A HEURISTIC APPROACH BASED ON GOLDEN SECTION SIMULATION-OPTIMIZATION FOR RECONFIGURABLE REMANUFACTURING INVENTORY SPACE PLANNING

A Dissertation Presented

by

Aysegul Topcu

to

The Department of Mechanical and Industrial Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in the field of

Industrial Engineering

Northeastern University
Boston, Massachusetts

August 2009
Abstract

Manufacturing a product for the first time is generally quite different than remanufacturing the same item. The number of usable parts retrieved from returned products most often varies significantly, causing fluctuations in inventory capacity and configuration requirements. Consequently, remanufacturing requires storage designs that not only minimize warehousing space and inventory-holding costs, but also facilitate effective coordination of facilities planning and remanufacturing decisions. A well designed facility will affect the efficiency of operations and influence the operating costs and profits of a company.

The focus of this dissertation is on decision-making and modeling issues that arise in facility and warehouse designs in a remanufacturing context. Key components that decision-support systems need to address in such settings include uncertainty in yield rates and demand, reconfigurable and flexible designs, interdependencies between returned products, and type of inventory control system. In order to address the above important issues, a mixed-integer, multi-period and multi-component stochastic programming recourse (SPR) optimization model has been developed which identifies optimal schedules of internal, external, and reconfigured amounts of inventory space for a given time period. Extending the SPR model to two products with interdependent parts in a multi-period setting increases its size significantly due to the explosion of possible scenarios and the number of variables and constraints. It is recommended that heuristic methodologies be used to overcome the resulting problems of solving large combinatorial optimization models. In addition, results from these models are compared to the expected value
formulations which result in a much higher minimum cost solution, underscoring the potential value in the SPR modeling approach.

In order to better emulate a generalized remanufacturing facility with random receiving patterns, component yields, and refurbished demand over multiple time periods, a Monte Carlo simulation model has been developed. Inventory storage space capacity is reconfigured as space needs change at a specified cost following a set of reconfiguration logic rules. Finally, a heuristic approach based on a multi-dimensional golden section search algorithm is implemented to identify the optimal storage capacities and reconfiguration decisions that minimize long-term expected total costs. The computation time with the heuristic approach is successfully reduced by 97% from 49.2 hours to 83.9 minutes with a higher number of inventory capacities. In several cases, total cost with this approach tends to be only 0.67% higher, which is sufficient in practical applications. Using the heuristic approach, the savings from reconfiguration are calculated under different yield rate and cost scenarios. The results demonstrate that reconfiguration becomes very important and can save a company substantial sums when the difference in yield rates among part types is high. In addition, experimental design analysis and response surface models are used to examine the impact of each inventory storage capacity on the total cost, and to develop useful heuristics for practitioners.
Acknowledgement

To my parents for their endless love and belief in me...

A great many people have contributed to the production of this dissertation. I owe my deepest gratitude to the following people who have made its completion possible.

I would like to express my deep gratitude and thanks to Professor Thomas P. Cullinane who always expressed a high spirit for scholarship throughout the time it took me to complete this research. He continuously held me to a high research standard, encouraged and guided me to meet those standards which taught me how to do research. Without his continuous encouragement and guidance, this dissertation would not have been possible.

I am also deeply grateful to Professor James C. Benneyan for not only his guidance but also sharing his meticulous research insights and creative thinking which made this dissertation a reality. The valuable and long discussions with him improved my knowledge in the area better. His insightful comments and constructive criticisms at all stages were always perceptive, thought-provoking and helpful.

I would like to acknowledge my thanks to Professor Emanuel Melachrinoudis who has generously given his time and expertise to better my work on lingo formulations of stochastic programming section of my dissertation.
I need to express my thanks to Musa Sirma, a former graduate student in computer science at Northeastern University, whose expertise on programming helped me learn the key points in VB.NET, and provided me with some very useful insights. He spent long hours teaching me the basics I needed to develop the programming part of my research.

I am also thankful to my friends, Gulen, Aysun S., Erkan, Gokce, Zeynep, Onder and Aysun T., whose knowledge, wisdom and friendship have enlightened me through my graduate studies at Northeastern. I greatly value their support and care which helped me overcome setbacks and stay focused throughout this endeavor.

Most importantly, my parents, to whom this dissertation is dedicated, have been a constant source of love and support during the long years of my education. I would like to mention also my brother, Taskin, whose enthusiasm helped me through some very difficult times. Their support and words of encouragement helped me overcome many crisis situations during my graduate studies. I deeply appreciate their belief in me.
Table of Contents

Abstract ............................................................................................................................................ i
Acknowledgement ......................................................................................................................... iii
Table of Contents ............................................................................................................................ v
List of Figures ................................................................................................................................ix
List of Tables ...............................................................................................................................xiii
List of Abbreviations ................................................................................................................... xvi
Chapter 1 – Introduction ............................................................................................................... 17
   1.1 Overview ............................................................................................................................. 17
   1.2 Important Issues in Remanufacturing Facility Layout .................................................... 20
      1.2.1 Traditional Facility Design ....................................................................................... 20
      1.2.2 Remanufacturing Considerations ............................................................................. 22
Chapter 2 – Literature Review ...................................................................................................... 32
   2.1 Inventory Control ................................................................................................................ 32
      2.1.1 Periodic Review Models .......................................................................................... 34
      2.1.2 Continuous Review Models .................................................................................. 36
   2.2 Disassembly Operation ..................................................................................................... 38
   2.3 Material Handling Systems ............................................................................................... 39
   2.4 Facility Layout .................................................................................................................... 41
      2.4.1 Traditional Manufacturing ....................................................................................... 41
      2.4.2 Remanufacturing ..................................................................................................... 44
   2.5 Stochastic Programming and Applications on Production and Capacity Planning .......... 45
Appendix A – The Lingo® Code of the Stochastic Programming Formulations ......................... 187
Appendix B – Pseudo-code of Monte Carlo Simulation Model ................................................. 214
Appendix C – Numerical Results from DOE Analysis .............................................................. 222
List of Figures

Figure 1.1 Typical operations in manufacturing processes .......................................................... 21
Figure 1.2 Typical operations in remanufacturing processes ....................................................... 23
Figure 1.3 Typical disassembly processes .................................................................................... 24
Figure 1.4 Box-arrow modeling.................................................................................................... 25
Figure 1.5 Product breakdown ...................................................................................................... 26
Figure 1.6 Wash and clean parts ................................................................................................... 27
Figure 1.7 Inspect and test parts .................................................................................................... 28
Figure 1.8 Repair and test parts .................................................................................................... 29
Figure 3.1 Remanufacturing product flow example ..................................................................... 51
Figure 3.2 Sample of replications for inventory build of part A and B ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$) 54
Figure 3.3 Inventory builds ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$), 100 replications ....................... 54
Figure 3.4 Average number of parts in inventory ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$) ....................... 55
Figure 3.5 Maximum number of parts in inventory ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$) ............... 55
Figure 3.6 Inventory builds ($\gamma_A = .82 \gamma_B = .78; \alpha_A = .80 \alpha_B = .80$), 100 replications ............... 56
Figure 3.7 Average number of parts in inventory ($\gamma_A = .82 \gamma_B = .78; \alpha_A = .80 \alpha_B = .80$) ......... 56
Figure 3.8 Maximum number of parts in inventory ($\gamma_A = .82 \gamma_B = .78; \alpha_A = .80 \alpha_B = .80$) ......... 56
Figure 3.9 Number of remanufactured products ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$), 100 replications ...... 57
Figure 3.10 Number of remanufactured products ($\gamma_A = .82 \gamma_B = .78; \alpha_A = .80 \alpha_B = .80$), 100 replications ............................................................................................................................ 57
Figure 4.1 Stochastic programming representation assuming single-part, two-period, and constant demand.................................................................................................................... 69

Figure 4.2 Example of discretization of probability space for the number of salvaged reusable components ........................................................................................................................... 70

Figure 4.3 Yield probabilities for three-period SPR model........................................................................................................................... 84

Figure 4.4 Increase in number of constraints and variables........................................................................................................................... 93

Figure 4.5 Comparison of objective values ........................................................................................................................... 93

Figure 4.6 Comparison of run times ..................................................................................................................................................... 94

Figure 5.1 Generalized reconfigurable remanufacturing process and W.I.P. inventory........................................................................... 98

Figure 5.2 Material flows in and out of each inventory storage space ........................................................................................................................... 100

Figure 5.3 Overall optimization framework ..................................................................................................................................................... 102

Figure 5.4 Input vs. output variables ..................................................................................................................................................... 104

Figure 5.5 Simulation model inputs..................................................................................................................................................... 106

Figure 5.6 Part related inputs ..................................................................................................................................................... 106

Figure 5.7 Cost output tab ..................................................................................................................................................... 107

Figure 5.8 Arrival logic for returned products..................................................................................................................................................... 108

Figure 5.9 Disassembly and sorting process logic ..................................................................................................................................................... 109

Figure 5.10 Inspection, testing and repair logic ..................................................................................................................................................... 110

Figure 5.11 Finished parts ..................................................................................................................................................... 111

Figure 5.12 Demand fulfillment logic ..................................................................................................................................................... 112

Figure 5.13 Assembly process ..................................................................................................................................................... 114

Figure 5.14 Flowchart of multi-dimensional golden section search algorithm ........................................................................................................................... 119

Figure 5.15 Pseudo code of multi-dimensional golden section search algorithm ........................................................................................................................... 121
Figure 5.16 Pseudo code of multi-dimensional golden section search algorithm, assuming 2 inventory capacities ................................................................. 122

Figure 5.17 Example of capacity output tab ................................................................. 123

Figure 5.18 Runtime vs. number of inventories ......................................................... 123

Figure 5.19 Heuristic pseudo code with 2 inventory capacities, searching on each inventory separately rather than nested ................................................................. 124

Figure 5.20 Convergence of the heuristic approach .................................................. 125

Figure 6.1 Savings from reconfiguration vs. ratio .................................................... 140

Figure 6.2 Normal plots of the standardized effects ................................................... 143

Figure 6.3 Pareto charts of the standardized effects .................................................. 143

Figure 6.4 Residual plots for all three cases ............................................................... 146

Figure 6.5 Main effects plots for all three cases ......................................................... 146

Figure 6.6 Normal plot of the standardized effects - Low difference in yield rates ....... 151

Figure 6.7 Normal plot of the standardized effects - Moderate difference in yield rates .... 154

Figure 6.8 Normal plot of the standardized effects - High difference in yield rates ......... 157

Figure 6.9 Response surface and contour plots for total cost vs. return and new inventories .... 158

Figure 6.10 Response surface and contour plots for total cost vs. usable and disposable inventories ......................................................................................... 158

Figure 6.11 Response surface and contour plots for total cost vs. usable and finished inventories ......................................................................................... 159

Figure 6.12 Response surface and contour plots for total cost vs. return and disposable inventories ......................................................................................... 159

Figure 6.13 Response surface and contour plots for total cost vs. usable and new inventories . 160
Figure 6.14 Response surface and contour plots for total cost vs. usable and finished inventories
............................................................................................................................................. 160

Figure 6.15 Response surface and contour plots for total cost vs. disposable and new inventories
............................................................................................................................................. 161

Figure 6.16 Response surface and contour plots for total cost vs. new and finished inventories161
List of Tables

Table 3.1 Simulation Model Analysis ........................................................................................................... 53
Table 4.1 Comparison of two-period mixed-integer SPR vs. EV formulations (Recourse variables are shown in blue) .................................................................................................................. 84
Table 4.2 Possible 3-period yield scenarios .................................................................................................... 84
Table 4.3 Comparison of mixed-integer single component three-period SPR vs. EV formulations ................................................................................................................................... 86
Table 4.4 Mixed-integer single component SPR model - Summary of runtime and objective value ..................................................................................................................................... 87
Table 4.5 Mixed-integer single component EV model - Summary of runtime and objective value ..................................................................................................................................... 87
Table 4.6 Comparison of mixed-integer two-component SPR vs. EV formulations in a two-period setting ........................................................................................................................................ 88
Table 4.7 Mixed-integer two-component SPR model - Summary of runtime and objective value .................................................................................................................................... 89
Table 4.8 Mixed-integer two-component EV model - Summary of runtime and objective value 89
Table 4.9 Comparison of mixed-integer two-product with overlapping parts SPR vs. EV formulations in a two-period setting ........................................................................................................................................ 90
Table 4.10 Mixed-integer two-product with overlapping parts SPR model - Summary of runtime and objective value .................................................................................................................................... 91
Table 4.11 Mixed-integer two-product with overlapping parts EV model - Summary of runtime and objective value .................................................................................................................................... 91
Table 5.1 Variables in remanufacturing and storage processes
Table 5.2 Example 1 inputs and outputs (Cell phones, two recaptured parts)
Table 5.3 Example 2 output
Table 5.4 Lower and upper capacity bounds
Table 5.5 Uncontrollable variables
Table 5.6 Sensitivity analysis on cost inputs
Table 6.1 Values of uncontrollable variables
Table 6.2 Initial lower and upper levels for inventory capacities
Table 6.3 Cost structures of the five scenarios
Table 6.4 With reconfiguration - Low difference in yield rates
Table 6.5 No reconfiguration - Low difference in yield rates
Table 6.6 With reconfiguration - Moderate difference in yield rates
Table 6.7 No reconfiguration - Moderate difference in yield rates
Table 6.8 With reconfiguration - High difference in yield rates
Table 6.9 No reconfiguration - High difference in yield rates
Table 6.10 Reconfiguration cost by inventory type - Low difference in yield rates
Table 6.11 Reconfiguration cost by inventory type - Moderate difference in yield rates
Table 6.12 Reconfiguration cost by inventory type - High difference in yield rates
Table 6.13 Total cost summary for the three scenarios
Table 6.14 Summary of savings from reconfiguration
Table 6.15 $2^{(7-4)}$ design table - Factors and responses
Table 6.16 Estimated effects and coefficients for total cost - Low difference in yield rates
Table 6.17 Analysis of variance for total cost - Low difference in yield rates
Table 6.18 Estimated effects and coefficients for total cost - Moderate difference in yield rates
........................................................................................................................................................................... 145
Table 6.19 Analysis of variance for total cost - Moderate difference in yield rates.................... 145
Table 6.20 Estimated effects and coefficients for total cost - High difference in yield rates..... 145
Table 6.21 Analysis of variance for total cost - High difference in yield rates....................... 145
Table 6.22 2^5 design table - Factors and responses ................................................................. 148
Table 6.23 Estimated effects and coefficients - Low difference in yield rates......................... 149
Table 6.24 Analysis of variance for total cost - Low difference in yield rates....................... 150
Table 6.25 Estimated effects and coefficients - Moderate difference in yield rates................ 152
Table 6.26 Analysis of variance for total cost - Moderate difference in yield rates................. 153
Table 6.27 Estimated effects and coefficients - High difference in yield rates....................... 155
Table 6.28 Analysis of variance for total cost - High difference in yield rates...................... 156
Table 6.29 Regression analysis - Low difference in yield rates.............................................. 163
Table 6.30 Analysis of variance for total cost - Low difference in yield rates....................... 163
Table 6.31 Regression analysis - Moderate difference in yield rates.................................... 164
Table 6.32 Analysis of variance for total cost - Moderate difference in yield rates............... 164
Table 6.33 Regression analysis - High difference in yield rates.......................................... 164
Table 6.34 Analysis of variance for total cost - High difference in yield rates...................... 164
List of Abbreviations

SPR  Stochastic Programming Recourse
EV   Expected Value
PCB  Printed Circuit Board
StDev Standard Deviation
MC   Monte Carlo
Opt  Optimization
Recon Reconfigured
DOE  Design of Experiments
min. Minutes
Chapter 1 – Introduction

1.1 Overview

Over the last two decades, supply chain management has received remarkable attention from both the business world and academia. The Council of Supply Chain Management Professionals (formerly known as The Council of Logistics Management) defines supply chain management as “the process of planning, implementing and controlling the efficient, cost-effective flow of raw materials, in-process inventory, finished goods and related information from the point of origin to the point of consumption for the purpose of conforming to customer requirements.” Most research has focused on the forward movement of materials from the supplier or manufacturer to the end customer. Reverse supply chain management, however, defined as “the effective and efficient management of the series of activities required to retrieve a product from a customer and either dispose of it or recover value (Prahinski and Kocabasoglu 2006)”, has not received as much attention. According to the Reverse Logistics Executive Council, the cost of handling, transporting, and determining the disposition of returned products is $35 billion annually for U.S. firms (Meyer 1999). Reverse supply chain management includes activities such as remanufacturing, reuse, reconditioning, recycling, and repair. Remanufacturing includes the processes of disassembling a used product, assessing the condition of its components, repairing or reworking them into refurbished components to satisfy exactly the same or higher quality
standards as new components, and using these units in new product manufacture (Topcu and Cullinane 2005, Lund 1984).

Since 1955, an estimated eight percent of the U.S. gross national product has been annually spent on new facilities (Rosenblatt 1986). Tompkins et al. (2003) estimate that approximately $250 billion is spent to plan or to replan facilities in the United States. Effective facilities planning and design can reduce these expenses significantly and improve productivity, efficiency, responsiveness, and profitability. Conversely, a poor design will be costly, time-consuming, and cause delays, damaged product, congestion, and lost product.

Remanufacturing is becoming very important in the economies of the world and is attractive to both manufacturers and consumers. One reason is that rapid development in technology has resulted in increased demand for new consumer goods shortening the use time of many products, and increasing the quantity of salvageable, used and scrap products. The anticipation of environmental and end-of-life take-back laws in the United States and the European Union over the last decade also has increased the importance of remanufacturing and required businesses to manage the entire life of the product (Toffel 2003). In recent years, the environment has been seriously threatened by the continuous growth in consumer waste. According to the U.S. Environmental Protection Agency (EPA), 196 million tons of waste were generated by the United States in 1990. Growing consumer awareness and preferences for environmentally conscious products and concern about the exhaustion of nonrenewable resources also have induced many firms to design products for reuse and more environmentally benign products and processes (Canon 2005, Xerox 2005, Eastman Kodak 2005, Sharp 2001). Finally, sales
opportunities in secondary and global markets have increased revenue generation for returned products (Meyer 1999). With remanufacturing, manufacturers can realize significant savings on raw materials and the time required to obtain the component parts of a product. In summary, the major benefits of remanufacturing include significant reductions in raw materials, human resources, energy, and pollution.

Logistically, however, remanufacturing can be labor intensive and a source of high variability in the overall supply chain. Processing times often depend on the age, amount of wear, and uncertain condition of returned products. When a returned item arrives at a remanufacturing facility, the condition of its component parts usually is not known until the item is disassembled. After disassembly, parts typically are tested or inspected to determine their functionality, and then put into storage. Uncertainty as to when these parts will be needed creates uncertainty in how much storage space should be allocated for each part type, which in turn impacts the layout of a remanufacturing facility. Other factors that influence the layout are characteristics (e.g. dimensions/size, volume, weight, fragility) of the recovered and remanufactured units, work to be done on the returned products, storage requirements (e.g. temperature or humidity range), and demand for remanufactured units.

The objective of this dissertation is to provide insights into the structure of an effective coordination of facilities planning and remanufacturing decisions. Identifying a good, if not optimal, facility layout and storage space design assumes a particular relevance since the layout itself must accommodate efficient operations under varying conditions. Inherent uncertainty and variability due to (i) the number of returned products, (ii) the type and number of parts reclaimed
from each returned product, (iii) the type of processes required to remanufacture a part, (iv) the
flow of parts and materials, and (v) the demand for the remanufactured part or final product, are
always present and require designs that minimize warehouse space and inventory holding costs
while also facilitating effective coordination of facilities planning and remanufacturing
decisions.

1.2 Important Issues in Remanufacturing Facility Layout

1.2.1 Traditional Facility Design

A key attribute of a successful facilities plan is its adaptability and ability to be, or to become,
suitable for some new use. Good facilities ideally have the characteristics of flexibility,
modularity, upgradability, adaptability, and selective operability (Tompkins et al. 2003). Three
of the most popular types of layouts are called product layouts, process layouts, and hybrid
layouts. The best type of layout for a particular application depends on the volumes and types of
products to be remanufactured. Product layouts (a.k.a. production line product layouts) are
based on the operation sequence and capacity requirements of the units being produced. An
automated soft drink bottling plant is a good example of the flow in a product layout since the
routing of bottles of soda must proceed from washing to filling to capping, such that washing and
filling should be placed next to each other in the layout. This type of layout makes it easy to
decide where to locate stations, and is appropriate for high-volume, low-variety production.

While product layouts are efficient in terms of faster processing rates, lower inventories, and less
unproductive time lost to changeovers and materials handling, they are less flexible and may
require redesign especially for products with short or uncertain lives (Krajewski and Ritzman
In contrast, process layouts (a.k.a. functional layouts or job shops) group machines according to function, such that equipment of the same type are grouped together. Process layouts are best for low-volume and high-variety production, such as job shops. Hybrid layouts are a combination of product and process layouts and are found in situations where production lines are routed through automated machining and assembly cells. More information on all these types of layouts can be found in Krajewski and Ritzman (1999) and Tompkins et al. (2003).

![Figure 1.1 Typical operations in manufacturing processes](image)

The primary processes in a traditional manufacturing facility are illustrated in Figure 1.1, typically including assembly, subassembly, inspection for certification, and shipping. The first steps involve the acquisition of raw materials, components, and parts that are then assembled into
a finished product, inspected, and designated as repairable or good products. Good products are stored in warehouse facilities and shipped to customers, whereas repairable products are cycled through repair processes.

1.2.2 Remanufacturing Considerations

In contrast, in a remanufacturing facility the inputs also include components and products recovered from the field. As shown in Figure 1.2, when used products are returned from the consumer to the remanufacturing plant, they are inspected and sorted into two categories: acceptable and non-repairable products. Non-repairable products are directly disposable of. In turn, after acceptable products are disassembled, the resulting components are classified as usable, non-usable, or recycle-only parts (non-usable). Non-usable parts are directly disposed of. Recycle-only parts are sent to a third party for recycling. Usable parts are inspected and, if acceptable, cleaned, processed, and sent to a parts inventory. The traditional manufacturing processes then are carried out using the cleaned parts, new parts from the parts supplier, or some combination of both. The inspection takes place for re-certification rather than certification and therefore may be slightly different in nature. If the parts are not acceptable, meaning that they require some repair, they may first be put in inventory unrepaired. Once demand arises for these parts, they are repaired and sent to parts inventory. When replacement parts are not readily available in repaired inventory for a subsequent product being remanufactured, they are acquired from a parts supplier to complete the assembly process. These parts may be new or recycled depending upon the suppliers’ business practices.
Figure 1.2 Typical operations in remanufacturing processes
Comparing the operations in Figure 1.1 with those in Figure 1.2 clearly reveals that the disassembly of returned products increases the number of operations required to produce a saleable product. This disassembly process can be viewed as the breakdown of a product and the processing, cleaning, inspecting, testing, and repairing of the resultant parts (see Figure 1.3). When products are returned, the condition of the parts is uncertain until the products have been disassembled and tested. After cleaning and testing, parts are put into storage. At what point in time these parts will be needed usually is not precisely known. This creates uncertainty in how much inventory space should be allocated for each part type. A good storage system can significantly affect the material flow in terms of minimizing travel time between stations and material handling equipment.

Utilizing a box-arrow type of modeling methodology that is often used in process modeling, the additional processes required in the remanufacturing of products can be assessed in terms of their impact on the facilities design. An illustration of the modeling method is shown in Figure 1.4. Arrows entering the box from the top serve to represent a control on the process in the box. Arrows entering from the left represent inputs to the process under study and the arrows leaving
a box from the right denote materials or data (outputs) that have obtained from processing the inputs. Arrows entering from the bottom of the box serve to identify the mechanism or tools used to facilitate the conversion of inputs to outputs.

The product breakdown operation typically consists of receiving used products from the consumer and breaking them down into their component parts. As can be seen from Figure 1.5, the outputs from this operation consists of much more than components that are candidates for reuse in a remanufactured unit. When a consumer product that has been in use such as a washing machine, dishwasher or dryer is taken apart, there are a great many seals, insulators and other parts that cannot be recovered. These materials become scrap and must be disposed of. Another output from the product recovery operation is a report on the number of parts recovered. This information is generally entered into the materials control system data base. The first additional impact of this operation on the facility design is the need for space to accommodate the temporary storage of scrap, good components and those components needing repair. A second
impact is the need for space to perform the operations of taking the reclaimed units apart. Calculation of this floor space requires that the movement of workers and the use of tools to perform the work be taken into account. This operation will also add an additional burden to the material handling system since the reclaimed products will have to be moved into the work areas and the component parts and scrap materials transported away.

![Product breakdown diagram](image)

Figure 1.5 Product breakdown

Figure 1.6 illustrates in the box arrow format the function of washing and cleaning of parts. As illustrated in Figure 1.2, unless a part is obviously damaged, before the part can be used, it must be cleaned and tested. The cleaning process depends upon the part being recovered. For most metal parts, power washing and drying are enough. For electronic equipment, cleaning with high pressure air blasts and chemically etching the leads is adequate. Some parts that are subject to grease and oil require baths or ultrasonic cleaning.
The input to this operation is typically units that have a high probability of being recovered and remanufactured as well as the cleaning fluids and other cleaning materials. The main output from this operation is clean components. Similar to the product breakdown operation, junk and scrap are generated by this work. Also since some chemical cleaning can take place, the generation of environmentally regulated scrap is commonplace when reclaiming component electronic parts. The facility will likewise be impacted by the actual washing and cleaning operations. If solvents are involved with the cleaning operation, certain safeguards will have to be included in the physical structure of the building. For example, basins below the areas where the work is performed may be required to prevent spills of hazardous materials from entering the building or worse the soil. Hoods or ventilation systems could also be required to prevent workers from being exposed to hazardous fumes or vapors. Even when a cleaning operation involves steps as simple as power washing parts, it will require the installation of hot water, drains and drying units to be integrated into the facilities design whereas these steps would not
be required for a traditional manufacturing operation. Material handling equipment selection, equipment operation, operator training, maintenance requirements, maintenance equipment and inventory control at the washing and cleaning workstation can be rather complicated since parts that have not been cleaned must be isolated from the units that have been cleaned. The handling of parts through the cleaning and drying process can be a very costly if the process is not planned properly. Achieving effective labor utilization, providing effective space utilization, providing high inventory turnover at the workstation, preventing product contamination and providing a safe work environment are the challenges faced at this operation.

Inspecting and testing parts as illustrated in Figure 1.7 is reasonably straightforward for the work performed. The parts are moved into the inspection area of the facility, inspected and moved out of the area. The inspection processes may require some specialized equipment or the use of some specialized material such as Zyglo for testing for cracks in large metal parts. The main impact on the facility when compared to the traditional manufacturing operations is the space
consumed by the operations, the workers and the inspection tools. The added material handling operations within the inspection operation and in and out of the area where the operation is performed will require coordination and additional equipment beyond the handling requirements for traditional manufacturing.

The repair and test operation shown in the box arrow model in Figure 1.8 will only be required when a part is found to be defective but repairable during the inspect and test operations. These operations generally do not involve the buildup of any substantial levels of inventory since the parts have already been examined before reaching this stage of the remanufacturing process. However, the repair processes can involve substantial areas of floor space for some classes of products and must be carefully planned for. Since the volume of units flowing to and through this operation is normally low, the materials handling requirement is not usually very substantial.

Figure 1.8 Repair and test parts
As discussed above, remanufacturing processes typically include additional operations such as disassembly, inspection, functionality testing, and cleaning that each require some amount of storage capacity as inventory flows through the process. Typically multiple components are salvaged from each returned item, each with different yields and each having storage requirements. For example, Gupta et al (2004) describes a cellular phone remanufacturing facility in which 21 components are recaptured and refurbished (battery, printer circuit board, covers, antennas, etc). These types of systems are partially governed by several endogenous and exogenous variables, including (uncontrollable) yield rates, returned product volumes, and refurbished demand and (controllable) storage capacities for each type of part and at each location in the process. Storage demands often may be off-setting over time, between part types, and at various locations in the process.

This dissertation examines how the total inventory storage cost is affected by an effective coordination of storage space planning and reconfiguration decisions in remanufacturing systems. Chapter 2 presents an overview of the state-of-the art on inventory control, disassembly operation, material handling systems and both the manufacturing and remanufacturing facility layouts. Chapter 3 states the problem and research objectives, including a simple remanufacturing model to illustrate the main issues in remanufacturing and their impact on space requirements. In Chapter 4, a stochastic programming recourse modeling approach is discussed, and its performance is evaluated. In Chapter 5, a heuristic approach based on multi-dimensional golden section simulation-optimization is developed, applying to a generalized Monte Carlo remanufacturing inventory model. This approach identifies the optimal inventory levels and reconfiguration decisions over a user-specified number of time periods that minimizes total long-
term expected costs. Later, in Chapter 6, using the heuristic approach developed in Chapter 5, savings from reconfiguration are shown, and experimental design and regression analysis are used to illustrate the impact of the inventory capacities on the total cost.
Chapter 2 – Literature Review


2.1 Inventory Control

To come to a full understanding of inventory control and management in a remanufacturing environment, it is useful to distinguish the characteristics of a remanufacturing environment with the traditional manufacturing environment. For an analysis of traditional inventory models without return flows, one can refer to the work of Federgruen and Zheng (1992) and Zheng (1992). The most basic characteristic of a remanufacturing environment is the existence of return flows of used products. By recognizing this, the impact of product return flows on inventory management is addressed. As early as 1967, Schrady (1967) proposes an extension to classical EOQ models that includes product returns without a disposal option. He assumes constant (deterministic) demand and return rates and fixed leadtimes for external orders and recovery. He proposes a control policy with fixed lotsizes serving demand as far as possible from recovered products. Mabini et al. (1992) develop EOQ formulae for the
manufacturing/remanufacturing system with deterministic demand and lead times. This article is one of the first articles that consider multi-item inventories in the context of reverse logistics. More recently, variants to this model have been discussed by Richter (1996) that includes the disposal option. He gives expressions for the optimal control parameter values and discusses their dependence on the return rate. This literature review on remanufacturing focuses on three main areas: PUSH vs. PULL control strategies, periodic vs. continuous review models and single-component vs. multiple-component products. To make things simple, periodic/continuous review models and simple/multiple-component products under PUSH vs. PULL control strategies are further explained. With PUSH control, the timing of the remanufacturing operations is completely return driven: As soon as sufficient returned products are in remanufacturable inventory, these products are batched and pushed into the remanufacturing process (Van der Laan et al. 1999a). On the other hand, PULL control remanufacturing only starts if items are needed to satisfy demands (Kiesmuller 2003a). Therefore, the timing of the remanufacturing operations under PULL control depends on a composite of returns, future expected demands, and inventory positions (Van der Laan et al. 1999a). Furthermore, a classification of periodic review models, in which the system status is reviewed at discrete time periods vs. continuous review models, in which the system status is continuously reviewed (Van der Laan et al. 1999a) is done. In addition, a categorization of single-component products and complex hybrid systems, where products consist of multiple components is provided.

In addition to theoretical contributions to inventory control for remanufacturing, which is the main focus of this literature review, case studies have been reported on this subject. For example, Toktay et al. (2000) provides a case study on inventory management of single-use

2.1.1 Periodic Review Models

An investigation of a periodic review model for a single-component product was proposed by Simpson (1978). The timing and lotsizing of disposal, remanufacturing and outside procurements operations are controlled by a PULL control strategy. It assumes that demand and return processes are dependent and stochastic. Remanufacturable products are either remanufactured or disposed of if they are not needed. Outside procurement satisfy the demands that cannot be fulfilled from product returns. Limitations of this model are that remanufacturing and outside procurement leadtimes are assumed to be zero, and fixed manufacturing and outside procurement costs are not taken into account. Cohen et al. (1980) develop and investigate an inventory model for a single-component product in which the random returns depend explicitly on the demand stream. Like Simpson’s model, the drawback of this model is that it does not consider backorders, fixed costs and procurement leadtimes.

Inderfurth (1997) extends this model for a single-component product by accounting the non-zero leadtimes. The impact of leadtimes on the inventory control is elaborated, and it is demonstrated for which leadtime situations simple optimal policies can be derived. Kiesmuller and Van der Laan (2001) differ from Cohen et al. model by considering the backorders and the leadtimes for purchasing, respectively. They develop a periodic review inventory system for a single-component product by using a Markov-Chain approach in a finite planning horizon to determine the optimal order-up-to policy with respect to total average relevant cost. Teunter and Vlachos
(2002) analyze a hybrid production system with manufacturing and remanufacturing. Based on their results, they conclude that it is not generally necessary to include a disposal option for returned items if there are more demands than returns, and that remanufacturing is marginally profitable. Fleischmann and Kuik (2003) show that a traditional \((s, S)\) policy is optimal if demand and returns are independent, recovery has the shortest lead time of both channels, and there is no disposal option. Mahadevan et al. (2003) investigate the performance of a periodic review PUSH control system as a function of return rates, backorder costs and manufacturing and remanufacturing leadtimes, and develop approximate lower and upper bounds on the optimal solution. In their model, the decision variables for the remanufacturing facility are when to release returned products to the remanufacturing line and how many new products to manufacture.

Nakashima et al. (2002) deal with a product recovery system for a single class of product life cycle. They propose a new analytical approach to evaluate the system using a Markov chain and give numerical examples for various conditions. Nakashima et al. (2004) deal with the optimal control problem of a single-item remanufacturing system under stochastic demand. It considers two types of inventories: the actual product inventory in a factory and the virtual inventory used by consumers. The state of the remanufacturing system is defined by both inventory levels. The system obtains the optimal production policy that minimizes the expected average cost per period using the policy iteration method. However, the product life-time is not considered when modeling the system.
2.1.2 Continuous Review Models

One of the first continuous and single-component product models with PUSH control was proposed by Heyman (1977). It applies to a situation with stochastic uncorrelated demands and returns. A limitation of this model is that remanufacturing and outside procurement leadtimes are zero. Muckstadt and Isaac (1981) consider a continuous system with PUSH control that differs from Heyman. Their model applies to a situation with uncertain remanufacturing leadtimes, finite remanufacturing capacities and nonzero outside procurement leadtimes. They model the remanufacturing facility with a first-come, first-serve queueing system, and the production decision is determined by a continuous $(Q, r)$ policy, where $Q$ is the order quantity and $r$ is the reorder point. It is shown that the optimal policy parameters can be computed and the results are extended to a one warehouse N retailer situation. Extensions of the Muckstadt and Isaac model to include the disposal of returned products have been studied by Van der Laan et al. (1996).

Kelle and Silver (1989) formulate a continuous review model for a single-component product. It assumes that demand and the returns processes are totally independent; all remanufactured products are remanufactured, meaning that no disposal occurs. They determine an optimal purchasing policy for reusable containers by transferring the stochastic model into a deterministic one. Yuan and Cheung (1998) consider a model that the returns stream is dependent upon the demand stream, but without the leadtime for purchasing. An investigation of a continuous PULL control with a disposal option and random demands and returns can be found in Van der Laan et al. (1997). As one of the important results of this study, a PUSH control strategy should not be used if the return rate is large because it leads to a large average stock-on-
hand in the serviceable inventory inducing large average holding costs. This can be avoided by using a PULL control strategy, which means keeping items in the recoverable stock as long as possible and therefore, reduces holding costs. In case of low return rates the cost difference between PUSH and PULL control strategies is not worth mentioning. They determine formulas for the total system cost per unit of time and analyze the system numerically. The influence of the leadtimes on such continuous PUSH and PULL control strategies is studied in Van der Laan et al. (1999b). As the authors indicate, their PUSH and PULL control strategies are not very efficient if the remanufacturing leadtime is much smaller than the manufacturing leadtime. They draw this conclusion because they observe decreasing costs with increasing remanufacturing leadtime. In Inderfurth and Van der Laan (2001), this counter intuitive effect is called leadtime paradoxon. They suggest that the remanufacturing leadtimes should be considered as a decision variable rather than an exogenous model parameter/constant, and one should try to determine the optimal inventory position for varying leadtimes, so that there exists an opportunity to set the remanufacturing leadtime in an optimal way.

Fleischmann et al. (2002) derive an optimal control policy with stochastic item returns. They present a model extending the traditional single-item Poisson-demand inventory model with a Poisson return-flow of items. Kiesmuller (2003a) provides a new control approach in the context of a PULL control which uses two separate inventory positions for the production and remanufacturing decision instead of one. This approach is based on the fact that since the recovery inventory control problems are different for varying leadtime relations, they have to be discussed separately. Kiesmuller (2003b) also develops an optimal control of a product recovery system with lead time.
2.2 Disassembly Operation

Currently, there are several areas of disassembly that are being studied by researchers (Brennan et al. 1994, Jovane et al. 1993). Gupta and McLean (1996) group these areas into four broad categories: product design for ease of disassembly, disassembly process planning, design and implementation of disassembly systems and operations, and planning issues in the disassembly environment. For further research done in these areas, please refer to Gupta and Taleb (1994) and Gupta and McLean (1996).

An appropriate planning and control of disassembly processes is an essential prerequisite for an economic disassembly. Wiendahl et al. (1999) outline a specific disassembly planning and control concept based on an investigation of the suitability of common production planning and control methods for disassembly. The planning of manual disassembly systems has been a field of intensive research for more than a decade. The existing remanufacturing facilities are mainly based on manual disassembly operations. However, issues such as higher productivity and increasing cost of labor force improvements in the effectiveness of disassembly operations. Automation of disassembly operations is seen to be the solution to increase the productivity. For example, highly flexible disassembly equipment has been developed and commonly used manufacturing equipment has been applied for automated disassembly. Santochi et al. (2002a) compare the manual and automated disassembly operations and describes a complete and general computer-aided disassembly planning system. In addition to Santochi et al., O’Shea et al. (1998) provide the reader with a good literature review on disassembly planning. Franke et al. (2003) show an approach for planning, modeling and evaluation of a disassembly system from the specification of disassembly sequences via the rough system design, the automatic generation of
a simulation model and the evaluation of the system including the determination of repair stations for failed automated processes. They concentrate on the integration of a layout optimization algorithm into disassembly system design. Automated disassembly can receive valuable support from the development of special tools, improving the ability of robots to handle different shapes and materials. For example, in Santochi et al. (2002b), an experimental cell, equipped with an artificial vision system, for the automated disassembly of refrigerator compressors, aiming at recovering copper and aluminum from electric motors is shown.

2.3 Material Handling Systems

As stated by Tompkins and Reed (1976), facilities design is the joint selection of an integrated materials handling system and plant layout. Previously existing models based on the cost of the materials handling system criterion to evaluate alternative layouts include Hillier (1963), Armour et al. (1964), Hillier and Connors (1966), and Pritsker and Ghare (1970). Webster and Tyberghein (1980) consider the most flexible layout to be the one with the lowest material handling cost over a number of demand scenarios. Bullington and Webster (1987) extend this definition to the multi-period case and present a method for evaluating layout flexibility based on estimating the costs of future re-layout costs. They recommend that these costs can be used as an additional criterion in determining the most flexible layout. Gupta (1986) presents a simulation approach for measuring layout flexibility based on material flow. Askin et al. (1997) evaluate the layout alternatives for agile manufacturing based on material flow.

Lacksonen and Meller (2000) develop an IDEF0 functional model to describe the facility layout design process. They state that there are two key concepts in this process: 1) it is an iterative
process, and 2) the material handling system must be designed simultaneously with the facility layout, similar to what is stated by Tompkins and Reed (1976). The layout design process is considered in three steps: conceptual design, preliminary design, and detailed design. The iterative nature of the process is explained through these steps. A comprehensive understanding of the relationships between material handling system design and layout design as well as other critical activities is found by working through each individual step of the model.

Ferrer and Whybark (2001) describe the first fully integrated material planning system to facilitate the management of a remanufacturing facility. Ferrari et al. (2003) support the design activity of plant layout by means of an integrated approach that takes into account many criteria, both quantitative and qualitative, and present a global approach based on material flow and activity relationships. This approach is carried out using software called LRP (layout and re-layout program) that has a modular architecture and allows a continuous improvement with the design or in the testing phase. The authors conclude that the integration of LRP with modules of warehouse design and of manufacturing cells design appears very interesting. More recently, Jaramillo and McKendall (2004) generate a dynamic extended facility layout problem by simultaneously allocating machines to the plant floor, assigning part/product flow to machines, and defining department boundaries while minimizing total material handling cost. Braglia et al. (2005) presents an extended formulation of layout flexibility concept, when assuming that uncertainty in material handling costs may be described by the expected values and standard deviations of demand forecasts. The definition, analysis, and properties of layout flexibility are introduced with reference to a previous work (Braglia et al. 2003). The concept is discussed and
a procedure is formulated that is capable of characterizing the configuration that gives the best reduction in handling cost fluctuations.

2.4 Facility Layout

2.4.1 Traditional Manufacturing

For a general overview on facility layout problems in manufacturing, Meller and Gau (1996a) present recent and emerging trends, including new methodologies, objectives, and algorithms. They also compare the state of the art in facility layout software to the state of the art in facility layout research. Singh and Sharma (2006) also review different approaches to facility layout by formulations, solution methodologies, and current as well as emerging trends, including a detailed review of layout software packages. In most early research, the objective of these layout problems was to minimize the material handling cost, whereas several more recent studies have addressed the design of layouts in dynamic environments.

Balakrishnan and Cheng (1998) provide a comprehensive review of papers on the dynamic facility layout problem. Benjaafar et al. (2002) provide a comprehensive list of papers that are pertinent to the design of layouts in dynamic environments and define the dynamic facility layout problem as follows: “Assuming demand information for each period is available at the initial design stage, the objective is to identify a layout for each period such that both the material handling and re-layout costs are minimized over the planning horizon”. Shore and Tompkins (1980) were some of the first to consider the design of layouts under uncertainty. They use a facility penalty function that assesses the operation of a facility at levels different than the current one. A change in the departmental areas is not considered. Rosenblatt (1986) develops a
formal model and an optimal solution procedure for determining optimal layouts for multiple periods that deals with the dynamic nature of plant layout. Both material handling cost and the cost of relocating departments from one period to the other are taken into account in a different manner from the Shore and Tompkins’ approach. Since Rosenblatt, a number of researchers, such as Batta (1987), Urban (1992, 1998) and Balakrishnan (1993), improved on Rosenblatt’s solution procedure.

Afentakis and Millen (1990), Kouvelis and Kiran (1991), and Balakrishnan et al. (1992) studied variations of the basic dynamic layout problem. Montreuil and Venkatadri (1991) developed a methodology for designing dynamic layouts for the expansion of manufacturing systems. The stochastic plant layout problem also has been addressed by Montreuil and Laforge (1992), Yang and Peters (1998), and others. Montreuil and LaForge (1992) assume that future production scenarios and their probability of occurrence are known, and they propose a method for developing multiple-period layouts. Like Montreuil and Venkatadri’s approach (1991), a limitation of this method is that the relative positions of departments are fixed for all periods and only their sizes and shapes can vary. Yang and Peters (1998) develop a model that assumes the flow matrices and their probability of occurrence are known for multiple periods. First, the periods for which the layout is to remain static are determined, then, solved for each period. At the end, the results are combined to produce a layout for multiple periods.

