / \mu_2 ) \).
Thus, from (11.16), we get
\[
\frac{\lambda_1^{11}}{\lambda} P(D_1(0) \leq n_1 - 1)P(D_2(0) + D_1(0) \leq n_2 - 1) + \frac{\lambda_1^{10}}{\lambda} [P(D_3(0) \leq n_1 - 1)] \\
+ \frac{\lambda_1^{01}}{\lambda} [P(D_4(0) \leq n_2 - 1)]
\]

The fillrate can now be reduced to
\[
P(W = 0) = \frac{\lambda_1^{11}}{\lambda} [P(D_3(0) \leq n_1 - 1)P(D_4(0) \leq n_2 - 1)] \\
+ \frac{\lambda_1^{10}}{\lambda} [P(D_3(0) \leq n_1 - 1)] + \frac{\lambda_1^{01}}{\lambda} [P(D_4(0) \leq n_2 - 1)]
\]

(10.17)

**The Average Waiting Time**

Another important measure is the average waiting time, which can be calculated in two ways. The first way is by Little’s formula where \( \bar{W} = L / \lambda \) in which \( L \) is the Average Number in the system. But in our case, this is not known, and so we will calculate the average waiting time directly by \( \bar{W} = \int_0^\infty (1 - F(x))dx \). Obviously, from this, we will then also have the average number of customers in the system.

**The Distribution of the number of customers in the system**

In this section, we will outline a method to determine the distribution of the number of customers in the system. Here we will show the framework of how to proceed toward the correct formulas, but the work will be completed in a further research. As explained several times during this research, this was one of the motivations of this research, because Hausman and Cheung [5] failed to obtain correct formulas. Let us define some random variables which we will use in the analysis.

\( C \quad \text{The total number of customers waiting in the system.} \)
\( C_{01} \quad \text{The total number of customers of type 1 waiting in the system.} \)
The total number of CT 2 waiting in the system.

The total number of CT3 waiting in the system.

The number of items in repair facility 1 (ss1) from CT1.

The number of items in repair facility 2 (ss2) from CT2.

The number of items in repair facility 1 (ss1) from CT3.

The number of items in repair facility 2 (ss2) from CT3.

When customers enter the system and send their failed items to repair facilities ss1 and/or ss2, they may receive spares directly if available and leave, or may be required to wait for repair before leaving. Here there are items in the two repair facilities ss1 and ss2: In repair facility 1, there are i+k1 items of class 1, whereas in repair facility 2, there are j+k2 items of class 2. As used often in this thesis, we will use n1 for the number of spares in ss1 and n2 for the number of spares in ss2. In the following, we will focus our analysis on ss1, but obviously the same analysis is valid for ss2. In ss1, there are at the moment i+k1 items of class 1. Someone brought them. Some of the customers brought items and left and some of them are still waiting. In ss1 there are n1 spares of class 1; if i+k1>n1 then there are currently no spares on the shelf. But who took them? We will define m1 as the number of items of class 1 which were taken by CT3 from ss1. Thus, assuming all items are equally eligible to have received a spare immediately upon arrival at ss1, the number of available spares m1 which were taken by customers of type 3 from the n1 available has a hypergeometric distribution. Thus, we write this as

\[
P_1(m_1 \mid i, k_1) = \binom{k_1}{m_1} \binom{i}{n_1-m_1} \binom{k_1+i}{k_1+m_1}, \text{ for } m_1=0,1,2,\ldots, \min(k_1, n_1).
\]

Analogously, if m2 is the number of spares of class 2 which were taken by customers of type 3 in ss1, we have

\[
P_2(m_2 \mid j, k_2) = \binom{k_2}{m_2} \binom{j}{n_2-m_2} \binom{k_2+j}{k_2+m_2}, \text{ for } m_2=0,1,2,\ldots, \min(k_2, n_2).
\]
If CT3 took \( m_1 \) spares from the shelf in \( ss_1 \), then CT1 took the remaining \( n_1-m_1 \) spares and left the system. Also, if CT3 took \( m_2 \) spares from the shelf in \( ss_2 \), then customers of type 2 took \( n_2-m_2 \) spares and left. Thus, the other customers are still waiting. Hence,

\[
P(C^{10} = r_1 | i,k_1) = \sum_{m_1} P_{10}(r_1 = i - (n_1 - m_1) | i,k_1,m_1)P_1(m_1 | i,k_1) \quad r_1 \geq 0
\]

and

\[
P(C^{01} = r_2 | j,k_2) = \sum_{m_2} P_{01}(r_2 = j - (n_2 - m_2) | j,k_2,m_2)P_2(m_2 | j,k_2) \quad r_2 \geq 0
\]

But what is the probability of \( r_3 \) CT3 waiting? There are \( k_1-m_1 \) requests from type 3 items in \( ss_1 \) and \( k_2-m_2 \) from type 3 items in \( ss_2 \). Because of the FIFO-policy of satisfying customers, \( C^{11} = \max([k_1-m_1]^+, [k_2-m_2]^+) \equiv C^{11}(m_1,m_2,k_1,k_2) \).

Thus,

\[
P(C^{11} = r_3 | i,j,k_1,k_2) = \sum_{m_1} \sum_{m_2} P(C^{11} = r_3 | i,j,k_1,k_2,m_1,m_2)P(m_1,m_2 | i,j,k_1,k_2)
\]

where

\[
P(C^{11} = r_3 | i,j,k_1,k_2,m_1,m_2) = \begin{cases} 1 & r_3 = \max([k_1-m_1]^+, [k_2-m_2]^+) \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
P(m_1,m_2 | i,j,k_1,k_2) = P_1(m_1 | i,k_1)P_2(m_2 | j,k_2)
\]

which are determined above.

But what is the total number of customers waiting in the system?

\[C = C^{11} + C^{10} + C^{01}\]. Hence,

\[
P(C = r) = \sum_i \sum_j \sum_{k_1} \sum_{k_2} \sum_{m_1} \sum_{m_2} P(C = r | i,j,k_1,k_2,m_1,m_2) * P(i,j,k_1,k_2,m_1,m_2)
\]

where

\[
P(C = r | i,j,k_1,k_2,m_1,m_2) = \begin{cases} 1 & r = C^{10} + C^{01} + C^{11} \\ 0 & \text{otherwise} \end{cases}
\]

and

\[
P(i,j,k_1,k_2,m_1,m_2) = P(m_1,m_2 | i,j,k_1,k_2)P(i,j,k_1,k_2)
\]

\[
= P_1(m_1 | i,k_1)P_2(m_2 | j,k_2)P_{10}(i)P_{01}(k)P(k_1,k_2)
\]
where $P_{10}(i)$ is Poisson with parameter $\frac{\lambda_{10}}{\mu_1}$ and $P_{01}(j)$ Poisson with parameter $\frac{\lambda_{01}}{\mu_2}$.

$P(k_1,k_2)$ is still to do and will be done in an eventual research.

This approach helps avoid the dependencies between the subsystems. The only dependency appears in $P(k_1,k_2)$ which is the most difficult part in the analysis. But once calculated, we intend to complete the analysis of how to find the distribution of the total number of customers waiting in the system.

**The Average Number of Customers in the System**

There is no direct way to calculate the number of customers in the system. In general, we can say that the average number of customers in the system $\bar{C}$ is equal to the sum of number of customers of each type. That is, $\bar{C} = \bar{C}_{11} + \bar{C}_{01} + \bar{C}_{10}$. There are two ways to calculate the average number of customers of each type.