Several authors studied the robustness approach to the stochastic plant layout problem, where the most robust layout is the one with the highest frequency of being closest to the optimal solution for the largest number of scenarios. Rosenblatt and Lee (1987) are some of the first researchers
to introduce this concept in analyzing single period layouts. They consider an uncertain environment in which the exact values of probabilities of the different possible scenarios are unknown. For such an environment, layout flexibility is defined in terms of the robustness of the layout performance under different scenarios. Thus, the most flexible (robust) layout is the one whose cost performance remains close to the optimal layout for the largest number of scenarios. They propose a robust approach to the stochastic plant layout problem that was further elaborated by Rosenblatt and Kropp (1992), who presented an optimal solution procedure for the single period stochastic plant layout problem. This procedure only requires solving a deterministic flow-to-flow matrix, where the deterministic matrix is a weighted average of all possible flow matrices.

Because of the computational cost and solution quality, some researchers have turned away from mathematical programming techniques to heuristics (Armour and Buffa (1963), Palekar et al. (1992), Kouvelis et al. (1992), Lacksonen and Enscore 1993, Urban 1993, Conway and Venkatramanan 1994, Meller and Bozer (1996b), Kaku and Mazzola 1997, Kochhar and Heragu (1999), and Rawabdeh and Tahboub (2006)). Palekar et al. (1992) focus on the issue of modeling uncertainties in the plant layout problem. They solve such models using dynamic programming for small problems and heuristics for large ones. The proposed heuristics were able to generate good solutions in a reasonable amount of time for problems with up to 40 departments. Simulation studies indicate that a rolling horizon approach yields better results than a fixed horizon approach. Kouvelis et al. (1992) present heuristic strategies for developing robust layouts for multiple planning periods. Kochhar and Heragu (1999) describe a genetic-based algorithm for single- and multiple-period dynamic-layout problems that consider layout
changeover costs. Ferrari et al. (2003) classify such heuristics into construction, improvement, and hybrid types. Construction algorithms produce the solution without requiring any starting layout (Sly 1995). Improvement algorithms start with an initial layout and try to improve it with facility exchanges (Sly et al. 1996 and Tompkins and Reed 1976). Hybrid approaches provide a first construction phase and a final improvement arrangement (Ferrari et al. 2003). Additionally, multi-criteria models are proposed such as Rosenblatt (1979), Waghodekar and Sahu (1986), Housyar (1991), and Welgama and Gibson (1995).

2.4.2 Remanufacturing

Remanufacturing systems are more dynamic, variable, and complex than traditional manufacturing systems as a result of the variability associated with the routings, processing times, and demand. Kekre et al. (2003) develop a simulation-based line configuration model that simultaneously considers line balancing and line length to maximize the remanufacturing system’s effective throughput. More recently, Lim and Noble (2006) evaluate the performance of different layout alternatives, such as job shop, cellular, fractal, and holonic layouts in remanufacturing. They state that it is possible to improve overall system performance through logistical issues such as facility layout rather than through operational-based approaches using appropriate production planning and control techniques. The primary performance measures included were throughput time and work-in-progress. These two measures were used to examine how each layout scheme affects the performance and efficiency in a remanufacturing system. Based on the experimental and operational results, the authors found that each layout has unique performance characteristics that make it most suitable for different operating scenarios. Consequently, a multi-criteria perspective is useful to determine which layout organization
should be selected based on the criteria chosen by the decision maker. Franke et al. (2006) introduce a model that allows the continuous adaptation of remanufacturing facilities under quickly changing product, process, and market constraints by means of combinatorial optimization and discrete-event simulation. Uncertainties regarding quantity and conditions of mobile phones, reliability of capacities, processing times, and demand are considered. Capacity and remanufacturing program planning are determined by the optimization model, while the simulation allows the planner to determine the required transport, storage capacities, and performance of the remanufacturing system.

2.5 Stochastic Programming and Applications on Production and Capacity Planning

This literature review is restricted to stochastic programming with recourse. A great deal of attention has also been paid to the chance constrained stochastic programming in the literature. Agizy (1969) formulates a wide class of dynamic inventory models as a stochastic programming problem that can be reduced to a linear program with upper-bounded variables and shows that the special structure of the problem allows the use of network flow solution techniques in preference to the simplex method of linear programming. The model is based on the single commodity, multi-period inventory model with no back orders and with a finite horizon. He states that network techniques are preferable to dynamic programming because of the large number of state variables involved. Jagannathan (1991) applies stochastic programming to a multi-item production planning with continuous review of the stock on hand of various items.

Haurie and Moresino (2000) show that a stochastic programming approach could be used to approximate the solution of the associated stochastic control problem in relatively large scale
manufacturing flow control problems. Christie and Wu (2002) present a multistage stochastic programming model for strategic capacity planning in semiconductor manufacturing. They define the main sources of uncertainty as demand of different technologies and capacity estimations for each fabrication facility. The objective of their model is to minimize the gaps between product demands and the capacity allocated to the technology specified by each product. Gupta and Maranas (2003) describe a stochastic programming based approach to model the planning process as it reacts to demand realizations unfolding over time, and formulate the midterm production-planning model under demand uncertainty as a two-stage stochastic program. The manufacturing variables are considered as the first-stage, here-and-now decisions while logistics decisions are modeled as the second-stage wait-and-see decisions. Karabuk and Wu (2003) study strategic capacity planning in the semiconductor industry. Their approach differ from a typical two-stage stochastic program in that the decision makers who actually carry out the recourse are different from the ones who make the first-stage decisions, meaning that the recourse policy used under a particular scenario is not known exactly a priori and must be approximated at time zero. Barahona et al. (2005) present a mixed-integer, two-stage, stochastic programming model for capacity planning under demand uncertainty in semiconductor manufacturing. They then developed four different heuristic approaches based on cutting planes to produce good solutions in a reasonable amount of time, which in some cases found the near-optimal solution in less than three hours. Zimberg and Testuri (2006) present a simplified stochastic model for crude oil and processing. The model incorporates the uncertainty and provides purchasing decisions that are hedged against several scenarios. They conclude that the stochastic model offers additional value, however it is more expensive to formulate as it
incorporates statistical information and the number of variables increases proportionally with the number of scenarios.

There is an extensive literature on applications of stochastic programming on capacity and production planning. However, there exists little research, if any, on its applications in a remanufacturing setting.
Chapter 3 – Problem Statement and Research Objectives

3.1 Handling Uncertainty

3.1.1 Sources of Randomness in the Remanufacturing Supply Chain

A fundamental characteristic of a remanufacturing environment is the inbound flow of used products, where most of the components or subassemblies will have probabilistic yields in the sense that not all items will be suitable for re-use. Some salvaged components ultimately may cause a remanufactured unit to fail a subsequent reliability or functionality test. The stochastic nature of returned products affects predictability, safety stock, production targets, rework and waste (such as additional cleaning, testing, inspection, and reassembly).

The flow of parts through a remanufacturing facility is dependent on the necessary disassembly, cleaning, and testing processes, the age and amount of wear, and demand for immediate assembly or for inventory. Required materials or equipment, such as cleaning solvents or particular tools, depend on the condition and nature of a part, resulting in product flow and equipment needs that are uncertain until the unit arrives at the facility. Both remanufactured units and reclaimed components need to be stored in inventory until they are needed to satisfy demand. All these sources of uncertainty make the design of a remanufacturing facility more complex. Moreover, with multiple types of products or model variations being returned, some
products will share common components such that a recaptured part from one product may be used in another.

Different component yields within and between products further complicates effective inventory planning and warehouse design. As a simple example, consider the remanufacturing of a product consisting of three components, denoted by A, B, and C and with disassembly yield rates ($\gamma$) of .85, .65 and 0, respectively. That is, for any given unit received for remanufacturing, after disassembly the probabilities that part A, part B, and part C are salvageable are $\gamma_A = .85$, $\gamma_B = .65$, and $\gamma_C = 0$, respectively. If a remanufacturing facility receives a batch of $n = 100$ products per period for remanufacturing, the expected number of salvaged part A, part B, and part C are $n\gamma_A = 85$, $n\gamma_B = 65$, $n\gamma_C = 0$, respectively, although batch-to-batch there will be variability in each observed yield described by binomial probabilities. Figure 3.1 illustrates the flow of parts through this remanufacturing system. The processes are denoted by rectangles, and the storage requirements are denoted by triangles. The numbers outside of the parentheses on the top of each flow arrow indicate the expected number of parts that flow through each process and storage space in the first period, and the numbers inside parentheses denote the expected number of parts moving from one step in the process to another in the second period.

To remanufacture these 100 units, the non-salvageable parts need to be replaced either from inventory or purchased from an outside vendor, with an expected number of replacements of part $i$ equal to $n(1 - \gamma_i)$, $i = A, B, C$. Again, for any given batch the actual number of scrapped and replaced parts will be random with binomial probabilities and variances. In this illustration, for example, the observed yield of part B will have a standard deviation of $[n\gamma_B(1 - \gamma_B)]^{0.5} =
[100(.65).35]^{0.5} \approx 4.77$, implying roughly 95% of batches will yield between 56 and 74 useable pieces of part B, with the remaining 26 to 44 pieces drawn from inventory or sourced externally. Moreover, in many realistic scenarios, the incoming batch size would be a random variables as well, resulting in compound probability distributions and greater variability in the component yields (where $n$ has a prior distribution), such as a Poisson-binomial model.

After subsequent reliability testing of the salvageable A and B parts, a subset of these are available for reassembly, with expectations $n\gamma_a\alpha_A$ and $n\gamma_b\alpha_B$ and variances $n\gamma_A\alpha_A(1 - \gamma_A\alpha_A)$ and $n\gamma_B\alpha_B(1 - \gamma_B\alpha_B)$, respectively, where $\alpha_i$ is the reliability failure rate for part $i$. While the appropriate probability distributions, means, and variances can be derived for this simple example, in more complex and realistic scenarios the logic may become intractable. If for illustration one can assume that initially there are no parts of either type in inventory, then ultimately the number of remanufactured units from the original batch depends on the smaller number of A and B parts which pass the reliability test. The number of C parts would need to be forecasted, ordered from an outside manufacturer, and on hand in order to then remanufacture the units. Given long lead times for Part C, all A and B parts would need to be stored until the C units are received. Either way, the unused A or B parts would be stored in inventory for possible use in a subsequent period.
3.1.2 A Simulation Model Illustration

The following simulation model has been designed to provide insight into how the number of remanufactured products and the inventory builds of each part are affected by the disassembly and reliability yield rates, and how this affects appropriate storage policies planning. The goal of this simulation example is simply to show how the inventory of remanufactured product and the inventory build of parts A and B vary as the disassembly and reliability yield rates vary. (The simulation of a full-scale scenario would be very complex and is beyond the intended scope here, but could follow similar logic.) In the below scenarios, the batch size is held constant at 100 returned products per period and the model is run for 100 replications of 50 periods each. Both the disassembly and reliability testing yield rates follow binomial distributions. An extension of this model would allow the batch size to vary from period to period. This analysis is shown in Table 3.1.
Some of the key assumptions in the model are as follows. All returned products are disassembled immediately, meaning that no storage is necessary after the products are returned to the remanufacturing facility at the end of their life cycle. Other than disassembly, clean, repair and reliability testing, no other operation is performed on any part or remanufactured product. Scrap part inventory is fed by disassembly and reliability testing, stored in the same storage area. All other non-serviceable parts are disposed of immediately without storage. Only part C is purchased from an outside supplier, with the number of remanufactured items in any period equal to the minimum number of available (in inventory from past periods or recaptured in the current period) parts of type A or B. Inventory holding costs for returned products, disassembled products, and remanufactured products are constant during each period. The machines which perform remanufacturing operations are assumed to operate without any repair or maintenance interruption. No inspection before shipping the product to the customer is required.

Figure 3.2 shows one replication of inventory builds of part A and B over 50 time periods and assuming equal disassembly and reliability yield rates for both parts, illustrating the complementary nature of part accumulation and the variability in storage requirements over time. The mean and maximum of 100 replications per period are shown in Figure 3.3.
### Table 3.1 Simulation Model Analysis

<table>
<thead>
<tr>
<th>Period</th>
<th>Replication 1</th>
<th>Replication 2</th>
<th>Replication 99</th>
<th>Replication 100</th>
<th>Inventory Build of A, Inv(A)</th>
<th>Inventory Build of B, Inv(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X(RM) Inv(A) Inv(B)</td>
<td>X(RM) Inv(A) Inv(B)</td>
<td>X(RM) Inv(A) Inv(B)</td>
<td>X(RM) Inv(A) Inv(B)</td>
<td>X(RM) Inv(A) Inv(B)</td>
<td>X(RM) Inv(A) Inv(B)</td>
</tr>
<tr>
<td>1</td>
<td>56 0 8 60 0 3</td>
<td>64 0 8 64 0 11</td>
<td>61.82 3.13 63.46 5.88</td>
<td>62.70 5.56 68.68 8.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>66 0 11 64 0 4</td>
<td>61 0 14 63 9 0</td>
<td>62.67 4.12 61.53 63.81</td>
<td>54.59 70.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>53 0 22 67 2 0</td>
<td>66 0 9 66 0 3</td>
<td>62.36 3.96 61.52 63.72</td>
<td>54.86 70.38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>71 0 12 61 0 4</td>
<td>60 0 13 68 0 11</td>
<td>63.82 4.38 62.60 65.04</td>
<td>55.23 72.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>63 0 9 66 0 6</td>
<td>62 0 8 62 0 4</td>
<td>62.89 4.73 61.58 64.20</td>
<td>53.62 72.16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>68 0 10 63 0 5</td>
<td>67 0 5 67 4 0</td>
<td>63.53 4.38 62.32 64.74</td>
<td>54.95 72.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>67 0 12 71 0 3</td>
<td>60 1 60 5 0</td>
<td>63.29 3.81 62.23 64.35</td>
<td>55.83 70.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>68 0 14 61 0 2</td>
<td>62 0 0 61 4 0</td>
<td>63.18 4.22 62.01 64.35</td>
<td>54.90 71.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>64 0 11 61 0 12</td>
<td>65 8 0 67 0 0</td>
<td>63.41 4.67 62.17 64.65</td>
<td>54.65 72.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>69 0 11 57 0 23</td>
<td>62 5 0 63 1 0</td>
<td>63.17 4.10 62.03 64.31</td>
<td>55.13 71.21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Average:** 63.42 19.9 5.5 63.64 0.62 14.68 63.68 24.44 1.38 63.46 11.08 0.88

**Max Val:** 71 82 25 74 9 39 73 62 14 71 35 11

**Min Val:** 51 0 0 54 0 0 51 0 0 54 0 0

**StDev:** 5.22 23.35 7.32 4.45 1.83 12.96 4.50 21.46 3.39 4.02 9.75 2.47
Figures 3.4 and 3.5 illustrate the distribution of the average and maximum, respectively, of the number of parts in inventory at any time over each of 100 replications. As shown, the inventory behaviors of both parts are very similar, due to their similar yields. Conversely, Figures 3.6-3.8 illustrate the same information for an example with slightly different yield rates for part A and
part B, \((\gamma_A = .82, \gamma_B = .78)\). As illustrated, note that results change significantly for even slightly different yields with storage requirements quickly becoming more dissimilar and variable. In particular, the average and maximum number of storage spaces are dramatically impacted by several folds, having implications on fixed and flexible design planning. Because the inventory build of parts occurs when the demand for one is less than the other in any given period, unused parts are stored until there is demand in some following period. This process can cause a sudden build and depletion of inventory for some parts over a few periods, and the storage space needs to be adaptable to accommodate this phenomenon.

![Figure 3.4 Average number of parts in inventory \((\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80)\)](image)

![Figure 3.5 Maximum number of parts in inventory \((\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80)\)](image)
Chapter 3 Problem Statement and Research Objectives

Figure 3.6 Inventory builds ($\gamma_A=.82$, $\gamma_B=.78; \alpha_A=.80 \alpha_B=.80$), 100 replications

Figure 3.7 Average number of parts in inventory ($\gamma_A=.82$, $\gamma_B=.78; \alpha_A=.80 \alpha_B=.80$)

Figure 3.8 Maximum number of parts in inventory ($\gamma_A=.82$, $\gamma_B=.78; \alpha_A=.80 \alpha_B=.80$)
Finally, Figures 3.9 and 3.10 illustrate the impact of inventory availability on the number of remanufactured products produced per period. Note that this remanufacturing rate is almost the same for both examples, but will be lower if any given part has a lower yield rate.

Figure 3.9 Number of remanufactured products ($\gamma_A = \gamma_B = .80; \alpha_A = \alpha_B = .80$), 100 replications

Figure 3.10 Number of remanufactured products ($\gamma_A = .82 \gamma_B = .78; \alpha_A = .80 \alpha_B = .80$), 100 replications
3.2 Reconfigurable and Flexible Designs

3.2.1 Reconfigurable Facility Layout

Rapid changes in technology and markets require production systems that are themselves easily upgraded and into which new technologies and new functions can be integrated. These changes include (Koren et al. 1999) increasing introduction of new products, parts changes in existing products, product demand fluctuations, changes in government regulations and process technology. In Koren and Ulsoy (2002), a reconfigurable manufacturing system is defined as “one designed at the outset for rapid change in its structure, as well as its hardware and software components, in order to quickly adjust its production capacity and functionality within a part family in response to sudden market changes or intrinsic system changes”. For a manufacturing system to be reconfigurable, it needs certain characteristics.

The above authors define these characteristics as: (1) modularity (the ability to design all system components, both software and hardware to be modular), (2) scalability (the ability to easily change existing production capacity by rearranging an existing production system and changing the production capacity of reconfigurable components within that system), (3) integrability (the ability to design systems and components for both ready integration and future introduction of new technology), (4) convertibility (the ability to allow quick changeover between existing products and quick system adaptability for future products), (5) customizability (the ability to design the system capability and flexibility to match the applications), and (6) diagnosibility (the ability to identify quickly the sources of quality and reliability problems that occur in large systems). Mehrabi et al. (2000a) examine and identify the key interrelated technologies that should be developed and implemented to achieve these characteristics. They explain the issues
related to technology requirements of reconfigurable manufacturing systems at the system and machine design levels, and ramp-up time reduction.

Lee (1997) considers manufacturing system reconfiguration under the assumptions of (i) manufacturing systems are statistically or dynamically reconfigurable systems (ii) alternative designs are available for each component without compromise of component functionality. Lee concludes that designing components to have similar component routes reduces the relocation cost of machines while reconfiguring the system. Mehrabi et al. (2000b) present an overview of available reconfigurable manufacturing techniques, their key drivers and enablers, and their impacts, achievements and limitations. They state that reconfigurable manufacturing systems go beyond the objectives of mass, lean, and flexible manufacturing by: (1) the reduction of lead time for launching new systems and reconfiguring existing systems, and (2) the rapid manufacturing modification and quick integration of new technology and new functions into existing systems. They explain how reconfigurability differs from agility, but also conclude that agile manufacturing complements reconfigurable manufacturing, and define reconfigurability as “the set of methodologies and techniques that aid in design, diagnostic, and ramp-up of reconfigurable manufacturing systems and machines that give corporations the engineering tools that they need to be flexible and respond quickly to market opportunities and changes.” Reconfigurable manufacturing systems are defined as “…designed for rapid adjustment of production capacity and functionality, in response to new circumstances by rearrangement or change of its components.” The core idea is to design the production system around a part family to achieve a lower cost, and therefore a higher production rate with the required flexibility (Mellor 2002).
Elmaraghy (2005) outline and compare the characteristics of reconfigurable manufacturing systems and flexible manufacturing systems, linking the concept of a manufacturing system life cycle with aspects of manufacturing system flexibility and reconfigurability. There is little research, however, that focuses specifically on reconfigurable facility layout. Benjaafar et al. (2002) give examples that support the migration of next generation systems towards highly adaptable and quickly reconfigurable systems. Meng et al. (2004) apply the concept of reconfigurability in layout problems. They state that the reconfigurable layout problem differs from traditional robust and dynamic layout problems in two aspects: (1) it assumes that production data are available only for the current and upcoming production period, and (2) it considers queuing performance measures such as work in progress inventory and product lead time in the objective function of the layout problem.

3.2.2 Reconfigurable Warehouse and Inventory Design

As the simulation results in Section 3.3 illustrated, significant variability in inventory can occur in remanufacturing facilities. A reconfigurable storage system design should be able to adjust its capacity to accommodate these fluctuations beyond the objectives of dynamic or flexible facility layout problems by resolving the configurability problem more frequently in any real time. As an example, the number of usable parts from returned products changes from period to period, and a company may desire to modify its layout to store the extra parts until they are needed.

3.3 Research Objectives and Issues

The research objectives are:

1) To develop stochastic programming recourse models for reconfigurable multi-product and multi-period storage allocation
2) To implement a multi-dimensional golden section simulation-optimization algorithm to identify the optimal storage capacities and reconfiguration decisions

3) To develop pseudo-optimal heuristics for longitudinal storage space planning in remanufacturing facilities

In the next few paragraphs, a discussion on the issues in remanufacturing storage space planning is provided. The number of returned products and parts reclaimed from each product vary significantly. A low yield rate causes new parts to be purchased from an outside party to meet the demand, while with high yield rates the capacity planning decisions among internal, reconfigured and external storages become an important issue to store the extras. Decisions whether new parts are ordered in batches or on a need basis are also made. This causes changes on the layout due to space planning of storing the new parts. With high yield rates, a cost comparison in terms of storing internally, by reconfiguring space or externally is made to minimize the total inventory storage cost. The issues arising with reconfiguration are (1) determining the parts that have similar storage requirements, (2) for the parts which require different storage conditions, calculating the cost of reconfiguring space which was actually allocated to be storing another part type.

Apart from the variability in yield rates and its impact on total storage cost and layout, when a returned product is disassembled, the processes required by its parts differ, which affects the material flow throughout the remanufacturing facility. Its effects are also observed on the number of disposable parts that gets disposed in that cycle. The shipment size of disposable parts to the outside parties subsequently becomes a significant decision to be made.
Chapter 4 – A Modeling Approach: Stochastic Programming Recourse Models

4.1 Overview of Stochastic Programming

No modeling approaches have appeared in the literature that addresses reconfigurable remanufacturing facility designs that adapt to inherent uncertainties. To illustrate how such a model might work and the associated issues, a stochastic programming recourse model that optimizes the changing storage space requirements for multiple components over multiple time periods is described. Stochastic programming is one potential approach that has promise for modeling reconfigurable facility layout problems.

Inventory space requirements in remanufacturing facilities can vary significantly over time and by the type of space needed, due to variability in recaptured component quality, availability of refurbished components, remanufactured product demand, and returned product rates. Multi-period stochastic programming recourse models are developed to identify optimal schedules of internal, external, and reconfigured amounts of storage space in each time period. Results are compared to expected value models and non-integer models, and computational issues are discussed. The existence of different heuristic approaches, such as genetic algorithms and tabu search enables us to deal with the increased model size and computation duration. This study has enabled us to illustrate that stochastic programming when used with heuristic approaches is an efficient method to calculate the carrying cost in a remanufacturing facility.
As remanufacturing processes become increasingly important, reconfiguration of available storage capacity to satisfy anticipated needs has advantages for responding to unpredictable component recapture yields, product condition, availability of refurbished components, and remanufactured product demand. This work develops stochastic programming recourse (SPR) models that optimize changing storage space requirements over multiple time periods. Two central ideas of these models are (1) instead of replacing uncertain values with their expectations, the probability space of possible events is explicitly modeled and (2) the concept of recourse in later periods, i.e. the ability to take corrective action depending on these random outcomes in preceding periods.

The optimal decision produced to minimize total expected costs includes both the amount of each type of storage space to allocate in the current time period and a schedule of subsequent recourse actions in each latter period given each possible random set of outcomes. A two-period SPR model, for example, identifies an initial storage configuration for period 1 and recourse actions for period 2 that together minimize the a priori total expected cost. First introduced independently by Dantzig (1955) and Beale (1955), SP models have been applied to a wide variety of problems in which inherent randomness affects decision making, ranging from agriculture, energy, military, to manufacturing applications. While related models have been applied to capacity and production planning problems, little work has been published on reconfigurable remanufacturing applications. Capacity planning is typically implemented using spreadsheets, discrete-event simulation, and linear programming based techniques (Bermon and Hood 1999).
Network techniques (Agizy, 1969), stochastic decomposition (Higle and Sen, 1991), approximations based on Monte Carlo sampling (Dantzig and Glynn, 1990), Lagrangian relaxation technique (Takriti et al., 1996), Benders’ decomposition (MirHassani et al., 2000, Sen and Sherali, 2006), Monte Carlo simulation based approaches are some of the solution techniques utilized to solve stochastic programming with recourse models. Agizy (1969) formulates a single-commodity, multi-period inventory model with no backorders and with a finite horizon as a stochastic programming problem that can be reduced to a linear program with upper-bounded variables. He also shows that the special structure of the problem allows the use of network flow solution techniques in preference to the simplex method of linear programming. He states that network techniques are preferable to dynamic programming because of the large number of state variables involved. The second solution technique is the stochastic decomposition, which is applicable to problems governed by both discrete and continuous random variables allowing any integer restrictions on the first stage decisions if needed (Higle and Sen, 1991). Their method is motivated by Benders’ decomposition. It uses randomly generated observations of random variables to construct statistical estimates of the objective function. The third solution method is the approximations based on Monte Carlo sampling. Dantzig and Glynn (1990) support combining a nested dual-decomposition approach with Monte Carlo importance sampling and the assignment of the sampling tasks to parallel processors. They point out the role that parallel processors and importance sampling can play when combined with earlier results on decomposition. The fourth solution method is Lagrangian relaxation technique, namely progressive hedging (Takriti et al. 1996). They also formulate the same problem as a deterministic model. Both the stochastic and deterministic model can be implemented in parallel to reduce the computation time. The stochastic model is solved using a
Lagrangian relaxation type of technique. The fifth solution method is the Benders’ decomposition. MirHassani et al. (2000) apply Benders decomposition to solve a two-stage SP model. They show how a parallel Benders algorithm can be modified to achieve a convergence rate comparable with the serial Benders approach. They presented computational results on supply chain networks involving up to 8 plant sites, 15 distribution centers, 30 customer locations, and with 100 scenarios. Sen and Sherali (2006) discuss a class of problems in which the second-stage subproblems impose integer restrictions on some variables and they are solved using branch-and-cut methods. They discuss that this has the advantage of solving their two-stage stochastic mixed-integer programming problem by dividing it into smaller mixed-integer programming subproblems. Shapiro and Homem-de-Mello (1998) use Monte Carlo simulation based approaches in their problems where the objective function is given as an expected value function. The random data in recourse variables have a continuous distribution. One can also refer to Schultz (2003) for a general discussion on branch-and-bound and Lagrangian relaxation.

When the uncertain parameters with continuous distributions (meaning with an infinite number of scenarios) are involved, Santoso et al. (2005) integrate a sampling strategy, the sample average approximation scheme with an accelerated Benders decomposition algorithm to solve supply chain design problems.

Stochastic programming has been used in a wide range of areas, such as the automobile industry, electric utility industry, semiconductor industry, computer industry, oil processing industry and supply chain planning. As the literature review highlights, among all industries, there is a tremendous amount of research done in the semiconductor industry. Jordan and Graves (1995) use stochastic programming in the automobile industry to allocate capacity to automobile models.
given the uncertainty in demand, driven by the consumer. SP is also used in the electric utility industry to decide when to fire up another generator given the uncertainty in demand, driven by both weather and price (Takriti et al. 1996). Takriti et al. indicate significant savings in the cost of operating power generating systems when the stochastic model is used instead of the deterministic model. MirHassani et al. (2000) considered a two-stage model for multi-period capacity planning of supply chain networks. Their overall objective is to minimize the cost of the first-stage strategic decisions and the expected production and distribution costs over the uncertain demand scenarios. Tsiakis et al. (2001) considered a two-stage stochastic programming model for a supply chain network designed under demand uncertainty, and developed a large-scale mixed-integer linear programming model for this problem. Morton and Wood (1999), Hood et al. (2003), Barahona et al. (2005), Swaminathan (2000 and 2002), Christie and Wu (2002) and Karabuk and Wu (2003) use it in semiconductor industry. Swaminathan (2000 and 2002) models the tool capacity planning to have a good estimate of future tool requirements due to the change in products and technology in the semiconductor industry. Swaminathan (2002) differ from Swaminathan (2000) in that in the later article, one needs to plan production at the operations level instead of planning total wafer production on tools. Hence, the question in the later is which operations of each wafer type should be performed on which tools. Christie and Wu (2002) present a stochastic programming model for the multistage annual capacity planning using two different scenario-analysis constructs. Karabuk and Wu present a scenario-based stochastic programming model which produces a capacity configuration that protects against extreme outcomes of demand and capacity fluctuation. For additional work on capacity planning under uncertainty, please refer to Morton and Wood (1999), Hood et al. (2003) and Barahona et al. (2005). Swaminathan and Tayur
(1998) use stochastic programming in the computer industry. The approach finds the optimal configurations and inventory levels of the semi-finished products that can serve more than one final product in an environment where demands are stochastic. Zimberg and Testuri (2006) proposed a stochastic model when the demand for bunker fuel oil in oil processing industry is uncertain, and compared it with a deterministic model. Gupta and Maranas (2003) provide a stochastic programming based approach to model the midterm planning of multisite supply chains as it reacts to demand realizations unfolding over time. There were two sets of decisions made in their model: manufacturing decisions made before demand realizations and logistics decisions postponed in a wait-and-see model to optimize in the face of uncertainty. Their model evaluates the changes in inventory levels and profit margins to market uncertainties. Messina and Bosetti (2006) use stochastic programming in gradual land conversion problems to determine the optimal land portfolio in time, given uncertainty affecting the market. Schultz (2003) identifies two- and multi-stage stochastic integer programs that are large-scale block-structured mixed-integer linear programs. Jagannathan (1991) applies stochastic programming to multi-item production planning with continuous review of the stock on hand of various items. For additional reading on usage of stochastic programming in production planning, please refer to Jonsbraten et al. (1998).

In Topcu et al. (2008), a simple two-period SPR model with an extension to multiple periods is illustrated. The objective is to determine the optimal internal, reconfigured, and external storage space for the refurbished components in each future period that minimizes all associated costs. Different than the models to be illustrated in succeeding sections, if the internal space needed in period $p$ exceeds allocated availability, rather than having five options, two options are
considered: to reconfigure other space at a cost of $c_r$ per unit space or to pay for outside storage space at a cost of $c_o$ per unit space, whereas an opportunity cost $c_u$ is associated with each unused unit of internal storage. Please refer to Figure 4.1 on the next page for the model illustration.

While related models have been applied to all these different problems mentioned above, little work has been published on reconfigurable remanufacturing applications.

4.2 SPR Model Development

4.2.1 Model Assumptions and Notation

Remanufactured items consist of one or more components reclaimed from returned products. In each time period, a number of returned products are received by a remanufacturing facility, each consisting of one or more recapturable components. After disassembly, each component undergoes testing, cleaning, and refurbishing, with a probability based number ultimately yielded as acceptable for reuse. Disassembly, testing, remanufacturing, and reconfiguration durations are assumed small enough that they occur in the same period that components are salvaged. There is a penalty cost associated with unmet demand. Excess refurbished components are stored internally, externally or disposed of if capacity is exceeded; non-salvageable components are disposed of externally and do not need to be stored. There is a cost associated with disposal of refurbished components if capacity is exceeded.
Figure 4.1 Stochastic programming representation assuming single-part, two-period, and constant demand
The objective is to determine the optimal internal, reconfigured, and external storage space for the refurbished components as well as the disposal or selling cost of refurbished components and the penalty cost for unmet demand in each future period that minimizes all associated costs. If the internal space needed in period $p$ exceeds allocated availability, the five options available are (i) to reconfigure other space at a cost of $c_r$ per unit space, (ii) pay for outside storage space at a cost of $c_o$ per unit space, whereas an opportunity cost $c_u$ is associated with each unused unit of internal storage, (iii) scrap/dispose the extra refurbished components at a cost of $p_j$ per component (iv) sell the extra refurbished component at a price (same as disposing the extra refurbished components at a cost), (v) or some combination of the above. Upper bounds exist on the total available internal and external storage space in each time period. Sum of allocated internal space and reconfigured space cannot exceed the internal storage capacity. A decision made in the first period as to the amount of internal space to allocate for the next period, based on probabilities of future component yields, incoming supply, and demand for remanufactured products (with the facility assumed to start empty in period 0), and a schedule of recourse actions is identified for each possible outcome that, together with this allocation decision in period 1, minimize the total multi-period expected inventory storage cost.

Figure 4.2 Example of discretization of probability space for the number of salvaged reusable components

70
Given \( I_p \) returned products in period \( p \), Figure 4.2 illustrates all possible component yield volumes, \( X \), after disassembly and testing (a binomial random variable, with \( x = 0, 1, \ldots, I_p \)), which then is discretized into a fewer number of yield percentages in order to make the model tractable. Each tree path \( k \), or “scenario”, corresponds to a possible percent, \( Y_{k,p} \), of salvaged reusable components occurring with probability \( \pi_k \). In the above example, the remanufacturer recaptures \( Y_{1,1} = 60\% \) of a certain component with probability \( \pi_1 = 0.1 \), \( Y_{2,1} = 90\% \) with probability \( \pi_2 = 0.6 \), and \( Y_{3,1} = 75\% \) with probability \( \pi_3 = 0.3 \). Looking ahead to period 2, an initial decision is made in period 1 about how much internal storage space, \( S_1 \), to allocate. After demand is met in the second period, \( E_{k,2} \) excess refurbished components need to be stored, or discarded (based on the observed yield and demand). The decision variables in this model are:

\[
I_p = \text{Number of returned products to receive (‘pull’) into the facility in period } p,
\]

\[
I^{\alpha}_{k,p} = \text{Number of returned products to receive (‘pull’) into the facility in period } p \text{ if there exists extra components},
\]

\[
I^{\beta}_{k,p} = \text{Number of returned products to receive (‘pull’) into the facility in period } p \text{ if there is unmet demand},
\]

\[
S_1 = \text{Internal storage space allocated for the extra refurbished components in period 1},
\]

\[
R_{k,p} = \text{Amount of space to reconfigure for excess refurbished components in period } p \text{ under scenario } k,
\]

\[
O_{k,p} = \text{External storage space for excess refurbished components in period } p \text{ under scenario } k,
\]

\[
U_{k,p} = \text{Unused internal storage space in period } p \text{ under scenario } k,
\]

\[
J_{k,p} = \text{Number of disposed components in period } p \text{ under scenario } k, \text{ and}
\]
$X_{k,p} = \text{Unmet demand in period } p \text{ under scenario } k,$

where the subscripts $p$ and $k$ indicate the period and outcome, respectively, and with $R_{k,p}$, $O_{k,p}$, $U_{k,p}$, $J_{k,p}$ and $X_{k,p}$ being recourse variables. For simplification the subscript $k$ is dropped when the variables are scenario-independent, (e.g., $I_p$ and $S_1$). Two binary variables, $\alpha_{k,p}$ and $\beta_{k,p}$, indicate whether the demand for remanufactured products is met or not.

$\alpha_{k,p} = \text{Binary variable taking the value of 1 if there are extra refurbished components in period } p \text{ under scenario } k,$

$\beta_{k,p} = \text{Binary variable taking the value of 1 if demand is not met in period } p \text{ under scenario } k,$

The random variables are as follows:

$Y_{k,p} = \text{Remanufacturing yield percentage } k \text{ in period } p,$

$\pi_k = \text{Probability of outcome } k,$

$E_{k,p} = \text{Extra number of refurbished components in period } p \text{ under outcome } k.$ For a two-period model ($p = 2$), $E_{k,2} = \max(0, Y_{k,2} \cdot \alpha_{k,2} - D_{k,2} \cdot \pi_{k,2})$, where $E_{k,1} = 0$

Input cost, demand and capacity parameters include:

$D_p = \text{Product demand in period } p,$

$c_s = \text{Cost of internally storing each remaining refurbished component},$

$c_r = \text{Cost of reconfiguring one unit of storage space},$

$c_o = \text{Cost of externally storing each remaining refurbished component},$

$c_u = \text{Opportunity cost of each unused internal unit of space},$
4.2.2 Two-Period SPR Model

In a two-period model, after an initial space decision is made in the first period and the number of refurbished components is observed, reconfiguration, external storage and disposal recourse decisions are made in period 2. For illustration purposes, assuming the $k = 3$ possible yield scenarios illustrated in Figure 4.2, just one component type is salvageable per returned item, and demand is deterministic and known, then a two-period SPR model can be written as

Minimize \( (c, S_1) + \sum_{k=1}^{3} \pi_k \left( c_r R_{k,2} + c_o O_{k,2} + c_u U_{k,2} + c_j J_{k,2} + c_s X_{k,2} \right) \) \[1\]

subject to

\[ \alpha_{k,2} + \beta_{k,2} = 1 \] \hspace{1cm} \text{(binary constraint)} \[2\]

\[ I_{k,2}^\alpha + I_{k,2}^\beta = I_{2,2} \] \hspace{1cm} \text{(number of returned products)} \[3\]

\[ |I_{k,2}^\alpha - I_{k,2}^\beta| = I_{2,2} \] \hspace{1cm} \text{(extra components)} \[4\]

\[ I_{k,2}^\alpha Y_{k,2} - D_{2,2}^\alpha \alpha_{k,2} = E_{k,2} \] \hspace{1cm} \text{(extra components)} \[5\]

\[ -I_{k,2}^\beta Y_{k,2} + D_{2,2}^\beta \beta_{k,2} = X_{k,2} \] \hspace{1cm} \text{(unmet demand)} \[6\]

\[ S_{1,2} + R_{k,2} + O_{k,2} + J_{k,2} \geq E_{k,2} \] \hspace{1cm} \text{(storage of extra components)} \[7\]
where \( k = 1, 2, 3 \). The objective function minimizes the total two-period cost, i.e. the internal storage cost plus the expected value of the future reconfigured storage, external storage, opportunity, disposal, and penalty costs. Constraints 2, 3, and 4 ensure that either demand is met or not. The binary variables \( \alpha_{k,2} = 1 \) if the number of refurbished components exceeds the demand, and \( \beta_{k,2} = 1 \) if penalty is occurred for unmet demand. Constraints 5 and 6 compute the number of extra components, and the unmet demand in the second period, respectively. Constraint 7 ensures extra components are stored either internally, in reconfigured space, externally, or sent for disposal. Constraints 8 and 9 force any unused internal storage space to incur opportunity costs, constraints 10 and 11 specify limits for internal and external storage space, and constraint 12 ensures non-negativity of all variables. Constraint 13 sets the binary variables. While constraints 7 and 9 are given here for readability, they are redundant and can be removed while implementing the model in a solver software. Solving an equivalent expected value model adds an additional constraint to the model which is the expected yield for extra refurbished components:
4.2.3 Multiple Periods

The same general approach can be taken to extend the above model to more realistic scenarios with random demand and multiple time periods, components, and types of returned products with partially common components (although all significantly increasing the number of possible scenarios, model size, and run times as discussed below). In the multiple period case, as one example, given \( m \) outcomes per period, \( n \geq 3 \) periods, and \( m^{(n-1)} \) scenarios, the above model extends to:

\[
\text{Minimize } \sum_{k=1}^{n} \sum_{p=1}^{m-1} \pi_k \sum_{p=2}^{n} \left( c_r R_{k,p} + c_o O_{k,p} + c_u U_{k,p} + c_j J_{k,p} + c_x X_{k,p} \right) 
\]

subject to

\[
\alpha_{k,p} + \beta_{k,p} = 1 \quad \text{(binary constraint)} \]

\[
I^\alpha_{k,p} + I^\beta_{k,p} = I_p \quad \text{(number of returned products)} \]

\[
|I^\alpha_{k,p} - I^\beta_{k,p}| = I_p \quad \text{(unmet demand)} \]

\[
I^\alpha_{k,p} * Y_{k,p} + E_{k,(p-1)} * \alpha_{k,p} - D_p * \alpha_{k,p} = E_{k,p} \quad \text{(extra components)} \]

\[
-I^\beta_{k,p} * Y_{k,p} - E_{k,(p-1)} * \beta_{k,p} + D_p * \beta_{k,p} = X_{k,p} \quad \text{(unmet demand)} \]

\[
S_1 + R_{k,p} + O_{k,p} + J_{k,p} \geq E_{k,p} \quad \text{(storage of extra components)} \]

\[
S_1 + R_{k,p} + O_{k,p} + J_{k,p} - E_{k,p} = U_{k,p} \quad \text{(unused space)}
\]
\[ E_{k,p} - S_1 - R_{k,p} + U_{k,p} \geq 0 \]  \[ S_1 + R_{k,p} \leq C^s_p \]  \[ O_{k,p} \leq C^a_p \]  \[ R_{k,p} = R_{k+1,p} = \ldots = R_{m^n-p,p} \ldots \; \; \; R_{m^n-p+1,p} = R_{m^n-p+2,p} = \ldots = R_{m^{p-1}m^n-p,n-1} \]  \[ O_{k,p} = O_{k+1,p} = \ldots = O_{m^n-p,p} \ldots \; \; \; O_{m^n-p+1,p} = O_{m^n-p+2,p} = \ldots = O_{m^{p-1}m^n-p,n-1} \]  \[ U_{k,p} = U_{k+1,p} = \ldots = U_{m^n-p,p} \ldots \; \; \; U_{m^n-p+1,p} = U_{m^n-p+2,p} = \ldots = U_{m^{p-1}m^n-p,n-1} \]  \[ J_{k,p} = J_{k+1,p} = \ldots = J_{m^n-p,p} \ldots \; \; \; J_{m^n-p+1,p} = J_{m^n-p+2,p} = \ldots = J_{m^{p-1}m^n-p,n-1} \]  \[ X_{k,p} = X_{k+1,p} = \ldots = X_{m^n-p,p} \ldots \; \; \; X_{m^n-p+1,p} = X_{m^n-p+2,p} = \ldots = X_{m^{p-1}m^n-p,n-1} \]  \[ S_1, I^\alpha_{k,p}, I^\beta_{k,p}, X_{k,p}, E_{k,p}, R_{k,p}, O_{k,p}, U_{k,p}, J_{k,p} \geq 0 \]  \[ \alpha_{k,p}, \beta_{k,p} \]  \[ \text{(binary variables)} \]  \[ \beta_{k,p} \]  \[ \text{(non-negativity constraints)} \]  where \( k = 1, 2, 3, \ldots m^{(n-1)}, p = 2, 3, \ldots n \). As previously, the objective cost function minimizes the expected total cost, now over \( n \) periods, and the constraints in equations 16-25 and 31-32 act similarly as in the 2-period model. The constraints in equations 26-30 are non-anticipativity constraints to ensure that all scenarios with a common history have the same set of decisions up to the current time in the decision tree (i.e., essentially one cannot anticipate the future nor change the past). Similar to above, solving an equivalent expected value model adds an additional constraint to the model which is the expected yield for extra refurbished components:

\[ S_1 = \sum_{k=1}^{m^{(n-1)}} \sum_{p=2}^{n} (\pi_k E_{k,p}) \]  \[ \text{[33]} \]
4.3 Model Extensions

The above models can be extended to scenarios with multiple salvaged part types and products. The decision variables then become:

\[ p_I \]
Number of returned products to receive into the facility in period \( p \),

\[ I_{i,k,p}^{\alpha} \]
Number of returned products to receive (‘pull’) into the facility in period \( p \) if there exists extra components of type \( i \) in period \( p \) under scenario \( k \),

\[ I_{i,k,p}^{\beta} \]
Number of returned products to receive (‘pull’) into the facility in period \( p \) if the demand is not met for component type \( i \) in period \( p \) under scenario \( k \),

\[ S_{i,1} \]
Internal storage space allocated for the extra refurbished components per type \( i \) in period 1,

\[ R_{i,k,p} \]
Amount of space to reconfigure for excess refurbished components per type \( i \) in period \( p \) under scenario \( k \),

\[ O_{i,k,p} \]
External storage space for excess refurbished components per type \( i \) in period \( p \) under scenario \( k \),

\[ U_{i,k,p} \]
Unused internal storage space per type \( i \) in period \( p \) under scenario \( k \),

\[ J_{i,k,p} \]
Number of disposed components per type \( i \) in period \( p \) under scenario \( k \), and

\[ X_{i,k,p} \]
Unmet demand per component type \( i \) in period \( p \) under scenario \( k \).

Similarly, additional cost, demand, and capacity variables include

\[ \alpha_{k,p} \]
Binary variable if there are extra refurbished components in period \( p \) under scenario \( k \),

\[ \beta_{k,p} \]
Binary variable if demand is not met in period \( p \) under scenario \( k \),
\[ Y_{i,k,p} = \text{Remanufacturing yield percentage } k \text{ per type } i \text{ in period } p, \]
\[ \pi_k = \text{Probability of outcome } k, \]
\[ E_{i,k,p} = \text{Extra number of refurbished components per type } i \text{ in period } p, \max(0, Y_{i,k,p} I_p - D_{i,p}), \]
where \( E_{i,k,1} = 0 \)
\[ D_{i,p} = \text{Product demand per type } i \text{ in period } p, \]
\[ c_{i,s} = \text{Cost of internally storing each remaining refurbished component of type } i, \]
\[ c_{i,r} = \text{Cost of reconfiguring one unit of storage space for component type } i, \]
\[ c_{i,o} = \text{Cost of externally storing each remaining refurbished component of type } i, \]
\[ c_{i,u} = \text{Opportunity cost of each unused internal unit of space for component type } i, \]
\[ c_{i,j} = \text{Cost of disposing each remaining refurbished component of type } i, \]
\[ c_{i,x} = \text{Cost of unmet demand for component type } i, \]
\[ C^s_{i,p} = \text{Capacity for internal storage space per component type } i \text{ in period } p, \text{ and} \]
\[ C^o_{i,p} = \text{Capacity for external storage space per component type } i \text{ in period } p. \]

### 4.3.1 Two-Component and Two-Period SPR Model

These models are extended to more than one component. Denote \( i = \text{component type and } t = \text{total number of component types}. \) For the \( t = 2 \) component type case, the formulation becomes

\[
\text{Minimize } \sum_{i=1}^{2} c_{i,s} S_{i,1} + \sum_{k=1}^{3} \pi_k \left( c_{i,r} R_{i,k,2} + c_{i,o} U_{i,k,2} + c_{i,u} J_{i,k,2} + c_{i,j} X_{i,k,2} + c_{i,x} X_{i,k,2} \right) \quad [34]
\]
subject to
\[
\alpha_{i,k,2} + \beta_{i,k,2} = 1
\]  
(binary constraints)  
[35]

\[
I^\alpha_{i,k,2} + I^\beta_{i,k,2} = I_2
\]  
(number of returned products)  
[36]

\[
|I^\alpha_{i,k,2} - I^\beta_{i,k,2}| = I_2
\]  
[37]

\[
I^\alpha_{i,k,2} \cdot Y_{i,k,2} - D_{i,k,2} \cdot \alpha_{i,k,2} = E_{i,k,2}
\]  
(extra components)  
[38]

\[
-I^\beta_{i,k,2} \cdot Y_{i,k,2} + D_{i,k,2} \cdot \beta_{i,k,2} = X_{i,k,2}
\]  
(unmet demand)  
[39]

\[
S_{i,1} + R_{i,k,2} + O_{i,k,2} + J_{i,k,2} \geq E_{i,k,2}
\]  
(storage of extra components)  
[40]

\[
S_{i,1} + R_{i,k,2} + O_{i,k,2} + J_{i,k,2} - E_{i,k,2} = U_{i,k,2}
\]  
(unused space)  
[41]

\[
E_{i,k,2} - S_{i,1} - R_{i,k,2} + U_{i,k,2} \geq 0
\]  
[42]

\[
S_{i,1} + R_{i,k,2} \leq C^o_{i,k,2}
\]  
(internal space capacity)  
[43]

\[
O_{i,k,2} \leq C^o_{i,k,2}
\]  
(external space capacity)  
[44]

\[
S_{i,1} \cdot I^\alpha_{i,k,2} \cdot I^\beta_{i,k,2} \cdot X_{i,k,2} \cdot E_{i,k,2} \cdot R_{i,k,2} \cdot O_{i,k,2} \cdot U_{i,k,2} \cdot J_{i,k,2} \geq 0
\]  
(non-negativity)  
[45]

\[
\alpha_{i,k,2}, \beta_{i,k,2}
\]  
(binary variables)  
[46]

where \(k = 1, 2, 3\) and \(i = 1, 2\). Similar to above, while these constraints are given here for readability, they are redundant with constraints 40 and 42, and can be removed while implementing the model in a solver software.
4.3.2 Multi-Component and Multi-Period SPR Model

The mathematical formulation can be written for multiple components in a multi-period setting as follows:

Minimize \[
\sum_{i=1}^{l} c_{i,s} S_{i,1} + \sum_{k=1}^{m} \sum_{p}^{n-1} \sum_{i=1}^{p} \left( c_{i,r} R_{i,k,p} + c_{i,o} O_{i,k,p} + c_{i,u} U_{i,k,p} + c_{i,j} J_{i,k,p} + c_{i,x} X_{i,k,p} \right) \]

subject to

\[\alpha_{i,k,p} + \beta_{i,k,p} = 1\] (binary constraints) \[48\]

\[I^\alpha_{i,k,p} + I^\beta_{i,k,p} = I_{p}\] (number of returned products) \[49\]

\[|I^\alpha_{i,k,p} - I^\beta_{i,k,p}| = I_{p}\] \[50\]

\[I^\alpha_{i,k,p} * Y_{i,k,p} + E_{i,k,(p-1)} * \alpha_{i,k,p} - D_{i,p} * \alpha_{i,k,p} = E_{i,k,p}\] (extra components) \[51\]

\[-I^\beta_{i,k,p} * Y_{i,k,p} - E_{i,k,(p-1)} * \beta_{i,k,p} + D_{i,p} * \beta_{i,k,p} = X_{i,k,p}\] (unmet demand) \[52\]

\[S_{i,1} + R_{i,k,p} + O_{i,k,p} + J_{i,k,p} \geq E_{i,k,p}\] (storage of extra components) \[53\]

\[S_{i,1} + R_{i,k,p} + O_{i,k,p} + J_{i,k,p} - E_{i,k,p} = U_{i,k,p}\] (unused space) \[54\]

\[E_{i,k,p} - S_{i,1} - R_{i,k,p} + U_{i,k,p} \geq 0\] \[55\]

\[S_{i,1} + R_{i,k,p} \leq C^s_{i,p}\] (internal space capacity) \[56\]

\[O_{i,k,p} \leq C^o_{i,p}\] (external space capacity) \[57\]

\[R_{i,k,p} = R_{i,k+1,p} = \ldots = R_{i,m^n-p,p} = R_{i,m^n-p+1,p} = R_{i,m^n-p+2,p} = \ldots = R_{i,m^{p-1}m^n-p,n-1}\] \[58\]

\[O_{i,k,p} = O_{i,k+1,p} = \ldots = O_{i,m^n-p,p} = O_{i,m^n-p+1,p} = O_{i,m^n-p+2,p} = \ldots = O_{i,m^{p-1}m^n-p,n-1}\] \[59\]
Chapter 4 A Modeling Approach: Stochastic Programming Recourse Models

\[ U_{i,k,p} = U_{i,k+1,p} = \ldots = U_{i,m^{n-p},p} \ldots U_{i,m^{p-1},p} = U_{i,m^{n-p},m^{n-p}} \] [60]

\[ J_{i,k,p} = J_{i,k+1,p} = \ldots = J_{i,m^{n-p},p} \ldots J_{i,m^{p-1},p} = J_{i,m^{n-p},m^{n-p}} \] [61]

\[ X_{i,k,p} = X_{i,k+1,p} = \ldots = X_{i,m^{n-p},p} \ldots X_{i,m^{p-1},p} = X_{i,m^{n-p},m^{n-p}} \] [62]

\[ S_{i,k,p}, I^{\alpha}_{i,k,p} I^{\beta}_{i,k,p}, X_{i,k,p}, E_{i,k,p}, R_{i,k,p}, O_{i,k,p}, U_{i,k,p}, J_{i,k,p} \geq 0 \] (non-negativity) [63]

\[ \alpha_{i,k,p}, \beta_{i,k,p} \] (binary variables) [64]

where \( k = 1, 2, 3, \ldots m^{(n-1)}, p = 2, 3, \ldots n, \ i = 1, 2, 3, \ldots t \). Non-anticipativity constraints (58-62) are needed to ensure that all scenarios with a common history have the same set of decisions up to the current time period in the decision tree. Similar to above, while these constraints are given here for readability, they are redundant with constraints 53 and 55, and can be removed when implementing the model in a solver software.

4.3.3 Multiple Products with Overlapping Parts in a Multi-Period Setting

When more than one product is remanufactured in a facility and these products have common components, let’s denote \( z = \) product type, \( w = \) total number of product types:

Minimize \[ \sum_{i=1}^{m} \sum_{k=1}^{n} \sum_{p=1}^{m} \left( c_{i,k,p} R_{i,k,p} + c_{i,o} O_{i,k,p} + c_{i,u} U_{i,k,p} + c_{i,j} J_{i,k,p} + c_{i,x} X_{i,k,p} \right) \] [64]

subject to

\[ \alpha_{i,k,p} + \beta_{i,k,p} = 1 \] (binary constraint) [65]
\[ I^{\alpha, z}_{i, k, p} + I^{\beta, z}_{i, k, p} = I^z_p \]  
(number of returned products)  \[66\]

\[ I^{\alpha, z}_{i, k, p} - I^{\beta, z}_{i, k, p} = I^z_p \]  \[67\]

\[ \left( \sum_{z=1}^{w} I^{\alpha, z}_{i, k, p} * Y^z_{i, k, p} \right) + E_{i, k, (p-1)} * \alpha_{i, p} - D_{i, p} * \alpha_{i, k, p} = E_{i, k, p} \]  
(extra components)  \[68\]

\[ \left( \sum_{z=1}^{w} -I^{\beta, z}_{i, k, p} * Y^z_{i, k, p} \right) - E_{i, k, (p-1)} * \beta_{i, p} + D_{i, p} * \beta_{i, k, p} = X_{i, k, p} \]  
(unmet demand)  \[70\]

\[ S_{i,1} + R_{i, k, p} + O_{i, k, p} + J_{i, k, p} \geq E_{i, k, p} \]  
(storage of extra components)  \[71\]

\[ S_{i,1} + R_{i, k, p} + O_{i, k, p} + J_{i, k, p} - E_{i, k, p} = U_{i, k, p} \]  
(unused space)  \[72\]

\[ E_{i, k, p} - S_{i,1} - R_{i, k, p} + U_{i, k, p} \geq 0 \]  \[73\]

\[ S_{i,1} + R_{i, k, p} \leq C^{s}_{i, p} \]  
(internal space capacity)  \[74\]

\[ O_{i, k, p} \leq C^{o}_{i, p} \]  
(external space capacity)  \[75\]

\[ R_{i, k, p} = R_{i, k+1, p} = ... = R_{i, m^{n-p}, p} \ldots R_{i, m^{n-p}+1, p} = R_{i, m^{n-p}+2, p} = ... = R_{i, m^{p-1}m^{n-p}, n-1} \]  \[76\]

\[ O_{i, k, p} = O_{i, k+1, p} = ... = O_{i, m^{n-p}, p} \ldots O_{i, m^{n-p}+1, p} = O_{i, m^{n-p}+2, p} = ... = O_{i, m^{p-1}m^{n-p}, n-1} \]  \[77\]

\[ U_{i, k, p} = U_{i, k+1, p} = ... = U_{i, m^{n-p}, p} \ldots U_{i, m^{n-p}+1, p} = U_{i, m^{n-p}+2, p} = ... = U_{i, m^{p-1}m^{n-p}, n-1} \]  \[78\]

\[ J_{i, k, p} = J_{i, k+1, p} = ... = J_{i, m^{n-p}, p} \ldots J_{i, m^{n-p}+1, p} = J_{i, m^{n-p}+2, p} = ... = J_{i, m^{p-1}m^{n-p}, n-1} \]  \[79\]

\[ X_{i, k, p} = X_{i, k+1, p} = ... = X_{i, m^{n-p}, p} \ldots X_{i, m^{n-p}+1, p} = X_{i, m^{n-p}+2, p} = ... = X_{i, m^{p-1}m^{n-p}, n-1} \]  \[80\]

\[ S_{i,1} + I^z_p I^{\alpha, z}_{i, k, p} + I^{\beta, z}_{i, k, p} X_{i, k, p} E_{i, k, p} R_{i, k, p} O_{i, k, p} U_{i, k, p} J_{i, k, p} \geq 0 \]  
(non-negativity)  \[81\]

\[ \alpha_{i, k, p} \beta_{i, k, p} \]  
(binary variables)  \[82\]
where \( k = 1, 2, 3, \ldots m^{(n-1)}, p = 2, 3, 4, \ldots n, i = 1, 2, 3, \ldots t, \) and \( z = 1, 2, 3, \ldots w. \) Non-anticipativity constraints (76-80) are needed to ensure that all scenarios with a common history have the same set of decisions up to the current time period in the decision tree. Similar to above, while these constraints are given here for readability, they are redundant with constraints 71 and 73, and can be removed when implementing the model in a solver software.

### 4.4 Model Performance and Results

A cell phone remanufacturing company collects used phones, which are then disassembled to salvage three components: the printed circuit board, the antenna and a lithium-ion battery. As an example, assume that the company only needs to store the antennas. It costs the company $2 to store each unit of antennas internally. Extra refurbished antennas can also be stored by reconfiguring some of the available space allocated to store other components. The reconfiguration cost is $6. Storing each unit externally costs $10. The opportunity cost of not using the allocated space is $2. If there is not enough internal or external space to accommodate the extra refurbished antennas, then they are disposed of. The disposal cost is $16. On the other hand, if the demand for the refurbished antennas is not met, the penalty cost for unmet demand is $20. These costs do not reflect the real numbers, and are just chosen for analysis. Table 4.1 summarizes the results for the following given data:

\[
P(Y_{1,2} = 0.6) = 0.1, \ P(Y_{2,2} = 0.9) = 0.6, \ P(Y_{3,2} = 0.75) = 0.3, \ D_2 = 40, \ C_2^x = 50, \ C_2^o = 10,
\]
\[
c_s = $2, c_r = $6, c_o = $10, c_u = $2, c_f = $16, c_d = $20
\]

Solving the same model with the expected value formulation produces different results for the recourse variables and an optimal internal storage allocation of 4.8 spaces and a $40.96
Chapter 4 A Modeling Approach: Stochastic Programming Recourse Models

minimum expected total cost (6.7% higher due to the assumption of perfect information). The stochastic programming formulations by Lingo® are provided in Appendix A.

Table 4.1 Comparison of two-period mixed-integer SPR vs. EV formulations (Recourse variables are shown in blue)

<table>
<thead>
<tr>
<th>Model Variables (D = Decision, R = Random)</th>
<th>Type of variable</th>
<th>Period</th>
<th>Stochastic Programming Scenario k</th>
<th>Expected Value Scenario k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Internal storage space</td>
<td>D</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Number of returned products</td>
<td>D</td>
<td>2</td>
<td>53.3</td>
<td></td>
</tr>
<tr>
<td>Demand</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Component yield</td>
<td>R</td>
<td>2</td>
<td>60%</td>
<td>90%</td>
</tr>
<tr>
<td>Extra number of components</td>
<td>R</td>
<td>2</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Reconfigured storage space</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External storage space</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unused storage space</td>
<td>D</td>
<td>2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Disposed components</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Penalty</td>
<td>D</td>
<td>2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Total Expected Cost</td>
<td></td>
<td></td>
<td>$38.4</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3 Yield probabilities for three-period SPR model

Table 4.2 Possible 3-period yield scenarios
With $n \geq 3$ periods, “non-anticipativity” constraints (essentially meaning that one cannot anticipate the future nor change the past) also are needed to ensure all scenarios with a common history have the same set of decisions up to the current time in the decision tree. As an example, Figure 4.3 illustrates an $n = 3$ period model with demand and internal, reconfigured, external, and opportunity storage costs assumed constant in all periods.

Similarly discretizing the probability space to $m = 3$ yields in each period results in the $m^{n-1} = 3^2 = 9$ possible scenarios shown in Table 4.2, with the “non-anticipativity” constraints here being

\[
R_{1,2} = R_{2,2} = R_{3,2} \quad R_{4,2} = R_{5,2} = R_{6,2} \quad R_{7,2} = R_{8,2} = R_{9,2} \quad [83]
\]

\[
O_{1,2} = O_{2,2} = O_{3,2} \quad O_{4,2} = O_{5,2} = O_{6,2} \quad O_{7,2} = O_{8,2} = O_{9,2} \quad [84]
\]

\[
U_{1,2} = U_{2,2} = U_{3,2} \quad U_{4,2} = U_{5,2} = U_{6,2} \quad U_{7,2} = U_{8,2} = U_{9,2} \quad [85]
\]

\[
J_{1,2} = J_{2,2} = J_{3,2} \quad J_{4,2} = J_{5,2} = J_{6,2} \quad J_{7,2} = J_{8,2} = J_{9,2} \quad [86]
\]

\[
X_{1,2} = X_{2,2} = X_{3,2} \quad X_{4,2} = X_{5,2} = X_{6,2} \quad X_{7,2} = X_{8,2} = X_{9,2} \quad [87]
\]

Again, an initial decision is made about how much internal storage space to allocate based on all available information and scenario probabilities. After the second period, the component yield for that period becomes known and a second decision is made for the recourse variables – the amounts of reconfigured, external storages, and disposal to accommodate this yield, as well as the unused storage space and unmet demand for the current period, and the allocation of internal storage to prepare for the third period. Once the component yield in the third period becomes known, a decision is made again for the recourse variables and the number of returned products to pull into the facility.
Table 4.3 summarizes the results for the 3-period example and the relative improvement of the SPR model, with the number of products returned for remanufacturing being 62 and 67 in the second and third period, respectively, and a total expected cost of $83.93 versus $130.5 (a 48% increase). Table 4.4 and 4.5 summarize the results based on the following model inputs with number of periods up to 6:

\[ D_p = 40, C_p^{i} = 50, C_p^{o} = 10, c_s = 2, c_r = 2, c_o = 10, c_u = 2, c_j = 16, c_x = 20 \]

Table 4.3 Comparison of mixed-integer single component three-period SPR vs. EV formulations

<table>
<thead>
<tr>
<th>Model Variables (D = Decision, R = Random)</th>
<th>Type of variable</th>
<th>Period</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of outcomes</td>
<td>-</td>
<td>-</td>
<td>0.02 .05 .03 .06 .24 .3 .24 .03 .03</td>
</tr>
<tr>
<td>Internal storage space</td>
<td>D</td>
<td>1</td>
<td>16 (30.4)</td>
</tr>
<tr>
<td>Yield</td>
<td>R</td>
<td>2</td>
<td>60% 90% 75%</td>
</tr>
<tr>
<td>Demand</td>
<td>R</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>Number of returned products</td>
<td>D</td>
<td>2</td>
<td>62.2 (61.6)</td>
</tr>
<tr>
<td>Extra number of components</td>
<td>R</td>
<td>2</td>
<td>0 (0) 16 (15.4) 6.7 (6.2)</td>
</tr>
<tr>
<td>Reconfigured storage space</td>
<td>D</td>
<td>2</td>
<td>0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>External storage space</td>
<td>D</td>
<td>2</td>
<td>0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Unused storage space</td>
<td>D</td>
<td>2</td>
<td>16 (30.4) 0 (15.0) 9.3 (24.2)</td>
</tr>
<tr>
<td>Disposed components</td>
<td>D</td>
<td>2</td>
<td>0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Penalty</td>
<td>D</td>
<td>2</td>
<td>2.7 (3.0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Yield</td>
<td>R</td>
<td>3</td>
<td>75 60 90 90 65 55 95 75 50</td>
</tr>
<tr>
<td>Demand</td>
<td>R</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>Number of returned products</td>
<td>D</td>
<td>3</td>
<td>66.7 (67.6)</td>
</tr>
<tr>
<td>Extra number of components</td>
<td>R</td>
<td>3</td>
<td>10 (10.7) 0 (0.6) 20 (20.8) 36 (36.3) 19.3 (19.4) 12.7 (12.6) 30 (30.4) 16.7 (16.9) 0 (0)</td>
</tr>
<tr>
<td>Reconfigured storage space</td>
<td>D</td>
<td>3</td>
<td>0 (0) 0 (0) 4 (0) 20 (5.9) 3.3 (0) 0 (0) 14 (0) 0.7 (0) 0 (0)</td>
</tr>
<tr>
<td>External storage space</td>
<td>D</td>
<td>3</td>
<td>0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Unused storage space</td>
<td>D</td>
<td>3</td>
<td>6 (19.7) 16 (29.9) 0 (9.6) 0 (0) 0 (0) 3.3 (17.8) 0 (0) 0 (13.5) 16 (30.4)</td>
</tr>
<tr>
<td>Disposed components</td>
<td>D</td>
<td>3</td>
<td>0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Penalty</td>
<td>D</td>
<td>3</td>
<td>0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0) 0 (0)</td>
</tr>
<tr>
<td>Total Expected Cost</td>
<td></td>
<td></td>
<td>$83.93 ($130.5)</td>
</tr>
</tbody>
</table>
Table 4.4 Mixed-integer single component SPR model - Summary of runtime and objective value

<table>
<thead>
<tr>
<th>Period</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>32</td>
<td>183</td>
<td>814</td>
<td>3245</td>
<td>12156</td>
</tr>
<tr>
<td>Number of constraints</td>
<td>$31 (6)^*</td>
<td>211 (36)</td>
<td>1024 (162)</td>
<td>4385 (648)</td>
<td>16416 (2430)</td>
</tr>
<tr>
<td>Mean Demand</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>19.2</td>
<td>0.05</td>
<td>42.0</td>
<td>0.30</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>38.4</td>
<td>0.05</td>
<td>83.9</td>
<td>0.22</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>57.6</td>
<td>0.03</td>
<td>130.2</td>
<td>1.05</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>76.8</td>
<td>0.05</td>
<td>187.1</td>
<td>0.22</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>96.0</td>
<td>0.08</td>
<td>275.8</td>
<td>0.87</td>
</tr>
</tbody>
</table>

*The numbers in parentheses show the number of redundant constraints.

Table 4.5 Mixed-integer single component EV model - Summary of runtime and objective value

<table>
<thead>
<tr>
<th>Period</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>32</td>
<td>183</td>
<td>814</td>
<td>3245</td>
<td>12156</td>
</tr>
<tr>
<td>Number of constraints</td>
<td>32</td>
<td>212</td>
<td>1025</td>
<td>4386</td>
<td>16417</td>
</tr>
<tr>
<td>Mean Demand</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>20.5</td>
<td>0.05</td>
<td>62.0</td>
<td>0.02</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>41.0</td>
<td>0.02</td>
<td>130.5</td>
<td>0.22</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>61.4</td>
<td>0.08</td>
<td>196.9</td>
<td>0.33</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>81.9</td>
<td>0.07</td>
<td>267.7</td>
<td>0.43</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>102.4</td>
<td>0.02</td>
<td>282.8</td>
<td>0.27</td>
</tr>
</tbody>
</table>

If the company wants to calculate the total cost of storing the printed circuit board and lithium-ion battery, then the unit internal storage costs are $3 and $2, respectively. The results are based on the following other inputs:

\[
P(Y_{1,1,2} = 0.6) = 0.1, P(Y_{1,2,2} = 0.9) = 0.6, P(Y_{1,3,2} = 0.75) = 0.3
\]

\[
P(Y_{2,1,2} = 0.4) = 0.1, P(Y_{2,2,2} = 0.85) = 0.6, P(Y_{2,3,2} = 0.65) = 0.3
\]

\[D_{t,p} = 40, C_{i,p}^{x} = 50, C_{i,p}^{a} = 10,\]
\[ c_{1,s} = 3, c_{1,r} = 6, c_{1,o} = 10, c_{1,u} = 3, c_{1,j} = 16, c_{1,x} = 20 \]
\[ c_{2,s} = 2, c_{2,r} = 5, c_{2,o} = 8, c_{2,u} = 2, c_{2,j} = 14, c_{2,x} = 18 \]

Table 4.6 summarizes the results for the SPR and expected value models, with the number of products returned for remanufacturing being 62 and 67 respectively, and a total expected cost of $121.8 versus $141.8. The results for models with up to 6 periods are shown in Table 4.7 and 4.8.

Table 4.6 Comparison of mixed-integer two-component SPR vs. EV formulations in a two-period setting

<table>
<thead>
<tr>
<th>Model Variables (D = Decision, R = Random)</th>
<th>Type of variable</th>
<th>Period</th>
<th>Stochastic Programming</th>
<th>Expected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal storage space for type 1</td>
<td>D</td>
<td>1</td>
<td>6.2</td>
<td>15</td>
</tr>
<tr>
<td>Internal storage space for type 2</td>
<td>D</td>
<td>1</td>
<td>12.3</td>
<td>11</td>
</tr>
<tr>
<td>Number of returned products</td>
<td>D</td>
<td>2</td>
<td>61.5</td>
<td>66.7</td>
</tr>
<tr>
<td>Demand for type 1</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Demand for type 2</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Component yield for type 1</td>
<td>R</td>
<td>2</td>
<td>60% 90% 75%</td>
<td>60% 90% 75%</td>
</tr>
<tr>
<td>Component yield for type 2</td>
<td>R</td>
<td>2</td>
<td>40% 85% 65%</td>
<td>40% 85% 65%</td>
</tr>
<tr>
<td>Extra number of components for type 1</td>
<td>R</td>
<td>2</td>
<td>0 15.4 6.2</td>
<td>0 20 10</td>
</tr>
<tr>
<td>Extra number of components for type 2</td>
<td>R</td>
<td>2</td>
<td>0 12.3 0</td>
<td>0 16.7 3.3</td>
</tr>
<tr>
<td>Reconfigured storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>0 9.2 0</td>
<td>0 5 0</td>
</tr>
<tr>
<td>Reconfigured storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 5.7 0</td>
</tr>
<tr>
<td>External storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>External storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Unused storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>12.3 0 12.3</td>
<td>11 0 7.7</td>
</tr>
<tr>
<td>Unused storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Disposed components of type 1</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Disposed components of type 2</td>
<td>D</td>
<td>2</td>
<td>0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Penalty for type 1</td>
<td>D</td>
<td>2</td>
<td>3.1 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>Penalty for type 2</td>
<td>D</td>
<td>2</td>
<td>15.4 0 0</td>
<td>13.3 0 0</td>
</tr>
<tr>
<td>Total Expected Cost</td>
<td></td>
<td></td>
<td>$121.8</td>
<td>$141.8</td>
</tr>
</tbody>
</table>
Let’s assume that the company receives two types of cell phones. Each cell phone type has the same type circuit board, but different type of battery. Circuit boards can be stored together, whereas batteries are different, and stored separately. The subscript following $P$ denotes the product type, two types being $A$ and $B$. Table 4.9 summarizes the results for the SPR and expected value models, with a total expected cost of $367.7 versus $373.5, respectively.
### Table 4.9 Comparison of mixed-integer two-product with overlapping parts SPR vs. EV formulations in a two-period setting

<table>
<thead>
<tr>
<th>Model Variables (D = Decision, R = Random)</th>
<th>Type of variable</th>
<th>Period</th>
<th>Stochastic Programming Scenario k</th>
<th>Expected Value Scenario k</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Internal storage space for type 1</td>
<td>D</td>
<td>1</td>
<td>27.1</td>
<td>0</td>
</tr>
<tr>
<td>Internal storage space for type 2</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Internal storage space for type 3</td>
<td>D</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of returned product type A</td>
<td>D</td>
<td>2</td>
<td>47.1</td>
<td>47.1</td>
</tr>
<tr>
<td>Number of returned product type B</td>
<td>D</td>
<td>2</td>
<td>45.5</td>
<td>45.5</td>
</tr>
<tr>
<td>Demand for component type 1</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Demand for component type 2</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Demand for component type 3</td>
<td>R</td>
<td>2</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>Component yield for type 1</td>
<td>R</td>
<td>2</td>
<td>60% (55%)</td>
<td>90% (95%)</td>
</tr>
<tr>
<td>Component yield for type 2</td>
<td>R</td>
<td>2</td>
<td>(-) (35%)</td>
<td>(-) (88%)</td>
</tr>
<tr>
<td>Component yield for type 3</td>
<td>R</td>
<td>2</td>
<td>40% (+)</td>
<td>85% (+)</td>
</tr>
<tr>
<td>Extra number of component type 1</td>
<td>R</td>
<td>2</td>
<td>13.2</td>
<td>45.5</td>
</tr>
<tr>
<td>Extra number of component type 2</td>
<td>R</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Extra number of component type 3</td>
<td>R</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reconfigured storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>18.4</td>
</tr>
<tr>
<td>Reconfigured storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reconfigured storage space for type 3</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>External storage space for type 3</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unused storage space for type 1</td>
<td>D</td>
<td>2</td>
<td>13.9</td>
<td>0</td>
</tr>
<tr>
<td>Unused storage space for type 2</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Unused storage space for type 3</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disposed components of type 1</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disposed components of type 2</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disposed components of type 3</td>
<td>D</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Penalty for type 1</td>
<td>D</td>
<td>2</td>
<td>24.1</td>
<td>0</td>
</tr>
<tr>
<td>Penalty for type 2</td>
<td>D</td>
<td>2</td>
<td>21.2</td>
<td>9.4</td>
</tr>
<tr>
<td>Penalty for type 3</td>
<td>D</td>
<td>2</td>
<td>$367.7</td>
<td>$373.5</td>
</tr>
</tbody>
</table>

Total Expected Cost

$367.7

$373.5
Table 4.10 Mixed-integer two-product with overlapping parts SPR model - Summary of runtime and objective value

<table>
<thead>
<tr>
<th>Period</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>113</td>
<td>655</td>
<td>2925</td>
<td>11,675</td>
<td>43,753</td>
</tr>
<tr>
<td>Number of constraints</td>
<td>111 (18)</td>
<td>743 (108)</td>
<td>3553 (486)</td>
<td>14,733 (1944)</td>
<td>56,531 (7290)</td>
</tr>
<tr>
<td>Mean Demand</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
</tr>
<tr>
<td>20</td>
<td>0.02</td>
<td>183.9</td>
<td>0.17</td>
<td>486.6</td>
<td>1.17</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>367.7</td>
<td>1.25</td>
<td>1023.3</td>
<td>1.02</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>625.4</td>
<td>1.45</td>
<td>1860.7</td>
<td>1.77</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>959.8</td>
<td>2.18</td>
<td>2764.1</td>
<td>1.55</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1680.8</td>
<td>0.87</td>
<td>3402.9</td>
<td>1.12</td>
</tr>
</tbody>
</table>

Table 4.11 Mixed-integer two-product with overlapping parts EV model - Summary of runtime and objective value

<table>
<thead>
<tr>
<th>Period</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of variables</td>
<td>113</td>
<td>655</td>
<td>2925</td>
<td>11,675</td>
<td>43,753</td>
</tr>
<tr>
<td>Number of constraints</td>
<td>114</td>
<td>746</td>
<td>3556</td>
<td>14,736</td>
<td>56,534</td>
</tr>
<tr>
<td>Mean Demand</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
<td>Objective Value ($)</td>
<td>Runtime (min)</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>186.8</td>
<td>1.78</td>
<td>582.3</td>
<td>11.5</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>373.5</td>
<td>1.28</td>
<td>1695.6</td>
<td>6.20</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>662.9</td>
<td>2.60</td>
<td>2693.3</td>
<td>0.35</td>
</tr>
<tr>
<td>80</td>
<td>0.02</td>
<td>1088.8</td>
<td>1.98</td>
<td>3636.8</td>
<td>47.28</td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1514.7</td>
<td>3.62</td>
<td>4670.2</td>
<td>478.1</td>
</tr>
</tbody>
</table>

*Due to the long computation time, the six-period, two-product with overlapping parts SPR and EV models were not computed.

Table 4.10 and 4.11 summarize the results for models with up to 5 periods for the following given data:

\[
P_A(Y_{1,2} = 0.6) = 0.1, P_A(Y_{1,2} = 0.9) = 0.6, P_A(Y_{1,3} = 0.75) = 0.3
\]
\[
P_A(Y_{2,1} = 0.4) = 0.1, P_A(Y_{2,2} = 0.85) = 0.6, P_A(Y_{2,3} = 0.65) = 0.3
\]
\[
P_B(Y_{1,1} = 0.55) = 0.1, P_B(Y_{1,2} = 0.95) = 0.6, P_B(Y_{1,3} = 0.7) = 0.3
\]
\begin{align*}
P_B (Y_{2,1,2} = 0.35) &= 0.1, P_B (Y_{2,2,2} = 0.88) = 0.6, P_B (Y_{2,3,2} = 0.62) = 0.3 \\
D_{i,p} &= 40, C^s_{i,p} = 50, C^o_{i,p} = 10, \\
c_{1,s} &= \$3, c_{1,o} = \$6, c_{1,u} = \$10, c_{1,f} = \$16, c_{1,x} = \$20 \\
c_{2,s} &= \$2, c_{2,o} = \$5, c_{2,u} = \$8, c_{2,f} = \$2, c_{2,x} = \$14, c_{2,x} = \$18 \\
c_{3,s} &= \$2, c_{3,r} = \$7, c_{3,o} = \$11, c_{3,u} = \$2, c_{3,f} = \$17, c_{3,x} = \$22
\end{align*}

4.5 Discussion

The above models were implemented for 2 to 6 periods in Lingo® assuming random demand and supply, upper limits for internal and external storage in all periods and component types of $C^s_{i,p} = 50$ and $C^o_{i,p} = 10$ respectively, and all the other input data as previously. Figure 4.4-4.6 illustrates how model size, computation duration, and optimal solutions change as a function of the number of periods, assuming both random demand and supply. As shown in Figure 4.4, the number of constraints and variables increases dramatically as more periods are considered. Even a six-period model with a single component, for example, results in 16,416 constraints and 12,156 variables. When a model with two components is considered in a six-period setting, the number of constraints and variables increases to 32,826 and 24,307, respectively. The model size increases dramatically, resulting in 43,753 variables and 56,531 constraints for a two-product with overlapping parts SPR model.

92
Chapter 4 A Modeling Approach: Stochastic Programming Recourse Models

Figure 4.4 Increase in number of constraints and variables

Figure 4.5 Comparison of objective values
As previously, the expected value formulation results in a much more expensive solution, underscoring the value of SPR approaches as illustrated in Figure 4.5. As an example, when the mean demand is 40 per period, the total cost is $2433.0 for a five-period, two-product with overlapping parts SPR model, whereas the expected value formulation results in a higher cost of $2970.0. In most cases, the expected value formulation for the two-product with overlapping parts model is 12.7% higher than the SPR formulation. For the single and two-component models, it is 47.6% and 19.7% higher, respectively. As shown in Figure 4.6, the increase in run times is the most significant when the number of periods increases. As an example, in the single-component SPR model, the run time increases from 4.7 to 10,779 minutes when the model size increases from 5 to 6 periods. A similar jump in run time is observed in the two-component model as well (from 41.5 to 31,130.4 minutes). Hence, it is efficient to use the SPR approach in
the single and two-component cases for up to 5 periods. Because of the long computation
duration, two-product with overlapping parts SPR and EV models were set up for up to 5
periods. The dramatic increase from 1.02 to 15,994.6 minutes when the model size increases
from 4 to 5 periods shows that SPR approach can be used solving the two-product with
overlapping parts model for up to 4 periods.

Given the likely size of realistic models, heuristic and non-greedy solution approaches therefore
might be used, such as reducing the SPR to a linear program with upper-bounded variables or
use of network flow solution methods, heuristics based on cutting planes, genetic algorithms,
tabu search, or simulated annealing.

### 4.6 Drawbacks of Stochastic Programming

As previously noted, a simpler expected value formulation for this problem results in a much
higher minimum cost solution, underscoring the potential value in this type of modeling
approach. When the SPR model is further extended to two products with interdependent parts in
a multi-period setting, the number of variables increases significantly due to the explosion of
possible scenarios and size of the model. In order to overcome this complexity, once a large-
scale model is formulated something along the above lines, then heuristic methodologies such as
tabu search, genetic algorithms or simulated annealing might be used to overcome the resulting
problems of solving large combinatorial optimization models.
4.7 Conclusions

As previously, SPR approaches result in a much less total cost than the expected value formulation. In most cases, the solution with expected value formulation is 26.7% more expensive, underscoring the value of SPR approaches. However, the drawback of both these approaches is that the model size increases significantly due to the explosion of possible scenarios and the number of variables and constraints when it is extended to two products with interdependent parts in a multi-period setting. As illustrated previously, the six-period SPR model results in 43,753 variables and 56,531 constraints. Apart from this drawback, dramatic increases in run times are observed with higher number of periods. The run times for the single and two-component models increases from 4.7 to 10,779 minutes and from 41.5 to 31,130.4 minutes, respectively when the model size increases from 5 to 6 periods. In a two-product with overlapping parts case, increasing the number of periods from 4 to 5 periods results in a dramatic increase in run time from 1.02 to 15,994.6 minutes. Using heuristic methodologies is recommended for settings higher than 5 periods in the single and two-component models, and 4 periods in the two-product with overlapping parts model.

The main purpose of the SPR approach was to illustrate the implications of randomness in remanufacturing processes on facility design performance and space requirements and to illustrate general potential modeling approaches. Other possible modeling approaches that can address unique characteristics of remanufacturing include chance-constraints, option-based models, and no doubt others. Chapter 5 will cover the simulation of a more detailed remanufacturing model which demonstrates the effects of yield rates on storage space requirements and variability.
Chapter 5 – Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

The goal of this chapter is to develop a heuristic approach based on multi-dimensional golden section simulation-optimization to identify minimum cost policies for storage capacity allocation decisions. A simulation program is first used to emulate a generalized remanufacturing facility with random receiving patterns, component yields, and refurbished demand over multiple time periods, with spare capacity being reconfigured for other needs at a specified cost following a set of reconfiguration logic rules. A multi-dimensional golden section search algorithm is implemented to identify optimal storage capacities and reconfiguration decisions in each time period that minimize long-term expected total cost. Due to the long runtimes, a heuristic approach based on this algorithm is developed to solve the model in a more efficient way. The results show that under this approach the runtimes are successfully decreased by 78% with only 0.7% increase in total cost.

5.1 Model Definition and Characteristics

Figure 5.1 illustrates a generalized flow of recaptured material through a remanufacturing system and its various storage types. When a returned item arrives at a remanufacturing facility, the condition of its components usually is not fully known until the product is disassembled. After disassembly, parts typically are tested or inspected to determine their functionality, and then put
into storage. Uncertainty as to what the part yield rates are from each process in the system creates uncertainty in how much storage space should be allocated for each part type, which in turn impacts the layout of a remanufacturing facility. Other factors that influence the layout are characteristics (e.g. dimensions/size, volume, weight, fragility) of the recovered and remanufactured units, work to be provided on the returned products, and storage requirements (e.g. temperature or humidity range).

A fundamental characteristic of a remanufacturing environment is the inbound flow of used products, where most of the components or subassemblies will have probabilistic yields in the sense that not all parts will be suitable for re-use. This stochastic nature of returned products
affects predictability, safety stock, production targets, rework and waste (such as additional cleaning, testing, inspection, and reassembly). Required materials or equipment, such as cleaning solvents or particular tools, depend on the condition and nature of a part, resulting in product flow and equipment needs that are uncertain until the used products arrive at a facility. Both returned and remanufactured products, and reclaimed parts need to be stored in inventory until they are needed to satisfy demand. All these sources of uncertainty in the need for space make the design of a remanufacturing facility more complex.

5.2 Storage Needs and Capacities

As product flows through the remanufacturing process, items or components are stored in several locations. Generally upon receipt, returned products initially are routed to returned product storage, possibly with excess product stored externally (at a cost). After disassembly, the constituent parts are sorted into usable or disposable items, with the former stored in usable part storage until they are inspected and tested; parts exceeding this storage capacity are sent to external storage. Disposable parts exceeding the disposal shipping capacity are held at an external storage. After inspection, parts requiring repair are held in repairable part storage, accepted parts are routed to finished part storage, and non-repairable parts are sent to disposable part storage. If there is not enough space to store repairable parts, space is reallocated from elsewhere for this purpose, here from usable part storage, if capacity exists, with any overflow stored externally.

Once the necessary work on repairable parts is performed, they are held in finished part storage. In each time period, demand for remanufactured products either is completely met by some or all
of the available finished parts or additional new parts are purchased from an OEM. If demand is less than the available finished parts, the excess remains in finished part inventory until the next time period. However, if the latter is true, the new parts are ordered in batches of one or more (user specified). Extra new parts are stored either by reconfiguring space from finished parts inventory or externally. Finally, after assembly is complete, remanufactured products are stored in *remanufactured product inventory* before being shipped to the customer. If the capacity of remanufactured product inventory is less than the number of remanufactured products, then the excess is stored externally and shipped to the customer from that external storage. Shipments to the customers are made every period. Note that throughout this process storage space for a particular part type can be reallocated to the same use for a different part type or to another use. Each time period begins with the arrival of returned products in batches and ends when the demand for remanufactured products for that period is met. Figure 5.2 illustrates these material flows in and out of each of seven types of storage spaces.
5.3 Modeling Framework

In order to determine optimal storage capacities in each time period that minimize total expected cost, a simulation-optimization approach shown in Figure 5.3 was developed. A Monte Carlo program emulates the general logic and stochastic events as material flows through the remanufacturing facility, including random return volumes, component yields, and refurbished demand. The model runs for a user-specified number of time periods and replications, tracking the overall costs of total, internal, reconfigured, external, and unused space for any given set of inputs (storage capacities, costs). This simulation program can be used either to conduct what-if analysis or within an optimization framework. For the later, a multi-dimensional golden section search algorithm was developed that iteratively calls the Monte Carlo program, passing it a vector of storage capacities for evaluation, and repeating this logic to search for the combination of each decision variable (storage capacities) that minimizes long-term expected total cost.

Hybrid simulation-optimization has been used in similar cases where the stochastic and combinatorial aspects of the problem make other approaches somewhat intractable. Prior work discussed in Topcu et al. (2008) develops stochastic recourse models to determine optimal storage capacities and reconfiguration schedules which proved to be somewhat intractable for realistically-sized problems. The model size (e.g. the number of constraints and variables) and the computation duration both increase dramatically as a function of the number of periods. Given the likely size of realistic models, the use of non-greedy heuristic and solution approaches would be recommended - such as reducing the SPR to a linear program with upper-bounded variables, or the use of network flow solution methods, heuristics based on cutting planes, genetic algorithms, tabu search, or simulated annealing. Accounting for variability is important in returned product volumes and the randomness in yield rates from disassembly, inspection,
repair, and cleaning activities as illustrated by Topcu et al. (2007). Even a simple example for a product consisting of three components causes capacity requirements to fluctuate significantly. All these sources of randomness affect the inventory accumulation, resulting in capacity needs that are unknown and can substantially impact the layout of a remanufacturing facility.