- Through the Average waiting Time. By Little’s formula $\bar{C}_{11} = \lambda_{11} \bar{W}_{11}$, $\bar{W}_{11}$ can be calculated using the Waiting Time Distribution. This is the numerical way.
- Through the number of customers in the system. The problem with this way is that the number of customer distribution is still not known.

In this chapter, we calculated the waiting time distribution of a customer in an ERSMF-model. We also showed a method to calculate the distribution of the number of customers in the system. This is an important novelty of this research. As explained several times in this thesis, Hausman and Cheung [5] tried to develop a formula for the number of customers in system but failed to find correct formulas. Obviously, we only showed a method to get there, and we will eventually finish this work.
11. Model Extensions

11.0. Several model extensions

Obviously, the model need not stand alone. It may be part of a bigger system. In section 11.1, we analyze what happens if there is not one repair facility, but several, which we call a multi-echelon system. Sherbrooke [20-23], Albright [2-3] and almost all researchers in the area of spares have focused their analysis on this topic and therefore, we want to add this extensions to our model. In section 11.2, we will determine under what conditions the infinite servers assumption hold.

11.1. Scrapping in the basic model

To generalize the basic model, we tried to relax the assumption of scrapping. The first part of the research deals with the option to scrap an item. Each time a customer brings a failed item to the system there is a probability $p$ that the item can’t be repaired, and thus, it is scrapped. This means that since the customer must be satisfied, the system loses an item. To maintain system performance, that is, a certain level of spares on the shelf, scrapped items must somehow be replaced. This replacement can be accomplished in two ways. General replacement policies will be examined and divided into two parts: Replace an item when it is scrapped. This policy turns out to fit into the existing model by changing the terminology of repair time to reproduction time. This means that reproducing an item can be done either by repairing it or by ordering a new one. In this way, the newly defined reproduction time distribution $G(\cdot)$ will include the time of ordering a new one. Therefore, no new analysis need be done to include this option.

Order new items at a constant rate. The number of scrapped items must equal the number ordered on average. If the system orders more new items than the number scrapped, the number of spares on the shelf of the system will grow constantly and eventually new customers will not have to wait at all. If the system orders fewer items than the number scrapped, the system will become increasingly short and it will eventually not able to provide any repaired ones. The number of customers waiting for items will increase.
beyond the total number of items in the system and the queue will explode. Thus, to maintain balance, we need to order new items at one same rate that are scrapped; the average number scrapped accumulates to \( p\lambda t \) by time \( t \), where \( p \) is the probability of scrapping an item.

For example, consider a model where customers arrive at rate \( \lambda=10 \), while the scrapping rate \( p=0.1 \). This means that 1 item will be scrapped per hour on average. By ordering 1 new item each hour or 2 every two hours, the scrapped items will be replaced.

By time \( t \) there will \( R(t) \) scrapped items, where \( R(t) \) is the amount of items scrapped in \((0,t)\). Now, \( R(t) \) is Poisson with parameter \( p\lambda t \). If \( t \) is large, then \( R(t) \) is distributed normally with mean \( p\lambda t \) and variance \( p\lambda t \). On the other hand, by time \( t \), there are precisely \( p\lambda t \) replacements. Therefore, we want to know, \( \lim_{t \to \infty} (R(t) - p\lambda t) \), that is, the net change in the number of items in the system in the limit. Since \( R(t) \) is distributed normally,

\[
Z(t) = \frac{R(t) - p\lambda t}{\sqrt{p\lambda t}} \quad \text{is } N(0,1),
\]

so that,

\[
R(t) - p\lambda t = Z(t)^* \sqrt{p\lambda t}
\]

and

\[
\lim_{t \to \infty} (R(t) - p\lambda t) = \lim_{t \to \infty} (Z(t)^* \sqrt{p\lambda t}),
\]

which is \( +\infty \), when \( Z(t) \) is positive and \( -\infty \), when \( Z(t) \) is negative.

This property was first found by simulation, where the average waiting time varied from iteration to iteration, giving indication that the system is not stable.

In conclusion, policy 2 cannot be applied and only policy 1 is used. Thus, the basic model need not be changed to include scrapping.

11.2. Multi-echelon systems

Most of the articles in the area of repair system deal with multi-echelon systems, due to the fact that in reality, repair facilities are distributed among different locations.
The replacement time will be the time measured from when an item leaves system 1 until its subsequent return to system 1. For example, customers arrive to facility 1, where items are to be repaired. If an item cannot be repaired there it is sent on to facilities 3, 4 or 5, where it is repaired. $p_{13}$ will be the probability of an item to be sent from facility 1 to facility 3, $p_{11}$ the probability of being repaired at facility 1. Thus, $p_{11} + p_{12} + p_{13} + p_{14} = 1$.

When an item is sent to facility 2, 3 or 4, the waiting time at the specific facility will in fact be the replacement time of the item. Thus, the repair time distribution at facility 1 is:

$$H^1(x) = p_{11}G^1(x) + p_{12}F^2(x) + p_{13}F^3(x) + p_{14}F^4(x)$$

where $G^i(x)$ is the repair time distribution at facility $i$ and $F^i(x)$ is the waiting time distribution at facility $i$. Using this method, we can analyze and solve large multi-echelon systems separately.

### 11.3. Ample servers

One of the most important assumptions of our model is that of ample servers. This means that there is always a server available when a customer arrives. But, in reality there are no ample server systems. Each server costs money. So what is the minimum
number of servers so that we can define the system as an ample-server system? The number of servers in \( s_i \) is Poisson distributed with parameter \( \frac{\lambda_i}{\mu_i} \) and thus as a practical rule for each \( s_i \) we may want to know how many servers are needed to handle, say 95% of the demand on that server immediately. So what is the number of servers in each \( s_i \), \( v_i \) so that we can define our system an ample server assumption?

The probability that there are \( v_i \) servers occupied in \( s_i \) is

\[
\frac{(\lambda_i / \mu_i)^{v_i}}{v_i!} e^{-(\lambda_i / \mu_i)}, \text{ for } i = 1, \ldots, I. \tag{11.1}
\]

Thus, the probability that \( v_i \) servers will be sufficient is

\[
\sum_{r=0}^{v_i} \frac{(\lambda_i / \mu_i)^r}{r!} e^{-(\lambda_i / \mu_i)}, \text{ for } i = 1, \ldots, I. \tag{11.2}
\]

Thus, if each \( s_i \) satisfies the condition that \( \sum_{r=0}^{v_i} \frac{(\lambda_i / \mu_i)^r}{r!} e^{-(\lambda_i / \mu_i)} \geq 0.95 \), we can define our system as satisfying an ample server assumption. For the case that we cannot use the ample server assumption, we will need different formulas (see Berg and Posner [6]).
12. Summary and further research

In this thesis, we wanted to analyze a variety of exchangeable repair systems with spares. Obviously, we did not deal with ALL systems, but with some specific systems: The ERSOF-system and the ERSMF-system. For the ERSOF-system, there are a variety of models which were analyzed and presented in this research.

- **The basic optimization model**: The basic optimization model optimizes a service criterion under a budget constraint. As in every laboratory, we want to get the maximum out of our investment. We solved this model for the average waiting time as the objective function, including post-optimality analysis and after solving, we showed via an example how to get all the relevant output data from the model. In fact, we showed that when the optimal spares vector is determined, everything can be known from our model. Several methods such as new algorithms as well as dynamic programming are used when the fillrate and the total number of customers in system are described as the objective function.

- **Multiple constraints**: Our basic optimization model may not always satisfy the requests of the manager. In many cases, a manager may want to solve a model which includes multiple constraints. Mathematical proofs presented in this thesis lead to an algorithm which reduces, without loss of generality, a model with multiple constraints to the form of our basic optimization model.

- **The Dual model**: The basic optimization model has a service criterion as objective function and available budget as constraint. But what happens if the manager wants to optimize his budget allocation when a given service level is prescribed? A dual model of the basic optimization model is presented, including the average waiting time and the fillrate as constraints, when the budget required serves as the objective function. To show how we are able to manipulate all different optimization models, multiple service criteria based constraints are also added.

- **Optimization model with only one objective function**: In certain cases the basic optimization model may be inadequate. The manager may want to optimize a certain service criterion, but does not want to overspend. Every customer not
satisfied will incur a penalty and the goal is then to minimize the total cost of the system. Using this technique, we deal with one objective function without constraints.

- **Multiple Goals:** One of the objectives of this research was to analyze models containing multiple goals. In the literature, multiple goals are mentioned, but never in relation to spares provisioning. Using several techniques to combine the objectives, including Goal Programming, we show that various models with multiple objective functions can be solved.

- **Integer Programming:** Initially used to validate the results developed in this thesis, Integer Programming quickly demonstrates that it may serve as a method to solve problems directly. Obviously the problem is NP-Hard and thus, it is a tool for small size problems. Combining tools such as Visual Basic and the Lindo Solver for Excel, we solved difficult problems such as the conditional waiting time for an objective function. We demonstrated a method on how to transform a model with non-linear objective function into a linear binary integer problem which can be solved by a standard Solver such as Lindo. Extensions such as discounting were also introduced into this model.

- **Bulk:** One important extension made in the basic model is to accommodate bulk arrivals. A customer arrives bringing a group of items. Until now, the waiting time distribution for the customer was not known. In this thesis, we developed the waiting time distribution for customers bringing a group of items. Obviously, this extension can be easily introduced into the ERSOF-optimization model.

The second class of system analyzed in this research was called ERSMF-system. A customer brings a product which can fail via more than one item type. We developed formulas for the waiting time distribution and also a method to get the total number of customers waiting in the queue.

Finally, we relaxed other important assumptions. We showed how scrapping and multi-echelon systems can be introduced into the basic model.
Obviously, this research is not finished. There are multiple directions which can be explored further. We tried to present the background of a field which seemed to be solved by major researchers such as Hausman and Cheung, but which turned out to be incorrect. This shows how complex and how easy it is to make errors exploring this part of spares research. As explained in the introduction, spares provisioning is an important aspect in today’s repair policy development. Further research is planned in the following areas:

- The Hausman and Cheung problem. As mentioned several times in this thesis, Hausman and Cheung failed to create correct formulas for an ERSMF-model. We outlined a method to solve it, but didn't pursue it to completion. Thus, in a further research, we will provide correct formulas.

- We provided formulas for the ERSMF-model, but did not actually analyze any specific optimization models. This will be done in a further research, because ERSMF-models are more realistic than ERSOF-models.

- All the ERSMF-models in this work contain exactly two classes. But what happens when more classes are available. What are the formulas or what are the methods to calculate the service measures? This is still open.

- In this thesis, we presented a new method of incorporating integer programming to solve different nonlinear models. The problem is known to be NP-Hard. Thus, what are its limitations? Are there limitations with today's computers?

- In this thesis, the entire available budget went into spares. But in a system there may be other channels in which to invest: For example people, repair tools, etc. . In the literature, these other channels weren't considered. But in reality, managers do not only invest in spares but also in the factory, in modern repair tools and in training people. How does that affect the optimization model? Is there an optimization model which includes all these different aspects?

To summarize, we can say that important developments in research of spares provisioning were presented in this research. It showed different methods of how to deal with complicated models, either analytically, or in an applied using approximation and heuristic algorithms, integer programming or dynamic programming. In the end, the combination of mathematical tools and the computer made this research a success.
Notation summary

\[ c_i = \text{Cost of one item of class } i. \]

\[ F(t \mid \bar{n}) = \text{Waiting time distribution of a customer under spares configuration } \bar{n}. \]

\[ F_i(t \mid n_i) = \text{Waiting time distribution at ss}_i \text{ given } n_i \text{ spares.} \]

\[ FR_{\bar{n}} = \text{The fillrate of the system: The probability that a random customer gets his product repaired straight away.} \]

\[ FR_{i,n_i} = \text{The fillrate at ss}_i \text{: The probability that there is no waiting time at ss}_i. \]

\[ FR_{F,j_i,\bar{n}} = \text{The first fillrate of a specific customer: The probability that a specific customer gets at least one item repaired without waiting.} \]

\[ FR_{L,j_i,\bar{n}} = \text{The last fillrate of a specific customer: The probability that a specific customer gets his product repaired without waiting.} \]

\[ FR_{F,\bar{n}} = \text{The first fillrate of a random customer: The probability that a random customer gets at least one item repaired without waiting.} \]

\[ FR_{L,\bar{n}} = \text{The last fillrate of a specific customer: The probability that a specific customer gets his product repaired without waiting.} \]

\[ G(.) = \text{The cumulative repair time distribution.} \]

\[ G_{s_i}(.) = \text{The cumulative repair time distribution of ss}_i. \]

\[ I = \text{Number of different item classes in the system.} \]

\[ j_s = \text{A specific product containing different failed items.} \]

\[ J = \text{The set of all possible products. There are } 2^I-1 \text{ different possible products.} \]

\[ L_n, L_{\bar{n}} = \text{The average number of customers in a steady-state system with } n, \bar{n} \text{ spares.} \]

\[ L_{i,n_i} = \text{The average number of customers at ss}_i \text{ containing } n_i \text{ spares.} \]

\[ M = \text{The budget at our disposal to spend on spares.} \]

\[ n,\bar{n} = \text{The number of spares in the system. } \bar{n} = (n_1, n_2, n_3, \ldots n_I) \]

\[ N_i = \text{Number of spares in ss}_i \text{ (i=1,2,\ldots I)} \]
\[ T_{i,n_i} = \text{Waiting Time of a customer at ss}_i \text{ given } n_i \text{ spares.} \]
\[ \overline{T}_{i,n_i} = \text{The average waiting time of ss}_i \text{ given } n_i \text{ spares.} \]
\[ W_n, W_{\bar{n}} = \text{The steady-state waiting time of a customer in a system with } n, \bar{n} \text{ spares.} \]
\[ \overline{W}_n, \overline{W}_{\bar{n}} = \text{The average waiting time of a customer in a steady-state system with } n, \bar{n} \text{ spares.} \]
\[ W_{F,j,\bar{n}} = \text{First satisfaction time of a specific customer. The customer waits until he gets at least one item.} \]
\[ W_{F,\bar{n}} = \text{The first satisfaction time of a random customer.} \]
\[ W_{L,j,\bar{n}} = \text{Last satisfaction time of a specific customer. The customer waits until he gets all the items.} \]
\[ W_{L,\bar{n}} = \text{The last satisfaction time of a random customer.} \]
\[ \overline{W}_{F,\bar{n}} = \text{The expected first satisfaction time.} \]
\[ \overline{W}_{L,\bar{n}} = \text{The expected last satisfaction time.} \]
\[ X_n = \text{The steady-state number of customers in a system with n spares.} \]
\[ X_j = \text{The steady-state number of customers in ss}_i. \]
\[ Y_n = \text{The number of items in the repair facility in a system with n spares.} \]
\[ \lambda = \text{The arrival rate of customers to the system.} \]
\[ \lambda_i = \text{The rate of arrival of failed items at ss}_i, i=0,1,\ldots, I \]
\[ 1/\mu = \text{The mean repair time of a server.} \]
\[ 1/\mu_i = \text{The mean repair time of ss}_i. \]
Appendix A: Research to solve the ERSOF model using Normal Approximation

As an example, let us consider a system with only two sub-systems with spares levels $x_1$ and $x_2$.