![Figure 5.3 Overall optimization framework](image)

Table 5.1 summarizes the controllable and uncontrollable variables inherent in these types of systems, where the subscripts $p$, $t$, and $r$ indicate the part type ($1,\ldots, P$), time period ($1,\ldots, T$), and simulation replication ($1,\ldots, R$), respectively, and with uncontrollable variables highlighted in gray including yield percentages, number of returned products per time period, and demand for remanufactured items. Overall yield rates for each recaptured component type are functions of four specific yield rates: usable parts, repairable parts, repaired parts, and acceptable parts. The usable parts are the good parts upon disassembly, whereas the acceptable parts are upon inspection and testing. The decision variables are the capacities that are shaded with diagonal lines in Table 5.1. Process performance measures include space (internal, reconfigured, external, etc.).
unused) and associated storage cost (total, internal, reconfigured, external) measures, where total cost includes all storage costs over all periods plus the opportunity cost of unused space:

\[ TC_{p,t} = IC_{p,t} + RC_{p,t} + EC_{p,t} + US_{p,t} \]  \[1\]

### Table 5.1 Variables in remanufacturing and storage processes

<table>
<thead>
<tr>
<th>Returned products</th>
<th>Yield rate</th>
<th>Number of products or parts (without spare parts)</th>
<th>Spare parts</th>
<th>Total number of products or parts (with spare parts)</th>
<th>Unused space</th>
<th>Parts stored in reconfigured space</th>
<th>Storage capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( X^*_{RT,t} )</td>
<td>( e_{RT,t} )</td>
<td>( X_{RT,t} )</td>
<td>( \mu_{RT,t} )</td>
<td></td>
<td>( e_{RT,t} )</td>
</tr>
<tr>
<td>Usable parts</td>
<td></td>
<td>( \lambda^U_{p,t} )</td>
<td>( X^U_{p,t} )</td>
<td>( e_{US, p,t} )</td>
<td>( X^U_{US,p,t} )</td>
<td>( \mu_{US,p,t} )</td>
<td>( R^U_{US,p,t} )</td>
</tr>
<tr>
<td>Disposable parts</td>
<td></td>
<td>( \lambda^{DIS}<em>{p,t}, \lambda^{DIS}</em>{p,t-1} )</td>
<td>( \lambda^{DIS}_R )</td>
<td>( X^{DIS}_{p,t} )</td>
<td>( e_{DIS,p,t} )</td>
<td>( X^{DIS}_{p,t} )</td>
<td>( \mu^{DIS}_p,t )</td>
</tr>
<tr>
<td>Repairable parts</td>
<td></td>
<td>( \lambda^R_{p,t} )</td>
<td>( X^R_{p,t} )</td>
<td>( e_{p,t} )</td>
<td>( X^R_{p,t} )</td>
<td>( \mu^R_{p,t} )</td>
<td>( R^R_{p,t} )</td>
</tr>
<tr>
<td>Repaired parts</td>
<td></td>
<td>( \lambda^R_{p,t} )</td>
<td>( X^R_{p,t} )</td>
<td>( e_{p,t} )</td>
<td>( X^R_{p,t} )</td>
<td>( \mu^R_{p,t} )</td>
<td>( R^R_{p,t} )</td>
</tr>
<tr>
<td>Acceptable parts</td>
<td></td>
<td>( \lambda^{AC}_{p,t} )</td>
<td>( X^{AC}_{p,t} )</td>
<td>( e_{p,t} )</td>
<td>( X^{AC}_{p,t} )</td>
<td>( \mu^{AC}_{p,t} )</td>
<td>( R^{AC}_{p,t} )</td>
</tr>
<tr>
<td>Finished parts</td>
<td></td>
<td>( \lambda^{FIN}_{p,t} )</td>
<td>( X^{FIN}_{p,t} )</td>
<td>( e_{p,t} )</td>
<td>( X^{FIN}_{p,t} )</td>
<td>( \mu^{FIN}_{p,t} )</td>
<td>( R^{FIN}_{p,t} )</td>
</tr>
<tr>
<td>New parts</td>
<td></td>
<td>( \lambda^{NEW}_{p,t} )</td>
<td>( X^{NEW}_{p,t} )</td>
<td>( e_{p,t} )</td>
<td>( X^{NEW}_{p,t} )</td>
<td>( \mu^{NEW}_{p,t} )</td>
<td>( R^{NEW}_{p,t} )</td>
</tr>
<tr>
<td>Remanufactured products</td>
<td>( \lambda^{RM}_{p,t} )</td>
<td>( X^{RM}_{p,t} )</td>
<td>( e_{RM,t} )</td>
<td>( D_{p,t} )</td>
<td>( X^{RM}_{p,t} )</td>
<td>( \mu^{RM}_{p,t} )</td>
<td>( \lambda^{RM}_{p,t} )</td>
</tr>
</tbody>
</table>

**Model outputs**

- \( TC_{p,t} \) = total cost; \( IC_{p,t} \) = internal cost; \( RC_{p,t} \) = reconfigured cost; \( EC_{p,t} \) = external cost; \( US_{p,t} \) = unused space; Please note that \( p \) is dropped and becomes returned or remanufactured when referring to the products

**Unit cost inputs (part related)**

- \( c_{p,\text{internal}} \) = unit internal cost; \( c_{p,\text{recon}} \) = unit reconfiguration cost; \( c_{p,\text{external}} \) = unit external cost; \( c_{p,\text{usable}} \) = unit reconfiguration cost of repairable parts in usable part inventory; \( c_{p,\text{finished}} \) = unit reconfiguration cost of new parts in finished part inventory; \( c_{\text{unused}} \) = opportunity cost of unused space

**Unit cost inputs (product related)**

- \( c_{RT,\text{internal}} \) = unit internal cost of returned products; \( c_{RT,\text{external}} \) = unit external cost of returned products; \( c_{RM,\text{internal}} \) = unit internal cost of remanufactured products; \( c_{RM,\text{external}} \) = unit external cost of remanufactured products; \( c_{\text{unused}} \) = opportunity cost of unused space
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

The user-interfaced simulation model is fairly flexible and can be used for a product composed of any number of part types. The user can specify whether parts stored externally will be used in the current or next period, whether ordering new parts is an option to satisfy demand, the batch size of new parts, the type of distribution for the arrival of returned products to the remanufacturing facility and remanufactured product demand, and the unit costs of storing each part type internally, externally, or in reconfigured space.

![Figure 5.4 Input vs. output variables](image)

The reconfigurable remanufacturing system’s input variables are categorized into controllable and uncontrollable. The controllable variables are capacities of inventory storage spaces, whereas the uncontrollable variables are yield percentages, mean and standard deviation of returned products and remanufactured product demand. The yield percentages related to part types are usable, repairable, repaired and acceptable. The system’s responses are total, internal,
reconfigured, external storage space costs and unused space. This is all summarized in Figure 5.4.

5.4 Simulation Model

5.4.1 User Interface

The simulation model is implemented in Microsoft VB.NET, an object-oriented computer language based on Visual Basic on their .NET framework, with a front-end GUI user interface. The user enters the above inputs on two separate screens, one for the macro level information and the other for detailed cost data, as illustrated in Figures 5.5 and 5.6. On the first input screen, the user specifies the mean and standard deviation of the number of returned products, the demand for remanufactured products, the number of component types that is recaptured from each product, internal and external storage costs for returned and remanufactured items, and the simulation run conditions (number of time periods and replications). On the second screen, the user enters the number of recaptured parts per item, batch size of externally sourced replacement parts, and the yield rates and associated unit storage cost for each of the four processes parts flow through (disassembly, inspection, repair, reassembly). After the simulation runs, output is exported automatically to a Microsoft Excel file, as illustrated by the ten-period example in Figure 5.7, including the mean, standard deviation, minimum, and maximum of costs of total, internal, reconfiguration, external, unused space, and new part demand. A pseudo code of the simulation program is provided in Appendix B.
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

Figure 5.5 Simulation model inputs

Figure 5.6 Part related inputs

106
5.4.2 Program Logic

The Monte Carlo model operates according to the following assumptions and logic, where the circles in Figures 5.8 through 5.13 denote seven general capacitated storage locations: C.1 denoting returned products, C.2 usable parts, C.3 disposable parts, C.4 repairable parts, C.5 new parts, C.6 finished parts, and C.7 remanufactured products. Please note that these same capacity notations will be used throughout this and next chapter. As shown in Figure 5.8, a quantity of \( X_{RT,t} \) returned products arrives at the remanufacturing facility in each time period \( t \). The internal storage cost for returned products is \( X_{RT,t} \cdot c_{RT, \text{internal}} \), up to the capacity of the returned product storage area, with any excess items \( (e_{RT,t}) \) stored externally at a cost of \( e_{RT,t} \cdot c_{RT, \text{external}} \).

After disassembly of a returned product, component types eligible for recapture are sorted into either usable parts (stored in usable part inventory) or disposable parts (stored in disposable part inventory), as shown in Figure 5.9. If the number of usable parts is less than the usable part storage capacity (for that part type), the internal usable parts storage cost is \( X_{US,p,t} \cdot c_{p, \text{internal}} \); otherwise additional parts \( e_{US,p,t} \) are first stored in any available usable part storage for other component types, at a reconfiguration and storage cost of \( R_{US,p,t} \cdot c_{p, \text{recon}} \), with any remaining
usable parts (if not enough local “overflow” space is available) stored externally at a higher cost of \( \left( c_p^{US} - R_p^{US} \right) c_{p, \text{external}} \).

In addition to the above, material arrives at the disposable part inventory from two other activities: 1) inspection and testing and 2) repair; both are discussed below. The cost of storing these parts (in disposable part inventory) before they are disposed of (possibly shipped to an
external material processing or hazardous waste facility) is \( X_{p,t}^{DIS} \cdot c_{p, \text{internal}} \). If the disposable part storage capacity is exceeded, additional parts \( (e_{p,t}^{DIS}) \) are first stored in any available disposable part storage for other component types at a cost of \( R_{p,t}^{DIS} \cdot c_{p, \text{recon}} \), and then externally at a cost of \( (e_{p,t}^{DIS} - R_{p,t}^{DIS}) \cdot c_{p, \text{external}} \).

Figure 5.9 Disassembly and sorting process logic
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

Figure 5.10 Inspection, testing and repair logic
Usable parts are sent to cleaning, inspection, and testing operations, where they are deemed acceptable, repairable, or disposable. Accepted parts are sent to finished part inventory, disposable parts are sent to the process described above, and repairable parts are sent to the repairing area, as shown in Figure 5.10. The internal storage cost for repairable parts is $X_{p,t}^{REP} \cdot c_{p,\text{internal}}$. Parts beyond the repairable part storage capacity ($\varepsilon_{p,t}^{REP}$) are first stored in any available repairable part storage for other component types at a cost of $R_{p,t}^{REP} \cdot c_{p,\text{con}}$, and then in usable part storage space, if available, at a reconfiguring and storage cost of $R_{p,t}^{REP/US} \cdot c_{p,\text{usable}}$. Finally, any remaining parts are stored externally at a cost of $(\varepsilon_{p,t}^{REP} - R_{p,t}^{REP} - R_{p,t}^{REP/US}) \cdot c_{p,\text{external}}$.

Once repaired, these parts are stored in finished part inventory, along with accepted parts, at a cost of $X_{p,t}^{FIN} \cdot c_{p,\text{internal}}$; parts that cannot be repaired are disposed of following the earlier logic.

Figure 5.11 Finished parts
The new part demand is:
\[ X_{p,t}^{NEW} = D_{p,t} - X_{p,t}^{FIN} \]

If \( X_{p,t}^{NEW} < X_{p,t}^{ORD} \), NO

YES

Order new parts
\[ X_{p,t} = X_{p,t}^{NEW} - X_{p,t}^{ORD} \]

Meet the demand with new parts from previous period

Send new parts to assembly

The number of extra new parts:
\[ e_{p,t}^{NEW} = X_{p,t}^{ORD} - X_{p,t}^{NEW} \]

YES

NO

The unused space in new part inventory:
\[ \mu_{p,t}^{NEW} = e_{p,t}^{NEW} - X_{p,t}^{FIN} \]

Meet the demand with new parts

If \( e_{p,t}^{FIN} < 0 \), NO

YES

Store extra new parts

The number of extra new parts after demand is met due to batch size in ordering:
\[ e_{p,t}^{NEW} = X_{p,t}^{ORD} - 4e_{p,t}^{NEW} - X_{p,t}^{FIN} \]

Leaving the extras in new part inventory, and send the rest to assembly

If \( e_{p,t}^{FIN} > 0 \), YES

NO

Store extra new parts in finished part inventory

If \( e_{p,t}^{FIN} < 0 \), NO

YES

Store extra new parts externally

External storage

Figure 5.12 Demand fulfillment logic
As shown in Figure 5.11, if the number of finished parts is higher than the finished part storage capacity (for that part type), the additional parts ($F_{p,t}$) are first stored in any available finished part storage for other component types, at a reconfiguration and storage cost of $R_{p,t} \cdot c_{p, recon}$, with any remaining finished parts stored externally at a higher cost of $(R_{p,t} - F_{p,t}) \cdot c_{p, external}$.

If demand exceeds the supply of finished refurbished parts, the availability and number of any extra parts in new parts inventory from previous periods are checked as shown in Figure 5.12. If demand is still higher, new parts are ordered (in user-specified batch sizes) to meet demand and stored at a temporary storage cost of $X_{p,t} \cdot c_{p, internal}$. If the number of new parts ordered is higher than the new part storage capacity (for that part type), the additional parts ($N_{p,t}$) are first stored in any available new part storage for other component types, at a reconfiguration and storage cost of $R_{p,t} \cdot c_{p, recon}$, with any remaining new parts stored in finished part storage space, if available, at a reconfiguring and storage cost of $R_{p,t} \cdot c_{p, reconstruction}$. New parts are assumed to arrive within the same time period ordered, such that the remanufacturing process is not delayed by delivery lags.

Once finished refurbished and new (if any needed) parts are assembled into remanufactured products, they are stored in the remanufactured product inventory at a cost of $X_{RM,t} \cdot c_{RM, internal}$ before shipping to the customer, with extras stored externally at a cost of $R_{RM,t} \cdot c_{RM, external}$, as shown in Figure 5.13. Demand for remanufactured products every time period is always
satisfied and new part supply is assumed to be infinite (so no backordering occurs of either remanufactured items or replacement parts).

The total cost for the entire remanufacturing process is the sum of the internal, reconfigured, and external storage space costs plus the opportunity cost of unused space.

$$TC_{p,t} = c_{\text{unused}} \left( \mu_{RT,t} + \mu^{US}_{p,t} + \mu^{DIS}_{p,t} + \mu^{REP}_{p,t} + \mu^{FIN}_{p,t} + \mu^{NEW}_{p,t} + \mu_{RM,t} \right) + IC_{p,t} + RC_{p,t} + EC_{p,t} \quad [2]$$

where

$$IC_{p,t} = \left( X_{RT,t} \cdot c_{RT,\text{internal}} \right) + c_{\text{p, internal}} \left( X^{US}_{p,t} + X^{DIS}_{p,t} + X^{REP}_{p,t} + X^{FIN}_{p,t} + X^{NEW}_{p,t} \right) + \left( X_{RM,t} \cdot c_{RM,\text{internal}} \right) \quad [3]$$

$$RC_{p,t} = c^{\text{part}}_{\text{p, recon}} \left( R^{US}_{p,t} + R^{DIS}_{p,t} + R^{REP}_{p,t} + R^{FIN}_{p,t} + R^{NEW}_{p,t} \right) + \left( R^{\text{REP/US, \text{p, recon}}} \cdot c^{\text{p, recon}} \right) + \left( R^{\text{NEW/FIN, \text{p, recon}}} \cdot c^{\text{p, recon}} \right) \quad [4]$$
\[ EC_{p,t} = (c_{RT, i} \cdot c_{RT, external}) + \left( e_{US, t} - R_{US, t} \right) + \left( e_{DIS, t} - R_{DIS, t} \right) + \left( e_{REP, t} - R_{REP, t} - R_{REP/US, t} \right) + \ldots \]

In our examples, the opportunity cost of unused space is $6 (c_{\text{unused}} = $6). This is the cost associated with having the space but not using it in a period.

### 5.4.3 Examples

**Example 1: Cell phones (two recaptured parts)**

To illustrate the simulation model, an example will be presented. Gupta et al. (2004) describe a cellular phone remanufacturing facility in which 21 components are recaptured and refurbished, with each cellular phone requiring 4 processes: (1) disassembly and sorting, (2) inspection and testing, (3) repair and testing, and (4) assembly. To simplify this example, two parts are selected for the analysis: (1) batteries and (2) printed circuit boards (PCBs). The yield percentages for each part type at each process, the number of returned cellular phones, and the demand for remanufactured phones all are random. Returned phones arrive to the facility with mean 150 and a standard deviation of 10. Demand for remanufactured phones has a mean of 100 and a standard deviation of 5. When the phones are disassembled, 90% of batteries and 70% of printed circuit boards are usable, with the remainder disposed of. Of the usable parts, 80% and 65% of batteries and PCBs, respectively, pass the inspection and testing stages. 75% of unacceptable batteries and 95% of unacceptable PCBs are repairable, with the remainder disposed of. The rest is disposable. After parts are sent to the repair process, 80% of batteries and 90% of PCBs are repaired, and the rest is sent for disposal. Capacities are set at the values shown in Table 5.2 for each analysis. For example, in the first analysis, returned product capacity is 150; (2) the usable,
disposable, repairable, new and finished part inventory capacities are 100, 30, 10, 5 and 80, respectively; (3) and the remanufactured product capacity is 100. Internal and external storage costs per returned and remanufactured phones are $3 and $6, respectively. Internal, reconfiguration, and external storage costs per battery are $3, $3, and $6, respectively, whereas the same costs for PCB are $1, $1, and $2, respectively. These costs have been selected for illustrative purposes only, and do not reflect the costs associated with the work in the paper by Gupta et al (2004).

For each analysis run, the numbers in the first row in the capacity columns denote the battery storage capacity and the numbers in the second rows denote the printed circuit board capacity. The numbers in parentheses are the corresponding standard deviations. As can be seen by comparing the first two runs in Table 5.2, an increase of 50% in the finished part C.6 inventory capacity (from 80 to 120 storage spaces) causes the total cost decrease by 9%. Increasing usable part C.2 capacity from 100 to 130 in the third run decreases the total cost by 7.3%, from $4,820 to $4,567. In the fourth run, an increase from 10 to 40 in the repairable part C.4 inventory capacity decreases total cost to $4,402 from $4,567.

Table 5.2 Example 1 inputs and outputs (Cell phones, two recaptured parts)

<table>
<thead>
<tr>
<th>Run</th>
<th>Capacities (fixed)</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C.1</td>
<td>C.2</td>
</tr>
<tr>
<td>1</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>130</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>130</td>
</tr>
</tbody>
</table>
Example 2: Cell phones (four recaptured parts)

As a larger example, suppose there are four recap tured parts: (1) battery, (2) printed circuit board, (3) front and back cover, and (4) antenna. Returned products and demand for remanufactured products occur with the same rates as in the first example. When phones are disassembled, 90% of batteries, 70% of printed circuit boards, 81% of covers, and 73% of antennas are usable. These 80%, 65%, 72%, and 92% of batteries, PCBs, covers and antennas, respectively, pass inspection and testing. 75% of unacceptable batteries are sent for repair, 95% of PCBs, 89% of covers, and 71% of antennas, of which 80% of batteries, 90% of PCBs, 95% of covers, and 78% of antennas actually are successfully repaired, with the rest sent to disposal.

Table 5.3 summarizes inventory capacities for several scenarios (runs). In the first run, for example, returned product capacity is 150; (2) usable, disposable, repairable, new, and finished part inventory capacities are 100, 30, 10, 5, and 80, respectively; (3) and remanufactured product capacity is 100. Internal and external storage costs for returned and remanufactured phones are $3 and $6 per item, respectively, and internal, reconfiguration, and external storage costs for batteries are $3, $3, and $6; $1, $1, and $2 for the PCBs; $2, $2, and $4 for covers; $3, $3, and $9 for antennas.

For each analysis run, the first row denotes battery inventory capacities, the second row denotes printed circuit board capacities, the third row denotes cover capacities, and the fourth row denotes antenna capacities. Standard deviations are shown in parentheses. As can be seen by comparing the first two runs in Table 5.3, an increase of 50% in finished part C.6 inventory capacity results in a decrease of 2.9% in total cost from $9,264 to $8,992. Increasing usable part C.2 inventory capacity from 100 to 130 decreases total cost by 7.6%. In the fourth run, an
increase from 10 to 40 in the repairable part C.4 capacity decreases total cost by 3.2% to $8,041.

As these two examples indicate, total cost can be significantly affected by changes in inventory capacities in some cases. In the next section, the multi-dimensional golden section search is used to find the optimal capacity values.

Table 5.3 Example 2 output

<table>
<thead>
<tr>
<th>Run</th>
<th>Capacities (fixed)</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C.1 C.2 C.3 C.4 C.5 C.6 C.7</td>
<td>Total cost ($)</td>
</tr>
<tr>
<td>1</td>
<td>150 100 30 10 5 80</td>
<td>$9,264 ($789)</td>
</tr>
<tr>
<td>2</td>
<td>150 100 30 10 5 120</td>
<td>$8,992 ($574)</td>
</tr>
<tr>
<td>3</td>
<td>150 130 30 10 5 120</td>
<td>$8,306 ($353)</td>
</tr>
<tr>
<td>4</td>
<td>150 130 30 40 5 120</td>
<td>$8,041 ($209)</td>
</tr>
</tbody>
</table>
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

Figure 5.14 Flowchart of multi-dimensional golden section search algorithm
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

5.5 Optimization Model

5.5.1 Golden Section Search Algorithm

The golden section search is a technique used to find the minimum or maximum (extremum) of a unimodal function over the interval \([a, b]\). It is assumed that the function has only one minimum in \([a, b]\). For \(x_1\) and \(x_2\) in the interval \([a, b]\) computed as \(x_1 = c \cdot a + (1 - c) \cdot b\) and \(x_2 = (1 - c) \cdot a + c \cdot b\), the functions \(f(x_1)\) and \(f(x_2)\) are evaluated. \(c\) is the constant reduction factor for the size of the interval in each step, and equal to \((-1 + \sqrt{5}) / 2\). If \(f(x_1)\) is less than \(f(x_2)\), then \(x_2\) becomes the new upper bound, \(x_1\) becomes the new \(x_2\), and a new \(x_1\) is computed as \(c \cdot a + (1 - c) \cdot b\), so the new search interval is \([a, x_2]\). Otherwise, \(x_1\) becomes the new lower bound, \(x_2\) becomes the new \(x_1\), and a new \(x_2\) is calculated as \((1 - c) \cdot a + c \cdot b\), so the new search interval is \([x_1, b]\). In each step, a new narrower search interval is constructed that is guaranteed to contain the function minimum. The search stops when the difference between the lower (\(a\)) and upper bounds (\(b\)) is less than some very small \(\alpha\). The derivation of the golden section search method can be found in Gerald et al. (2004).

To identify optimal storage capacities at each location in the above process, a simple multi-dimensional golden section search algorithm is implemented. This routine efficiently searches for the optimal capacities of each storage and part type, within upper and lower bounds that minimize total cost over a user-specified number of time periods. These optimal capacities are achieved by reducing the interval under consideration in each step continuing until the difference between the lower (\(\text{CapLow}\)) and upper bounds (\(\text{CapHigh}\)) is less than some very small value \(\alpha\), searching across each part type as illustrated in Figure 5.14. For each part type \(p\), the lower and upper capacity bounds for each inventory type \(C\) (the generalized remanufacturing system has
seven inventory types) are specified by the user. The total cost is estimated by running the simulation model at two points defined as \( x_1 \) and \( x_2 \), computed as \( x_1 = c \times \text{CapLow} + (1 - c) \times \text{CapHigh} \), and \( x_2 = (1 - c) \times \text{CapLow} + c \times \text{CapHigh} \). If the total cost at \( x_1 \) is lower than at \( x_2 \), a new upper search bound is set to \( x_2 \), \( x_1 \) becomes the new \( x_2 \), and a new \( x_1 \) is computed as \( c \times \text{CapLow} + (1 - c) \times \text{CapHigh} \). Otherwise, a new lower search bound is set to \( x_1 \), \( x_2 \) becomes the new \( x_1 \), and a new \( x_2 \) is computed as \( (1 - c) \times \text{CapLow} + c \times \text{CapHigh} \). This search procedure continues until the difference between the lower and upper capacity bounds is less than some user specified small value \( \alpha \), repeating for all optimal inventory capacities for all part types. A pseudo code of the golden section search algorithm is provided in Figure 5.15. A simplified version of the pseudo code with two inventory capacities, as an example, is also shown in Figure 5.16.

For \( p = 0 \) To number of part types

  Initialize capacity bounds for each inventory type
  \( \text{CapLow} = --- \)
  \( \text{CapHigh} = --- \)
  \( x_1 = c \times \text{CapLow} + (1 - c) \times \text{CapHigh} \)
  \( x_2 = (1 - c) \times \text{CapLow} + c \times \text{CapHigh} \)

  For \( C = 0 \) To number of inventory capacity types

    Do until \( \text{CapHigh} - \text{CapLow} < \alpha \)
    Calculate the total cost at \( x_1 \) and \( x_2 \)
    If \( \text{TotalCost}(x_1) < \text{TotalCost}(x_2) \) Then
      \( \text{CapHigh} = x_2 \)
      \( x_2 = x_1 \)
      \( x_1 = c \times \text{CapLow} + (1 - c) \times \text{CapHigh} \)
    Else
      \( \text{CapLow} = x_1 \)
      \( x_1 = x_2 \)
      \( x_2 = (1 - c) \times \text{CapLow} + c \times \text{CapHigh} \)
    End If

    Loop

Next capacity \( C \)

Next part type \( p \)

Report optimal capacity for each inventory

\( \text{CapOpt} \)

Figure 5.15 Pseudo code of multi-dimensional golden section search algorithm
For \( p = 0 \) To number of part types

Initialize capacity bounds for each inventory type
CapLow = ---
CapHigh = ---
\( x_1 = c \cdot \text{CapLow} + (1 - c) \cdot \text{CapHigh} \)
\( x_2 = (1 - c) \cdot \text{CapLow} + c \cdot \text{CapHigh} \)

Do until Cap2High – Cap2Low < \( \alpha \)

Do until Cap1High – Cap1Low < \( \alpha \)

Calculate the total cost at \( x_1 \) and \( x_2 \)
If TotalCost(\( x_1 \)) < TotalCost(\( x_2 \)) Then
Cap1High = \( x_2 \)
\( x_2 = x_1 \)
\( x_1 = c \cdot \text{Cap1Low} + (1 - c) \cdot \text{Cap1High} \)
Else
Cap1Low = \( x_1 \)
\( x_1 = x_2 \)
\( x_2 = (1 - c) \cdot \text{Cap1Low} + c \cdot \text{Cap1High} \)
End If
Loop

If TotalCost(\( x_1 \)) < TotalCost(\( x_2 \)) Then
Cap2High = \( x_2 \)
\( x_2 = x_1 \)
\( x_1 = c \cdot \text{Cap2Low} + (1 - c) \cdot \text{Cap2High} \)
Else
Cap2Low = \( x_1 \)
\( x_1 = x_2 \)
\( x_2 = (1 - c) \cdot \text{Cap2Low} + c \cdot \text{Cap2High} \)
End If
Loop

Next part type \( p \)

Report optimal capacity for each inventory
Cap1Opt
Cap2Opt

Figure 5.16 Pseudo code of multi-dimensional golden section search algorithm, assuming 2 inventory capacities

Figure 5.17 illustrates a sample output screen with optimal capacity levels for a product with four parts. Note that although capacities for returned and remanufactured products (columns B and H) are given for each part, those two capacities are not part specific.
For a large number of part and inventory types, search computation times increase significantly due to the nested searching. As shown in Figure 5.18, the runtime jumps to 2,992 from 123 minutes with an increase in the number of inventory types from two to four. To achieve faster results, un-nested searches are conducted on each capacity one at a time, with the other capacities set to their current values (or their mid values on the first iteration), iterating through this process several times to improve the solution on each iteration. The iterative process stops
Chapter 5 Development of Heuristics Based on Multi-Dimensional Golden Section Simulation-Optimization

when the absolute difference in optimal capacity values (for each inventory type) found in the current and from previous period is less than or equal to a specified value $\varepsilon$. Figure 5.19 illustrates the implementation of this heuristic for two inventory types. Performance of this heuristic is compared to optimal results below, and often found to be very close with only a 0.07% to 2.09% increase in total cost while reducing the runtimes by roughly 97% with four inventory types.

```
For $i = 0$ To number of iterations
    For $p = 0$ To number of part types
        Initialize capacity bounds for each inventory type
        CapLow = ---
        CapHigh = ---
        $x_1 = c \cdot \text{CapLow} + (1 - c) \cdot \text{CapHigh}$
        $x_2 = (1 - c) \cdot \text{CapLow} + c \cdot \text{CapHigh}$
        Do until $\text{Cap1High} - \text{Cap1Low} < \alpha$
            If $\text{TotalCost}(x_1) < \text{TotalCost}(x_2)$ Then
                $\text{Cap1High} = x_2$
                $x_1 = x_2$
                $x_1 = c \cdot \text{Cap1Low} + (1 - c) \cdot \text{Cap1High}$
            Else
                $\text{Cap1Low} = x_1$
                $x_1 = x_2$
                $x_2 = (1 - c) \cdot \text{Cap1Low} + c \cdot \text{Cap1High}$
            End If
        Loop
        Set CapInv1 at the local optimal value found in the previous step
        Do until $\text{Cap2High} - \text{Cap2Low} < \alpha$
            If $\text{TotalCost}(x_1) < \text{TotalCost}(x_2)$ Then
                $\text{Cap2High} = x_2$
                $x_1 = x_2$
                $x_1 = c \cdot \text{Cap2Low} + (1 - c) \cdot \text{Cap2High}$
            Else
                $\text{Cap2Low} = x_1$
                $x_1 = x_2$
                $x_2 = (1 - c) \cdot \text{Cap2Low} + c \cdot \text{Cap2High}$
            End If
        Loop
    Next part type $p$
    Stopping criteria
    If $|\text{Cap1Opt}(i-1) - \text{Cap1Opt}(i)| \leq \varepsilon \text{ AND } |\text{Cap2Opt}(i-1) - \text{Cap2Opt}(i)| \leq \varepsilon$ Then
        Exit for
    End If
Next iteration $i$
```

Report optimal capacity for each inventory
Cap1Opt
Cap2Opt

Figure 5.19 Heuristic pseudo code with 2 inventory capacities, searching on each inventory separately rather than nested
Figure 5.20 illustrates the convergence of the heuristic approach under different cases. X-axis being the iterations, the y-axis shows the maximum difference between iterations across capacities. In the first and second cases, the convergence occurs at the fourth iteration. Under the third case, it is faster, and occurs at the third iteration.

5.5.2 Sensitivity Analysis

The purpose of this sensitivity analysis is to explore how optimal capacities and total storage costs are affected by percentages, the number of returned products, and demand, with capacity requirements forecasted for 2 consecutive two-week periods, and the upper and lower bounds for inventory capacities summarized in Table 5.4. Table 5.5 summarizes the various yields, return rate, and product demand. For example, the low and high level for the returned product inventory is 120 and 180. For the usable part inventory, it is 80 and 170 for both the battery and PCB.
Table 5.4 Lower and upper capacity bounds

<table>
<thead>
<tr>
<th>Type of inventory</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned product inventory space</td>
<td>120</td>
<td>180</td>
</tr>
<tr>
<td>Usable part inventory space</td>
<td>80</td>
<td>170</td>
</tr>
<tr>
<td>Disposable part inventory space</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Repairable part inventory space</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>New part inventory space</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>Finished part inventory space</td>
<td>80</td>
<td>200</td>
</tr>
<tr>
<td>Remanufactured product inventory space</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

The returned cellular phones arrive at the facility with a mean of 150 and standard deviation of 10. The usable yield percentages of batteries and PCBs retrieved when these returned phones are disassembled are 90% and 70%, respectively. The remaining units are disposable. Once these usable parts go through certain processes, they are tested and inspected before being sent to the assembly. At this inspection stage, the parts are classified into three categories with the following yield percentages: 80% (battery)-65% (PCB) acceptable; 75% (battery)-95% (PCB) repairable and the rest being disposable. The acceptable parts are directly sent to the assembly, while repairable parts are first sent to the repair process. Out of the parts being sent to the repair, 80% (battery)-90% (PCB) are repaired and the rest is disposable. The demand for the remanufactured cellular phones has a mean of 100 with standard deviation of 5. This is summarized in Table 5.5.

Table 5.5 Uncontrollable variables

<table>
<thead>
<tr>
<th>Uncontrollable variables</th>
<th>Battery</th>
<th>PCB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean for number of returned products</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Standard deviation for number of returned products</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Usable part yield percentage</td>
<td>90%</td>
<td>70%</td>
</tr>
<tr>
<td>Acceptable part yield percentage</td>
<td>80%</td>
<td>65%</td>
</tr>
<tr>
<td>Repairable part yield percentage</td>
<td>75%</td>
<td>95%</td>
</tr>
<tr>
<td>Repaired part yield percentage</td>
<td>80%</td>
<td>90%</td>
</tr>
<tr>
<td>Mean for demand</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Standard deviation for demand</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.6 Sensitivity analysis on cost inputs

<table>
<thead>
<tr>
<th>Type of cost</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product related</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal storage</td>
<td>$3</td>
<td>$3</td>
<td>$3</td>
</tr>
<tr>
<td>External storage</td>
<td>$6</td>
<td>$6</td>
<td>$9</td>
</tr>
<tr>
<td>Part related</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal storage</td>
<td>$3$; $1^a$; $1^b$</td>
<td>$3$; $1$</td>
<td>$3$; $2$</td>
</tr>
<tr>
<td>Within part-type reconfiguration</td>
<td>$3$; $1$</td>
<td>$6$; $2$</td>
<td>$6$; $3$</td>
</tr>
<tr>
<td>Within inventory-type reconfiguration</td>
<td>$3$; $1$</td>
<td>$6$; $2$</td>
<td>$6$; $3$</td>
</tr>
<tr>
<td>External storage</td>
<td>$6$; $2$</td>
<td>$6$; $2$</td>
<td>$9$; $6$</td>
</tr>
<tr>
<td>Opportunity cost of unused internal space</td>
<td>$6$</td>
<td>$6$</td>
<td>$6$</td>
</tr>
</tbody>
</table>

Table 5.6 summarizes a trial sensitivity analysis on the relative storage costs. For the base-case scenario (scenario 1), the unit internal and reconfiguration storage costs are $3 and $1 for batteries and PCBs, respectively, whereas their unit external storage costs are $6 and $2. For returned and remanufactured cellular phones, the unit internal and external storage costs are $3 and $6, respectively. In the second scenario, the reconfiguration costs are doubled to $6 and $2 for batteries and PCBs, respectively, with all other costs unchanged. In the third scenario, the external storage cost is increased to $9, and internal and reconfiguration costs for PCBs are increased to $2 and $3, respectively, but unchanged for batteries. External storage costs for parts are increased to $9 (battery) and $6 (PCB). As indicated before, these costs do not reflect the real numbers, and are just chosen for analysis.

Table 5.7 summarizes the total cost and runtimes under each scenario. Due to long computation times, models with five, six and seven inventory capacities were not computed. The numbers in parentheses denote the corresponding standard deviation.

By searching across each part type separately, for example, run times were reduced by 74% to 84%. As shown by the first row in Table 5.7, the runtime varies between 115.7 and 127.1
minutes over three scenarios when the number of inventory capacities is 2. However, it increases significantly as the number of inventory capacities increases. For example, when the model has four inventory capacities, the runtime is computed to vary between 2,899.4 and 2,991.6 minutes, shown by the last row in Table 5.7. Hence, the runtime is expected to be 402 days when the number of inventory capacities is 7. Table 5.8-5.10 displays a detailed cost output, including the unused space and the capacity requirements for the MC optimization.

Table 5.7 MC Optimization – Summary of results for the three scenarios

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Scenario 1</th>
<th></th>
<th>Scenario 2</th>
<th></th>
<th>Scenario 3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total cost ($)</td>
<td>Runtime (min)</td>
<td>Optimal capacity values</td>
<td>Total cost ($)</td>
<td>Runtime (min)</td>
<td>Optimal capacity values</td>
</tr>
<tr>
<td>2</td>
<td>$5172.1</td>
<td>122.8</td>
<td>C1: 147, C2: 131-121</td>
<td>$5499.0</td>
<td>127.1</td>
<td>C1: 145, C2: 136-120</td>
</tr>
<tr>
<td></td>
<td>($134.8)</td>
<td></td>
<td></td>
<td>($141.5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>($152.3)</td>
<td></td>
<td></td>
<td>($156.3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>($312.0)</td>
<td></td>
<td></td>
<td>($389.2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.8 MC Optimization – Scenario 1

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>147 131 121</td>
<td>$5172.1</td>
<td>($134.8)</td>
<td>$3591.2</td>
<td>($131.5)</td>
<td>$98.7</td>
<td>($28.3)</td>
</tr>
<tr>
<td>3</td>
<td>145 136 117 8 22</td>
<td>$5099.3</td>
<td>($152.3)</td>
<td>$3465.1</td>
<td>($112.7)</td>
<td>$145.5</td>
<td>($33.7)</td>
</tr>
<tr>
<td>4</td>
<td>151 145 101 14 8 18</td>
<td>$5001.2</td>
<td>($312.0)</td>
<td>$3261.6</td>
<td>($62.2)</td>
<td>$144.2</td>
<td>($22.8)</td>
</tr>
</tbody>
</table>

Table 5.9 MC Optimization – Scenario 2

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>145 136 120</td>
<td>$5499.0</td>
<td>($141.5)</td>
<td>$3770.8</td>
<td>($100.5)</td>
<td>$301.2</td>
<td>($56.1)</td>
</tr>
<tr>
<td>3</td>
<td>138 131 120 4 19</td>
<td>$5414.8</td>
<td>($156.3)</td>
<td>$3542.1</td>
<td>($92.0)</td>
<td>$378.3</td>
<td>($68.4)</td>
</tr>
<tr>
<td>4</td>
<td>142 138 111 13 8 24</td>
<td>$5288.1</td>
<td>($389.2)</td>
<td>$3438.4</td>
<td>($133.7)</td>
<td>$315.0</td>
<td>($35.7)</td>
</tr>
</tbody>
</table>
Table 5.10 MC Optimization – Scenario 3

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>C.1 156</td>
<td>$6561.0 ($123.2)</td>
<td>$4188.3 ($165.2)</td>
<td>$277.0 ($54.4)</td>
<td>$399.4 ($69.5)</td>
<td>282.7</td>
<td>115.7</td>
</tr>
<tr>
<td></td>
<td>C.2 148</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.3 131</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C.1 151</td>
<td>$6450.2 ($143.0)</td>
<td>$4020.7 ($156.6)</td>
<td>$356.0 ($41.3)</td>
<td>$584.7 ($73.6)</td>
<td>248.1</td>
<td>453.5</td>
</tr>
<tr>
<td></td>
<td>C.2 127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.3 119</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>C.1 158</td>
<td>$6327.7 ($204.5)</td>
<td>$3960.9 ($191.3)</td>
<td>$391.3 ($48.0)</td>
<td>$563.1 ($58.2)</td>
<td>235.4</td>
<td>2899.4</td>
</tr>
<tr>
<td></td>
<td>C.2 149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.3 120</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.4 15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Under the heuristic approach, the computation time is successfully reduced by roughly 97% from 2,899.4 – 2,991.6 to 82.4 – 85.6 minutes for a four-inventory-type case. The total cost obtained with the heuristic approach is only 0.067% (minimum) to 2.14% (maximum) higher than the optimal total cost. Comparison of MC optimization and heuristics is provided in Table 5.11-5.13.