Thus,

$$c_1 x_1 + c_2 x_2 = M,$$

$$\frac{a_1 \bar{T}_1(x_1)}{c_1} = \frac{a_2 \bar{T}_2(x_2)}{c_2}.$$

Since $a_i \ast \bar{T}_i(n_i) = \frac{L_i(n_i)}{\lambda} = \frac{-\sum_{k=1}^{n_i} (\frac{\lambda}{\mu_1})^k e^{-\int \frac{\lambda}{\mu_1} \lambda}}{(k + n_i)!}.$

Thus,

$$\left[ \sum_{k=1}^{n_1} \frac{(\frac{\lambda}{\mu_1})^k}{(k + n_1)!} e^{-\int \frac{\lambda}{\mu_1} \lambda} \right] = \left[ \sum_{k=1}^{n_2} \frac{(\frac{\lambda}{\mu_2})^k}{(k + n_2)!} e^{-\int \frac{\lambda}{\mu_2} \lambda} \right].$$

This specific two problem can be solved by enumeration, but we used the approximation of the Normal distribution. The Normal distribution with mean $(\frac{\lambda}{\mu_1})$ and standard deviation $\sqrt{(\frac{\lambda}{\mu_1})}$ is a good approximation for the Poisson distribution with rate $(\frac{\lambda}{\mu_1})$ when $(\frac{\lambda}{\mu_1})>4$.

Thus,

$$P_{Normal}(X > x_i) = \frac{c_1}{c_2} * P_{Normal}(X > x_2),$$

$$x_2 = \frac{M - c_1 \ast x_1}{c_2}.$$

where $X$ is a random variable distributed normally.

Thus,
\[ P_{\text{Normal}}(Z > \frac{x_1 - (\lambda_1 / \mu_1)}{\sqrt{(\lambda_1 / \mu_1)}}) = \frac{c_1}{c_2} P_{\text{Normal}}(Z > \frac{M - c_1 \times x_1 - (\lambda_2 / \mu_2)}{\sqrt{(\lambda_2 / \mu_2)}}) \]

An approximation for the cumulative normal distribution was developed by Haim Shore [29]:
\[ P = 1 - \text{Exp}\{ - \ln(2) \exp\{B \exp(C z - 1) + D z\}\} , \]
with \( B = -1.8128, C = -0.472245, D = 0.294549 \).

Thus,
\[ \text{Exp}\{ - \ln(2) \exp\{B \left( \exp(C z - 1) - 1\right) + D z\}\} \]
\[ = \frac{c_1}{c_2} \text{Exp}\{ - \ln(2) \exp\{B \left( \exp\left( \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}} \right) - 1\right) + D \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}}\}\} \]
should be solved for \( x_i \).

Solving, we obtain:
1)  
\[ - \ln(2) \exp\{B \left( \exp\left( \frac{x_1 - (\lambda_1 / \mu_1)}{\sqrt{(\lambda_1 / \mu_1)}}\right) - 1\right) + D \frac{x_1 - (\lambda_1 / \mu_1)}{\sqrt{(\lambda_1 / \mu_1)}}\} \]
\[ = - \ln(2) \exp\{B \left( \exp\left( \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}}\right) - 1\right) + D \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}}\} + \ln\left(\frac{c_1}{c_2}\right) \]

2)  
\[ \exp\{B \left( \exp\left( \frac{x_1 - (\lambda_1 / \mu_1)}{\sqrt{(\lambda_1 / \mu_1)}}\right) - 1\right) + D \frac{x_1 - (\lambda_1 / \mu_1)}{\sqrt{(\lambda_1 / \mu_1)}}\} \]
\[ = \exp\{B \left( \exp\left( \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}}\right) - 1\right) + D \frac{c_2}{\sqrt{(\lambda_2 / \mu_2)}}\} - \frac{\ln(c_1)}{\ln(2)} \]

Obviously, this equation cannot yield a closed form solution for \( x \) and therefore heuristics are required.
Appendix B: Using the software developed for the basic model

The following appendix helps the reader to understand how to use the software interface. When starting the software, a window is opened which defines the data of the given model. Figure B.1 shows 3 columns:

1. The Name and the ID of the item
2. The data for the repair facility: The distribution, the mean and the standard deviation.
3. The failure rate of the products.

After defining the data the software starts and the data is presented on the screen. Figure B.2 shows the screen. On the left, different curves are presented which can be changed using F, f –buttons.
Figure B.2: The interface of the software

On the right, the data of item is presented. It includes all data of the input of the item, the fillrate, the average waiting time and the average number backlogged. On the menu, choose the optimization to choose the optimized service level. 5 different options are presented to the user. Choose one. We will choose the fillrate as an example
Figure B.3: Dialog to choose the fillrate as the optimization mode

Figure B.3 shows the window of the fillrate. Choose the values to optimize (e.g.: 0.95)

The results, the user can see in the result-matrix.

**Results of Sensitivity Analysis**

<table>
<thead>
<tr>
<th>Fillrate</th>
<th>Optimal Number [based on Optimization]</th>
<th>Mean Repair Time [hours]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>10</td>
<td>2.5</td>
</tr>
<tr>
<td>0.75</td>
<td>6</td>
<td>3.75</td>
</tr>
<tr>
<td>0.9</td>
<td>8</td>
<td>4.5</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>1.1</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>1.2</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>1.5</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure B.4: The result-matrix
The optimal number to choose is 10. The result matrix offers a sensitivity analysis for different values of arrival rate and the mean repair time.

The software adds other features to make the software user-friendly, but will not included here.
Appendix C: The error of Hausman and Cheung

On page 174 of [16], Hausman and Cheung assumed that $P(Q \leq m) = P(\max_{i} [O_i - S_i]^+ \leq m)$, where $Q$ is the number of backlogged jobs in steady state, $O_i$ is the number of parts i in repair in steady state and $S_i$ is the stocking level of part i.(Spares). Having item class 1 and item class 2 in repair, it is impossible to know if there are 2 jobs waiting, one for class 1 and one for class 2, or one job which waits for both items. In fact, the number of backlogged jobs $Q$ is between $\max_{i} [O_i - S_i]^+$, when all the jobs contain all classes of items, and $\sum_{i} [O_i - S_i]^+$, when each job contains exactly one class of item.
Appendix D: Simulating the numerical result of an ER Sof-model.

ARENA Simulation Results
bgui

Summary for Replication 1 of 1

Project: Unnamed Project
Analyst: bgui

Run execution date: 7/22/2004
Model revision date: 7/22/2004

Replication ended at time: 1000.0 Hours
Base Time Units: Hours

TALLY VARIABLES

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Average</th>
<th>Half Width</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time in System</td>
<td>.36806</td>
<td>.00841</td>
<td>.00000</td>
<td>1.8979</td>
<td>68224</td>
</tr>
<tr>
<td>Match 12 Q2.WaitingTime</td>
<td>.02221</td>
<td>.00707</td>
<td>.00000</td>
<td>2.6307</td>
<td>4962</td>
</tr>
<tr>
<td>Match 11 Q1.WaitingTime</td>
<td>.11366</td>
<td>.01229</td>
<td>.00000</td>
<td>1.0427</td>
<td>5924</td>
</tr>
<tr>
<td>Match 11 Q2.WaitingTime</td>
<td>.26091</td>
<td>.02672</td>
<td>.00000</td>
<td>2.3676</td>
<td>5924</td>
</tr>
<tr>
<td>Match 10 Q1.WaitingTime</td>
<td>.17006</td>
<td>.00587</td>
<td>.00000</td>
<td>.60054</td>
<td>4879</td>
</tr>
<tr>
<td>Match 10 Q2.WaitingTime</td>
<td>.08144</td>
<td>.00638</td>
<td>.00000</td>
<td>1.4892</td>
<td>4879</td>
</tr>
<tr>
<td>Match 9 Q1.WaitingTime</td>
<td>.80016</td>
<td>.02884</td>
<td>.00000</td>
<td>1.8549</td>
<td>4896</td>
</tr>
<tr>
<td>Match 9 Q2.WaitingTime</td>
<td>.02545</td>
<td>.00968</td>
<td>.00000</td>
<td>1.7485</td>
<td>4896</td>
</tr>
<tr>
<td>Match 8 Q1.WaitingTime</td>
<td>.10310</td>
<td>.01033</td>
<td>.00000</td>
<td>.85614</td>
<td>4997</td>
</tr>
<tr>
<td>Match 8 Q2.WaitingTime</td>
<td>.30528</td>
<td>.02769</td>
<td>.00000</td>
<td>2.7641</td>
<td>4997</td>
</tr>
<tr>
<td>Match 7 Q1.WaitingTime</td>
<td>.08843</td>
<td>.01338</td>
<td>.00000</td>
<td>1.3685</td>
<td>6030</td>
</tr>
<tr>
<td>Match 7 Q2.WaitingTime</td>
<td>.40707</td>
<td>.03940</td>
<td>.00000</td>
<td>2.6683</td>
<td>6030</td>
</tr>
<tr>
<td>Match 6 Q1.WaitingTime</td>
<td>.82421</td>
<td>.03135</td>
<td>.00000</td>
<td>1.8979</td>
<td>5004</td>
</tr>
<tr>
<td>Match 6 Q2.WaitingTime</td>
<td>.01949</td>
<td>.00640</td>
<td>.00000</td>
<td>1.8388</td>
<td>5004</td>
</tr>
<tr>
<td>Match 5 Q1.WaitingTime</td>
<td>.11183</td>
<td>.01515</td>
<td>.00000</td>
<td>.97642</td>
<td>5837</td>
</tr>
<tr>
<td>Identifier</td>
<td>Average</td>
<td>Half Width</td>
<td>Minimum</td>
<td>Maximum</td>
<td>Final Value</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------</td>
<td>-----------</td>
<td>---------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>NQ(Match 13 Q1)+NQ(Match 12 Q1)+NQ(Match 1)</td>
<td>25.119</td>
<td>.73783</td>
<td>.00000</td>
<td>73.000</td>
<td>22.000</td>
</tr>
<tr>
<td>Match 12 Q2.NumberInQueue</td>
<td>.11182</td>
<td>.03177</td>
<td>.00000</td>
<td>6.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>Match 11 Q1.NumberInQueue</td>
<td>.67335</td>
<td>.08235</td>
<td>.00000</td>
<td>10.000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 11 Q2.NumberInQueue</td>
<td>1.5469</td>
<td>.13513</td>
<td>.00000</td>
<td>8.0000</td>
<td>4.0000</td>
</tr>
<tr>
<td>Match 10 Q1.NumberInQueue</td>
<td>.83020</td>
<td>.04599</td>
<td>.00000</td>
<td>7.0000</td>
<td>2.0000</td>
</tr>
<tr>
<td>Match 10 Q2.NumberInQueue</td>
<td>.39735</td>
<td>.02563</td>
<td>.00000</td>
<td>2.0000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 9 Q1.NumberInQueue</td>
<td>3.9193</td>
<td>.24872</td>
<td>.00000</td>
<td>16.000</td>
<td>5.0000</td>
</tr>
<tr>
<td>Match 9 Q2.NumberInQueue</td>
<td>.12461</td>
<td>.04257</td>
<td>.00000</td>
<td>5.0000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 8 Q1.NumberInQueue</td>
<td>.51522</td>
<td>.06459</td>
<td>.00000</td>
<td>9.0000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 8 Q2.NumberInQueue</td>
<td>1.5255</td>
<td>.09872</td>
<td>.00000</td>
<td>6.0000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 7 Q1.NumberInQueue</td>
<td>.53323</td>
<td>.10268</td>
<td>.00000</td>
<td>13.000</td>
<td>.00000</td>
</tr>
<tr>
<td>Match 7 Q2.NumberInQueue</td>
<td>2.4559</td>
<td>.17058</td>
<td>.00000</td>
<td>11.000</td>
<td>4.0000</td>
</tr>
<tr>
<td>Match 6 Q1.NumberInQueue</td>
<td>4.1289</td>
<td>.27371</td>
<td>.00000</td>
<td>16.000</td>
<td>6.0000</td>
</tr>
</tbody>
</table>
Match 6 Q2.NumberInQueue    .09752   .02800    .00000   5.0000     .00000
Match 5 Q1.NumberInQueue    .65272   .10601    .00000  11.0000    .00000
Match 5 Q2.NumberInQueue    1.6419    .15661    .00000  8.0000     .00000
Match 4 Q1.NumberInQueue    .82045   .05030    .00000  8.0000     2.0000
Match 4 Q2.NumberInQueue    .38861   .02230    .00000  2.0000     .00000
Match 3 Q1.NumberInQueue    4.1432    .26754    .00000 17.0000    1.0000
Match 3 Q2.NumberInQueue    .08676   .02081    .00000  4.0000     .00000
Match 2 Q1.NumberInQueue    .47545   .06665    .00000  9.0000     .00000
Match 13 Q1.NumberInQueue   3.9451    .24568    .00000 17.0000    1.0000
Match 2 Q2.NumberInQueue    1.5121    .11257    .00000  6.0000     3.0000
Match 1 Q1.NumberInQueue    .43916   .07758    .00000 10.0000    6.0000
Match 13 Q2.NumberInQueue   .11008   .02628    .00000  4.0000     1.0000
Match 12 Q1.NumberInQueue   4.0434    .22393    .00000 15.0000    1.0000
Match 1 Q2.NumberInQueue    2.5768    .18143    .00000 10.0000    .00000

OUTPUTS

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.NumberOut</td>
<td>1.3645E+05</td>
</tr>
</tbody>
</table>

Simulation run time: 2.28 minutes.
Simulation run complete.
Appendix E: Proof of the waiting time distribution

In this Appendix, the waiting distribution of an ERSOF-system where every customer brings exactly 2 equivalent items of the failed class is developed. We assume that the reader is familiar with the language presented in the chapter, so we can straightly forward to prove the formula.