Table 5.11 Comparison of MC optimization and heuristics – Scenario 1

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Model type</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
<th>Optimal capacity values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MC opt</td>
<td>$5172.1 ($134.8)</td>
<td>3591.2 (131.5)</td>
<td>$98.7 ($28.3)</td>
<td>$441.3 ($79.1)</td>
<td>173.5</td>
<td>122.8</td>
<td>C1: 147</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5207.5 ($129.1)</td>
<td>3611.1 ($112.2)</td>
<td>$100.4 ($15.3)</td>
<td>$385.5 ($51.0)</td>
<td>185.1</td>
<td>51.7</td>
<td>C1: 151</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C2: 131-121</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C2: 137-125</td>
</tr>
<tr>
<td>3</td>
<td>MC opt</td>
<td>$5099.3 ($152.3)</td>
<td>3465.1 ($112.7)</td>
<td>$145.5 ($33.7)</td>
<td>$421.5 ($88.0)</td>
<td>177.9</td>
<td>477.9</td>
<td>C1: 145</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5133.8 ($141.8)</td>
<td>3501.3 ($79.3)</td>
<td>$163.5 ($26.7)</td>
<td>$434.7 ($77.2)</td>
<td>172.4</td>
<td>71.2</td>
<td>C1: 154</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C2: 138-122</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C4: 10-16</td>
</tr>
<tr>
<td>4</td>
<td>MC opt</td>
<td>$5001.2 ($312.0)</td>
<td>$3261.6 ($62.2)</td>
<td>$144.2 ($22.8)</td>
<td>$962.3 ($112.7)</td>
<td>105.5</td>
<td>2991.6</td>
<td>C1: 151</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5108.0 ($267.8)</td>
<td>$3365.4 ($155.4)</td>
<td>$190.1 ($36.0)</td>
<td>$585.5 ($67.6)</td>
<td>161.2</td>
<td>83.6</td>
<td>C1: 156</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C2: 145-108</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C3: 17-40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C4: 8-22</td>
</tr>
</tbody>
</table>

129
### Table 5.12 Comparison of MC optimization and heuristics – Scenario 2

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Model type</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Recon cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
<th>Optimal capacity values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MC opt</td>
<td>$5499.0 ($141.5)</td>
<td>$3770.8 ($100.3)</td>
<td>$301.2 ($56.1)</td>
<td>$445.6 ($81.5)</td>
<td>163.6 (32.2)</td>
<td>127.1</td>
<td>C1: 145 C2: 136-120</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5502.7 ($164.3)</td>
<td>$3458.6 ($139.9)</td>
<td>$264.4 ($49.2)</td>
<td>$472.2 ($71.3)</td>
<td>217.9 (39.9)</td>
<td>52.0</td>
<td>C1: 149 C2: 138-119</td>
</tr>
<tr>
<td>3</td>
<td>MC opt</td>
<td>$5414.8 ($156.3)</td>
<td>$3542.1 ($92.0)</td>
<td>$378.3 ($68.4)</td>
<td>$621.3 ($92.1)</td>
<td>145.5 (28.3)</td>
<td>513.2</td>
<td>C1: 138 C2: 131-120 C4: 4-19</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5450.4 ($146.6)</td>
<td>$3466.0 ($125.1)</td>
<td>$156.6 ($23.5)</td>
<td>$500.9 ($83.7)</td>
<td>221.2 (40.3)</td>
<td>69.6</td>
<td>C1: 141 C2: 132-122 C4: 4-18</td>
</tr>
<tr>
<td>4</td>
<td>MC opt</td>
<td>$5288.1 ($389.2)</td>
<td>$3438.4 ($133.7)</td>
<td>$315.0 ($35.7)</td>
<td>$915.0 ($84.4)</td>
<td>103.3 (29.1)</td>
<td>2970.5</td>
<td>C1: 142 C2: 138-111 C3: 13-39 C4: 8-24</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$5292.4 ($135.5)</td>
<td>$3577.3 ($116.8)</td>
<td>$305.4 ($45.0)</td>
<td>$804.0 ($77.0)</td>
<td>101.0 (28.6)</td>
<td>82.4</td>
<td>C1: 145 C2: 136-115 C3: 11-35 C4: 10-24</td>
</tr>
</tbody>
</table>

### Table 5.13 Comparison of MC optimization and heuristics – Scenario 3

<table>
<thead>
<tr>
<th>Number of capacities optimized</th>
<th>Model type</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Recon cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
<th>Optimal capacity values</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>MC opt</td>
<td>$6561.0 ($123.2)</td>
<td>$4188.3 ($165.2)</td>
<td>$277.0 ($54.4)</td>
<td>$399.4 ($69.5)</td>
<td>282.7 (47.6)</td>
<td>115.7</td>
<td>C1: 156 C2: 148-131</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$6604.6 ($161.4)</td>
<td>$4370.4 ($171.3)</td>
<td>$399.4 ($52.7)</td>
<td>$340.5 ($49.7)</td>
<td>249.1 (51.5)</td>
<td>49.7</td>
<td>C1: 157 C2: 146-131</td>
</tr>
<tr>
<td>3</td>
<td>MC opt</td>
<td>$6450.2 ($143.0)</td>
<td>$4020.7 ($156.6)</td>
<td>$356.0 ($41.3)</td>
<td>$584.7 ($73.6)</td>
<td>248.1 (41.1)</td>
<td>453.5</td>
<td>C1: 151 C2: 127-119 C4: 29-18</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$6489.5 ($153.8)</td>
<td>$4340.3 ($117.2)</td>
<td>$301.5 ($44.0)</td>
<td>$463.4 ($92.4)</td>
<td>230.7 (47.3)</td>
<td>67.9</td>
<td>C1: 149 C2: 134-115 C4: 18-30</td>
</tr>
<tr>
<td>4</td>
<td>MC opt</td>
<td>$6327.7 ($204.5)</td>
<td>$3960.9 ($191.3)</td>
<td>$391.3 ($48.0)</td>
<td>$563.1 ($58.2)</td>
<td>235.4 (53.2)</td>
<td>2899.4</td>
<td>C1: 158 C2: 149-120 C3: 15-49 C4: 7-23</td>
</tr>
<tr>
<td></td>
<td>Heuristics</td>
<td>$6411.9 ($252.4)</td>
<td>$4104.5 ($157.9)</td>
<td>$315.5 ($38.4)</td>
<td>$416.0 ($67.4)</td>
<td>262.7 (42.0)</td>
<td>85.6</td>
<td>C1: 155 C2: 146-125 C3: 18-43 C4: 9-23</td>
</tr>
</tbody>
</table>
5.6 Conclusions

Total inventory storage cost in remanufacturing can be significantly affected by inventory capacities. In this chapter, a multi-dimensional golden section simulation-optimization algorithm is used to identify the optimal storage capacities and reconfiguration decisions over a user-specified number of time periods that minimize total long-term expected costs. As the results indicate, this algorithm is efficient while solving for the optimal storage capacity values with fewer number of inventory capacities. The runtime is computed to vary between 115.7 and 127.1 minutes with two inventory capacities. Increasing the model size to four inventory capacities increases the runtime significantly, varying around 49.2 hours. With the heuristic approach based on this algorithm, the computation duration is successfully reduced by 97% from 49.2 hours to 83.9 minutes for higher number of inventory capacities. In several cases, results with this approach tend to be only 0.665% higher, which is sufficient in most practical applications.
Chapter 6 – Analysis

In this chapter, two types of analyses are performed using the cellular phone example of Chapter 5. The intention of this work is to further understand reconfiguration and its impact on cost. The first analysis examines how much the reconfiguration of space saves a company under different yield rate and cost scenarios. The results show that the reconfiguration becomes very important and can save the company substantial sums when the difference in yield rates among parts is higher than 20%. In this analysis, the savings were determined to be 33.4%. Further analysis is performed using experimental design to identify which of the inventory capacities in the system causes the highest impact on total cost. It is shown that the finished part inventory capacity impacts the total cost in all three yield rate scenarios. Its effect is highest for difference in yield rates higher than 20% due to reconfiguring space for the extra new parts in the finished part inventory. The regression analysis is further performed to show how much of a decrease or increase in total cost is observed by a change in inventory capacities. The regression equation for yield rate difference higher than 20% shows that due to the increased number of parts disposed of (low usable part yield), increasing the disposable part inventory by a unit decreases the total cost by $15.9.

6.1 An Application

Both the reconfiguration and experimental design analyses are based on three cases of the yield rate difference among parts, the number of parts being 2 (battery and PCB). The yield rate
difference among parts can be less than or equal to 5%, moderate between 5% and 20%, or higher than 20%. In all three cases, the returned cellular phones arrive at the facility with a mean of 150 and standard deviation of 10. The usable yield percentages of batteries and PCBs retrieved when these returned phones are disassembled are 95% and 90% for the first case, 90% and 70% for the second case, and 48% and 97% for the third case, respectively. The rest are disposable. Once these usable parts go through certain processes, they are tested and inspected before being sent to the assembly. At this inspection stage, the parts are classified into three categories with the following yield percentages: 70% (battery)-65% (PCB) acceptable in the first case, 80% and 65% in the second case, 72% and 24% in the third case; 60% (battery)-65% (PCB) repairable in the first case, 75% and 95% in the second case, 32% and 68% in the third case and the rest being disposable. The acceptable parts are directly sent to the assembly area, while repairable parts are first sent to the repair process. Out of the parts being sent to the repair, 80% (battery)-85% (PCB) in the first case, 80% and 90% in the second case, 92% and 31% in the third case are repaired and the rest are disposable. The demand for the remanufactured cellular phones is assumed to have a mean of 100 with a standard deviation of 5. This is summarized in Table 6.1.

The capacity requirements are forecasted for two consecutive two-week periods. The number of replications is 10. The optimal capacity values of the inventories are searched within an initial low and high level. These initial levels of inventory capacities are displayed in Table 6.2. For example, the low and high level for the returned product inventory is 120 and 180. For the usable part inventory, it is 80 and 170 for both the battery and PCB.
Table 6.1 Values of uncontrollable variables

<table>
<thead>
<tr>
<th>Uncontrollable variables</th>
<th>Low difference in yield rates (≤ 5%)</th>
<th>Moderate difference in yield rates (between 5% and 20%)</th>
<th>High difference in yield rates (&gt; 20%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean for number of returned products</td>
<td>Battery: 150</td>
<td>Battery: 150</td>
<td>Battery: 150</td>
</tr>
<tr>
<td>StDev for number of returned products</td>
<td>Battery: 10</td>
<td>Battery: 10</td>
<td>Battery: 10</td>
</tr>
<tr>
<td>Usable part yield percentage</td>
<td>Battery: 95%</td>
<td>Battery: 90%</td>
<td>Battery: 70%</td>
</tr>
<tr>
<td>Acceptable part yield percentage</td>
<td>Battery: 70%</td>
<td>Battery: 65%</td>
<td>Battery: 65%</td>
</tr>
<tr>
<td>Repairable part yield percentage</td>
<td>Battery: 60%</td>
<td>Battery: 90%</td>
<td>Battery: 95%</td>
</tr>
<tr>
<td>Repaired part yield percentage</td>
<td>Battery: 80%</td>
<td>Battery: 85%</td>
<td>Battery: 90%</td>
</tr>
<tr>
<td>Mean for demand</td>
<td>Battery: 100</td>
<td>Battery: 100</td>
<td>Battery: 100</td>
</tr>
<tr>
<td>Standard deviation for demand</td>
<td>Battery: 5</td>
<td>Battery: 5</td>
<td>Battery: 5</td>
</tr>
</tbody>
</table>

Table 6.2 Initial lower and upper levels for inventory capacities

<table>
<thead>
<tr>
<th>Inventory capacities</th>
<th>Low level</th>
<th>High level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned product inventory space</td>
<td>120</td>
<td>180</td>
</tr>
<tr>
<td>Usable part inventory space</td>
<td>80</td>
<td>170</td>
</tr>
<tr>
<td>Disposable part inventory space</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Repairable part inventory space</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>New part inventory space</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>Finished part inventory space</td>
<td>80</td>
<td>200</td>
</tr>
<tr>
<td>Remanufactured product inventory space</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

The cost of storing the products and parts internally, externally and by reconfiguring is shown in Table 6.3. In the analysis, five different cost scenarios are used. The way the scenarios are set up is that the ratio of unit external cost to unit reconfiguration cost is incremented by 0.5. In the first scenario, they are equal, meaning that reconfiguring space to store a single unit is equal to storing the same unit externally. In the second scenario, this ratio is 1.5. The internal storage cost is assumed to be $3 per unit per period for both the returned products and refurbished parts. The cost of reconfiguring space is twice as much as storing a part internally. The within part and inventory-type reconfiguration costs per period are kept the same at $6 and $2 per battery and PCB, respectively in all scenarios. There is also a cost associated with unused space, being equal
to $6 per space per period. Same as in chapter 5, these costs do not reflect the real numbers, and are just selected for analysis.

Table 6.3 Cost structures of the five scenarios

<table>
<thead>
<tr>
<th>Type of cost</th>
<th>Internal storage</th>
<th>External storage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product related</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal storage</td>
<td>$3</td>
<td>$3</td>
</tr>
<tr>
<td>External storage</td>
<td>$6</td>
<td>$9</td>
</tr>
<tr>
<td><strong>Part related</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal storage</td>
<td>$3;$1</td>
<td>$3;$1</td>
</tr>
<tr>
<td>Within part-type reconfiguration</td>
<td>$6;$2</td>
<td>$6;$2</td>
</tr>
<tr>
<td>Within inv-type reconfiguration</td>
<td>$6;$2</td>
<td>$6;$2</td>
</tr>
<tr>
<td>External storage</td>
<td>$6;$2</td>
<td>$9;$3</td>
</tr>
<tr>
<td>Opportunity cost of unused internal space</td>
<td>$6</td>
<td>$6</td>
</tr>
</tbody>
</table>

a: battery; b: PCB

6.2 Savings from Reconfiguration

The focus of this analysis is to compare the total cost between two systems, one with reconfiguration, and the other with no reconfiguration. In the latter, the reconfiguration among part and inventory types does not take place and the parts are stored only either internally or externally. The results including the optimal values of inventory capacities and the costs are illustrated in Table 6.4, 6.6 and 6.8 for a system with reconfiguration. On the other hand, Table 6.5, 6.7 and 6.9 show the results from a system without reconfiguration. With yield rate difference between battery and PCB lower than 5%, the reconfiguration cost ranges from $62.4 to $104.7 across scenarios. It doubles in most cases when the difference is moderate, ranging from $125.4 to $191.4. With differences higher than 20%, the reconfiguration cost jumps to $1,143.2 from $185.7 due to reconfiguring space for new parts in the finished part inventory, and excess PCBs sharing space with battery in repairable part inventory with any overflow in the usable part inventory.
Table 6.4 With reconfiguration - Low difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C.1 C.2 C.3 C.4 C.5 C.6 C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>157 149 152 150 144 138 144</td>
<td>$5150.0 (1114.7)</td>
<td>$4780.7 (146.0)</td>
<td>$62.4 (13.6)</td>
<td>$27.9 (4.4)</td>
<td>24.0</td>
<td>154.3</td>
</tr>
<tr>
<td>2</td>
<td>149 152 142 143 136 136 136</td>
<td>$5133.6 (128.7)</td>
<td>$4884.9 (125.0)</td>
<td>$78.5 (14.7)</td>
<td>$36.4 (5.1)</td>
<td>23.2</td>
<td>211.8</td>
</tr>
<tr>
<td>3</td>
<td>152 155 152 150 139 136 136</td>
<td>$5229.9 (243.7)</td>
<td>$4955.6 (216.7)</td>
<td>$94.6 (24.5)</td>
<td>$21.7 (5.8)</td>
<td>28.5</td>
<td>157.4</td>
</tr>
<tr>
<td>4</td>
<td>155 155 154 151 141 137 137</td>
<td>$5345.8 (218.0)</td>
<td>$5100.5 (194.7)</td>
<td>$101.9 (20.2)</td>
<td>$29.4 (4.8)</td>
<td>19.0</td>
<td>250.7</td>
</tr>
<tr>
<td>5</td>
<td>153 153 152 150 142 137 137</td>
<td>$5266.0 (211.3)</td>
<td>$4967.7 (140.0)</td>
<td>$104.7 (25.8)</td>
<td>$28.0 (6.9)</td>
<td>27.6</td>
<td>155.0</td>
</tr>
</tbody>
</table>

Table 6.5 No reconfiguration - Low difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C.1 C.2 C.3 C.4 C.5 C.6 C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>157 159 150 148 143 133 133</td>
<td>$5121.4 (175.0)</td>
<td>$4821.7 (154.3)</td>
<td>0</td>
<td>$97.5 (12.3)</td>
<td>33.7</td>
<td>179.9</td>
</tr>
<tr>
<td>2</td>
<td>159 157 151 148 143 143 143</td>
<td>$5250.1 (162.9)</td>
<td>$4913.9 (213.9)</td>
<td>0</td>
<td>$96.8 (15.7)</td>
<td>39.9</td>
<td>252.0</td>
</tr>
<tr>
<td>3</td>
<td>148 152 141 148 143 133 133</td>
<td>$5402.7 (199.8)</td>
<td>$5076.6 (182.2)</td>
<td>0</td>
<td>$78.3 (18.1)</td>
<td>41.3</td>
<td>187.1</td>
</tr>
<tr>
<td>4</td>
<td>152 154 144 138 137 137 137</td>
<td>$5589.9 (206.7)</td>
<td>$5214.2 (196.8)</td>
<td>0</td>
<td>$124.3 (12.5)</td>
<td>41.9</td>
<td>247.7</td>
</tr>
<tr>
<td>5</td>
<td>154 154 152 140 139 137 137</td>
<td>$5152.4 (104.9)</td>
<td>$4853.1 (143.0)</td>
<td>$125.4 (30.8)</td>
<td>$24.5 (4.8)</td>
<td>24.9</td>
<td>257.1</td>
</tr>
</tbody>
</table>

Table 6.6 With reconfiguration - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C.1 C.2 C.3 C.4 C.5 C.6 C.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>152 154 140 139 105 50 50</td>
<td>$5152.4 (104.9)</td>
<td>$4853.1 (143.0)</td>
<td>$125.4 (30.8)</td>
<td>$24.5 (4.8)</td>
<td>24.9</td>
<td>257.1</td>
</tr>
<tr>
<td>2</td>
<td>154 158 139 137 108 50 50</td>
<td>$5294.5 (126.0)</td>
<td>$4971.9 (124.8)</td>
<td>$136.9 (33.3)</td>
<td>$29.7 (7.4)</td>
<td>26.0</td>
<td>241.6</td>
</tr>
<tr>
<td>3</td>
<td>158 154 143 143 113 50 50</td>
<td>$5403.2 (194.7)</td>
<td>$5040.3 (134.6)</td>
<td>$185.7 (53.3)</td>
<td>$22.4 (6.0)</td>
<td>25.8</td>
<td>206.0</td>
</tr>
<tr>
<td>4</td>
<td>147 147 136 132 102 49 49</td>
<td>$5716.4 (140.6)</td>
<td>$5333.3 (124.0)</td>
<td>$191.4 (52.3)</td>
<td>$29.7 (6.2)</td>
<td>27.0</td>
<td>256.3</td>
</tr>
<tr>
<td>5</td>
<td>149 149 139 131 101 50 50</td>
<td>$6182.0 (203.3)</td>
<td>$5805.5 (144.1)</td>
<td>$186.7 (48.9)</td>
<td>$21.2 (4.6)</td>
<td>28.1</td>
<td>181.2</td>
</tr>
</tbody>
</table>
Table 6.7 No reconfiguration - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>145 135 108 48 30 8 100</td>
<td>$5325.7 ($126.7)</td>
<td>$4966.4 ($145.7)</td>
<td>$0</td>
<td>$184.7 ($25.2)</td>
<td>29.1 (3.0)</td>
<td>248.7</td>
</tr>
<tr>
<td>2</td>
<td>153 140 110 28 50 30 8 105</td>
<td>$5442.6 ($203.4)</td>
<td>$4963.2 ($191.7)</td>
<td>$0</td>
<td>$327.0 ($18.3)</td>
<td>25.4 (4.6)</td>
<td>194.4</td>
</tr>
<tr>
<td>3</td>
<td>157 144 115 27 50 30 4 116</td>
<td>$5599.3 ($141.6)</td>
<td>$5080.3 ($146.2)</td>
<td>$0</td>
<td>$323.4 ($19.8)</td>
<td>32.6 (6.2)</td>
<td>190.3</td>
</tr>
<tr>
<td>4</td>
<td>146 138 106 29 49 30 4 133</td>
<td>$5899.3 ($184.3)</td>
<td>$5347.6 ($164.8)</td>
<td>$0</td>
<td>$383.7 ($20.9)</td>
<td>28.0 (5.7)</td>
<td>181.3</td>
</tr>
<tr>
<td>5</td>
<td>157 139 107 25 49 30 4 149</td>
<td>$6465.7 ($190.6)</td>
<td>$5947.7 ($184.9)</td>
<td>$0</td>
<td>$368.6 ($21.8)</td>
<td>24.9 (6.1)</td>
<td>149.4</td>
</tr>
</tbody>
</table>

Table 6.8 With reconfiguration - High difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>143 81 166 50 30 30 113</td>
<td>$6196.7 ($286.7)</td>
<td>$3586.7 ($119.7)</td>
<td>$647.3 ($174.6)</td>
<td>$1908.7 ($82.3)</td>
<td>9.0 (1.5)</td>
<td>190.8</td>
</tr>
<tr>
<td>2</td>
<td>156 84 169 50 30 30 126</td>
<td>$6370.4 ($324.0)</td>
<td>$3203.2 ($134.6)</td>
<td>$917.6 ($126.0)</td>
<td>$2140.4 ($58.4)</td>
<td>20.2 (4.0)</td>
<td>217.6</td>
</tr>
<tr>
<td>3</td>
<td>162 89 170 50 30 30 132</td>
<td>$6948.5 ($301.8)</td>
<td>$3444.0 ($93.0)</td>
<td>$1143.2 ($117.8)</td>
<td>$2273.3 ($68.0)</td>
<td>15.0 (2.7)</td>
<td>171.0</td>
</tr>
<tr>
<td>4</td>
<td>157 87 170 50 30 30 127</td>
<td>$7639.9 ($526.0)</td>
<td>$4270.1 ($134.8)</td>
<td>$827.9 ($136.4)</td>
<td>$2430.9 ($80.1)</td>
<td>16.0 (5.1)</td>
<td>257.9</td>
</tr>
<tr>
<td>5</td>
<td>148 85 163 50 30 30 118</td>
<td>$8654.7 ($511.3)</td>
<td>$4470.9 ($126.2)</td>
<td>$911.0 ($104.1)</td>
<td>$3127.6 ($92.8)</td>
<td>23.0 (6.2)</td>
<td>158.6</td>
</tr>
</tbody>
</table>

Table 6.9 No reconfiguration - High difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Capacities</th>
<th>Total cost ($)</th>
<th>Internal cost ($)</th>
<th>Reconfigured cost ($)</th>
<th>External cost ($)</th>
<th>Unused space</th>
<th>Runtime (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>159 88 157 50 30 6 30 80</td>
<td>$6780.3 ($382.7)</td>
<td>$4133.0 ($142.6)</td>
<td>0</td>
<td>$2571.1 ($169.0)</td>
<td>12.7 (2.6)</td>
<td>167.2</td>
</tr>
<tr>
<td>2</td>
<td>157 87 155 50 30 6 30 81</td>
<td>$7502.9 ($495.8)</td>
<td>$3384.7 ($153.8)</td>
<td>0</td>
<td>$3943.6 ($155.2)</td>
<td>29.1 (5.7)</td>
<td>252.9</td>
</tr>
<tr>
<td>3</td>
<td>147 84 149 50 30 5 30 86</td>
<td>$8011.6 ($599.0)</td>
<td>$3173.4 ($120.4)</td>
<td>0</td>
<td>$4684.0 ($574.3)</td>
<td>25.7 (3.0)</td>
<td>154.9</td>
</tr>
<tr>
<td>4</td>
<td>143 80 141 50 30 5 30 84</td>
<td>$10,342.7 ($642.9)</td>
<td>$4951.7 ($193.1)</td>
<td>0</td>
<td>$5243.4 ($583.6)</td>
<td>24.6 (2.5)</td>
<td>170.7</td>
</tr>
<tr>
<td>5</td>
<td>152 87 149 50 30 6 30 88</td>
<td>$12,999.4 ($622.3)</td>
<td>$5142.3 ($167.3)</td>
<td>0</td>
<td>$7680.1 ($437.9)</td>
<td>29.5 (1.9)</td>
<td>194.3</td>
</tr>
</tbody>
</table>
As displayed by the usable column in Table 6.10 and 6.11, 42% to 68% of reconfiguration takes place in the usable part inventory when the difference in yield rates is low and moderate. The reason is that space is reconfigured in the usable part inventory to accommodate the extra repairable PCBs. In both cases, due to the higher number of PCBs disposed of, reconfiguring space from battery to accommodate the extra PCBs causes the reconfiguration cost in the disposable part inventory, which is almost twice as much in the moderate case.

Table 6.10 Reconfiguration cost by inventory type - Low difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Usable</th>
<th>Disposable</th>
<th>Repairable</th>
<th>Finished</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$32.3</td>
<td>$9.5</td>
<td>$15.3</td>
<td>$5.3</td>
<td>$0</td>
</tr>
<tr>
<td>2</td>
<td>$43.7</td>
<td>$12.5</td>
<td>$18.2</td>
<td>$4.1</td>
<td>$0</td>
</tr>
<tr>
<td>3</td>
<td>$60.3</td>
<td>$7.9</td>
<td>$20.3</td>
<td>$6.1</td>
<td>$0</td>
</tr>
<tr>
<td>4</td>
<td>$64.2</td>
<td>$4.7</td>
<td>$28.2</td>
<td>$4.8</td>
<td>$0</td>
</tr>
<tr>
<td>5</td>
<td>$71.7</td>
<td>$3.6</td>
<td>$22.2</td>
<td>$7.2</td>
<td>$0</td>
</tr>
</tbody>
</table>

Table 6.11 Reconfiguration cost by inventory type - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Usable</th>
<th>Disposable</th>
<th>Repairable</th>
<th>Finished</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$53.1</td>
<td>$16.2</td>
<td>$32.6</td>
<td>$23.5</td>
<td>$0</td>
</tr>
<tr>
<td>2</td>
<td>$86.6</td>
<td>$8.6</td>
<td>$25.6</td>
<td>$16.1</td>
<td>$0</td>
</tr>
<tr>
<td>3</td>
<td>$107.1</td>
<td>$13.8</td>
<td>$28.0</td>
<td>$36.8</td>
<td>$0</td>
</tr>
<tr>
<td>4</td>
<td>$105.2</td>
<td>$10.4</td>
<td>$35.4</td>
<td>$40.4</td>
<td>$0</td>
</tr>
<tr>
<td>5</td>
<td>$90.6</td>
<td>$11.0</td>
<td>$44.8</td>
<td>$40.3</td>
<td>$0</td>
</tr>
</tbody>
</table>

Table 6.12 Reconfiguration cost by inventory type - High difference in yield rates

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Usable</th>
<th>Disposable</th>
<th>Repairable</th>
<th>Finished</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$183.9</td>
<td>$0</td>
<td>$106.4</td>
<td>$357.0</td>
<td>$0</td>
</tr>
<tr>
<td>2</td>
<td>$107.2</td>
<td>$0</td>
<td>$102.7</td>
<td>$707.7</td>
<td>$0</td>
</tr>
<tr>
<td>3</td>
<td>$221.3</td>
<td>$0</td>
<td>$118.1</td>
<td>$803.8</td>
<td>$0</td>
</tr>
<tr>
<td>4</td>
<td>$210.1</td>
<td>$0</td>
<td>$116.3</td>
<td>$501.5</td>
<td>$0</td>
</tr>
<tr>
<td>5</td>
<td>$192.5</td>
<td>$0</td>
<td>$113.8</td>
<td>$604.7</td>
<td>$0</td>
</tr>
</tbody>
</table>
As shown in Table 6.12, under the high difference case, the reconfiguration cost in the finished part inventory is 66% of the total reconfiguration cost. This sudden increase, from roughly $5.5 (low difference) to $594.9, is mainly due to reconfiguring space for new parts in the finished part inventory. The low yield rates cause an increased demand for new parts. Since there is not enough space in the new part inventory, the overflow is stored in the finished part inventory. A sudden jump in the repairable reconfiguration cost is associated with reconfiguring space from battery to store the excess number of repairable PCBs, with any overflow stored by reconfiguring space from usable part inventory.

Table 6.13 Total cost summary for the three scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total cost ($)</th>
<th>With reconfiguration</th>
<th>Without reconfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
<td>Moderate</td>
<td>High</td>
</tr>
<tr>
<td>1</td>
<td>$5015.0</td>
<td>$5152.4</td>
<td>$6196.7 ($286.7)</td>
</tr>
<tr>
<td>2</td>
<td>$5133.6</td>
<td>$5294.5</td>
<td>$6370.4 ($324.0)</td>
</tr>
<tr>
<td>3</td>
<td>$5242.9</td>
<td>$5403.2</td>
<td>$6948.5 ($301.8)</td>
</tr>
<tr>
<td>4</td>
<td>$5345.8</td>
<td>$5716.4</td>
<td>$7639.9 ($526.0)</td>
</tr>
<tr>
<td>5</td>
<td>$5266.0</td>
<td>$6182.0</td>
<td>$8654.7 ($511.3)</td>
</tr>
</tbody>
</table>

Table 6.14 Summary of savings from reconfiguration

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Savings from reconfiguration</th>
<th>Low</th>
<th>Moderate</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Savings ($)</td>
<td>Savings / Total cost w/o reconfiguration (%)</td>
<td>Savings ($)</td>
<td>Savings / Total cost w/o reconfiguration (%)</td>
</tr>
<tr>
<td>1</td>
<td>$106.4</td>
<td>2.1%</td>
<td>$143.3</td>
<td>2.7%</td>
</tr>
<tr>
<td>1.5</td>
<td>$116.5</td>
<td>2.2%</td>
<td>$148.1</td>
<td>2.7%</td>
</tr>
<tr>
<td>2</td>
<td>$159.8</td>
<td>3.0%</td>
<td>$196.1</td>
<td>3.5%</td>
</tr>
<tr>
<td>2.5</td>
<td>$172.6</td>
<td>3.1%</td>
<td>$182.9</td>
<td>3.1%</td>
</tr>
<tr>
<td>3</td>
<td>$323.9</td>
<td>5.8%</td>
<td>$283.7</td>
<td>4.4%</td>
</tr>
</tbody>
</table>
Table 6.13 summarizes the total cost for the ‘with’ vs. ‘without’ reconfiguration cases. Within each ratio (scenario), as the difference in yield rates gets higher, the higher the savings are. As an example, if the ratio of unit external cost to unit reconfiguration cost is 1, the savings in dollars have a jump from $106.4 to $583.6 with higher difference in yield rates. The highest jump in savings from $172.6 (low) to $2702.8 (high) is observed when the ratio is 2.5. Overall, the savings are higher as the ratio of unit external to unit reconfiguration increases. Under low and moderate differences in yield rates, the savings are up to 6%. For the high yield rate differences, the saving from reconfiguration is higher and reaches to 33.4% as indicated by the highlighted column in blue in Table 6.14.

Figure 6.1 Savings from reconfiguration vs. ratio

Figure 6.1 displays the savings across five ratio scenarios. Within each ratio, the highest savings are observed when the difference in yield rates is high. In order to explain what is happening within each yield rate difference, the slopes of lines are further explored. The slope of the dotted
line is high, indicating the importance of reconfiguration when the ratio of unit external cost to unit reconfiguration cost increases. On the other hand, slopes of the dash and solid lines for the low and moderate cases (respectively) are low, indicating that the impact of a ratio change on the savings is not as significant as in the case with higher difference in yield rates. In other words, savings from reconfiguration are not higher by an increase in the unit external storage cost.

6.3 Design of Experiments (DOE)

Experimental design analysis is used to explore which of the inventory capacities have the highest impact on total cost. The factors (effects) are the seven inventory capacities in the system identified as the returned product (C.1), usable part (C.2), disposable part (C.3), repairable part (C.4), new part (C.5), finished part (C.6) and remanufactured product (C.7). The low and high levels for each inventory capacity were specified in Table 6.2. The response variables are the total costs obtained under each scenario of yield rate difference among parts: low, moderate and high (each case was explained in previous section). The number of parts is 2, being the battery and PCB. The unit internal and reconfiguration storage costs are $3 and $1 for batteries and PCBs, respectively, whereas their unit external storage costs are $6 and $2. For returned and remanufactured cellular phones, the unit internal and external storage costs are $3 and $6, respectively.

Due to the high number of factors, a screening design is first performed, examining just the main effects. The screening design helps reducing the number of factors to five. Then, a full factorial design is done so that both the main and interaction effects can be considered. The interaction effects are further explored using the response surface and contour plots. After identifying the
factors which have the highest impact on the total cost, the regression equations are written for each case, exploring how the total inventory storage cost increases or decreases by a change in the inventory capacities.

6.3.1 Screening Process

A two-level design with seven factors has 128 possible factor combinations. Rather than choosing a full factorial design with all possible combinations, a 1/16 fractional factorial design with 8 runs, $2^{(7-4)}$ (resolution III) is chosen, considering only the main effects (so that fewer runs can be performed). The first stage in the DOE analysis is to screen the design and select the factors that have the largest effects. The fractional design table is shown in Table 6.15.

<table>
<thead>
<tr>
<th>Run</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>Storage space</th>
<th>Total cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low difference ($\leq 5$)</td>
<td>Moderate difference (between 5 and 20)</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$8182.2$</td>
<td>$8112.8$</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$7979.9$</td>
<td>$7747.3$</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$6692.6$</td>
<td>$6868.9$</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$6495.1$</td>
<td>$6770.3$</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$6397.8$</td>
<td>$5843.8$</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$6942.0$</td>
<td>$6905.5$</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$7574.1$</td>
<td>$7558.8$</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$8119.5$</td>
<td>$8666.2$</td>
</tr>
</tbody>
</table>

To determine which of the effects are significant, the P-value columns of ‘estimated effects and coefficients’ tables are used. These tables are displayed in Appendix C. In all three cases, the return, usable, disposable, new and finished inventory capacities are significant since their P-
values are less than 0.05, using $\alpha = 0.05$. On the other hand, both the repairable part and remanufactured product inventory capacities have p-values higher than the alpha value of 0.05.

In the normal plot of the effects as displayed in Figure 6.2, the points that do not fit the line well signal the significant effects, which are further from the fitted line than the insignificant effects. The insignificant effects, on the other hand, are centered around zero. For all three responses, the repairable and remanufactured inventory capacities are identified as insignificant. They are closer to the fitted line. Due to unit internal and reconfiguration storage costs being the same, the extra repairable parts can be stored by reconfiguring space from usable part inventory. The repairable part inventory is therefore identified as insignificant.

The pareto charts which display the absolute value of the effects can also be used to determine which effects are significant. These charts are shown in Figure 6.3. The effect of the finished part inventory capacity is significantly higher than the rest of the effects. It is the highest when
Next, the model is analyzed without using the repairable and remanufactured inventory capacity factors identified as insignificant. In other words, the insignificant factors are screened out and it is investigated whether the model is a good one for further exploration. A good standard by which to evaluate the model is to look at the P-values for each factor (Table 6.16, 6.18 and 6.20). The P-values for each factor in the model are less than $\alpha$ level (0.05), indicating that the model is good for further exploration and validation. The analysis of variance is also shown for each case in Table 6.17, 6.19 and 6.21. A high $T$ value and a low $P$ value for the main effects reinforce the fact that the model is a good one.

### Table 6.16 Estimated effects and coefficients for total cost - Low difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>7308.43</td>
<td>9.215</td>
<td>793.14</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>202.47</td>
<td>101.24</td>
<td>9.215</td>
<td>10.99</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>-173.37</td>
<td>-86.69</td>
<td>9.215</td>
<td>-9.41</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-88.70</td>
<td>-44.35</td>
<td>9.215</td>
<td>-4.81</td>
<td>0.011</td>
</tr>
<tr>
<td>New Cap</td>
<td>407.77</td>
<td>203.89</td>
<td>9.215</td>
<td>22.13</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>1334.52</td>
<td>667.26</td>
<td>9.215</td>
<td>72.41</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### Table 6.17 Analysis of variance for total cost - Low difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>8104641</td>
<td>1620928</td>
<td>1193.16</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>10</td>
<td>13585</td>
<td>1359</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of fit</td>
<td>2</td>
<td>649</td>
<td>325</td>
<td>0.20</td>
<td>0.822</td>
</tr>
<tr>
<td>Pure Error</td>
<td>8</td>
<td>12936</td>
<td>1617</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>8118226</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$S = 36.8581$       $PRESS = 34778.2$

$R-Sq = 99.83\%$     $R-Sq(pred) = 99.57\%$     $R-Sq(adj) = 99.75\%$
### Table 6.18 Estimated effects and coefficients for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>7303.26</td>
<td>22.08</td>
<td>330.74</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>463.21</td>
<td>231.61</td>
<td>22.08</td>
<td>10.49</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>392.24</td>
<td>196.12</td>
<td>22.08</td>
<td>8.88</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-156.26</td>
<td>-78.13</td>
<td>22.08</td>
<td>-3.54</td>
<td>0.005</td>
</tr>
<tr>
<td>New Cap</td>
<td>683.34</td>
<td>341.67</td>
<td>22.08</td>
<td>15.47</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>1413.71</td>
<td>706.86</td>
<td>22.08</td>
<td>32.01</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### Table 6.19 Analysis of variance for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>11433469</td>
<td>2286694</td>
<td>293.11</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>10</td>
<td>78014</td>
<td>7801</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of fit</td>
<td>2</td>
<td>3920</td>
<td>1960</td>
<td>0.21</td>
<td>0.814</td>
</tr>
<tr>
<td>Pure Error</td>
<td>8</td>
<td>74094</td>
<td>9262</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>11511483</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S = 88.3254 \quad \text{PRESS} = 199715 \]

\[ R-Sq = 99.32\% \quad R-Sq(\text{pred}) = 98.27\% \quad R-Sq(\text{adj}) = 98.98\% \]

### Table 6.20 Estimated effects and coefficients for total cost - High difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>8269.0</td>
<td>18.03</td>
<td>458.70</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>513.7</td>
<td>256.8</td>
<td>18.03</td>
<td>14.25</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>344.0</td>
<td>172.0</td>
<td>18.03</td>
<td>9.54</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-694.7</td>
<td>-347.3</td>
<td>18.03</td>
<td>-19.27</td>
<td>0.000</td>
</tr>
<tr>
<td>New Cap</td>
<td>80.6</td>
<td>40.3</td>
<td>18.03</td>
<td>2.24</td>
<td>0.049</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>2723.4</td>
<td>1361.7</td>
<td>18.03</td>
<td>75.53</td>
<td>0.000</td>
</tr>
</tbody>
</table>

### Table 6.21 Analysis of variance for total cost - High difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>33151806</td>
<td>6630361</td>
<td>1275.15</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>10</td>
<td>51997</td>
<td>5200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack of fit</td>
<td>2</td>
<td>26611</td>
<td>13306</td>
<td>4.19</td>
<td>0.057</td>
</tr>
<tr>
<td>Pure Error</td>
<td>8</td>
<td>25386</td>
<td>3173</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>33203803</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ S = 72.1089 \quad \text{PRESS} = 133112 \]

\[ R-Sq = 99.84\% \quad R-Sq(\text{pred}) = 99.60\% \quad R-Sq(\text{adj}) = 99.77\% \]
Further checking of the model can be done by using the residual plots as illustrated in Figure 6.4. The fitted values are the results predicted by the model, and the residuals are the actual total costs minus the predicted total costs. The residual plots also look satisfactory and show no cause for concern. The next step is to generate the main effect plots for the five inventory capacities as shown in Figure 6.5. The main purpose of these plots is to explore the impact of significant effects (inventory capacities) on the total cost individually.

![Residual Plots for Low](image)

**Figure 6.4 Residual plots for all three cases**

![Main Effects Plot for Low](image)

**Figure 6.5 Main effects plots for all three cases**

When the difference in yield rates is less than 5%, the finished part inventory capacity has the highest effect on the total cost as the slope of the line is very steep. The effect of the remaining capacities is small as seen by a smaller slope. Increasing the returned product and new part inventory capacities causes only a slight increase on the total cost. Increasing the usable and...
disposable part inventory capacities, on the other hand, causes only a slight decrease on the total cost.

Under moderate difference in yield rates, the finished part inventory capacity still has the highest impact on the total cost. Increasing the usable part inventory capacity in this case increases the total cost. The returned product, usable and new part inventory capacities have a higher impact on the total cost as seen by a slightly steeper slope, when compared with the case of low difference in yield rates.

The effect of the finished part inventory capacity is still the highest when the difference in yield rates is higher than 20%. Increasing the disposable part inventory capacity causes a higher decrease on the total cost than the cases with the low and moderate differences in yield rates. Increasing the returned product, usable and new part inventory capacities affect the total cost less when compared with the moderate difference in yield rates.

6.3.2 A Full Factorial Design

After concluding that the return, usable, disposable, new and finished part inventory capacities are the significant effects in the model, the next step is to perform a full factorial design with all the possible combinations. Hence, the results can be drawn based on effects free from confounding. A two-level design with five factors has $2^5 = 32$ possible factor combinations. Table 6.22 shows how the design looks like.
### Table 6.22 2^5 design table - Factors and responses

<table>
<thead>
<tr>
<th>Run</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C5</th>
<th>C6</th>
<th>Inventory capacity</th>
<th>Total cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low difference (&lt;= 5)</td>
<td>Moderate difference (between 5 and 20)</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$6450.8</td>
<td>$6062.6</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$6871.1</td>
<td>$6419.5</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$6116.7</td>
<td>$6422.3</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>$6320.4</td>
<td>$6636.1</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$6598.5</td>
<td>$5923.0</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$6763.5</td>
<td>$6048.9</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$6201.5</td>
<td>$6248.4</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>$6451.7</td>
<td>$6432.8</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$7231.0</td>
<td>$6535.0</td>
</tr>
<tr>
<td>10</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$7314.7</td>
<td>$6849.0</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$6773.0</td>
<td>$6975.4</td>
</tr>
<tr>
<td>12</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>$6978.7</td>
<td>$7237.6</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$7188.5</td>
<td>$6370.2</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$7417.9</td>
<td>$6573.5</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$6854.8</td>
<td>$6905.1</td>
</tr>
<tr>
<td>16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>$7055.2</td>
<td>$7008.0</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$7632.1</td>
<td>$7466.8</td>
</tr>
<tr>
<td>18</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$7993.1</td>
<td>$7556.0</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$7320.9</td>
<td>$7811.1</td>
</tr>
<tr>
<td>20</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>$7484.3</td>
<td>$8027.6</td>
</tr>
<tr>
<td>21</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>$7634.9</td>
<td>$7260.7</td>
</tr>
<tr>
<td>22</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>$7796.4</td>
<td>$7495.7</td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>$7232.7</td>
<td>$7668.6</td>
</tr>
<tr>
<td>24</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>$7389.9</td>
<td>$7981.8</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$8507.6</td>
<td>$7995.7</td>
</tr>
<tr>
<td>26</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>$8432.1</td>
<td>$8189.3</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>$7851.9</td>
<td>$8373.1</td>
</tr>
<tr>
<td>28</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>$7911.8</td>
<td>$8707.5</td>
</tr>
<tr>
<td>29</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$8255.5</td>
<td>$7809.3</td>
</tr>
<tr>
<td>30</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$8507.3</td>
<td>$7997.5</td>
</tr>
<tr>
<td>31</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$7880.0</td>
<td>$8288.1</td>
</tr>
<tr>
<td>32</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>$8075.1</td>
<td>$8348.3</td>
</tr>
</tbody>
</table>
Table 6.23 Estimated effects and coefficients - Low difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>7323.2</td>
<td>7.515</td>
<td>974.50</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td><strong>Return</strong></td>
<td>193.7</td>
<td>96.9</td>
<td>7.515</td>
<td>12.89</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Usable</strong></td>
<td>-403.9</td>
<td>-201.9</td>
<td>7.515</td>
<td>-26.87</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Disposable</strong></td>
<td>11.7</td>
<td>5.8</td>
<td>7.515</td>
<td>0.78</td>
<td>0.444</td>
</tr>
<tr>
<td><strong>New</strong></td>
<td>610.9</td>
<td>305.5</td>
<td>7.515</td>
<td>40.65</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Finished</strong></td>
<td>1096.6</td>
<td>548.3</td>
<td>7.515</td>
<td>72.96</td>
<td>0.000</td>
</tr>
<tr>
<td>Return*Usable</td>
<td>14.9</td>
<td>7.5</td>
<td>7.515</td>
<td>0.99</td>
<td>0.329</td>
</tr>
<tr>
<td>Return*Disposable</td>
<td>10.5</td>
<td>5.2</td>
<td>7.515</td>
<td>0.70</td>
<td>0.491</td>
</tr>
<tr>
<td><strong>Return*New</strong></td>
<td>-42.3</td>
<td>-21.1</td>
<td>7.515</td>
<td>-2.81</td>
<td>0.008</td>
</tr>
<tr>
<td>Return*Finished</td>
<td>-19.4</td>
<td>-9.7</td>
<td>7.515</td>
<td>-1.29</td>
<td>0.206</td>
</tr>
<tr>
<td><strong>Usable*Disposable</strong></td>
<td>59.3</td>
<td>29.7</td>
<td>7.515</td>
<td>3.95</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable*New</td>
<td>-20.6</td>
<td>-10.3</td>
<td>7.515</td>
<td>-1.37</td>
<td>0.180</td>
</tr>
<tr>
<td><strong>Usable*Finished</strong></td>
<td>-35.9</td>
<td>-18.0</td>
<td>7.515</td>
<td>-2.39</td>
<td>0.023</td>
</tr>
<tr>
<td>Disposable*New</td>
<td>2.6</td>
<td>1.3</td>
<td>7.515</td>
<td>0.17</td>
<td>0.863</td>
</tr>
<tr>
<td>Disposable*Finished</td>
<td>-36.5</td>
<td>-18.2</td>
<td>7.515</td>
<td>-2.43</td>
<td>0.021</td>
</tr>
<tr>
<td>New*Finished</td>
<td>1.9</td>
<td>0.9</td>
<td>7.515</td>
<td>0.12</td>
<td>0.902</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable</td>
<td>-4.3</td>
<td>-2.2</td>
<td>7.515</td>
<td>-0.29</td>
<td>0.776</td>
</tr>
<tr>
<td>Return<em>Usable</em>New</td>
<td>38.1</td>
<td>19.0</td>
<td>7.515</td>
<td>2.53</td>
<td>0.016</td>
</tr>
<tr>
<td>Return<em>Usable</em>Finished</td>
<td>-20.1</td>
<td>-10.0</td>
<td>7.515</td>
<td>-1.34</td>
<td>0.191</td>
</tr>
<tr>
<td>Return<em>Disposable</em>New</td>
<td>38.3</td>
<td>19.2</td>
<td>7.515</td>
<td>2.55</td>
<td>0.016</td>
</tr>
<tr>
<td>Return<em>Disposable</em>Finished</td>
<td>8.1</td>
<td>4.0</td>
<td>7.515</td>
<td>0.54</td>
<td>0.594</td>
</tr>
<tr>
<td>Return<em>New</em>Finished</td>
<td>18.6</td>
<td>9.3</td>
<td>7.515</td>
<td>1.24</td>
<td>0.226</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>New</td>
<td>19.3</td>
<td>9.7</td>
<td>7.515</td>
<td>1.29</td>
<td>0.208</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>Finished</td>
<td>-16.7</td>
<td>-8.4</td>
<td>7.515</td>
<td>-1.11</td>
<td>0.274</td>
</tr>
<tr>
<td><strong>Usable<em>New</em>Finished</strong></td>
<td>-41.1</td>
<td>-20.5</td>
<td>7.515</td>
<td>-2.73</td>
<td>0.010</td>
</tr>
<tr>
<td>Disposable<em>New</em>Finished</td>
<td>-4.8</td>
<td>-2.4</td>
<td>7.515</td>
<td>-0.32</td>
<td>0.751</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable*New</td>
<td>-51.1</td>
<td>-25.6</td>
<td>7.515</td>
<td>-3.40</td>
<td>0.002</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable*Finished</td>
<td>10.5</td>
<td>5.3</td>
<td>7.515</td>
<td>0.70</td>
<td>0.489</td>
</tr>
<tr>
<td>Return<em>Usable</em>New*Finished</td>
<td>-14.8</td>
<td>-7.4</td>
<td>7.515</td>
<td>-0.99</td>
<td>0.332</td>
</tr>
<tr>
<td>Return<em>Disposable</em>New*Finished</td>
<td>0.6</td>
<td>0.3</td>
<td>7.515</td>
<td>0.04</td>
<td>0.969</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>New*Finished</td>
<td>21.6</td>
<td>10.8</td>
<td>7.515</td>
<td>1.44</td>
<td>0.160</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable<em>New</em>Finished</td>
<td>9.1</td>
<td>4.6</td>
<td>7.515</td>
<td>0.61</td>
<td>0.548</td>
</tr>
</tbody>
</table>
Yield rate differences less than 5%

The values in the P column of Table 6.23 are used to determine which of the effects are significant. The alpha value ($\alpha$) is kept the same at 0.05. The blue highlighted factors in Table 6.23 are identified as significant since they have P-values less than 0.05. The T value for the finished part inventory is high with a value of 72.96 than the rest of the main effects, meaning that its impact is higher on the total cost. The low P-value for the main effects and the two, three and four-way interactions indicates that they are significant as shown on the analysis of variance table (Table 6.24). On the other hand, the P-value is 0.548 which is higher than 0.05 for the five-way interaction which indicates that its effect on the total cost is not significant.