\begin{equation*}
W_i \leq x = \{2N(t) - S_1(t+x) - Z(t,x) \leq n - 2\}
\end{equation*}

\begin{equation*}
N_1(t,x) - N_2(t,x) \leq n + Z(t,x) - 2
\end{equation*}

\begin{equation*}
F_t(x) = P(W_i \leq x) =
\end{equation*}

\begin{equation*}
P(Z(t,x) = 0)P(N_1(t,x) - N_2(t,x) \leq n - 2)
+ P(Z(t,x) = 1)P(N_1(t,x) - N_2(t,x) \leq n - 1)
+ P(Z(t,x) = 2)P(N_1(t,x) - N_2(t,x) \leq n)
\end{equation*}

\begin{equation*}
= [1 - G(x)]^2 P(N_1(t,x) - N_2(t,x) \leq n - 2)
+ 2G(x)[1 - G(x)]P(N_1(t,x) - N_2(t,x) \leq n - 1)
+ G(x)^2 P(N_1(t,x) - N_2(t,x) \leq n)
\end{equation*}

\begin{equation*}
= P(N_1(t,x) - N_2(t,x) \leq n - 2)[1 - G(x)]^2
+ 2G(x)[1 - G(x)]G(x)^2
+ G(x)^2\end{equation*}

\begin{equation*}
= P(N_1(t,x) - N_2(t,x) = n - 1)[2G(x)[1 - G(x)] + G(x)^2]
+ P(N_1(t,x) - N_2(t,x) = n)[G(x)^2]
\end{equation*}

\begin{equation*}
= P(N_1(t,x) - N_2(t,x) = n - 2)[1 + G(x)^2 - 2G(x) + 2G(x) - 2G(x)^2 + G(x)^2]
+ P(N_1(t,x) - N_2(t,x) = n - 1)[2G(x) - 2G(x)^2 + G(x)^2]
+ P(N_1(t,x) - N_2(t,x) = n)[G(x)^2]
\end{equation*}

\begin{equation*}
= P(N_1(t,x) - N_2(t,x) \leq n - 2)
+ P(N_1(t,x) - N_2(t,x) = n - 1)[2G(x) - G(x)^2]
+ P(N_1(t,x) - N_2(t,x) = n)[G(x)^2]
\end{equation*}

\begin{equation*}
F_t(x) \equiv P(W_i \leq x) =
\end{equation*}

\begin{equation*}
P(D(t,x) \leq n - 2) + P(D(t,x) = n - 1)[2G(x) - G(x)^2]
+ P(D(t,x) = n)[G(x)^2]
\end{equation*}
Appendix F: Proof of the Waiting Time Distribution of a ERSMF-model

In this appendix, we prove the formula of the waiting time distribution of an ERSMF-model where every customer brings exactly two items from two different classes. Like Appendix E, we assume that the reader is familiar with the notation of this research and thus, we want straightforward to develop the formula.

\[ P(W_t \leq x) = \]

\[ \overline{G}_1(x)G_2(x) \left\{ \begin{array}{l} N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \leq n_1 - 1 \\ N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \end{array} \right\} \]

\[ + \overline{G}_1(x)G_2(x) \left\{ \begin{array}{l} N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \leq n_1 - 1 \\ N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 \end{array} \right\} \]

\[ + G_1(x)G_2(x) \left\{ \begin{array}{l} N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \leq n_1 \\ N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \end{array} \right\} \]

\[ + G_1(x)G_2(x) \left\{ \begin{array}{l} N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) \leq n_1 \\ N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 \end{array} \right\} \]

\[ = \overline{G}_1(x)G_2(x) \sum_{k=-\infty}^{n_1-1} P(N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k) \]

\[ P \left( N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \]

\[ + \overline{G}_1(x)G_2(x) \sum_{k=-\infty}^{n_1-1} P(N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k) \]

\[ P \left( N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 \right) \]

\[ + G_1(x)G_2(x) \sum_{k=-\infty}^{n_1} P(N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k) \]

\[ P \left( N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \]

\[ + G_1(x)G_2(x) \sum_{k=-\infty}^{n_1} P(N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k) \]

\[ P \left( N_{1,00}(t, x) + N_{1,10}(t, x) - N_{2,11}(t, x) - N_{2,01}(t, x) \leq n_2 \right) \]
\[
= \overline{G}_1(x)\overline{G}_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k) 
+ P(k + N_{2,10}(t,x) - N_{1,01}(t,x) + N_{1,10}(t,x) - N_{2,01}(t,x) \leq n_2 - 1) \right] 
+ \overline{G}_1(x)G_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k) 
+ P(k + N_{2,10}(t,x) + N_{1,10}(t,x) - N_{1,01}(t,x) - N_{2,01}(t,x) \leq n_2 - 1) \right] 
+ G_1(x)\overline{G}_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}(t,x) + N_{1,10}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k) 
+ P(k + N_{2,10}(t,x) - N_{1,01}(t,x) + N_{1,10}(t,x) - N_{2,01}(t,x) \leq n_2 - 1) \right] 
+ G_1(x)G_2(x) \sum_{k=-\infty}^{n-1} \left[ P(N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = k) 
+ P(k + N_{2,10}(t,x) + N_{1,10}(t,x) - N_{1,01}(t,x) - N_{2,01}(t,x) \leq n_2 - 1) \right] 
+ \overline{G}_1(x)\overline{G}_2(x)P(N_{1,00}(t,x) + N_{1,01}(t,x) - N_{2,11}(t,x) - N_{2,10}(t,x) = n_1) 
P(n_1 + N_{1,10}(t,x) + N_{2,10}(t,x) - N_{1,01}(t,x) - N_{2,01}(t,x) \leq n_2 - 1) 
\]
\[
G_2(x) \sum_{k=-\infty}^{n_2-1} \left[ P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k \right) \\
+ P\left( k + N_{2,10}(t, x) - N_{1,01}(t, x) + N_{1,10}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \right]
\]

\[
+ G_2(x) \sum_{k=-\infty}^{n_2-1} \left[ P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k \right) \\
+ P\left( k + N_{2,10}(t, x) + N_{1,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \right]
\]

\[
+ G_1(x) P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = n_1 \right)
\]

\[
G_2(x) \left[ P\left( n_1 + N_{2,10}(t, x) + N_{1,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \\
+ P\left( n_1 + N_{2,10}(t, x) + N_{1,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) = n_2 \right) \right]
\]

\[
+ (1 - G_2(x)) P\left( n_1 + N_{1,10}(t, x) + N_{2,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right)
\]

\[
= \sum_{k=-\infty}^{n_2-1} \left[ P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k \right) \\
+ P\left( k + N_{2,10}(t, x) - N_{1,01}(t, x) + N_{1,10}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right) \right]
\]

\[
+ G_2(x) \sum_{k=-\infty}^{n_2-1} \left[ P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = k \right) \\
+ P\left( k + N_{2,10}(t, x) + N_{1,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) = n_2 \right) \right]
\]

\[
+ G_1(x) P\left( N_{1,00}(t, x) + N_{1,01}(t, x) - N_{2,11}(t, x) - N_{2,10}(t, x) = n_1 \right)
\]

\[
G_2(x) P\left( n_1 + N_{2,10}(t, x) + N_{1,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) = n_2 \right)
\]

\[
+ P\left( n_1 + N_{1,10}(t, x) + N_{2,10}(t, x) - N_{1,01}(t, x) - N_{2,01}(t, x) \leq n_2 - 1 \right)
\]
Appendix G: Developing formula

\[ TC = \left( \sum_{i=1}^{I} b_i T_{i,n_i} \sum_{k=0}^{\alpha} \frac{1}{(1 + \frac{\alpha}{\lambda})^k} \right) + \left( \sum_{i=1}^{I} c_i n_i \right) \]

\[ = \left( \sum_{i=1}^{I} b_i T_{i,n_i} \frac{1}{(1 + \frac{\alpha}{\lambda})} \sum_{k=0}^{\alpha} \frac{1}{(1 + \frac{\alpha}{\lambda})^k} \right) + \left( \sum_{i=1}^{I} c_i n_i \right) \]

\[ = \left( \sum_{i=1}^{I} b_i T_{i,n_i} \frac{1}{(1 + \frac{\alpha}{\lambda})} \frac{1}{1 - \frac{1}{(1 + \frac{\alpha}{\lambda})}} \right) + \left( \sum_{i=1}^{I} c_i n_i \right) \]

\[ = \left( \sum_{i=1}^{I} b_i T_{i,n_i} \frac{1 + \frac{\alpha}{\lambda}}{(1 + \frac{\alpha}{\lambda})} \right) + \left( \sum_{i=1}^{I} c_i n_i \right) \]

\[ = \left( \sum_{i=1}^{I} b_i T_{i,n_i} \frac{\lambda_i}{\alpha} \right) + \left( \sum_{i=1}^{I} c_i n_i \right) \]
Appendix H: Proof of the waiting time distribution of a ERSMF-model

In this appendix, we prove the formula of waiting time distribution of an ERSMF with up to two classes - formula, when the customer brings both items.

\[
\{W_i \leq x\} = \begin{cases} 
    n_1 + S_1(t + x) + Z_{11}(t, x) \geq 1 + N_{11}(t) + N_{10}^1(t), \\
    n_2 + S_2(t + x) + Z_{11}(t, x) \geq 1 + N_{11}^1(t) + N_{01}^1(t)
\end{cases}
\]

\[
= \begin{cases} 
    n_1 + (N_{11}^1(t) - N_{100}^1(t, x) - N_{101}^1(t, x)) + N_{2,11}^1(t, x) + N_{2,10}^1(t, x) + N_{10}^1(t) - N_{10}^1(t, x) + N_{2}^1(t, x) + Z_{11}(t, x) \geq 1 + N_{11}^1(t) + N_{01}^1(t) \\
    n_2 + (N(t) - N_{100}^1(t, x) - N_{101}^1(t, x) + N_{2,11}^1(t, x) + N_{2,10}^1(t, x) + N_{01}^1(t) - N_{10}^1(t, x) + N_{2}^1(t, x) + Z(t, x) \geq 1 + N_{11}^1(t) + N_{01}^1(t)
\end{cases}
\]

\[
P(W_i \leq x) = \sum \left( \begin{array}{c}
    n_1 + N_{1,00}^1(t, x) - N_{1,01}^1(t, x) - N_{2,10}^1(t, x) + N_1^0(t) - N_2^0(t, x) \leq n_1 - 1 \\
    n_1 + N_{1,00}^1(t, x) + N_{1,10}^1(t, x) - N_{1,11}^1(t, x) - N_{2,10}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_2 - 1 \\
    n_1 + N_{2,10}^1(t, x) + N_{2,01}^1(t, x) - N_{2,11}^1(t, x) - N_{2,01}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_2 - 1 \\
    n_1 + N_{1,00}^1(t, x) + N_{1,10}^1(t, x) - N_{1,11}^1(t, x) - N_{2,10}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_1 - 1 \\
    n_1 + N_{2,10}^1(t, x) + N_{2,01}^1(t, x) - N_{2,11}^1(t, x) - N_{2,01}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_2 - 1 \\
    n_1 + N_{2,01}^1(t, x) - N_{2,11}^1(t, x) - N_{2,01}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_2 - 1 \\
    n_1 + N_{1,00}^1(t, x) + N_{1,10}^1(t, x) - N_{1,11}^1(t, x) - N_{2,10}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_1 - 1 \\
    n_1 + N_{2,10}^1(t, x) + N_{2,01}^1(t, x) - N_{2,11}^1(t, x) - N_{2,01}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_1 - 1 \\
    n_1 + N_{2,01}^1(t, x) - N_{2,11}^1(t, x) - N_{2,01}^1(t, x) + N_1^0(t, x) - N_2^0(t, x) \leq n_1 - 1 \\
\end{array} \right) 
\]

168
\[
\begin{align*}
&= \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = k\right) \\
&+ P\left(k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) \leq n_2 - 1\right) \right] \\
&+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = k\right) \\
&+ P\left(n_1 + k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) = n_2 \right) \right] \\
&+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = n_1\right) \\
&+ P\left(n_1 + k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) \leq n_2 - 1\right) \right]
\end{align*}
\]

Thus, we get when \( t \to \infty \), we get

\[
\begin{align*}
P(W \leq x) &= \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = k\right) \\
&+ P\left(k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) \leq n_2 - 1\right) \right] \\
&+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = k\right) \\
&+ P\left(n_1 + k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) = n_2 \right) \right] \\
&+ G_2(x) \sum_{k=-\infty}^{n-1} \left[ P\left(N_{1,00}^{11}(t,x) + N_{1,01}^{11}(t,x) - N_{2,11}^{11}(t,x) - N_{2,10}^{11}(t,x) + N_{1,10}^{10}(t) - N_{2,10}^{10}(t,x) = n_1\right) \\
&+ P\left(n_1 + k + N_{1,10}^{11}(t,x) - N_{2,01}^{11}(t,x) + N_{1,01}^{01}(t,x) - N_{2,01}^{01}(t,x) + \\
&- N_{1,01}^{11}(t,x) + N_{2,10}^{11}(t,x) - N_{1,10}^{10}(t) + N_{2,10}^{10}(t,x) \leq n_2 - 1\right) \right]
\end{align*}
\]
Appendix I: Check mechanism for the ERMF-formula

We will now check what happen if one service time is instant; thus, take $G_2(x) = 1$. It is obvious, that in such a case, when $x \geq 0$ we don't need spares and therefore $n_2 = 0$. We will now look at any given $x \geq 0$. Thus,

$$P(W \leq x) = \sum_{k=0}^{n_1-1} \left[ P(N_2(x) = k) \right] + \sum_{k=-\infty}^{n_1-1} \left[ P(N_4(x) = -k) \right] + G_1(x)P(N_1(x) - N_2(x) = n_1) \left[ P(N_3(x) - N_4(x) = -n_1) \right]$$

Thus,

$$\sum_{k=-\infty}^{n_1-1} \left[ P(N_2(x) = k) \right]$$

$$+ G_1(x)P(N_1(x) - N_2(x) = n_1) \left[ P(N_3(x) - N_4(x) \leq -n_1) \right]$$

Now, we want to look at $N_1(x), N_2(x), N_3(x), N_4(x)$

$N_1(x)$ is distributed Poisson with parameter $(\lambda^{11} + \lambda^{10}) \int_{x}^{\infty} G_1(v)dv$

$N_2(x)$ is distributed Poisson with parameter $(\lambda^{11} + \lambda^{10}) \int_{0}^{x} G_1(v)dv$

$N_3(x)$ is distributed Poisson with parameter $(\lambda^{11} + \lambda^{10}) \int_{0}^{x} G_1(v)dv$

$N_4(x)$ is distributed Poisson with parameter $(\lambda^{11} + \lambda^{10}) \int_{x}^{\infty} G_1(v)dv$

Thus,

$$\sum_{k=-\infty}^{n_1-1} \left[ P(N_2(x) = k) \right]$$

$$+ G_1(x)P(N_1(x) - N_2(x) = n_1) \left[ P(N_3(x) - N_4(x) \leq -n_1) \right]$$

But
\[ P(N_1(x) - N_2(x) = n_1) \] \[ \times [P(N_1(x) - N_2(x) \geq n_1)] = P(N_1(x) - N_2(x) = n_1) \]

\[ P(W \leq x) = \sum_{k=-\infty}^{n_1-1} [P(N_1(x) - N_2(x) = k) \] \[ + G_1(x)P(N_1(x) - N_2(x) = n_1) \]
\[ = P(N_1(x) - N_2(x) \leq n_1 - 1) \]
\[ + G_1(x)P(N_1(x) - N_2(x) = n_1) \]

This is the Berg and Posner formula, which we expected due to the fact that the second sub-system repair is instant.