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>28425322</td>
<td>5685064</td>
<td>1572.97</td>
<td>0.000</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>10</td>
<td>145147</td>
<td>14515</td>
<td>4.02</td>
<td>0.001</td>
</tr>
<tr>
<td>3-Way Interactions</td>
<td>10</td>
<td>97837</td>
<td>9784</td>
<td>2.71</td>
<td>0.016</td>
</tr>
<tr>
<td>4-Way Interactions</td>
<td>5</td>
<td>54584</td>
<td>10917</td>
<td>3.02</td>
<td>0.024</td>
</tr>
<tr>
<td>5-Way Interactions</td>
<td>1</td>
<td>1330</td>
<td>1330</td>
<td>0.37</td>
<td>0.548</td>
</tr>
<tr>
<td>Residual Error</td>
<td>32</td>
<td>115655</td>
<td>3614</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>32</td>
<td>115655</td>
<td>3614</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>28839877</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 60.1185  
PRESS = 462622

R-Sq = 99.60%  
R-Sq(pred) = 98.40%  
R-Sq(adj) = 99.21%

As shown in Figure 6.6, the normal plot of the standardized effects shows that the main effects have a higher impact on the total cost than the interaction effects as they are further from the fitted line. The interaction effects are centered around zero. The returned product, usable, new and finished part inventory capacities are identified as significant main effects. The reason the new part inventory turns out to be a significant effect is that even though there is almost no
demand for new parts, the opportunity cost of unused space is $6 which in turn impacts the total cost if the capacity of the new part inventory were to be increased. From the two-way interaction effects, the interactions between (1) usable and disposable, (2) usable and finished, (3) disposable and finished part inventory capacities are significant. From the three-way interaction effects, the interactions among (1) returned product, disposable and new, (2) returned product, usable and new, (3) usable, new and finished part inventory capacities are significant. From the four-way interaction effects, only the interaction among returned product, usable, disposable and new part inventory capacities is identified to be significant.

![Normal Plot of the Standardized Effects](image)

**Figure 6.6 Normal plot of the standardized effects - Low difference in yield rates**

**Yield rate differences between 5% and 20%**

In the case where the differences in yield rates are moderate, all the main effects are significant as highlighted in blue in Table 6.25. A high T value (154.95) with the finished part inventory
capacity is observed again which is similar to the case with yield rates differences less than 5%.

As also shown in Table 6.25, the significant interaction effects with a P value less than 0.05 are given.

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>7240.30</td>
<td>4.540</td>
<td>1594.70</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Return</td>
<td>215.14</td>
<td>107.57</td>
<td>4.540</td>
<td>23.69</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable</td>
<td>395.83</td>
<td>197.92</td>
<td>4.540</td>
<td>43.59</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable</td>
<td>-190.12</td>
<td>-95.06</td>
<td>4.540</td>
<td>-20.94</td>
<td>0.000</td>
</tr>
<tr>
<td>New</td>
<td>562.36</td>
<td>281.18</td>
<td>4.540</td>
<td>61.93</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished</td>
<td>1407.06</td>
<td>703.53</td>
<td>4.540</td>
<td>154.95</td>
<td>0.000</td>
</tr>
</tbody>
</table>

| Return*Usable               | -3.42   | -1.71       | 4.540          | -0.38  | 0.709  |
| Return*Disposable           | -33.20  | -16.60      | 4.540          | -3.66  | 0.001  |
| Return*New                  | 11.26   | 5.63        | 4.540          | 1.24   | 0.224  |
| Return*Finished             | -9.75   | -4.88       | 4.540          | -1.07  | 0.291  |
| Usable*Disposable           | 15.56   | 7.78        | 4.540          | 1.71   | 0.096  |
| Usable*New                  | 22.15   | 11.08       | 4.540          | 2.44   | 0.020  |
| Usable*Finished             | 23.92   | 11.96       | 4.540          | 2.63   | 0.013  |
| Disposable*New              | -23.94  | -11.97      | 4.540          | -2.64  | 0.013  |
| Disposable*Finished         | 8.66    | 4.33        | 4.540          | 0.95   | 0.348  |
| New*Finished                | 2.67    | 1.33        | 4.540          | 0.29   | 0.771  |
| Return*Usable*Disposable    | -5.78   | -2.89       | 4.540          | -0.64  | 0.529  |
| Return*Usable*New           | -24.56  | -12.28      | 4.540          | -2.70  | 0.011  |
| Return*Usable*Finished      | 16.20   | 8.10        | 4.540          | 1.78   | 0.084  |
| Return*Disposable*New       | -21.51  | -10.75      | 4.540          | -2.37  | 0.024  |
| Return*Disposable*Finished  | 25.13   | 12.57       | 4.540          | 2.77   | 0.009  |
| Return*New*Finished         | -0.56   | -0.28       | 4.540          | -0.06  | 0.952  |
| Usable*Disposable*New       | 0.96    | 0.48        | 4.540          | 0.11   | 0.917  |
| Usable*Disposable*Finished  | -28.19  | -14.10      | 4.540          | -3.10  | 0.004  |
| Usable*New*Finished         | -37.93  | -18.97      | 4.540          | -4.18  | 0.000  |
| Disposable*New*Finished     | -21.76  | -10.88      | 4.540          | -2.40  | 0.023  |
| Return*Usable*Disposable*New| -33.99  | -17.00      | 4.540          | -3.74  | 0.001  |
| Return*Usable*Disposable*Finished| -17.86| -8.93      | 4.540          | -1.97  | 0.058  |
| Return*Usable*New*Finished  | -1.84   | -0.92       | 4.540          | -0.20  | 0.840  |
| Return*Disposable*New*Finished| -24.10| -12.05      | 4.540          | -2.65  | 0.012  |
| Return*Usable*Disposable*New*Finished| 1.18 | 0.59       | 4.540          | 0.13   | 0.898  |
As it was true in the case with low differences in yield rates, the main effects and two, three, four-way interactions have a significant effect on the total cost as shown in Table 6.26. Among the significant interactions, the three and four-way interactions have a higher impact on the total cost than the two-way interactions because of the higher F values. The P-value for the five-way interaction is 0.898 which is higher than 0.05. Hence, its impact on the total cost is determined not to be important.

Table 6.26 Analysis of variance for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>40563091</td>
<td>8112618</td>
<td>6149.27</td>
<td>0.000</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>10</td>
<td>52743</td>
<td>5274</td>
<td>4.00</td>
<td>0.001</td>
</tr>
<tr>
<td>3-Way Interactions</td>
<td>10</td>
<td>75227</td>
<td>7523</td>
<td>5.70</td>
<td>0.000</td>
</tr>
<tr>
<td>4-Way Interactions</td>
<td>5</td>
<td>39322</td>
<td>7864</td>
<td>5.96</td>
<td>0.001</td>
</tr>
<tr>
<td>5-Way Interactions</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td>0.02</td>
<td>0.898</td>
</tr>
<tr>
<td>Residual Error</td>
<td>32</td>
<td>42217</td>
<td>1319</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>32</td>
<td>42217</td>
<td>1319</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>40772622</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 36.3219      PRESS = 168868
R-Sq = 99.90%     R-Sq(pred) = 99.59%     R-Sq(adj) = 99.80%

As shown in Figure 6.7, the returned product, usable, disposable, new and finished part inventory capacities are the significant main effects. Similar to the case with low differences, the new part inventory is identified as significant due to the opportunity of unused space. Even though there is almost no demand for new parts, the opportunity cost of unused space is $6 which in turn impacts the total cost if the capacity of the new part inventory were to be increased. From the two-way interactions, the interactions between (1) usable and new, (2) usable and finished, (3) disposable and new, (4) returned product and disposable part inventory capacities are significant. The following three-way interactions have an impact on the total cost: (1) returned product,
disposable and finished, (2) returned product, disposable and new, (3) disposable, new and finished, (4) returned product, usable and new, (5) usable, disposable and finished, and (6) usable, new and finished. From the four-way interactions, the interactions among (1) usable, disposable, new and finished, (2) returned product, disposable, new and finished, and (3) returned product, usable, disposable and new part inventory capacities are significant.

![Normal Plot of the Standardized Effects](image)

Figure 6.7 Normal plot of the standardized effects - Moderate difference in yield rates

**Yield rate differences higher than 20%**

In the case when the difference in yield rates is high, the impact of the finished part inventory capacity on total cost is even higher than the low and moderate cases. As shown in Table 6.27, the T value for the finished part inventory capacity is 134.6. The only two-way interaction effect is between new and finished part inventory capacities. From the three-way interactions, the interactions among (1) the returned product, usable and new, (2) returned product, disposable
and new part inventory capacities are significant. The interaction among returned product, usable, disposable and finished part inventory capacities is the only significant four-way interaction effect. The estimated effects and coefficients are summarized in Table 6.27.

Table 6.27 Estimated effects and coefficients - High difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>8298.0</td>
<td>9.966</td>
<td>832.67</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Return</td>
<td>166.7</td>
<td>83.4</td>
<td>9.966</td>
<td>8.36</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable</td>
<td>323.4</td>
<td>161.7</td>
<td>9.966</td>
<td>16.23</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable</td>
<td>-636.4</td>
<td>-318.2</td>
<td>9.966</td>
<td>-31.93</td>
<td>0.000</td>
</tr>
<tr>
<td>New</td>
<td>-16.9</td>
<td>-8.4</td>
<td>9.966</td>
<td>-0.85</td>
<td>0.403</td>
</tr>
<tr>
<td>Finished</td>
<td>2683.3</td>
<td>1341.7</td>
<td>9.966</td>
<td>134.63</td>
<td>0.000</td>
</tr>
<tr>
<td>Return*Usable</td>
<td>1.1</td>
<td>0.6</td>
<td>9.966</td>
<td>0.06</td>
<td>0.954</td>
</tr>
<tr>
<td>Return*Disposable</td>
<td>2.3</td>
<td>1.1</td>
<td>9.966</td>
<td>0.11</td>
<td>0.909</td>
</tr>
<tr>
<td>Return*New</td>
<td>13.6</td>
<td>6.8</td>
<td>9.966</td>
<td>0.68</td>
<td>0.501</td>
</tr>
<tr>
<td>Return*Finished</td>
<td>-11.0</td>
<td>-5.5</td>
<td>9.966</td>
<td>-0.55</td>
<td>0.586</td>
</tr>
<tr>
<td>Usable*Disposable</td>
<td>-5.6</td>
<td>-2.8</td>
<td>9.966</td>
<td>-0.28</td>
<td>0.779</td>
</tr>
<tr>
<td>Usable*New</td>
<td>15.3</td>
<td>7.6</td>
<td>9.966</td>
<td>0.77</td>
<td>0.450</td>
</tr>
<tr>
<td>Usable*Finished</td>
<td>0.6</td>
<td>0.3</td>
<td>9.966</td>
<td>0.03</td>
<td>0.977</td>
</tr>
<tr>
<td>Disposable*New</td>
<td>10.9</td>
<td>5.5</td>
<td>9.966</td>
<td>0.55</td>
<td>0.587</td>
</tr>
<tr>
<td>Disposable*Finished</td>
<td>-14.0</td>
<td>-7.0</td>
<td>9.966</td>
<td>-0.70</td>
<td>0.489</td>
</tr>
<tr>
<td>New*Finished</td>
<td>192.4</td>
<td>96.2</td>
<td>9.966</td>
<td>9.65</td>
<td>0.000</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable</td>
<td>3.7</td>
<td>1.8</td>
<td>9.966</td>
<td>0.18</td>
<td>0.856</td>
</tr>
<tr>
<td>Return<em>Usable</em>New</td>
<td>-48.3</td>
<td>-24.2</td>
<td>9.966</td>
<td>-2.42</td>
<td>0.021</td>
</tr>
<tr>
<td>Return<em>Usable</em>Finished</td>
<td>3.2</td>
<td>1.6</td>
<td>9.966</td>
<td>0.16</td>
<td>0.875</td>
</tr>
<tr>
<td>Return<em>Disposable</em>New</td>
<td>57.5</td>
<td>28.7</td>
<td>9.966</td>
<td>2.88</td>
<td>0.007</td>
</tr>
<tr>
<td>Return<em>Disposable</em>Finished</td>
<td>6.2</td>
<td>3.1</td>
<td>9.966</td>
<td>0.31</td>
<td>0.757</td>
</tr>
<tr>
<td>Return<em>New</em>Finished</td>
<td>38.8</td>
<td>19.4</td>
<td>9.966</td>
<td>1.95</td>
<td>0.060</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>New</td>
<td>9.8</td>
<td>4.9</td>
<td>9.966</td>
<td>0.49</td>
<td>0.627</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>Finished</td>
<td>16.4</td>
<td>8.2</td>
<td>9.966</td>
<td>0.82</td>
<td>0.417</td>
</tr>
<tr>
<td>Usable<em>New</em>Finished</td>
<td>-7.4</td>
<td>-3.7</td>
<td>9.966</td>
<td>-0.37</td>
<td>0.713</td>
</tr>
<tr>
<td>Disposable<em>New</em>Finished</td>
<td>-5.6</td>
<td>-2.8</td>
<td>9.966</td>
<td>-0.28</td>
<td>0.781</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable*New</td>
<td>-37.0</td>
<td>-18.5</td>
<td>9.966</td>
<td>-1.86</td>
<td>0.072</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable*Finished</td>
<td>-42.3</td>
<td>-21.1</td>
<td>9.966</td>
<td>-2.12</td>
<td>0.042</td>
</tr>
<tr>
<td>Return<em>Usable</em>New*Finished</td>
<td>-7.8</td>
<td>-3.9</td>
<td>9.966</td>
<td>-0.39</td>
<td>0.699</td>
</tr>
<tr>
<td>Return<em>Disposable</em>New*Finished</td>
<td>17.5</td>
<td>8.8</td>
<td>9.966</td>
<td>0.88</td>
<td>0.386</td>
</tr>
<tr>
<td>Usable<em>Disposable</em>New*Finished</td>
<td>20.5</td>
<td>10.2</td>
<td>9.966</td>
<td>1.03</td>
<td>0.311</td>
</tr>
<tr>
<td>Return<em>Usable</em>Disposable<em>New</em>Finished</td>
<td>17.9</td>
<td>8.9</td>
<td>9.966</td>
<td>0.90</td>
<td>0.377</td>
</tr>
</tbody>
</table>
Table 6.28 shows that the three, four and five-way interaction effects have a lower impact on the total cost as their P-values are higher than 0.05. The high F and low P values for the main and two-way interactions emphasize the significance of these effects.

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>5</td>
<td>123807323</td>
<td>24761465</td>
<td>3895.76</td>
<td>0.000</td>
</tr>
<tr>
<td>2-Way Interactions</td>
<td>10</td>
<td>606656</td>
<td>60666</td>
<td>9.54</td>
<td>0.000</td>
</tr>
<tr>
<td>3-Way Interactions</td>
<td>10</td>
<td>122502</td>
<td>12250</td>
<td>1.93</td>
<td>0.078</td>
</tr>
<tr>
<td>4-Way Interactions</td>
<td>5</td>
<td>63148</td>
<td>12630</td>
<td>1.99</td>
<td>0.107</td>
</tr>
<tr>
<td>5-Way Interactions</td>
<td>1</td>
<td>5110</td>
<td>5110</td>
<td>0.80</td>
<td>0.377</td>
</tr>
<tr>
<td>Residual Error</td>
<td>32</td>
<td>203392</td>
<td>6356</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pure Error</td>
<td>32</td>
<td>203392</td>
<td>6356</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>124808131</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 79.7245    PRESS = 813568
R-Sq = 99.84%  R-Sq(pred) = 99.35%  R-Sq(adj) = 99.68%

As explained previously by the high T value, the normal plot in Figure 6.8 reinforces the higher impact of finished inventory capacity on total cost compared to the other main and interaction effects. The reason is that with low yields of finished parts, the demand for new parts increases significantly. Since there is not enough space in the new part inventory to store them, extras are stored in the finished part inventory by reconfiguring space.

Upon identifying the significant effects, next step is to fit a model by screening out the insignificant to see whether the design is a good one for further exploration. By examining the P-values from the reduced model, and the residual plots, the model turns out to be a satisfactory one.
The effects of significant two-way interactions are further explored using response surface and contour plots. As explained before, when the difference in yield rates is less than 5%, the two-way interactions between (1) returned product and new, (2) usable and disposable, (3) usable and finished part inventory capacities are the significant ones.

When the capacities of the returned product and new part inventories are set at their low settings, the total cost is less than $7000 as displayed in Figure 6.9. Increasing both inventory capacities increases the total cost. The total cost can be kept low while keeping the returned product inventory capacity at a value less than approximately 140 storage spaces and the new part inventory capacity at a value less than approximately 8 storage spaces.
When the interaction between usable and disposable part inventory capacities is examined in Figure 6.10, the total cost is less than $7100 when the usable part inventory capacity is set at its highest setting while the disposable part inventory capacity is at its lowest setting. Decreasing the usable part inventory capacity causes a higher increase on the total cost than increasing the disposable part inventory capacity.

As seen in Figure 6.11, the total cost is less than $6600 when both the usable and finished part inventory capacities are set at their highest settings. Decreasing the usable part inventory capacity below approximately 95 storage spaces increases the total cost while the finished part inventory capacity is kept at its lowest setting.
In Figure 6.12-6.15, the significant 2-way interactions are examined when the yield rate differences vary between 5% and 20%. The (1) returned product and disposable, (2) usable and new, (3) usable and finished, (4) disposable and new part inventory capacities are identified as the significant 2-way interactions.

As seen in Figure 6.12, the total cost is less than $7100 when the returned product inventory capacity is set at its lowest setting, while the capacity for the disposable part inventory is at its highest setting. A decrease in the disposable part inventory increases the total cost, whereas the total cost decreases if the returned product inventory capacity is decreased.
The interaction between the usable and new part inventory capacities is shown in Figure 6.13. The total cost is lower than $6800 when both the usable and new part inventory capacities are set at their lowest settings. Increasing the new part inventory capacity causes a higher increase on the total cost than increasing the disposable part inventory capacity.

The total cost is less $6500 when both the usable and finished part inventory capacities are set at their lowest settings. The impact of increasing the capacity of the usable part inventory is less on the total cost than an increase in the finished part inventory capacity as shown in Figure 6.14.
Figure 6.15 Response surface and contour plots for total cost vs. disposable and new inventories

The last interaction effect in the case of moderate differences is between the disposable and new part inventory capacities as illustrated in Figure 6.15. The total cost is lower than $7000 when the capacity of the disposable part inventory is set at its highest setting while the new part inventory at its lowest setting. The effect of an increase in the new part inventory capacity is higher on the total cost than a decrease in the capacity of disposable part inventory.

Figure 6.16 Response surface and contour plots for total cost vs. new and finished inventories

When the differences in yield rates are higher than 20%, the only two-way interaction is between the new and finished part inventory capacities. Setting the new part inventory at its highest setting and the finished part inventory at its lowest setting obtains a total cost less than $7000 as
shown in Figure 6.16. An increase in the finished part inventory capacity causes a higher increase on the total cost than increasing the capacity of new part inventory. This also explains the high coefficient value for the finished part inventory capacity.

In the above discussion, response surface and contour plots are used to further explore the two-way interactions. In general, one can observe that the main effects are more significant than the interaction effects in all three cases. In other words, the coefficients of main effects are considerably higher than the coefficients of interaction effects. Next, the regression analysis is performed to identify how much of a change in total cost is observed by one storage space increase or decrease in inventory capacities.

6.3.3 Regression Analysis

General insights on the regression analysis are that the impact of new part inventory capacity is high when the yield rate differences are low and moderate. This is observed by the coefficients of 23.5 and 21.6 in the low and moderate difference cases, respectively as shown in equations 1 and 2. The reason the new part inventory has a high impact is that even though there is almost no demand for new parts, the opportunity cost of unused space is $6 which in turn impacts the total cost if the capacity of the new part inventory were to be increased. When the differences in yield rates are higher than 20%, the finished part inventory capacity has the highest impact on the total cost. A coefficient of 22.4 for the term is observed as shown in equation 3. Due to low part yields, the demand for new parts increases significantly. The new parts are stored by reconfiguring space from the finished part inventory if space is not available in the new part inventory. Another important insight on the regression analysis is that the disposable part
inventory capacity is negatively correlated with the total cost. Increasing the capacity of the disposable part inventory by one storage space decreases the total cost by $4.75 and $15.9 in the moderate and high difference cases, respectively. A similar trend is observed with the usable part inventory capacity when the difference in yield rates is less than 5%. The total cost decreases by $4.49 when the capacity of the usable part inventory is increased by one storage space. Table 6.29-6.34 displays the results from the regression analysis, including the coefficients, T and P values.

*Yield rate differences less than 5%*

\[
TC = 5721 + 3.23 \text{ Return} - 4.49 \text{ Usable} + 23.5 \text{ New} + 9.14 \text{ Finished} \quad [1]
\]

*Yield rate differences between 5% and 20%*

\[
TC = 4286 + 3.59 \text{ Return} + 4.40 \text{ Usable} - 4.75 \text{ Disposable} + 21.6 \text{ New} + 11.7 \text{ Finished} \quad [2]
\]

*Yield rate differences higher than 20%*

\[
TC = 4779 + 2.78 \text{ Return} + 3.59 \text{ Usable} - 15.9 \text{ Disposable} + 22.4 \text{ Finished} \quad [3]
\]

Table 6.29 Regression analysis - Low difference in yield rates

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>5720.92</td>
<td>67.16</td>
<td>85.18</td>
<td>0.000</td>
</tr>
<tr>
<td>Return</td>
<td>3.2290</td>
<td>0.3502</td>
<td>9.22</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable</td>
<td>-4.4875</td>
<td>0.2335</td>
<td>-19.22</td>
<td>0.000</td>
</tr>
<tr>
<td>New</td>
<td>23.4978</td>
<td>0.8081</td>
<td>29.08</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished</td>
<td>9.1384</td>
<td>0.1751</td>
<td>52.19</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 6.30 Analysis of variance for total cost - Low difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>4</td>
<td>28423148</td>
<td>7105787</td>
<td>1006.03</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>59</td>
<td>416728</td>
<td>7063</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>28839877</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[S = 84.0428\] \[\text{R-Sq} = 98.6\%\] \[\text{R-Sq(adj)} = 98.5\%\]
Table 6.31 Regression analysis - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>4286.00</td>
<td>49.34</td>
<td>86.87</td>
<td>0.000</td>
</tr>
<tr>
<td>Return</td>
<td>3.5857</td>
<td>0.2504</td>
<td>14.32</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable</td>
<td>4.3981</td>
<td>0.1670</td>
<td>26.34</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable</td>
<td>-4.7531</td>
<td>0.3757</td>
<td>-12.65</td>
<td>0.000</td>
</tr>
<tr>
<td>New</td>
<td>21.6293</td>
<td>0.5779</td>
<td>37.43</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished</td>
<td>11.7255</td>
<td>0.1252</td>
<td>93.64</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.32 Analysis of variance for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>5</td>
<td>40563091</td>
<td>8112618</td>
<td>2245.64</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>58</td>
<td>209531</td>
<td>3613</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>4077262</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 60.1049 R-Sq = 99.5% R-Sq(adj) = 99.4%

Table 6.33 Regression analysis - High difference in yield rates

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>4778.7</td>
<td>105.0</td>
<td>45.51</td>
<td>0.000</td>
</tr>
<tr>
<td>Return</td>
<td>2.7786</td>
<td>0.5439</td>
<td>5.11</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable</td>
<td>3.5936</td>
<td>0.3626</td>
<td>9.91</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable</td>
<td>-15.9098</td>
<td>0.8159</td>
<td>-19.50</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished</td>
<td>22.3611</td>
<td>0.2720</td>
<td>82.22</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 6.34 Analysis of variance for total cost - High difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>4</td>
<td>123802755</td>
<td>30950689</td>
<td>1816.33</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>59</td>
<td>1005376</td>
<td>17040</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>63</td>
<td>124808131</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 130.538 R-Sq = 99.2% R-Sq(adj) = 99.1%

6.4 Conclusions

In this chapter, two analyses were performed using the heuristic approach developed in Chapter 5. The first analysis examines how much the reconfiguration saves the company under different
yield rate and cost scenarios. The results show that savings from reconfiguration can be as high as 33.4% when the difference in yield rates between battery and PCB is higher than 20%, and keep increasing as the ratio of unit external to reconfiguration cost increases. For yield rate differences less than 20% (low and moderate cases), the savings will still up to 6%. Using experimental design analysis, it is examined which of the inventory capacities in the system have the highest impact on total cost. The results indicate that the finished part inventory capacity has a high impact in all three yield rate scenarios. For yield rate differences higher than 20%, its effect is highest due to reconfiguring space for extra new parts in the finished part inventory. The regression analysis finally determines how the total inventory storage cost increases or decreases by a change in inventory capacities. For yield rate differences higher than 20%, it is shown that due to the increased number of parts disposed of (low part yields), increasing the disposable part inventory by one storage space decreases the total cost by $15.9. It is also determined that one storage space increase in finished part inventory capacity affects the total cost by an increase of $22.4.
Chapter 7 – Conclusions and Future Work

This dissertation has examined how the total inventory storage cost is affected by an effective coordination of inventory space planning and reconfiguration decisions in remanufacturing systems. It addresses the issues related to the fundamental stochastic nature of the remanufacturing environment, and how identifying an optimal storage space design is affected by the stochastic conditions, such as unpredictable component recapture yields, product condition, availability of refurbished components, and remanufactured product demand.

In Chapter 1, a general understanding of remanufacturing was first provided. The comparison between the traditional manufacturing and remanufacturing facilities were highlighted, including the new operations that must be considered with remanufacturing that go beyond those involved in first time manufacturing operation. Remanufacturing additionally involves the disassembly, inspection, functionality testing and cleaning operations that each requires storage space capacity as inventory flows through the process. Finally, the issues regarding the remanufacturing facility layout and its requirements were discussed.

The extensive literature review in Chapter 2 discussed the work done on inventory control, disassembly operations, material handling systems and both the manufacturing and
remanufacturing facility layouts. Additionally, the research done on stochastic programming with recourse was provided with a focus of its applications on production and capacity planning.

The simulation model designed in Chapter 3 provided insight into how the inventory builds of each part and the number of remanufactured products is affected by the disassembly and reliability yield rates, and how these rates affect appropriate storage policy planning. Most parts have probabilistic yields in the sense that not all returned products are suitable for re-use. The simulation results illustrate that significant variability in inventory can occur in remanufacturing facilities. The inventory build of parts occurs when the demand for one is less than the other in any given period; therefore, unused parts are stored until there is demand for them in some following period. This process can cause a sudden build or depletion of inventory for parts over a few periods. A reconfigurable storage system design should be able to adjust its space availability to accommodate these fluctuations beyond the objectives of dynamic or flexible facility layout problems. A discussion of reconfigurable and flexible designs including a summary of research on reconfigurable systems is presented.

In Chapter 4, a stochastic programming recourse model was developed to identify the optimal schedules of internal, external, and reconfigured amounts of storage space for multiple components over multiple time periods. The decision being made to minimize total expected costs includes both the amount of each type of storage space to allocate in the current time period and a schedule of subsequent recourse actions in each latter period given possible random set of outcomes. Results are then compared to expected value models, and computational issues are discussed. The expected value formulation for this problem results in a much higher minimum
cost solution, underscoring the potential value in this type of modeling approach. In most cases, the solution with the expected value formulation is 26.7% more expensive. Further extension of the SPR model to multi products with interdependent parts in a multi-period setting causes a dramatic increase in the model size due to the explosion of scenarios and the number of variables and constraints. Even a two-product with overlapping parts SPR model results in 43,753 variables and 56,531 constraints (for 6 periods). Another complexity with these types of formulations is the dramatic increase in run times with higher number of periods. The run times for the single and two-component models increases from 4.7 to 10,779 minutes and from 41.5 to 31,130.4 minutes, respectively when the model size increases from 5 to 6 periods. In a two-product with overlapping parts case, increasing the number of periods from 4 to 5 periods results in a dramatic increase in run time from 6.20 to 16,007.4 minutes. In order to overcome this complexity, using heuristic methodologies, such as tabu search, genetic algorithms or simulated annealing, is recommended for settings higher than 5 periods in the single and two-component models, and 4 periods in the two-product with overlapping parts model.

In Chapter 5, a Monte Carlo simulation-optimization model was developed to identify the minimum cost policies for storage capacity allocation decisions. The simulation program emulates the general logic and stochastic events as material flows through a generalized remanufacturing facility, including random return patterns, component yields, and refurbished demand over multiple time periods, with spare capacity being reconfigured for other needs at a specified cost following a set of reconfiguration logic rules. The program runs for a user-specified number of time periods and replications, tracking the overall costs of total, internal, reconfigured, external and unused space for any given set of inputs (storage capacities, costs). A
heuristic approach based on multi-dimensional golden section search algorithm was implemented to identify optimal storage capacities and reconfiguration decisions in each period that minimize long-term expected total cost. With the heuristic approach, the computation duration is successfully reduced by 97% from 49.2 hours to 83.9 minutes for higher number of inventory capacities. In several cases, results with this approach tend to be only 0.665% higher, which is sufficient in most applications.

Later, in chapter 6, two analyses were performed using the heuristic approach developed in Chapter 5. The first analysis examines how much the reconfiguration of storage space saves a company under different yield rate and cost scenarios. The results show that reconfiguration becomes very important and can save a company substantial sums when the difference in yield rates between parts (battery and PCB) is higher than 20%. In examples, the savings are shown to be as high as 33.4% and keep increasing as the ratio of unit external to reconfiguration cost increases. An experimental design analysis is performed with the goal of illustrating the impact of each inventory capacity on total cost. Results from the analysis show that the finished part inventory capacity impacts the total cost in all three yield rate scenarios. Its effect is highest for when the yield rate difference is higher than 20% due to reconfiguring space for extra new parts in the finished part inventory. The regression analysis finally determines how the total inventory storage cost increases or decreases by a change in inventory capacities. When the difference in yield rates is higher than 20%, it is shown that due to the increased number of parts disposed of (low part yields), increasing the disposable part inventory by one storage space decreases the total cost by $15.9. One storage space increase in finished part inventory capacity, on the other hand, affects the total cost by an increase of $22.4.
One possible extension of this research is testing reconfiguration on systems other than remanufacturing, such as hospital and emergency room settings. Another possible future research project is to examine the interdependencies among different products and their effects on storage space and cost. The simulation program could be expanded to include more than one product type. Apart from these, the future work on the stochastic programming portion of this research could be solving the models by using heuristic methodologies, such as tabu search, genetic algorithms or simulated annealing.
References


Chains." Omega 34(6): 519-532.


References


References


Appendix A – The Lingo® Code of the Stochastic Programming Formulations

MODEL:
!Single-component and two-period stochastic programming formulation;

SETS:
SCENARIO/1 .. 3/:PROB,RATE,REC,OUT,LOSS,JUNK,EXTRA,PEN,T1,T2,TB1,TB2;
ENDSETS

MIN=Cin*S1+@SUM(SCENARIO(K):PROB(K)*(Crec*REC(K)+Cout*OUT(K)+Cjunk*JUNK(K)+
Closs*LOSS(K)+Cpenalty*PEN(K))); !Objective Function;

!Demand met or not;
@FOR(SCENARIO(K):T1(K) + T2(K) = 1);
@FOR(SCENARIO(K):TB1(K) + TB2(K) = RETURN);
@FOR(SCENARIO(K):@ABS(TB1(K) - TB2(K)) = RETURN);

!Extra components and unmet demand;
@FOR(SCENARIO(K):EXTRA(K) = TB2(K)*RATE(K)-DEMAND*T2(K));
@FOR(SCENARIO(K):PEN(K) = -TB1(K)*RATE(K)+DEMAND*T1(K));

!Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR(SCENARIO(K):S1+REC(K)+OUT(K)+JUNK(K) >= EXTRA(K));

!Unused space;
@FOR(SCENARIO(K):LOSS(K) = S1+REC(K)+OUT(K)+JUNK(K)-EXTRA(K));
@FOR(SCENARIO(K):EXTRA(K)-S1-REC(K)+LOSS(K) >= 0);

!Capacity constraints;
@FOR(SCENARIO(K):S1+REC(K) <= CAP1);
@FOR(SCENARIO(K):OUT(K) <= CAP2);

!Binary variables;
@FOR(SCENARIO(K):@BIN(T1)));
@FOR(SCENARIO(K):@BIN(T2));

DATA:
PROB = 0.1
         0.6
         0.3;

RATE = 0.6
         0.9
         0.75;

DEMAND = 40;
Cin = 2; Crec = 6; Cout = 10; Closs = 2; Cjunk = 16; Cpenalty = 20;
MODEL:
!Two-component and two-period stochastic programming formulation;
SETS:
SCENARIO/1 .. 3/:PROB;
COMP/1,2/:Cin,S1,Crec,Cjunk,Closs,Cpenalty,CAP1,CAP2,DEMAND;
LINKS(SCENARIO,COMP):RATE,REC,OUT,LOSS,JUNK,EXTRA,PEN,T1,T2,TB1,TB2;
ENDSETS
MIN=@SUM(COMP(M):Cin(M)*S1(M))+@SUM(LINKS(K,M):PROB(K)*(Crec(M)*REC(K,M)+Cout(M)*OUT(K,M)+Cjunk(M)*JUNK(K,M)+Closs(M)*LOSS(K,M)+Cpenalty(M)*PEN(K,M)));
!Objective Function;
!Demand met or not;
@FOR(LINKS(K,M):T1(K,M) + T2(K,M) = 1);
@FOR(LINKS(K,M):TB1(K,M) + TB2(K,M) = RETURN);
@FOR(LINKS(K,M):@ABS(TB1(K,M) - TB2(K,M)) = RETURN);
!Extra components and unmet demand;
@FOR(LINKS(K,M):EXTRA(K,M) = TB2(K,M)*RATE(K,M)-DEMAND(M)*T2(K,M)));
@FOR(LINKS(K,M):PEN(K,M) = -TB1(K,M)*RATE(K,M)+DEMAND(M)*T1(K,M));
!Ensuring all extras are stored internally, externally, by reconfiguring or they are disposed of;
@FOR(LINKS(K,M):S1(M)+REC(K,M)+OUT(K,M)+JUNK(K,M) >= EXTRA(K,M));
!Unused space;
@FOR(LINKS(K,M):LOSS(K,M) = S1(M)+REC(K,M)+OUT(K,M)+JUNK(K,M)-EXTRA(K,M));
@FOR(LINKS(K,M):EXTRA(K,M)-S1(M)-REC(K,M)+LOSS(K,M) >= 0);
!Capacity constraints;
@FOR(LINKS(K,M):S1(M)+REC(K,M) <= CAP1(M));
@FOR(LINKS(K,M):OUT(K,M) <= CAP2(M));
!Binary variables;
@FOR(LINKS(K,M):@BIN(T1));
@FOR(LINKS(K,M):@BIN(T2));
DATA:
PROB = 0.1
  0.6
  0.3;
RATE = 0.6 0.4
  0.9 0.85
  0.75 0.65;
DEMAND = 40 40;
Cin = 3 2;
MODEL:
!Two-product with overlapping parts and two-period stochastic programming formulation;

SETS:
SCENARIO/1 .. 3/:PROB;
COMP/1,2,3/:Cin,S1,Crec,Cout,Cjunk,Closs,Cpenalty,CAP1,CAP2,DEMAND;
LINKS(SCENARIO,COMP):RATE_A,RATE_B,REC,OUT,LOSS,JUNK,EXTRA,PEN,
T1,T2,TB1_A,TB1_B,TB2_A,TB2_B;
ENDSETS

MIN=@SUM(COMP(M):Cin(M)*S1(M))+@SUM(LINKS(K,M):PROB(K)*
(Crec(M)*REC(K,M)+Cout(M)*OUT(K,M)+Cjunk(M)*JUNK(K,M)+Closs(M)*LOSS(K,M)+
Cpenalty(M)*PEN(K,M)))); !Objective Function;

!Demand met or not;
@FOR(LINKS(K,M):T1(K,M) + T2(K,M) = 1);
@FOR(LINKS(K,M):TB1_A(K,M) + TB2_A(K,M) = RET_A);
@FOR(LINKS(K,M):@ABS(TB1_A(K,M) - TB2_A(K,M)) = RET_A);
@FOR(LINKS(K,M):TB1_B(K,M) + TB2_B(K,M) = RET_B);
@FOR(LINKS(K,M):@ABS(TB1_B(K,M) - TB2_B(K,M)) = RET_B);

!Extra components and unmet demand;
@FOR(LINKS(K,M):EXTRA(K,M) = TB2_A(K,M)*RATE_A(K,M)+TB2_B(K,M)*RATE_B(K,M) -
DEMAND(M)*T2(K,M));
@FOR(LINKS(K,M):PEN(K,M) = -TB1_A(K,M)*RATE_A(K,M)-TB1_B(K,M)*RATE_B(K,M) +
DEMAND(M)*T1(K,M));

!Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR(LINKS(K,M):S1(M)+REC(K,M)+OUT(K,M)+JUNK(K,M) >= EXTRA(K,M));

!Unused space;
@FOR(LINKS(K,M):LOSS(K,M) = S1(M)+REC(K,M)+OUT(K,M)+JUNK(K,M) -EXTRA(K,M));
@FOR(LINKS(K,M):EXTRA(K,M) -S1(M) -REC(K,M) +LOSS(K,M) >= 0);

!Capacity constraints;
@FOR(LINKS(K,M):S1(M)+REC(K,M) <= CAP1(M));
@FOR(LINKS(K,M):OUT(K,M) <= CAP2(M));

!Binary variables;
@FOR(LINKS(K,M):@BIN(T1));
@FOR(LINKS(K,M):@BIN(T2));

DATA:
PROB = 0.1
 0.6
 0.3;
RATE_A = 0.6 0 0.4
 0.9 0 0.85
 0.75 0 0.65;
RATE_B = 0.55 0.35 0
 0.95 0.88 0
 0.70 0.62 0;
DEMAND = 40 40 40;
Cin = 3 2 2; Crec = 6 5 7; Cout = 10 8 11; Closs = 3 2 2; Cjunk = 16 14 17;
Cpenalty = 20 18 22;
CAP1 = 50 50 50; CAP2 = 10 10 10;
ENDDATA

MODEL:
!Single-component and three-period stochastic programming formulation;
SETS:
SCENARIO/1 .. 9/:PROB,RATE2,RATE3,T1,T2,T3,T4,TB1,TB2,TB3,TB4;
PERIOD/2,3/;
LINKS(SCENARIO,PERIOD):REC,OUT,LOSS,PEN,JUNK,EXTRA;
ENDSETS
MIN = Cin*S1 + @SUM(LINKS(K,L):PROB(K)*(Crec*REC(K,L) + Cout*OUT(K,L) +
Closs*LOSS(K,L) + Cjunk*JUNK(K,L)+Cpenalty*PEN(K,L)));
!Demand met or not;
@FOR(SCENARIO(K):T1(K)+T2(K) = 1);
@FOR(SCENARIO(K):T3(K)+T4(K) = 1);
@FOR(SCENARIO(K):TB1(K)+TB2(K) = RET2);
@FOR(SCENARIO(K):TB3(K)+TB4(K) = RET3);
@FOR(SCENARIO(K):@ABS(TB1(K)-TB2(K)) = RET2);
@FOR(SCENARIO(K):@ABS(TB3(K)-TB4(K)) = RET3);
!Extra components and unmet demand;
@FOR(SCENARIO(K):PEN(K,1) = -TB1(K)*RATE2(K)+DEM2*T1(K));
@FOR(SCENARIO(K):PEN(K,2) = -TB3(K)*RATE3(K)-T3(K)*EXTRA(K,1)+T3(K)*DEM3);
@FOR(SCENARIO(K):EXTRA(K,2) = TB4(K)*RATE3(K)+T4(K)*EXTRA(K,1)-T4(K)*DEM3);
!Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR(LINKS(K,L):S1+REC(K,L)+OUT(K,L)+JUNK(K,L) >= EXTRA(K,L)));
!Unused space;
Appendix A

@FOR (LINKS(K,L):LOSS(K,L) = S1+REC(K,L)+OUT(K,L)+JUNK(K,L)-EXTRA(K,L));
@FOR (LINKS(K,L):EXTRA(K,L) - S1 - REC(K,L) + LOSS(K,L) >= 0);

!Capacity constraints;
@FOR (LINKS(K,L):S1 + REC(K,L) <= CAP1);
@FOR (LINKS(K,L):OUT(K,L) <= CAP2);

!Binary variables;
@FOR (SCENARIO(K):@BIN(T1));
@FOR (SCENARIO(K):@BIN(T2));
@FOR (SCENARIO(K):@BIN(T3));
@FOR (SCENARIO(K):@BIN(T4));

!Non-anticipativity constraints;
REC(1,1)=REC(2,1); REC(2,1)=REC(3,1);
REC(4,1)=REC(5,1); REC(5,1)=REC(6,1);
REC(7,1)=REC(8,1); REC(8,1)=REC(9,1);
OUT(1,1)=OUT(2,1); OUT(2,1)=OUT(3,1);
OUT(4,1)=OUT(5,1); OUT(5,1)=OUT(6,1);
OUT(7,1)=OUT(8,1); OUT(8,1)=OUT(9,1);
LOSS(1,1)=LOSS(2,1); LOSS(2,1)=LOSS(3,1);
LOSS(4,1)=LOSS(5,1); LOSS(5,1)=LOSS(6,1);
LOSS(7,1)=LOSS(8,1); LOSS(8,1)=LOSS(9,1);
JUNK(1,1)=JUNK(2,1); JUNK(2,1)=JUNK(3,1);
JUNK(4,1)=JUNK(5,1); JUNK(5,1)=JUNK(6,1);
JUNK(7,1)=JUNK(8,1); JUNK(8,1)=JUNK(9,1);
PEN(1,1)=PEN(2,1); PEN(2,1)=PEN(3,1);
PEN(4,1)=PEN(5,1); PEN(5,1)=PEN(6,1);
PEN(7,1)=PEN(8,1); PEN(8,1)=PEN(9,1);

DATA:
PROB = 0.02 0.05 0.03 0.06 0.24 0.3 0.24 0.03 0.03;
RATE2 = 0.6 0.6 0.6 0.9 0.9 0.9 0.75 0.75 0.75;
RATE3 = 0.75 0.6 0.9 0.9 0.65 0.55 0.95 0.75 0.5;
DEM2 = 40;
DEM3 = 40;
Cin = 2;
Crec = 6;
Cout = 10;
Closs = 2;
Cjunk = 16;
Cpenalty = 20;
CAP1 = 50;
CAP2 = 10;
ENDDATA
END
MODEL:
!Two-component and three-period stochastic programming formulation;

SETS:
SCENARIO/1 .. 9/:PROB;
PERIOD/2,3/;
COMP/1, 2/: Cin, S1, CREC, Cout, Cjunk, Closs, Cpenalty, CAP1, CAP2, DEM2, DEM3;
LINKS(SCENARIO, PERIOD, COMP): REC, OUT, LOSS, PEN, JUNK, EXTRA;
LINKS_2(SCENARIO, COMP): T1, T2, T3, T4, TB1, TB2, TB3, TB4, RATE2, RATE3;
LINKS_3(SCENARIO, PERIOD);
ENDSETS

MIN = @SUM(COMP(M):Cin(M)*S1(M)) +
@SUM(LINKS(K,L,M):PROB(K)*(CREC(M)*REC(K,L,M) + Cout(M)*OUT(K,L,M) +
Closs(M)*LOSS(K,L,M) + Cjunk(M)*JUNK(K,L,M) + Cpenalty(M)*PEN(K,L,M)));

!Demand met or not;
@FOR(LINKS_2(K,M):T1(K,M)+T2(K,M) = 1);
@FOR(LINKS_2(K,M):T3(K,M)+T4(K,M) = 1);
@FOR(LINKS_2(K,M):TB1(K,M)+TB2(K,M) = RET2);
@FOR(LINKS_2(K,M):TB3(K,M)+TB4(K,M) = RET3);

@FOR(LINKS_2(K,M):@ABS(TB1(K,M)-TB2(K,M)) = RET2);
@FOR(LINKS_2(K,M):@ABS(TB3(K,M)-TB4(K,M)) = RET3);

!Extra components and unmet demand;
@FOR(LINKS_2(K,M):PEN(K,1,M) = -TB1(K,M)*RATE2(K,M)+DEM2(M)*T1(K,M));
@FOR(LINKS_2(K,M):EXTRA(K,1,M) = TB2(K,M)*RATE2(K,M)-DEM2(M)*T2(K,M));
@FOR(LINKS_2(K,M):PEN(K,2,M) = -TB3(K,M)*RATE3(K,M)-
T3(K,M)*EXTRA(K,1,M)+T3(K,M)*DEM3(M));
@FOR(LINKS_2(K,M):EXTRA(K,2,M) = TB4(K,M)*RATE3(K,M)+T4(K,M)*EXTRA(K,1,M)-
T4(K,M)*DEM3(M));

!Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR(LINKS(K,L,M):S1(M)+REC(K,L,M)+OUT(K,L,M)+JUNK(K,L,M) >= EXTRA(K,L,M));

!Unused space;
@FOR(LINKS(K,L,M):LOSS(K,L,M) = S1(M)+REC(K,L,M)+OUT(K,L,M)+JUNK(K,L,M) -
EXTRA(K,L,M));
@FOR(LINKS(K,L,M):EXTRA(K,L,M) - S1(M) - REC(K,L,M) + LOSS(K,L,M) >= 0);

!Capacity constraints;
@FOR(LINKS(K,L,M):S1(M)+REC(K,L,M) <= CAP1(M));
@FOR(LINKS(K,L,M):OUT(K,L,M) <= CAP2(M));

!Binary variables;
@FOR(LINKS_2(K,M):@BIN(T1));
@FOR(LINKS_2(K,M):@BIN(T2));
@FOR(LINKS_2(K,M):@BIN(T3));
@FOR(LINKS_2(K,M):@BIN(T4));

!Non-anticipativity constraints;
REC(1,1,1)=REC(2,1,1); REC(2,1,1)=REC(3,1,1);
REC(4,1,1)=REC(5,1,1); REC(5,1,1)=REC(6,1,1);
Appendix A

\begin{verbatim}
REC(7,1,1) = REC(8,1,1); REC(8,1,1) = REC(9,1,1);
OUT(1,1,1) = OUT(2,1,1); OUT(2,1,1) = OUT(3,1,1);
OUT(4,1,1) = OUT(5,1,1); OUT(5,1,1) = OUT(6,1,1);
OUT(7,1,1) = OUT(8,1,1); OUT(8,1,1) = OUT(9,1,1);
LOSS(1,1,1) = LOSS(2,1,1); LOSS(2,1,1) = LOSS(3,1,1);
LOSS(4,1,1) = LOSS(5,1,1); LOSS(5,1,1) = LOSS(6,1,1);
LOSS(7,1,1) = LOSS(8,1,1); LOSS(8,1,1) = LOSS(9,1,1);
JUNK(1,1,1) = JUNK(2,1,1); JUNK(2,1,1) = JUNK(3,1,1);
JUNK(4,1,1) = JUNK(5,1,1); JUNK(5,1,1) = JUNK(6,1,1);
JUNK(7,1,1) = JUNK(8,1,1); JUNK(8,1,1) = JUNK(9,1,1);
PEN(1,1,1) = PEN(2,1,1); PEN(2,1,1) = PEN(3,1,1);
PEN(4,1,1) = PEN(5,1,1); PEN(5,1,1) = PEN(6,1,1);
PEN(7,1,1) = PEN(8,1,1); PEN(8,1,1) = PEN(9,1,1);
REC(1,1,2) = REC(2,1,2); REC(2,1,2) = REC(3,1,2);
REC(4,1,2) = REC(5,1,2); REC(5,1,2) = REC(6,1,2);
REC(7,1,2) = REC(8,1,2); REC(8,1,2) = REC(9,1,2);
OUT(1,1,2) = OUT(2,1,2); OUT(2,1,2) = OUT(3,1,2);
OUT(4,1,2) = OUT(5,1,2); OUT(5,1,2) = OUT(6,1,2);
OUT(7,1,2) = OUT(8,1,2); OUT(8,1,2) = OUT(9,1,2);
LOSS(1,1,2) = LOSS(2,1,2); LOSS(2,1,2) = LOSS(3,1,2);
LOSS(4,1,2) = LOSS(5,1,2); LOSS(5,1,2) = LOSS(6,1,2);
LOSS(7,1,2) = LOSS(8,1,2); LOSS(8,1,2) = LOSS(9,1,2);
JUNK(1,1,2) = JUNK(2,1,2); JUNK(2,1,2) = JUNK(3,1,2);
JUNK(4,1,2) = JUNK(5,1,2); JUNK(5,1,2) = JUNK(6,1,2);
JUNK(7,1,2) = JUNK(8,1,2); JUNK(8,1,2) = JUNK(9,1,2);
PEN(1,1,2) = PEN(2,1,2); PEN(2,1,2) = PEN(3,1,2);
PEN(4,1,2) = PEN(5,1,2); PEN(5,1,2) = PEN(6,1,2);
PEN(7,1,2) = PEN(8,1,2); PEN(8,1,2) = PEN(9,1,2);
\end{verbatim}

\textbf{DATA}:

\begin{verbatim}
PROB = 0.02
    0.05
    0.03
    0.06
    0.24
    0.3
    0.24
    0.03
    0.03;
RATE2 = 0.6  0.4
    0.6  0.4
    0.9  0.85
    0.9  0.85
    0.75  0.65
    0.75  0.65
\end{verbatim}
MODEL:
!Two-product with overlapping parts and three-period stochastic programming
formulation;

SETS:
SCENARIO/1 .. 9/:PROB;
PERIOD/2,3/;
COMP/1 .. 3/:Cin, S1, Crec, Cout, Cjunk, Closs, Cpenalty, CAP1, CAP2, DEM2, DEM3;
LINKS(SCENARIO,PERIOD,COMP):REC,OUT,LOSS,PEN,JUNK,EXTRA;
LINKS_2(SCENARIO,COMP):T1,T2,T3,T4,TB1_A,TB1_B,TB2_A,TB2_B,TB3_A,TB3_B,
TB4_A,TB4_B,RATE2_A,RATE2_B,RATE3_A,RATE3_B;
LINKS_3(SCENARIO,PERIOD);
ENDSETS

MIN = @SUM(COMP(M):Cin(M)*S1(M)) +
@SUM(LINKS(K,L,M):PROB(K)*(Crec(M)*REC(K,L,M) + Cout(M)*OUT(K,L,M) +
Closs(M)*LOSS(K,L,M) + Cjunk(M)*JUNK(K,L,M)+Cpenalty(M)*PEN(K,L,M)));  

!Demand met or not;
@FOR(LINKS_2(K,M):T1(K,M)+T2(K,M) = 1);
@FOR(LINKS_2(K,M):T3(K,M)+T4(K,M) = 1);
@FOR(LINKS_2(K,M):TB1_A(K,M)+TB2_A(K,M) = RET2_A);
@FOR(LINKS_2(K,M):TB3_A(K,M)+TB4_A(K,M) = RET3_A);

@FOR(LINKS_2(K,M):@ABS(TB1_B(K,M) + TB2_B(K,M) = RET2_B);
@FOR(LINKS_2(K,M):@ABS(TB3_B(K,M) + TB4_B(K,M) = RET3_B);

@FOR(LINKS_2(K,M):@ABS(TB1_A(K,M)-TB2_A(K,M) = RET2_A));
@FOR(LINKS_2(K,M):@ABS(TB3_A(K,M)-TB4_A(K,M) = RET3_A);
Appendix A

@FOR (LINKS_2(K,M) : @ABS(TB1_B(K,M) - TB2_B(K,M)) = RET2_B);
@FOR (LINKS_2(K,M) : @ABS(TB3_B(K,M) - TB4_B(K,M)) = RET3_B);

! Extra components and unmet demand;
@FOR (LINKS_2(K,M) : PEN(K,1,M) = -TB1_A(K,M)*RATE2_A(K,M) -
        TB1_B(K,M)*RATE2_B(K,M) + DEM2(M)*T1(K,M));
@FOR (LINKS_2(K,M) : EXTRA(K,1,M) =
        TB2_A(K,M)*RATE2_A(K,M) + TB2_B(K,M)*RATE2_B(K,M) - DEM2(M)*T2(K,M));
@FOR (LINKS_2(K,M) : PEN(K,2,M) = -TB3_A(K,M)*RATE3_A(K,M) -
        TB3_B(K,M)*RATE3_B(K,M) - T3(K,M)*EXTRA(K,1,M) + T3(K,M)*DEM3(M));
@FOR (LINKS_2(K,M) : EXTRA(K,2,M) =
        TB4_A(K,M)*RATE3_A(K,M) + TB4_B(K,M)*RATE3_B(K,M) + T4(K,M)*EXTRA(K,1,M) -
        T4(K,M)*DEM3(M));

! Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR (LINKS(K,L,M) : S1(M) + REC(K,L,M) + OUT(K,L,M) + JUNK(K,L,M) >= EXTRA(K,L,M));

! Unused space;
@FOR (LINKS(K,L,M) : LOSS(K,L,M) = S1(M) + REC(K,L,M) + OUT(K,L,M) + JUNK(K,L,M) -
        EXTRA(K,L,M));
@FOR (LINKS(K,L,M) : EXTRA(K,L,M) - S1(M) - REC(K,L,M) + LOSS(K,L,M) >= 0);

! Capacity constraints;
@FOR (LINKS(K,L,M) : S1(M) + REC(K,L,M) <= CAP1(M));
@FOR (LINKS(K,L,M) : OUT(K,L,M) <= CAP2(M));

! Binary variables;
@FOR (LINKS_2(K,M) : @BIN(T1));
@FOR (LINKS_2(K,M) : @BIN(T2));
@FOR (LINKS_2(K,M) : @BIN(T3));
@FOR (LINKS_2(K,M) : @BIN(T4));

! Non-anticipativity constraints;
REC(1,1,1) = REC(2,1,1); REC(2,1,1) = REC(3,1,1);
REC(4,1,1) = REC(5,1,1); REC(5,1,1) = REC(6,1,1);
REC(7,1,1) = REC(8,1,1); REC(8,1,1) = REC(9,1,1);
OUT(1,1,1) = OUT(2,1,1); OUT(2,1,1) = OUT(3,1,1);
OUT(4,1,1) = OUT(5,1,1); OUT(5,1,1) = OUT(6,1,1);
OUT(7,1,1) = OUT(8,1,1); OUT(8,1,1) = OUT(9,1,1);
LOSS(1,1,1) = LOSS(2,1,1); LOSS(2,1,1) = LOSS(3,1,1);
LOSS(4,1,1) = LOSS(5,1,1); LOSS(5,1,1) = LOSS(6,1,1);
LOSS(7,1,1) = LOSS(8,1,1); LOSS(8,1,1) = LOSS(9,1,1);
JUNK(1,1,1) = JUNK(2,1,1); JUNK(2,1,1) = JUNK(3,1,1);
JUNK(4,1,1) = JUNK(5,1,1); JUNK(5,1,1) = JUNK(6,1,1);
JUNK(7,1,1) = JUNK(8,1,1); JUNK(8,1,1) = JUNK(9,1,1);
PEN(1,1,1) = PEN(2,1,1); PEN(2,1,1) = PEN(3,1,1);
PEN(4,1,1) = PEN(5,1,1); PEN(5,1,1) = PEN(6,1,1);
PEN(7,1,1) = PEN(8,1,1); PEN(8,1,1) = PEN(9,1,1);
REC(1,1,2) = REC(2,1,2); REC(2,1,2) = REC(3,1,2);
Appendix A

\begin{verbatim}
REC(4,1,2) = REC(5,1,2); REC(5,1,2) = REC(6,1,2);
REC(7,1,2) = REC(8,1,2); REC(8,1,2) = REC(9,1,2);
OUT(1,1,2) = OUT(2,1,2); OUT(2,1,2) = OUT(3,1,2);
OUT(4,1,2) = OUT(5,1,2); OUT(5,1,2) = OUT(6,1,2);
OUT(7,1,2) = OUT(8,1,2); OUT(8,1,2) = OUT(9,1,2);
LOSS(1,1,2) = LOSS(2,1,2); LOSS(2,1,2) = LOSS(3,1,2);
LOSS(4,1,2) = LOSS(5,1,2); LOSS(5,1,2) = LOSS(6,1,2);
LOSS(7,1,2) = LOSS(8,1,2); LOSS(8,1,2) = LOSS(9,1,2);
JUNK(1,1,2) = JUNK(2,1,2); JUNK(2,1,2) = JUNK(3,1,2);
JUNK(4,1,2) = JUNK(5,1,2); JUNK(5,1,2) = JUNK(6,1,2);
JUNK(7,1,2) = JUNK(8,1,2); JUNK(8,1,2) = JUNK(9,1,2);
PEN(1,1,2) = PEN(2,1,2); PEN(2,1,2) = PEN(3,1,2);
PEN(4,1,2) = PEN(5,1,2); PEN(5,1,2) = PEN(6,1,2);
PEN(7,1,2) = PEN(8,1,2); PEN(8,1,2) = PEN(9,1,2);
REC(1,1,3) = REC(2,1,3); REC(2,1,3) = REC(3,1,3);
REC(4,1,3) = REC(5,1,3); REC(5,1,3) = REC(6,1,3);
REC(7,1,3) = REC(8,1,3); REC(8,1,3) = REC(9,1,3);
OUT(1,1,3) = OUT(2,1,3); OUT(2,1,3) = OUT(3,1,3);
OUT(4,1,3) = OUT(5,1,3); OUT(5,1,3) = OUT(6,1,3);
OUT(7,1,3) = OUT(8,1,3); OUT(8,1,3) = OUT(9,1,3);
LOSS(1,1,3) = LOSS(2,1,3); LOSS(2,1,3) = LOSS(3,1,3);
LOSS(4,1,3) = LOSS(5,1,3); LOSS(5,1,3) = LOSS(6,1,3);
LOSS(7,1,3) = LOSS(8,1,3); LOSS(8,1,3) = LOSS(9,1,3);
JUNK(1,1,3) = JUNK(2,1,3); JUNK(2,1,3) = JUNK(3,1,3);
JUNK(4,1,3) = JUNK(5,1,3); JUNK(5,1,3) = JUNK(6,1,3);
JUNK(7,1,3) = JUNK(8,1,3); JUNK(8,1,3) = JUNK(9,1,3);
PEN(1,1,3) = PEN(2,1,3); PEN(2,1,3) = PEN(3,1,3);
PEN(4,1,3) = PEN(5,1,3); PEN(5,1,3) = PEN(6,1,3);
PEN(7,1,3) = PEN(8,1,3); PEN(8,1,3) = PEN(9,1,3);

DATA:
PROB = 0.02
  0.05
  0.03
  0.06
  0.24
  0.3
  0.3
  0.03
  0.03;
RATE2_A = 0.6 0.4 0.6 0.4 0.9 0.85 0.9 0.85 0.9 0.85 0.75 0.65
\end{verbatim}
0.75 0 0.65
0.75 0 0.65;
RATE3_A = 0.75 0 0.67
0.6 0 0.46
0.9 0 0.88
0.9 0 0.92
0.65 0 0.75
0.55 0 0.67
0.95 0 0.84
0.75 0 0.71
0.5 0 0.68;

RATE2_B = 0.55 0.35 0
0.55 0.35 0
0.55 0.35 0
0.95 0.88 0
0.95 0.88 0
0.95 0.88 0
0.70 0.62 0
0.70 0.62 0
0.70 0.62 0;

RATE3_B = 0.55 0.62 0
0.68 0.46 0
0.95 0.83 0
0.95 0.98 0
0.54 0.75 0
0.55 0.64 0
0.85 0.84 0
0.45 0.65 0
0.68 0.75 0;

DEM2 = 40 40 40;
DEM3 = 40 40 40;
Cin = 3 2 2;
Crec = 6 5 7;
Cout = 10 8 11;
Closs = 3 2 2;
Cjunk = 16 14 17;
Cpenalty = 20 18 22;
CAP1 = 50 50 50;
CAP2 = 10 10 10;
ENDDATA
END

MODEL:
!Single-component and four-period stochastic programming formulation;

SETS:
SCENARIO/1 ..
27/:PROB,RATE2,RATE3,RATE4,T1,T2,T3,T4,T5,T6,TB1,TB2,TB3,TB4,TB5,TB6;
PERIOD/2,3,4/;;
LINKS(SCENARIO,PERIOD):REC,OUT,LOSS,PEN,JUNK,EXTRA;
ENDSETS

MIN = Cin*S1 + @SUM(LINKS(K,L):PROB(K)*(Crec*REC(K,L) + Cout*OUT(K,L) + Closs*LOSS(K,L) + Cjunk*JUNK(K,L) + Cpenalty*PEN(K,L))) ;

!Demand met or not;
@FOR(SCENARIO(K):T1(K)+T2(K) = 1);
@FOR(SCENARIO(K):T3(K)+T4(K) = 1);
@FOR(SCENARIO(K):T5(K)+T6(K) = 1);
@FOR(SCENARIO(K):TB1(K)+TB2(K) = RET2);
@FOR(SCENARIO(K):TB3(K)+TB4(K) = RET3);
@FOR(SCENARIO(K):TB5(K)+TB6(K) = RET4);

@FOR(SCENARIO(K):@ABS(TB1(K)-TB2(K)) = RET2);
@FOR(SCENARIO(K):@ABS(TB3(K)-TB4(K)) = RET3);
@FOR(SCENARIO(K):@ABS(TB5(K)-TB6(K)) = RET4);

!Extra components and unmet demand;
@FOR(SCENARIO(K):PEN(K,1) = -TB1(K)*RATE2(K)+DEM2*T1(K));
@FOR(SCENARIO(K):EXTRA(K,1) = TB2(K)*RATE2(K)-DEM2*T2(K));
@FOR(SCENARIO(K):PEN(K,2) = -TB3(K)*RATE3(K)-T3(K)*EXTRA(K,1)+DEM3*T3(K));
@FOR(SCENARIO(K):EXTRA(K,2) = TB4(K)*RATE3(K)+T4(K)*EXTRA(K,1)-DEM3*T4(K));
@FOR(SCENARIO(K):PEN(K,3) = -TB5(K)*RATE4(K)-T5(K)*EXTRA(K,2)+DEM4*T5(K));
@FOR(SCENARIO(K):EXTRA(K,3) = TB6(K)*RATE4(K)+T6(K)*EXTRA(K,2)-DEM4*T6(K));

!Ensuring all extras are stored internally, externally, by reconfiguring or they are disposed of;
@FOR(LINKS(K,L):S1+REC(K,L)+OUT(K,L)+JUNK(K,L) >= EXTRA(K,L));

!Unused space;
@FOR(LINKS(K,L):LOSS(K,L) = S1+REC(K,L)+OUT(K,L)+JUNK(K,L) - EXTRA(K,L));
@FOR(LINKS(K,L):EXTRA(K,L)-S1-REC(K,L)+LOSS(K,L) >= 0);

!Capacity constraints;
@FOR(LINKS(K,L):S1+REC(K,L) <= CAP1);
@FOR(LINKS(K,L):OUT(K,L) <= CAP2);

!Binary variables;
@FOR(SCENARIO(K):@BIN(T1));
@FOR(SCENARIO(K):@BIN(T2));
@FOR(SCENARIO(K):@BIN(T3));
@FOR(SCENARIO(K):@BIN(T4));
@FOR(SCENARIO(K):@BIN(T5));
@FOR(SCENARIO(K):@BIN(T6));

!Non-anticipativity constraints;
REC(1,2)=REC(2,2); REC(2,2)=REC(3,2);
REC(4,2)=REC(5,2); REC(5,2)=REC(6,2);
REC(7,2)=REC(8,2); REC(8,2)=REC(9,2);
REC(10,2)=REC(11,2); REC(11,2)=REC(12,2);
REC(13,2)=REC(14,2); REC(14,2)=REC(15,2);
REC(16,2)=REC(17,2); REC(17,2)=REC(18,2);
REC(19,2)=REC(20,2); REC(20,2)=REC(21,2);
REC(22,2)=REC(23,2); REC(23,2)=REC(24,2);
REC(25,2)=REC(26,2); REC(26,2)=REC(27,2);
REC(1,1) = REC(2,1); REC(2,1) = REC(3,1); REC(3,1) = REC(4,1); REC(4,1) = REC(5,1); 
REC(5,1) = REC(6,1); REC(6,1) = REC(7,1); REC(7,1) = REC(8,1); REC(8,1) = REC(9,1); 
REC(10,1) = REC(11,1); REC(11,1) = REC(12,1); REC(12,1) = REC(13,1); 
REC(13,1) = REC(14,1); REC(14,1) = REC(15,1); REC(15,1) = REC(16,1); 
REC(16,1) = REC(17,1); REC(17,1) = REC(18,1); 
REC(19,1) = REC(20,1); REC(20,1) = REC(21,1); REC(21,1) = REC(22,1); 
REC(22,1) = REC(23,1); REC(23,1) = REC(24,1); REC(24,1) = REC(25,1); 
REC(25,1) = REC(26,1); REC(26,1) = REC(27,1); 

OUT(1,2) = OUT(2,2); OUT(2,2) = OUT(3,2); 
OUT(4,2) = OUT(5,2); OUT(5,2) = OUT(6,2); 
OUT(7,2) = OUT(8,2); OUT(8,2) = OUT(9,2); 
OUT(10,2) = OUT(11,2); OUT(11,2) = OUT(12,2); 
OUT(13,2) = OUT(14,2); OUT(14,2) = OUT(15,2); 
OUT(16,2) = OUT(17,2); OUT(17,2) = OUT(18,2); 
OUT(19,2) = OUT(20,2); OUT(20,2) = OUT(21,2); 
OUT(22,2) = OUT(23,2); OUT(23,2) = OUT(24,2); 
OUT(25,2) = OUT(26,2); OUT(26,2) = OUT(27,2); 

OUT(1,1) = OUT(2,1); OUT(2,1) = OUT(3,1); OUT(3,1) = OUT(4,1); OUT(4,1) = OUT(5,1); 
OUT(5,1) = OUT(6,1); OUT(6,1) = OUT(7,1); OUT(7,1) = OUT(8,1); OUT(8,1) = OUT(9,1); 
OUT(10,1) = OUT(11,1); OUT(11,1) = OUT(12,1); OUT(12,1) = OUT(13,1); 
OUT(13,1) = OUT(14,1); OUT(14,1) = OUT(15,1); OUT(15,1) = OUT(16,1); 
OUT(16,1) = OUT(17,1); OUT(17,1) = OUT(18,1); 
OUT(19,1) = OUT(20,1); OUT(20,1) = OUT(21,1); OUT(21,1) = OUT(22,1); 
OUT(22,1) = OUT(23,1); OUT(23,1) = OUT(24,1); OUT(24,1) = OUT(25,1); 
OUT(25,1) = OUT(26,1); OUT(26,1) = OUT(27,1); 

LOSS(1,2) = LOSS(2,2); LOSS(2,2) = LOSS(3,2); 
LOSS(4,2) = LOSS(5,2); LOSS(5,2) = LOSS(6,2); 
LOSS(7,2) = LOSS(8,2); LOSS(8,2) = LOSS(9,2); 
LOSS(10,2) = LOSS(11,2); LOSS(11,2) = LOSS(12,2); 
LOSS(13,2) = LOSS(14,2); LOSS(14,2) = LOSS(15,2); 
LOSS(16,2) = LOSS(17,2); LOSS(17,2) = LOSS(18,2); 
LOSS(19,2) = LOSS(20,2); LOSS(20,2) = LOSS(21,2); 
LOSS(22,2) = LOSS(23,2); LOSS(23,2) = LOSS(24,2); 
LOSS(25,2) = LOSS(26,2); LOSS(26,2) = LOSS(27,2); 

LOSS(1,1) = LOSS(2,1); LOSS(2,1) = LOSS(3,1); LOSS(3,1) = LOSS(4,1); 
LOSS(4,1) = LOSS(5,1); LOSS(5,1) = LOSS(6,1); LOSS(6,1) = LOSS(7,1); 
LOSS(7,1) = LOSS(8,1); LOSS(8,1) = LOSS(9,1); 
LOSS(10,1) = LOSS(11,1); LOSS(11,1) = LOSS(12,1); LOSS(12,1) = LOSS(13,1); 
LOSS(13,1) = LOSS(14,1); LOSS(14,1) = LOSS(15,1); LOSS(15,1) = LOSS(16,1); 
LOSS(16,1) = LOSS(17,1); LOSS(17,1) = LOSS(18,1); 
LOSS(19,1) = LOSS(20,1); LOSS(20,1) = LOSS(21,1); LOSS(21,1) = LOSS(22,1); 
LOSS(22,1) = LOSS(23,1); LOSS(23,1) = LOSS(24,1); LOSS(24,1) = LOSS(25,1); 
LOSS(25,1) = LOSS(26,1); LOSS(26,1) = LOSS(27,1); 

JUNK(1,2) = JUNK(2,2); JUNK(2,2) = JUNK(3,2); 
JUNK(4,2) = JUNK(5,2); JUNK(5,2) = JUNK(6,2); 
JUNK(7,2) = JUNK(8,2); JUNK(8,2) = JUNK(9,2); 
JUNK(10,2) = JUNK(11,2); JUNK(11,2) = JUNK(12,2); 
JUNK(13,2) = JUNK(14,2); JUNK(14,2) = JUNK(15,2); 
JUNK(16,2) = JUNK(17,2); JUNK(17,2) = JUNK(18,2); 
JUNK(19,2) = JUNK(20,2); JUNK(20,2) = JUNK(21,2); 
JUNK(22,2) = JUNK(23,2); JUNK(23,2) = JUNK(24,2);
Appendix A

JUNK(25,2) = JUNK(26,2); JUNK(26,2) = JUNK(27,2);

JUNK(1,1) = JUNK(2,1); JUNK(2,1) = JUNK(3,1); JUNK(3,1) = JUNK(4,1);
JUNK(4,1) = JUNK(5,1); JUNK(5,1) = JUNK(6,1); JUNK(6,1) = JUNK(7,1);
JUNK(7,1) = JUNK(8,1); JUNK(8,1) = JUNK(9,1);
JUNK(10,1) = JUNK(11,1); JUNK(11,1) = JUNK(12,1); JUNK(12,1) = JUNK(13,1);
JUNK(13,1) = JUNK(14,1); JUNK(14,1) = JUNK(15,1); JUNK(15,1) = JUNK(16,1);
JUNK(16,1) = JUNK(17,1); JUNK(17,1) = JUNK(18,1);
JUNK(19,1) = JUNK(20,1); JUNK(20,1) = JUNK(21,1); JUNK(21,1) = JUNK(22,1);
JUNK(22,1) = JUNK(23,1); JUNK(23,1) = JUNK(24,1); JUNK(24,1) = JUNK(25,1);
JUNK(25,1) = JUNK(26,1); JUNK(26,1) = JUNK(27,1);

PEN(1,2) = PEN(2,2); PEN(2,2) = PEN(3,2);
PEN(4,2) = PEN(5,2); PEN(5,2) = PEN(6,2);
PEN(7,2) = PEN(8,2); PEN(8,2) = PEN(9,2);
PEN(10,2) = PEN(11,2); PEN(11,2) = PEN(12,2);
PEN(13,2) = PEN(14,2); PEN(14,2) = PEN(15,2);
PEN(16,2) = PEN(17,2); PEN(17,2) = PEN(18,2);
PEN(19,2) = PEN(20,2); PEN(20,2) = PEN(21,2);
PEN(22,2) = PEN(23,2); PEN(23,2) = PEN(24,2);
PEN(25,2) = PEN(26,2); PEN(26,2) = PEN(27,2);

PEN(1,1) = PEN(2,1); PEN(2,1) = PEN(3,1); PEN(3,1) = PEN(4,1); PEN(4,1) = PEN(5,1);
PEN(5,1) = PEN(6,1); PEN(6,1) = PEN(7,1); PEN(7,1) = PEN(8,1); PEN(8,1) = PEN(9,1);
PEN(10,1) = PEN(11,1); PEN(11,1) = PEN(12,1); PEN(12,1) = PEN(13,1);
PEN(13,1) = PEN(14,1); PEN(14,1) = PEN(15,1); PEN(15,1) = PEN(16,1);
PEN(16,1) = PEN(17,1); PEN(17,1) = PEN(18,1);
PEN(19,1) = PEN(20,1); PEN(20,1) = PEN(21,1); PEN(21,1) = PEN(22,1);
PEN(22,1) = PEN(23,1); PEN(23,1) = PEN(24,1); PEN(24,1) = PEN(25,1);
PEN(25,1) = PEN(26,1); PEN(26,1) = PEN(27,1);

DATA:
PROB = @OLE('F:\PROB.xls');
RATE2 = @OLE('F:\RATE.xls');
RATE3 = @OLE('F:\RATE.xls');
RATE4 = @OLE('F:\RATE.xls');
DEM2 = 40;
DEM3 = 40;
DEM4 = 40;
Cin = 2;
Crec = 6;
Cout = 10;
Closs = 2;
Cjunk = 16;
Cpenalty = 20;
CAP1 = 50;
CAP2 = 10;
ENDDATA

END
MODEL:
!Two-component and four-period stochastic programming formulation;

SETS:
SCENARIO/1 .. 27/:PROB;
PERIOD/2,3,4/:;
COMP/1,2/: Cin,S1,Crec,Cout,Cjunk,Closs,Cpenalty,CAP1,CAP2,DEM2,DEM3,DEM4;
LINKS(SCENARIO,PERIOD,COMP):REC,OUT,LOSS,PEN,JUNK,EXTRA;
LINKS_2(SCENARIO,COMP):T1,T2,T3,T4,T5,T6,TB1,TB2,TB3,TB4,TB5,TB6,
RATE2,RATE3,RATE4;
LINKS_3(SCENARIO,PERIOD);
ENDSETS

MIN = @SUM(COMP(M):Cin(M)*S1(M)) +
@SUM(LINKS(K,L,M):PROB(K)*(Crec(M)*REC(K,L,M) + Cout(M)*OUT(K,L,M) +
Closs(M)*LOSS(K,L,M) + Cjunk(M)*JUNK(K,L,M)+Cpenalty(M)*PEN(K,L,M)))

!Demand met or not;
@FOR(LINKS_2(K,M):T1(K,M)+T2(K,M) = 1);
@FOR(LINKS_2(K,M):T3(K,M)+T4(K,M) = 1);
@FOR(LINKS_2(K,M):T5(K,M)+T6(K,M) = 1);
@FOR(LINKS_2(K,M):TB1(K,M)+TB2(K,M) = RET2);
@FOR(LINKS_2(K,M):TB3(K,M)+TB4(K,M) = RET3);
@FOR(LINKS_2(K,M):TB5(K,M)+TB6(K,M) = RET4);

@FOR(LINKS_2(K,M):@ABS(TB1(K,M)-TB2(K,M)) = RET2);
@FOR(LINKS_2(K,M):@ABS(TB3(K,M)-TB4(K,M)) = RET3);
@FOR(LINKS_2(K,M):@ABS(TB5(K,M)-TB6(K,M)) = RET4);

!Extra components and unmet demand;
@FOR(LINKS_2(K,M):PEN(K,1,M) = -TB1(K,M)*RATE2(K,M)+DEM2(M)*T1(K,M));
@FOR(LINKS_2(K,M):EXTRA(K,1,M) = TB2(K,M)*RATE2(K,M)-DEM2(M)*T2(K,M));
@FOR(LINKS_2(K,M):PEN(K,2,M) = -TB3(K,M)*RATE3(K,M)-
T3(K,M)*EXTRA(K,1,M)+T3(K,M)*DEM3(M));
@FOR(LINKS_2(K,M):EXTRA(K,2,M) = TB4(K,M)*RATE3(K,M)+T4(K,M)*EXTRA(K,1,M)-
T4(K,M)*DEM3(M));
@FOR(LINKS_2(K,M):PEN(K,3,M) = -TB5(K,M)*RATE4(K,M)-
T5(K,M)*EXTRA(K,2,M)+T5(K,M)*DEM4(M));
@FOR(LINKS_2(K,M):EXTRA(K,3,M) = TB6(K,M)*RATE4(K,M)+T6(K,M)*EXTRA(K,2,M)-
T6(K,M)*DEM4(M));

!Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
@FOR(LINKS(K,L,M):S1(M)+REC(K,L,M)+OUT(K,L,M) + JUNK(K,L,M) >= EXTRA(K,L,M));

!Unused space;
@FOR(LINKS(K,L,M):LOSS(K,L,M) = S1(M)+REC(K,L,M)+OUT(K,L,M) + JUNK(K,L,M) -
EXTRA(K,L,M));
@FOR(LINKS(K,L,M):EXTRA(K,L,M)-S1(M)-REC(K,L,M)-LOSS(K,L,M) >= 0);

!Capacity constraints;
@FOR(LINKS(K,L,M):S1(M)+REC(K,L,M) <= CAP1(M));
@FOR(LINKS(K,L,M):OUT(K,L,M) <= CAP2(M));

!Binary variables;
Appendix A

@FOR (LINKS_2 (K, M) : @BIN (T1));
@FOR (LINKS_2 (K, M) : @BIN (T2));
@FOR (LINKS_2 (K, M) : @BIN (T3));
@FOR (LINKS_2 (K, M) : @BIN (T4));
@FOR (LINKS_2 (K, M) : @BIN (T5));
@FOR (LINKS_2 (K, M) : @BIN (T6));

!Non-anticipativity constraints;
REC (1, 2, 1) = REC (2, 2, 1); REC (2, 2, 1) = REC (3, 2, 1);
REC (4, 2, 1) = REC (5, 2, 1); REC (5, 2, 1) = REC (6, 2, 1);
REC (7, 2, 1) = REC (8, 2, 1); REC (8, 2, 1) = REC (9, 2, 1);
REC (10, 2, 1) = REC (11, 2, 1); REC (11, 2, 1) = REC (12, 2, 1);
REC (13, 2, 1) = REC (14, 2, 1); REC (14, 2, 1) = REC (15, 2, 1);
REC (16, 2, 1) = REC (17, 2, 1); REC (17, 2, 1) = REC (18, 2, 1);
REC (19, 2, 1) = REC (20, 2, 1); REC (20, 2, 1) = REC (21, 2, 1);
REC (22, 2, 1) = REC (23, 2, 1); REC (23, 2, 1) = REC (24, 2, 1);
REC (25, 2, 1) = REC (26, 2, 1); REC (26, 2, 1) = REC (27, 2, 1);
REC (1, 1, 1) = REC (2, 1, 1); REC (2, 1, 1) = REC (3, 1, 1); REC (3, 1, 1) = REC (4, 1, 1);
REC (4, 1, 1) = REC (5, 1, 1); REC (5, 1, 1) = REC (6, 1, 1); REC (6, 1, 1) = REC (7, 1, 1);
REC (7, 1, 1) = REC (8, 1, 1); REC (8, 1, 1) = REC (9, 1, 1);
REC (10, 1, 1) = REC (11, 1, 1); REC (11, 1, 1) = REC (12, 1, 1); REC (12, 1, 1) = REC (13, 1, 1);
REC (13, 1, 1) = REC (14, 1, 1); REC (14, 1, 1) = REC (15, 1, 1); REC (15, 1, 1) = REC (16, 1, 1);
REC (16, 1, 1) = REC (17, 1, 1); REC (17, 1, 1) = REC (18, 1, 1);
REC (19, 1, 1) = REC (20, 1, 1); REC (20, 1, 1) = REC (21, 1, 1); REC (21, 1, 1) = REC (22, 1, 1);
REC (22, 1, 1) = REC (23, 1, 1); REC (23, 1, 1) = REC (24, 1, 1); REC (24, 1, 1) = REC (25, 1, 1);
REC (25, 1, 1) = REC (26, 1, 1); REC (26, 1, 1) = REC (27, 1, 1);
REC (1, 2, 2) = REC (2, 2, 2); REC (2, 2, 2) = REC (3, 2, 2);
REC (4, 2, 2) = REC (5, 2, 2); REC (5, 2, 2) = REC (6, 2, 2);
REC (7, 2, 2) = REC (8, 2, 2); REC (8, 2, 2) = REC (9, 2, 2);
REC (10, 2, 2) = REC (11, 2, 2); REC (11, 2, 2) = REC (12, 2, 2);
REC (13, 2, 2) = REC (14, 2, 2); REC (14, 2, 2) = REC (15, 2, 2);
REC (16, 2, 2) = REC (17, 2, 2); REC (17, 2, 2) = REC (18, 2, 2);
REC (19, 2, 2) = REC (20, 2, 2); REC (20, 2, 2) = REC (21, 2, 2);
REC (22, 2, 2) = REC (23, 2, 2); REC (23, 2, 2) = REC (24, 2, 2);
REC (25, 2, 2) = REC (26, 2, 2); REC (26, 2, 2) = REC (27, 2, 2);
REC (1, 1, 2) = REC (2, 1, 2); REC (2, 1, 2) = REC (3, 1, 2); REC (3, 1, 2) = REC (4, 1, 2);
REC (4, 1, 2) = REC (5, 1, 2); REC (5, 1, 2) = REC (6, 1, 2); REC (6, 1, 2) = REC (7, 1, 2);
REC (7, 1, 2) = REC (8, 1, 2); REC (8, 1, 2) = REC (9, 1, 2);
REC (10, 1, 2) = REC (11, 1, 2); REC (11, 1, 2) = REC (12, 1, 2); REC (12, 1, 2) = REC (13, 1, 2);
REC (13, 1, 2) = REC (14, 1, 2); REC (14, 1, 2) = REC (15, 1, 2); REC (15, 1, 2) = REC (16, 1, 2);
REC (16, 1, 2) = REC (17, 1, 2); REC (17, 1, 2) = REC (18, 1, 2);
REC (19, 1, 2) = REC (20, 1, 2); REC (20, 1, 2) = REC (21, 1, 2); REC (21, 1, 2) = REC (22, 1, 2);
REC (22, 1, 2) = REC (23, 1, 2); REC (23, 1, 2) = REC (24, 1, 2); REC (24, 1, 2) = REC (25, 1, 2);
REC (25, 1, 2) = REC (26, 1, 2); REC (26, 1, 2) = REC (27, 1, 2);
OUT (1, 2, 1) = OUT (2, 2, 1); OUT (2, 2, 1) = OUT (3, 2, 1);
OUT (4, 2, 1) = OUT (5, 2, 1); OUT (5, 2, 1) = OUT (6, 2, 1);
OUT (7, 2, 1) = OUT (8, 2, 1); OUT (8, 2, 1) = OUT (9, 2, 1);
OUT (10, 2, 1) = OUT (11, 2, 1); OUT (11, 2, 1) = OUT (12, 2, 1);
OUT (13, 2, 1) = OUT (14, 2, 1); OUT (14, 2, 1) = OUT (15, 2, 1);
OUT (16, 2, 1) = OUT (17, 2, 1); OUT (17, 2, 1) = OUT (18, 2, 1);
OUT (19, 2, 1) = OUT (20, 2, 1); OUT (20, 2, 1) = OUT (21, 2, 1);
OUT (22, 2, 1) = OUT (23, 2, 1); OUT (23, 2, 1) = OUT (24, 2, 1);
OUT(25,2,1) = OUT(26,2,1); OUT(26,2,1) = OUT(27,2,1);