We will now check what happen if the other service time is instant,

\[ G_2(x) = \begin{cases} 
1 & x \geq \frac{1}{\mu_1} \\
0 & x < \frac{1}{\mu_1} 
\end{cases} \]

Thus,

\[ P(W \leq x) = \sum_{k=-\infty}^{n_1-1} [P(N_1(x) - N_2(x) = k) \]
\[ + G_2(x) \sum_{k=-\infty}^{n_1-1} [P(N_1(x) - N_2(x) = k) \]
\[ \times [P(N_3(x) - N_4(x) \leq n_2 - k - 1)] \]
\[ + G_2(x)P(N_1(x) - N_2(x) = n_1) \]
\[ + P(N_3(x) - N_4(x) \leq n_2 - n_1 - 1) \]
\[ = \sum_{k=-\infty}^{n_1-1} [P(N_1(x) - N_2(x) = k) \]
\[ + G_2(x) \sum_{k=-\infty}^{n_1-1} [P(N_1(x) - N_2(x) = k) \]
\[ \times [P(N_3(x) - N_4(x) = n_2 - k)] \]

We defined the random variables as follow. We will now check each term to see, what will happen if \( G_2(x) = 1 \).
\[ N_1(x) = N_{1,00}^{11}(x) + N_{1,01}^{11}(x) + N_1^{10}(x) \]
\[ N_2(x) = N_{2,11}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) \]
\[ N_3(x) = N_{1,10}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) + N_1^{01}(x) \]
\[ N_4(x) = N_{2,01}^{11}(x) + N_{1,01}^{11}(x) + N_2^{01}(x) + N_1^{10}(x) \]

From the definitions, we know that \( N_{1,00}^{11}(x) = N_{1,01}^{11}(x) = N_1^{10}(x) = N_{2,01}^{11}(x) = 0 \)

We now check what happens to the following term.
\[
P(N_1(x) - N_2(x) = k)P(N_3(x) - N_4(x) = n_2 - k)
= P(N_{2,11}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) = -k)P(N_{1,10}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) + N_1^{01}(x) - N_2^{01}(x) = n_2 - k)
= P(N_{1,10}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x) + N_1^{01}(x) - N_2^{01}(x) = n_2 + N_{2,11}^{11}(x) + N_{2,10}^{11}(x) + N_2^{10}(x))
= P(N_{1,10}^{11}(x) + N_1^{01}(x) - N_2^{01}(x) - N_{2,11}^{11}(x) = n_2)\]

Now, let’s define the following random variables:
\[ N_1^*(x) = N_{1,10}^{11}(x) + N_1^{01}(x) \]
\[ N_2^*(x) = N_2^{01}(x) + N_{2,11}^{11}(x) \]

Thus,
\[
P(W \leq x) = P(N_1^*(x) - N_2^*(x) \leq n_2 - 1) + G_2(x)P(N_1^*(x) - N_2^*(x) = n_2)\]

Let us check how \( N_1^*(x) \) and \( N_2^*(x) \) are distributed
\[ N_1^*(x) \text{ is distributed Poisson with parameter} \]
\[ \lambda_{11} \int_x^\infty G_2(v)dv + \lambda_{01} \int_x^\infty G_2(v)dv = (\lambda_{11} + \lambda_{01}) \int_x^\infty G_2(v)dv \]
\[ N_2^*(x) \text{ is distributed Poisson with parameter} \]
\[ \lambda_{11} \int_0^x G_2(v)dv + \lambda_{01} \int_0^x G_2(v)dv = (\lambda_{11} + \lambda_{01}) \int_0^x G_2(v)dv \]

Therefore, in both cases, we in fact have not a system with two classes but a system with one class. The formula reduces to Berg and Posner’s formula of a single class.

Q.E.D
References


מטרת המחקר היא לנתח מודלים לתיקון פריטים שונים שכוללים מספר גדול של שרתים והטיפוח התוקן הכללי. המטרה של המחקר היא למצוא את מספר חלקי חילוף שמייעל פונקציה נתונה. המחקר מתמקד בחישוב התנהגות שתי מערכות שונות שהמערכת כוללת סוגים שונים של פריטים. המערכת הראשונה מניחה שהלקוח מביא מוצר (שכולל מספר רב של פריטים) אחד (ERSOF). המערכת השנייה מניחה שהלקוח מביא מוצר (שכולל מספר רב של פריטים) יותר מפריט אחד (ERSMF).

בذلك נחקרו מודלים שונים למערכת הראשונה. סוג אחד של מודל מייעל קריטריונים של רמת שירות כדוגמת fillrate (אחוז של לקוחות שמקבלים שירות ללא המתנה), זמן המתנה ממוצע, מספר ממוצע של לקוחות בשירות, ועוד. במטרהquisar את התנהגות השדרוגים we hear בתแลนด์ התוקן ואת המרחב התוקן, מצאנו מודלים שונים של אילוצים לנטישה של יוזמות.

ב군יו השני של מודל, אנחנו מייעלים את סכום הכסף המושקע בחלקי חילוף תחת אילוצים של רמות שירות שונות (GOAL PROGRAMMING). בתうこと השלישי של מודל, אנחנו מיעלים מטרות שונות תחת אילוצים שונים (GOAL PROGRAMMING).

המחקר הזה כולל פתרונות מלאים, דוגמאות שונות ומראה אלגוריתמים שונים לפתרון המודלים. אנחנו הוסיפו ניתוח רגישות העוזר להבנת המודלים. פיתחנו נוסחאות שונות למערכת השנייה כגון התפלגות זמן המתנה. כל זה נעשה למודל של שני סוגים של פריטים. המגבילה זאת ניתנת לשחרור במחקר עתידי.

הוספנו תוספות כמו הגעתה בקבוצות (BULK, זריקה), שהוסיפה כל המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקר光伏 המודלים של המחקרה ו-E C. Hausman ו-Cheung
העבודה נעשתה בהדרכת
פרופסור פוזנר
פרופסור קורח

במחלקה להנדסת תעשייה וניהול
בפקולטה להנדסה
עיבוד וידאות של מערכות לתיקון פריטים

מתקק לשק מחולק של תדרישות להלבת תחביר "דוקטור לפילוסופיה"

מאת

מיכאל דרייפוס

הוגש לסיאט אוניברסיטת בן גוריון בנגב

7/22/2015

ירшение
עיבוד נתונים של מערכות לחריגות מפרטים

നתח

מיכאל דרייפוס

7/22/2015