OUT(1,1,1) = OUT(2,1,1); OUT(2,1,1) = OUT(3,1,1); OUT(3,1,1) = OUT(4,1,1);
OUT(4,1,1) = OUT(5,1,1); OUT(5,1,1) = OUT(6,1,1); OUT(6,1,1) = OUT(7,1,1);
OUT(7,1,1) = OUT(8,1,1); OUT(8,1,1) = OUT(9,1,1);
OUT(10,1,1) = OUT(11,1,1); OUT(11,1,1) = OUT(12,1,1); OUT(12,1,1) = OUT(13,1,1);
OUT(13,1,1) = OUT(14,1,1); OUT(14,1,1) = OUT(15,1,1); OUT(15,1,1) = OUT(16,1,1);
OUT(16,1,1) = OUT(17,1,1); OUT(17,1,1) = OUT(18,1,1);
OUT(19,1,1) = OUT(20,1,1); OUT(20,1,1) = OUT(21,1,1); OUT(21,1,1) = OUT(22,1,1);
OUT(22,1,1) = OUT(23,1,1); OUT(23,1,1) = OUT(24,1,1); OUT(24,1,1) = OUT(25,1,1);
OUT(25,1,1) = OUT(26,1,1); OUT(26,1,1) = OUT(27,1,1);

OUT(1,2,2) = OUT(2,2,2); OUT(2,2,2) = OUT(3,2,2);
OUT(4,2,2) = OUT(5,2,2); OUT(5,2,2) = OUT(6,2,2);
OUT(7,2,2) = OUT(8,2,2); OUT(8,2,2) = OUT(9,2,2);
OUT(10,2,2) = OUT(11,2,2); OUT(11,2,2) = OUT(12,2,2);
OUT(13,2,2) = OUT(14,2,2); OUT(14,2,2) = OUT(15,2,2);
OUT(16,2,2) = OUT(17,2,2); OUT(17,2,2) = OUT(18,2,2);
OUT(19,2,2) = OUT(20,2,2); OUT(20,2,2) = OUT(21,2,2);
OUT(22,2,2) = OUT(23,2,2); OUT(23,2,2) = OUT(24,2,2);
OUT(25,2,2) = OUT(26,2,2); OUT(26,2,2) = OUT(27,2,2);

OUT(1,1,2) = OUT(2,1,2); OUT(2,1,2) = OUT(3,1,2); OUT(3,1,2) = OUT(4,1,2);
OUT(4,1,2) = OUT(5,1,2); OUT(5,1,2) = OUT(6,1,2); OUT(6,1,2) = OUT(7,1,2);
OUT(7,1,2) = OUT(8,1,2); OUT(8,1,2) = OUT(9,1,2);
OUT(10,1,2) = OUT(11,1,2); OUT(11,1,2) = OUT(12,1,2); OUT(12,1,2) = OUT(13,1,2);
OUT(13,1,2) = OUT(14,1,2); OUT(14,1,2) = OUT(15,1,2); OUT(15,1,2) = OUT(16,1,2);
OUT(16,1,2) = OUT(17,1,2); OUT(17,1,2) = OUT(18,1,2);
OUT(19,1,2) = OUT(20,1,2); OUT(20,1,2) = OUT(21,1,2); OUT(21,1,2) = OUT(22,1,2);
OUT(22,1,2) = OUT(23,1,2); OUT(23,1,2) = OUT(24,1,2); OUT(24,1,2) = OUT(25,1,2);
OUT(25,1,2) = OUT(26,1,2); OUT(26,1,2) = OUT(27,1,2);

LOSS(1,2,1) = LOSS(2,2,1); LOSS(2,2,1) = LOSS(3,2,1);
LOSS(4,2,1) = LOSS(5,2,1); LOSS(5,2,1) = LOSS(6,2,1);
LOSS(7,2,1) = LOSS(8,2,1); LOSS(8,2,1) = LOSS(9,2,1);
LOSS(10,2,1) = LOSS(11,2,1); LOSS(11,2,1) = LOSS(12,2,1);
LOSS(13,2,1) = LOSS(14,2,1); LOSS(14,2,1) = LOSS(15,2,1);
LOSS(16,2,1) = LOSS(17,2,1); LOSS(17,2,1) = LOSS(18,2,1);
LOSS(19,2,1) = LOSS(20,2,1); LOSS(20,2,1) = LOSS(21,2,1);
LOSS(22,2,1) = LOSS(23,2,1); LOSS(23,2,1) = LOSS(24,2,1);
LOSS(25,2,1) = LOSS(26,2,1); LOSS(26,2,1) = LOSS(27,2,1);

LOSS(1,1,1) = LOSS(2,1,1); LOSS(2,1,1) = LOSS(3,1,1); LOSS(3,1,1) = LOSS(4,1,1);
LOSS(4,1,1) = LOSS(5,1,1); LOSS(5,1,1) = LOSS(6,1,1); LOSS(6,1,1) = LOSS(7,1,1);
LOSS(7,1,1) = LOSS(8,1,1); LOSS(8,1,1) = LOSS(9,1,1);
LOSS(10,1,1) = LOSS(11,1,1); LOSS(11,1,1) = LOSS(12,1,1);
LOSS(12,1,1) = LOSS(13,1,1); LOSS(13,1,1) = LOSS(14,1,1);
LOSS(14,1,1) = LOSS(15,1,1); LOSS(15,1,1) = LOSS(16,1,1);
LOSS(16,1,1) = LOSS(17,1,1); LOSS(17,1,1) = LOSS(18,1,1);
LOSS(19,1,1) = LOSS(20,1,1); LOSS(20,1,1) = LOSS(21,1,1);
LOSS(21,1,1) = LOSS(22,1,1); LOSS(22,1,1) = LOSS(23,1,1);
LOSS(23,1,1) = LOSS(24,1,1); LOSS(24,1,1) = LOSS(25,1,1);
LOSS(25,1,1) = LOSS(26,1,1); LOSS(26,1,1) = LOSS(27,1,1);

LOSS(1,2,2) = LOSS(2,2,2); LOSS(2,2,2) = LOSS(3,2,2);
LOSS(4,2,2) = LOSS(5,2,2); LOSS(5,2,2) = LOSS(6,2,2);
Appendix A

\[
\begin{align*}
JUNK(12,1,2) &= JUNK(13,1,2); \\
JUNK(13,1,2) &= JUNK(14,1,2); \\
JUNK(14,1,2) &= JUNK(15,1,2); \\
JUNK(15,1,2) &= JUNK(16,1,2); \\
JUNK(16,1,2) &= JUNK(17,1,2); \\
JUNK(17,1,2) &= JUNK(18,1,2); \\
JUNK(18,1,2) &= JUNK(20,1,2); \\
JUNK(20,1,2) &= JUNK(21,1,2); \\
JUNK(21,1,2) &= JUNK(22,1,2); \\
JUNK(22,1,2) &= JUNK(23,1,2); \\
JUNK(23,1,2) &= JUNK(24,1,2); \\
JUNK(24,1,2) &= JUNK(25,1,2); \\
JUNK(25,1,2) &= JUNK(26,1,2); \\
JUNK(26,1,2) &= JUNK(27,1,2); \\

PEN(1,2,1) &= PEN(2,2,1); \\
PEN(2,2,1) &= PEN(3,2,1); \\
PEN(4,2,1) &= PEN(5,2,1); \\
PEN(5,2,1) &= PEN(6,2,1); \\
PEN(7,2,1) &= PEN(8,2,1); \\
PEN(8,2,1) &= PEN(9,2,1); \\
PEN(10,2,1) &= PEN(11,2,1); \\
PEN(11,2,1) &= PEN(12,2,1); \\
PEN(13,2,1) &= PEN(14,2,1); \\
PEN(14,2,1) &= PEN(15,2,1); \\
PEN(16,2,1) &= PEN(17,2,1); \\
PEN(17,2,1) &= PEN(18,2,1); \\
PEN(19,2,1) &= PEN(20,2,1); \\
PEN(20,2,1) &= PEN(21,2,1); \\
PEN(22,2,1) &= PEN(23,2,1); \\
PEN(23,2,1) &= PEN(24,2,1); \\
PEN(25,2,1) &= PEN(26,2,1); \\
PEN(26,2,1) &= PEN(27,2,1); \\

PEN(1,1,1) &= PEN(2,1,1); \\
PEN(2,1,1) &= PEN(3,1,1); \\
PEN(3,1,1) &= PEN(4,1,1); \\
PEN(4,1,1) &= PEN(5,1,1); \\
PEN(5,1,1) &= PEN(6,1,1); \\
PEN(6,1,1) &= PEN(7,1,1); \\
PEN(7,1,1) &= PEN(8,1,1); \\
PEN(8,1,1) &= PEN(9,1,1); \\
PEN(10,1,1) &= PEN(11,1,1); \\
PEN(11,1,1) &= PEN(12,1,1); \\
PEN(12,1,1) &= PEN(13,1,1); \\
PEN(13,1,1) &= PEN(14,1,1); \\
PEN(14,1,1) &= PEN(15,1,1); \\
PEN(15,1,1) &= PEN(16,1,1); \\
PEN(16,1,1) &= PEN(17,1,1); \\
PEN(17,1,1) &= PEN(18,1,1); \\
PEN(18,1,1) &= PEN(19,1,1); \\
PEN(19,1,1) &= PEN(20,1,1); \\
PEN(20,1,1) &= PEN(21,1,1); \\
PEN(21,1,1) &= PEN(22,1,1); \\
PEN(22,1,1) &= PEN(23,1,1); \\
PEN(23,1,1) &= PEN(24,1,1); \\
PEN(24,1,1) &= PEN(25,1,1); \\
PEN(25,1,1) &= PEN(26,1,1); \\
PEN(26,1,1) &= PEN(27,1,1); \\

PEN(1,2,2) &= PEN(2,2,2); \\
PEN(2,2,2) &= PEN(3,2,2); \\
PEN(4,2,2) &= PEN(5,2,2); \\
PEN(5,2,2) &= PEN(6,2,2); \\
PEN(7,2,2) &= PEN(8,2,2); \\
PEN(8,2,2) &= PEN(9,2,2); \\
PEN(10,2,2) &= PEN(11,2,2); \\
PEN(11,2,2) &= PEN(12,2,2); \\
PEN(13,2,2) &= PEN(14,2,2); \\
PEN(14,2,2) &= PEN(15,2,2); \\
PEN(16,2,2) &= PEN(17,2,2); \\
PEN(17,2,2) &= PEN(18,2,2); \\
PEN(19,2,2) &= PEN(20,2,2); \\
PEN(20,2,2) &= PEN(21,2,2); \\
PEN(22,2,2) &= PEN(23,2,2); \\
PEN(23,2,2) &= PEN(24,2,2); \\
PEN(25,2,2) &= PEN(26,2,2); \\
PEN(26,2,2) &= PEN(27,2,2); \\

PEN(1,1,2) &= PEN(2,1,2); \\
PEN(2,1,2) &= PEN(3,1,2); \\
PEN(3,1,2) &= PEN(4,1,2); \\
PEN(4,1,2) &= PEN(5,1,2); \\
PEN(5,1,2) &= PEN(6,1,2); \\
PEN(6,1,2) &= PEN(7,1,2); \\
PEN(7,1,2) &= PEN(8,1,2); \\
PEN(8,1,2) &= PEN(9,1,2); \\
PEN(10,1,2) &= PEN(11,1,2); \\
PEN(11,1,2) &= PEN(12,1,2); \\
PEN(12,1,2) &= PEN(13,1,2); \\
PEN(13,1,2) &= PEN(14,1,2); \\
PEN(14,1,2) &= PEN(15,1,2); \\
PEN(15,1,2) &= PEN(16,1,2); \\
PEN(16,1,2) &= PEN(17,1,2); \\
PEN(17,1,2) &= PEN(18,1,2); \\
PEN(18,1,2) &= PEN(19,1,2); \\
PEN(19,1,2) &= PEN(20,1,2); \\
PEN(20,1,2) &= PEN(21,1,2); \\
PEN(21,1,2) &= PEN(22,1,2); \\
PEN(22,1,2) &= PEN(23,1,2); \\
PEN(23,1,2) &= PEN(24,1,2); \\
PEN(24,1,2) &= PEN(25,1,2); \\
PEN(25,1,2) &= PEN(26,1,2); \\
PEN(26,1,2) &= PEN(27,1,2); \\

DATA: \\
PROB = @OLE('F:\PROB.xls'); \\
RATE2 = @OLE('F:\RATE Comp.xls'); \\
RATE3 = @OLE('F:\RATE Comp.xls'); \\
RATE4 = @OLE('F:\RATE Comp.xls'); \\
DEM2 = 40 40; \\
DEM3 = 40 40; \\
DEM4 = 40 40; \\
\end{align*}
\]
MODEL:
!Two-product with overlapping parts and four-period stochastic programming formulation;

SETS:
SCENARIO/1 .. 27/:PROB;
PERIOD/2,3,4/;
COMP/1 .. 3/:Cin,S1,Crec,Cout,Cjunk,Closs,Cpenalty,CAP1,CAP2,DEM2,DEM3,DEM4;
LINKS(SCENARIO,PERIOD,COMP):REC,OUT,LOSS,PEN,JUNK,EXTRA;
LINKS_2(SCENARIO,COMP):T1,T2,T3,T4,T5,T6,TB1_A,TB1_B,TB2_A,TB2_B,TB3_A,TB3_B,
TB4_A,TB4_B,TB5_A,TB5_B,TB6_A,TB6_B,RATE2_A,RATE2_B,RATE3_A,RATE3_B,RATE4_A,
RATE4_B;
LINKS_3(SCENARIO,PERIOD);
ENDSETS

MIN = @SUM(COMP(M):Cin(M)*S1(M)) +
@SUM(LINKS(K,L,M):PROB(K)*(Crec(M)*REC(K,L,M) + Cout(M)*OUT(K,L,M) +
Closs(M)*LOSS(K,L,M) + Cjunk(M)*JUNK(K,L,M)+Cpenalty(M)*PEN(K,L,M)));

!Demand met or not;
@FOR(LINKS_2(K,M):T1(K,M)+T2(K,M) = 1);
@FOR(LINKS_2(K,M):T3(K,M)+T4(K,M) = 1);
@FOR(LINKS_2(K,M):T5(K,M)+T6(K,M) = 1);
@FOR(LINKS_2(K,M):TB1_A(K,M)+TB2_A(K,M) = RET2_A);
@FOR(LINKS_2(K,M):TB3_A(K,M)+TB4_A(K,M) = RET3_A);
@FOR(LINKS_2(K,M):TB5_A(K,M)+TB6_A(K,M) = RET4_A);
@FOR(LINKS_2(K,M):TB1_B(K,M)+TB2_B(K,M) = RET2_B);
@FOR(LINKS_2(K,M):TB3_B(K,M)+TB4_B(K,M) = RET3_B);
@FOR(LINKS_2(K,M):TB5_B(K,M)+TB6_B(K,M) = RET4_B);

@FOR(LINKS_2(K,M):@ABS(TB1_A(K,M)-TB2_A(K,M)) = RET2_A);
@FOR(LINKS_2(K,M):@ABS(TB3_A(K,M)-TB4_A(K,M)) = RET3_A);
@FOR(LINKS_2(K,M):@ABS(TB5_A(K,M)-TB6_A(K,M)) = RET4_A);
@FOR(LINKS_2(K,M):@ABS(TB1_B(K,M)-TB2_B(K,M)) = RET2_B);
@FOR(LINKS_2(K,M):@ABS(TB3_B(K,M)-TB4_B(K,M)) = RET3_B);
@FOR(LINKS_2(K,M):@ABS(TB5_B(K,M)-TB6_B(K,M)) = RET4_B);

!Extra components and unmet demand;
@FOR(LINKS_2(K,M):PEN(K,1,M) = -TB1_A(K,M)*RATE2_A(K,M)-
TB1_B(K,M)*RATE2_B(K,M)+DEM2(M)*T1(K,M));
@FOR(LINKS_2(K,M):EXTRA(K,1,M) =
Appendix A

\[ TB2_A(K,M) \times RATE2_A(K,M) + TB2_B(K,M) \times RATE2_B(K,M) - DEM2(M) \times T2(K,M); \]
\[ @FOR \text{(LINKS}_2(K,M): PEN(K,2,M) = -TB3_A(K,M) \times RATE3_A(K,M) - \]
\[ TB3_B(K,M) \times RATE3_B(K,M) - T3(K,M) \times EXTRA(K,1,M) + T3(K,M) \times DEM3(M); \]
\[ @FOR \text{(LINKS}_2(K,M): EXTRA(K,2,M) = \]
\[ TB4_A(K,M) \times RATE3_A(K,M) + TB4_B(K,M) \times RATE3_B(K,M) + T4(K,M) \times EXTRA(K,1,M) - \]
\[ T4(K,M) \times DEM3(M); \]
\[ @FOR \text{(LINKS}_2(K,M): PEN(K,3,M) = -TB5_A(K,M) \times RATE4_A(K,M) - \]
\[ TB5_B(K,M) \times RATE4_B(K,M) - T5(K,M) \times EXTRA(K,2,M) + T5(K,M) \times DEM4(M); \]
\[ @FOR \text{(LINKS}_2(K,M): EXTRA(K,3,M) = \]
\[ TB6_A(K,M) \times RATE4_A(K,M) + TB6_B(K,M) \times RATE4_B(K,M) + T6(K,M) \times EXTRA(K,2,M) - \]
\[ T6(K,M) \times DEM4(M); \]

! Ensuring all extras are stored internally, externally, by reconfiguring or
they are disposed of;
\[ @FOR \text{(LINKS}(K,L,M): S1(M) + REC(K,L,M) + OUT(K,L,M) + JUNK(K,L,M) \geq EXTRA(K,L,M)); \]

! Unused space;
\[ @FOR \text{(LINKS}(K,L,M): LOSS(K,L,M) = S1(M) + REC(K,L,M) + OUT(K,L,M) + JUNK(K,L,M) - \]
\[ EXTRA(K,L,M)); \]
\[ @FOR \text{(LINKS}(K,L,M): EXTRA(K,L,M) - S1(M) - OUT(K,L,M) \geq 0); \]

! Capacity constraints;
\[ @FOR \text{(LINKS}(K,L,M): S1(M) + REC(K,L,M) \leq CAP1(M)); \]
\[ @FOR \text{(LINKS}(K,L,M): OUT(K,L,M) \leq CAP2(M)); \]

! Binary variables;
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T1)); \]
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T2)); \]
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T3)); \]
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T4)); \]
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T5)); \]
\[ @FOR \text{(LINKS}_2(K,M): @BIN(T6)); \]

! Non-anticipativity constraints;
\[ REC(1,2,1) = REC(2,2,1); \]
\[ REC(2,2,1) = REC(3,2,1); \]
\[ REC(4,2,1) = REC(5,2,1); \]
\[ REC(5,2,1) = REC(6,2,1); \]
\[ REC(7,2,1) = REC(8,2,1); \]
\[ REC(8,2,1) = REC(9,2,1); \]
\[ REC(10,2,1) = REC(11,2,1); \]
\[ REC(11,2,1) = REC(12,2,1); \]
\[ REC(13,2,1) = REC(14,2,1); \]
\[ REC(14,2,1) = REC(15,2,1); \]
\[ REC(16,2,1) = REC(17,2,1); \]
\[ REC(17,2,1) = REC(18,2,1); \]
\[ REC(19,2,1) = REC(20,2,1); \]
\[ REC(20,2,1) = REC(21,2,1); \]
\[ REC(22,2,1) = REC(23,2,1); \]
\[ REC(23,2,1) = REC(24,2,1); \]
\[ REC(25,2,1) = REC(26,2,1); \]
\[ REC(26,2,1) = REC(27,2,1); \]
\[ REC(1,1,1) = REC(2,1,1); \]
\[ REC(2,1,1) = REC(3,1,1); \]
\[ REC(3,1,1) = REC(4,1,1); \]
\[ REC(4,1,1) = REC(5,1,1); \]
\[ REC(5,1,1) = REC(6,1,1); \]
\[ REC(6,1,1) = REC(7,1,1); \]
\[ REC(7,1,1) = REC(8,1,1); \]
\[ REC(8,1,1) = REC(9,1,1); \]
\[ REC(10,1,1) = REC(11,1,1); \]
\[ REC(11,1,1) = REC(12,1,1); \]
\[ REC(12,1,1) = REC(13,1,1); \]
\[ REC(13,1,1) = REC(14,1,1); \]
\[ REC(14,1,1) = REC(15,1,1); \]
\[ REC(15,1,1) = REC(16,1,1); \]
\[ REC(16,1,1) = REC(17,1,1); \]
\[ REC(17,1,1) = REC(18,1,1); \]
\[ REC(19,1,1) = REC(20,1,1); \]
\[ REC(20,1,1) = REC(21,1,1); \]
\[ REC(21,1,1) = REC(22,1,1); \]
\[ REC(22,1,1) = REC(23,1,1); \]
\[ REC(23,1,1) = REC(24,1,1); \]
\[ REC(24,1,1) = REC(25,1,1); \]
\[ REC(25,1,1) = REC(26,1,1); \]
\[ REC(26,1,1) = REC(27,1,1); \]
\[ REC(1,2,2) = REC(2,2,2); \]
\[ REC(2,2,2) = REC(3,2,2); \]
\[ REC(4,2,2) = REC(5,2,2); \]
\[ REC(5,2,2) = REC(6,2,2); \]
REC(7,2,2) = REC(8,2,2); REC(8,2,2) = REC(9,2,2);
REC(10,2,2) = REC(11,2,2); REC(11,2,2) = REC(12,2,2);
REC(13,2,2) = REC(14,2,2); REC(14,2,2) = REC(15,2,2);
REC(16,2,2) = REC(17,2,2); REC(17,2,2) = REC(18,2,2);
REC(19,2,2) = REC(20,2,2); REC(20,2,2) = REC(21,2,2);
REC(22,2,2) = REC(23,2,2); REC(23,2,2) = REC(24,2,2);
REC(25,2,2) = REC(26,2,2); REC(26,2,2) = REC(27,2,2);
REC(1,1,2) = REC(2,1,2); REC(2,1,2) = REC(3,1,2); REC(3,1,2) = REC(4,1,2);
REC(4,1,2) = REC(5,1,2); REC(5,1,2) = REC(6,1,2); REC(6,1,2) = REC(7,1,2);
REC(7,1,2) = REC(8,1,2); REC(8,1,2) = REC(9,1,2);
REC(10,1,2) = REC(11,1,2); REC(11,1,2) = REC(12,1,2); REC(12,1,2) = REC(13,1,2);
REC(13,1,2) = REC(14,1,2); REC(14,1,2) = REC(15,1,2); REC(15,1,2) = REC(16,1,2);
REC(16,1,2) = REC(17,1,2); REC(17,1,2) = REC(18,1,2);
REC(19,1,2) = REC(20,1,2); REC(20,1,2) = REC(21,1,2); REC(21,1,2) = REC(22,1,2);
REC(22,1,2) = REC(23,1,2); REC(23,1,2) = REC(24,1,2); REC(24,1,2) = REC(25,1,2);
REC(25,1,2) = REC(26,1,2); REC(26,1,2) = REC(27,1,2);
REC(1,1,3) = REC(2,1,3); REC(2,1,3) = REC(3,1,3); REC(3,1,3) = REC(4,1,3);
REC(4,1,3) = REC(5,1,3); REC(5,1,3) = REC(6,1,3); REC(6,1,3) = REC(7,1,3);
REC(7,1,3) = REC(8,1,3); REC(8,1,3) = REC(9,1,3);
REC(10,1,3) = REC(11,1,3); REC(11,1,3) = REC(12,1,3); REC(12,1,3) = REC(13,1,3);
REC(13,1,3) = REC(14,1,3); REC(14,1,3) = REC(15,1,3); REC(15,1,3) = REC(16,1,3);
REC(16,1,3) = REC(17,1,3); REC(17,1,3) = REC(18,1,3);
REC(19,1,3) = REC(20,1,3); REC(20,1,3) = REC(21,1,3); REC(21,1,3) = REC(22,1,3);
REC(22,1,3) = REC(23,1,3); REC(23,1,3) = REC(24,1,3); REC(24,1,3) = REC(25,1,3);
REC(25,1,3) = REC(26,1,3); REC(26,1,3) = REC(27,1,3);

OUT(1,2,1) = OUT(2,2,1); OUT(2,2,1) = OUT(3,2,1);
OUT(4,2,1) = OUT(5,2,1); OUT(5,2,1) = OUT(6,2,1);
OUT(7,2,1) = OUT(8,2,1); OUT(8,2,1) = OUT(9,2,1);
OUT(10,2,1) = OUT(11,2,1); OUT(11,2,1) = OUT(12,2,1);
OUT(13,2,1) = OUT(14,2,1); OUT(14,2,1) = OUT(15,2,1);
OUT(16,2,1) = OUT(17,2,1); OUT(17,2,1) = OUT(18,2,1);
OUT(19,2,1) = OUT(20,2,1); OUT(20,2,1) = OUT(21,2,1);
OUT(22,2,1) = OUT(23,2,1); OUT(23,2,1) = OUT(24,2,1);
OUT(25,2,1) = OUT(26,2,1); OUT(26,2,1) = OUT(27,2,1);

OUT(1,1,1) = OUT(2,1,1); OUT(2,1,1) = OUT(3,1,1); OUT(3,1,1) = OUT(4,1,1);
OUT(4,1,1) = OUT(5,1,1); OUT(5,1,1) = OUT(6,1,1); OUT(6,1,1) = OUT(7,1,1);
OUT(7,1,1) = OUT(8,1,1); OUT(8,1,1) = OUT(9,1,1);
OUT(10,1,1) = OUT(11,1,1); OUT(11,1,1) = OUT(12,1,1); OUT(12,1,1) = OUT(13,1,1);
OUT(13,1,1) = OUT(14,1,1); OUT(14,1,1) = OUT(15,1,1); OUT(15,1,1) = OUT(16,1,1);
OUT(16,1,1) = OUT(17,1,1); OUT(17,1,1) = OUT(18,1,1);
OUT(19,1,1) = OUT(20,1,1); OUT(20,1,1) = OUT(21,1,1); OUT(21,1,1) = OUT(22,1,1);
<table>
<thead>
<tr>
<th>OUT</th>
<th>LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(22,1,1) = OUT(23,1,1)</td>
<td>LOSS(1,2,1) = LOSS(2,2,1)</td>
</tr>
<tr>
<td>(23,1,1) = OUT(24,1,1)</td>
<td>LOSS(4,2,1) = LOSS(5,2,1)</td>
</tr>
<tr>
<td>(24,1,1) = OUT(25,1,1)</td>
<td>LOSS(7,2,1) = LOSS(8,2,1)</td>
</tr>
<tr>
<td>(25,1,1) = OUT(26,1,1)</td>
<td>LOSS(10,2,1) = LOSS(11,2,1)</td>
</tr>
<tr>
<td>(26,1,1) = OUT(27,1,1)</td>
<td>LOSS(13,2,1) = LOSS(14,2,1)</td>
</tr>
<tr>
<td>(27,1,1) = OUT(28,1,1)</td>
<td>LOSS(16,2,1) = LOSS(17,2,1)</td>
</tr>
<tr>
<td>(1,2,2) = OUT(2,2,2)</td>
<td>LOSS(19,2,1) = LOSS(20,2,1)</td>
</tr>
<tr>
<td>(2,2,2) = OUT(3,2,2)</td>
<td>LOSS(22,2,1) = LOSS(23,2,1)</td>
</tr>
<tr>
<td>(3,2,2) = OUT(4,2,2)</td>
<td>LOSS(25,2,1) = LOSS(26,2,1)</td>
</tr>
<tr>
<td>(4,2,2) = OUT(5,2,2)</td>
<td>LOSS(1,1,2) = LOSS(2,1,2)</td>
</tr>
<tr>
<td>(5,2,2) = OUT(6,2,2)</td>
<td>LOSS(4,1,2) = LOSS(5,1,2)</td>
</tr>
<tr>
<td>(6,2,2) = OUT(7,1,2)</td>
<td>LOSS(7,1,2) = LOSS(8,1,2)</td>
</tr>
<tr>
<td>(7,2,2) = OUT(8,2,2)</td>
<td>LOSS(10,1,2) = LOSS(11,1,2)</td>
</tr>
<tr>
<td>(8,2,2) = OUT(9,1,2)</td>
<td>LOSS(13,1,2) = LOSS(14,1,2)</td>
</tr>
<tr>
<td>(10,2,2) = OUT(11,2,2)</td>
<td>LOSS(16,1,2) = LOSS(17,1,2)</td>
</tr>
<tr>
<td>(11,2,2) = OUT(12,2,2)</td>
<td>LOSS(19,1,2) = LOSS(20,1,2)</td>
</tr>
<tr>
<td>(12,2,2) = OUT(13,1,2)</td>
<td>LOSS(22,1,2) = LOSS(23,1,2)</td>
</tr>
<tr>
<td>(13,2,2) = OUT(14,1,2)</td>
<td>LOSS(25,1,2) = LOSS(26,1,2)</td>
</tr>
<tr>
<td>(14,2,2) = OUT(15,1,2)</td>
<td>LOSS(1,2,3) = LOSS(2,2,3)</td>
</tr>
<tr>
<td>(15,2,2) = OUT(16,1,2)</td>
<td>LOSS(4,2,3) = LOSS(5,2,3)</td>
</tr>
<tr>
<td>(16,2,3) = OUT(17,2,3)</td>
<td>LOSS(7,2,3) = LOSS(8,2,3)</td>
</tr>
<tr>
<td>(17,2,3) = OUT(18,2,3)</td>
<td>LOSS(10,2,3) = LOSS(11,2,3)</td>
</tr>
<tr>
<td>(18,2,3) = OUT(19,2,3)</td>
<td>LOSS(13,2,3) = LOSS(14,2,3)</td>
</tr>
<tr>
<td>(19,2,3) = OUT(20,2,3)</td>
<td>LOSS(16,2,3) = LOSS(17,2,3)</td>
</tr>
<tr>
<td>(20,2,3) = OUT(21,2,3)</td>
<td>LOSS(22,1,3) = LOSS(23,1,3)</td>
</tr>
<tr>
<td>(21,2,3) = OUT(22,1,2)</td>
<td>LOSS(25,1,3) = LOSS(26,1,3)</td>
</tr>
<tr>
<td>(22,1,3) = OUT(23,1,3)</td>
<td>LOSS(1,1,3) = LOSS(2,1,3)</td>
</tr>
<tr>
<td>(23,1,3) = OUT(24,1,3)</td>
<td>LOSS(4,1,3) = LOSS(5,1,3)</td>
</tr>
<tr>
<td>(24,1,3) = OUT(25,1,3)</td>
<td>LOSS(7,1,3) = LOSS(8,1,3)</td>
</tr>
<tr>
<td>(25,1,3) = OUT(26,1,3)</td>
<td>LOSS(10,1,3) = LOSS(11,1,3)</td>
</tr>
<tr>
<td>(26,1,3) = OUT(27,1,3)</td>
<td>LOSS(13,1,3) = LOSS(14,1,3)</td>
</tr>
<tr>
<td>(27,1,3) = OUT(28,1,3)</td>
<td>LOSS(16,1,3) = LOSS(17,1,3)</td>
</tr>
<tr>
<td>(1,2,1) = LOSS(2,2,1)</td>
<td>LOSS(4,2,1) = LOSS(5,2,1)</td>
</tr>
<tr>
<td>(2,2,1) = LOSS(3,2,1)</td>
<td>LOSS(7,2,1) = LOSS(8,2,1)</td>
</tr>
<tr>
<td>(3,2,1) = LOSS(4,2,1)</td>
<td>LOSS(10,2,1) = LOSS(11,2,1)</td>
</tr>
<tr>
<td>(4,2,1) = LOSS(5,2,1)</td>
<td>LOSS(13,2,1) = LOSS(14,2,1)</td>
</tr>
<tr>
<td>(5,2,1) = LOSS(6,2,1)</td>
<td>LOSS(16,2,1) = LOSS(17,2,1)</td>
</tr>
<tr>
<td>(6,2,1) = LOSS(7,2,1)</td>
<td>LOSS(19,2,1) = LOSS(20,2,1)</td>
</tr>
<tr>
<td>(7,2,1) = LOSS(8,2,1)</td>
<td>LOSS(22,2,1) = LOSS(23,2,1)</td>
</tr>
<tr>
<td>(8,2,1) = LOSS(9,2,1)</td>
<td>LOSS(25,2,1) = LOSS(26,2,1)</td>
</tr>
<tr>
<td>(9,2,1) = LOSS(10,2,1)</td>
<td>LOSS(1,1,2) = LOSS(2,1,2)</td>
</tr>
<tr>
<td>(10,2,1) = LOSS(11,2,1)</td>
<td>LOSS(4,1,2) = LOSS(5,1,2)</td>
</tr>
<tr>
<td>(11,2,1) = LOSS(12,2,1)</td>
<td>LOSS(7,1,2) = LOSS(8,1,2)</td>
</tr>
<tr>
<td>(12,2,1) = LOSS(13,2,1)</td>
<td>LOSS(16,1,2) = LOSS(17,1,2)</td>
</tr>
<tr>
<td>(13,2,1) = LOSS(14,2,1)</td>
<td>LOSS(19,1,2) = LOSS(20,1,2)</td>
</tr>
<tr>
<td>(14,2,1) = LOSS(15,2,1)</td>
<td>LOSS(22,1,2) = LOSS(23,1,2)</td>
</tr>
<tr>
<td>(15,2,1) = LOSS(16,1,2)</td>
<td>LOSS(25,1,2) = LOSS(26,1,2)</td>
</tr>
<tr>
<td>(16,2,1) = LOSS(17,2,1)</td>
<td>LOSS(1,1,3) = LOSS(2,1,3)</td>
</tr>
<tr>
<td>(17,2,1) = LOSS(18,2,1)</td>
<td>LOSS(4,1,3) = LOSS(5,1,3)</td>
</tr>
</tbody>
</table>
LOSS(10,1,1) = LOSS(11,1,1); LOSS(11,1,1) = LOSS(12,1,1); LOSS(12,1,1) = LOSS(13,1,1); LOSS(13,1,1) = LOSS(14,1,1); LOSS(14,1,1) = LOSS(15,1,1); LOSS(15,1,1) = LOSS(16,1,1); LOSS(16,1,1) = LOSS(17,1,1); LOSS(17,1,1) = LOSS(18,1,1); LOSS(19,1,1) = LOSS(20,1,1); LOSS(20,1,1) = LOSS(21,1,1); LOSS(21,1,1) = LOSS(22,1,1); LOSS(22,1,1) = LOSS(23,1,1); LOSS(23,1,1) = LOSS(24,1,1); LOSS(24,1,1) = LOSS(25,1,1); LOSS(25,1,1) = LOSS(26,1,1); LOSS(26,1,1) = LOSS(27,1,1);

LOSS(1,2,2) = LOSS(2,2,2); LOSS(2,2,2) = LOSS(3,2,2); LOSS(3,2,2) = LOSS(4,2,2); LOSS(4,2,2) = LOSS(5,2,2); LOSS(5,2,2) = LOSS(6,2,2); LOSS(6,2,2) = LOSS(7,2,2); LOSS(7,2,2) = LOSS(8,2,2); LOSS(8,2,2) = LOSS(9,2,2); LOSS(9,2,2) = LOSS(10,2,2); LOSS(10,2,2) = LOSS(11,2,2); LOSS(11,2,2) = LOSS(12,2,2); LOSS(12,2,2) = LOSS(13,2,2); LOSS(13,2,2) = LOSS(14,2,2); LOSS(14,2,2) = LOSS(15,2,2); LOSS(15,2,2) = LOSS(16,2,2); LOSS(16,2,2) = LOSS(17,2,2); LOSS(17,2,2) = LOSS(18,2,2); LOSS(18,2,2) = LOSS(19,2,2); LOSS(19,2,2) = LOSS(20,2,2); LOSS(20,2,2) = LOSS(21,2,2); LOSS(21,2,2) = LOSS(22,2,2); LOSS(22,2,2) = LOSS(23,2,2); LOSS(23,2,2) = LOSS(24,2,2); LOSS(24,2,2) = LOSS(25,2,2); LOSS(25,2,2) = LOSS(26,2,2); LOSS(26,2,2) = LOSS(27,2,2);

LOSS(1,1,2) = LOSS(2,1,2); LOSS(2,1,2) = LOSS(3,1,2); LOSS(3,1,2) = LOSS(4,1,2); LOSS(4,1,2) = LOSS(5,1,2); LOSS(5,1,2) = LOSS(6,1,2); LOSS(6,1,2) = LOSS(7,1,2); LOSS(7,1,2) = LOSS(8,1,2); LOSS(8,1,2) = LOSS(9,1,2); LOSS(9,1,2) = LOSS(10,1,2); LOSS(10,1,2) = LOSS(11,1,2); LOSS(11,1,2) = LOSS(12,1,2); LOSS(12,1,2) = LOSS(13,1,2); LOSS(13,1,2) = LOSS(14,1,2); LOSS(14,1,2) = LOSS(15,1,2); LOSS(15,1,2) = LOSS(16,1,2); LOSS(16,1,2) = LOSS(17,1,2); LOSS(17,1,2) = LOSS(18,1,2); LOSS(18,1,2) = LOSS(19,1,2); LOSS(19,1,2) = LOSS(20,1,2); LOSS(20,1,2) = LOSS(21,1,2); LOSS(21,1,2) = LOSS(22,1,2); LOSS(22,1,2) = LOSS(23,1,2); LOSS(23,1,2) = LOSS(24,1,2); LOSS(24,1,2) = LOSS(25,1,2); LOSS(25,1,2) = LOSS(26,1,2); LOSS(26,1,2) = LOSS(27,1,2);

LOSS(1,2,3) = LOSS(2,2,3); LOSS(2,2,3) = LOSS(3,2,3); LOSS(3,2,3) = LOSS(4,2,3); LOSS(4,2,3) = LOSS(5,2,3); LOSS(5,2,3) = LOSS(6,2,3); LOSS(6,2,3) = LOSS(7,2,3); LOSS(7,2,3) = LOSS(8,2,3); LOSS(8,2,3) = LOSS(9,2,3); LOSS(9,2,3) = LOSS(10,2,3); LOSS(10,2,3) = LOSS(11,2,3); LOSS(11,2,3) = LOSS(12,2,3); LOSS(12,2,3) = LOSS(13,2,3); LOSS(13,2,3) = LOSS(14,2,3); LOSS(14,2,3) = LOSS(15,2,3); LOSS(15,2,3) = LOSS(16,2,3); LOSS(16,2,3) = LOSS(17,2,3); LOSS(17,2,3) = LOSS(18,2,3); LOSS(18,2,3) = LOSS(19,2,3); LOSS(19,2,3) = LOSS(20,2,3); LOSS(20,2,3) = LOSS(21,2,3); LOSS(21,2,3) = LOSS(22,2,3); LOSS(22,2,3) = LOSS(23,2,3); LOSS(23,2,3) = LOSS(24,2,3); LOSS(24,2,3) = LOSS(25,2,3); LOSS(25,2,3) = LOSS(26,2,3); LOSS(26,2,3) = LOSS(27,2,3);

LOSS(1,1,3) = LOSS(2,1,3); LOSS(2,1,3) = LOSS(3,1,3); LOSS(3,1,3) = LOSS(4,1,3); LOSS(4,1,3) = LOSS(5,1,3); LOSS(5,1,3) = LOSS(6,1,3); LOSS(6,1,3) = LOSS(7,1,3); LOSS(7,1,3) = LOSS(8,1,3); LOSS(8,1,3) = LOSS(9,1,3); LOSS(9,1,3) = LOSS(10,1,3); LOSS(10,1,3) = LOSS(11,1,3); LOSS(11,1,3) = LOSS(12,1,3); LOSS(12,1,3) = LOSS(13,1,3); LOSS(13,1,3) = LOSS(14,1,3); LOSS(14,1,3) = LOSS(15,1,3); LOSS(15,1,3) = LOSS(16,1,3); LOSS(16,1,3) = LOSS(17,1,3); LOSS(17,1,3) = LOSS(18,1,3); LOSS(18,1,3) = LOSS(19,1,3); LOSS(19,1,3) = LOSS(20,1,3); LOSS(20,1,3) = LOSS(21,1,3); LOSS(21,1,3) = LOSS(22,1,3); LOSS(22,1,3) = LOSS(23,1,3); LOSS(23,1,3) = LOSS(24,1,3); LOSS(24,1,3) = LOSS(25,1,3); LOSS(25,1,3) = LOSS(26,1,3); LOSS(26,1,3) = LOSS(27,1,3);

JUNK(1,2,1) = JUNK(2,2,1); JUNK(2,2,1) = JUNK(3,2,1); JUNK(4,2,1) = JUNK(5,2,1); JUNK(5,2,1) = JUNK(6,2,1); JUNK(7,2,1) = JUNK(8,2,1); JUNK(8,2,1) = JUNK(9,2,1);
JUNK(10,2,1) = JUNK(11,2,1); JUNK(11,2,1) = JUNK(12,2,1);
JUNK(13,2,1) = JUNK(14,2,1); JUNK(14,2,1) = JUNK(15,2,1);
JUNK(16,2,1) = JUNK(17,2,1); JUNK(17,2,1) = JUNK(18,2,1);
JUNK(19,2,1) = JUNK(20,2,1); JUNK(20,2,1) = JUNK(21,2,1);
JUNK(22,2,1) = JUNK(23,2,1); JUNK(23,2,1) = JUNK(24,2,1);
JUNK(25,2,1) = JUNK(26,2,1); JUNK(26,2,1) = JUNK(27,2,1);

JUNK(1,1,1) = JUNK(2,1,1); JUNK(2,1,1) = JUNK(3,1,1); JUNK(3,1,1) = JUNK(4,1,1);
JUNK(4,1,1) = JUNK(5,1,1); JUNK(5,1,1) = JUNK(6,1,1); JUNK(6,1,1) = JUNK(7,1,1);
JUNK(7,1,1) = JUNK(8,1,1); JUNK(8,1,1) = JUNK(9,1,1);
JUNK(10,1,1) = JUNK(11,1,1); JUNK(11,1,1) = JUNK(12,1,1);
JUNK(12,1,1) = JUNK(13,1,1); JUNK(13,1,1) = JUNK(14,1,1);
JUNK(14,1,1) = JUNK(15,1,1); JUNK(15,1,1) = JUNK(16,1,1);
JUNK(16,1,1) = JUNK(17,1,1); JUNK(17,1,1) = JUNK(18,1,1);
JUNK(19,1,1) = JUNK(20,1,1); JUNK(20,1,1) = JUNK(21,1,1);
JUNK(21,1,1) = JUNK(22,1,1); JUNK(22,1,1) = JUNK(23,1,1);
JUNK(23,1,1) = JUNK(24,1,1); JUNK(24,1,1) = JUNK(25,1,1);
JUNK(25,1,1) = JUNK(26,1,1); JUNK(26,1,1) = JUNK(27,1,1);

JUNK(1,2,2) = JUNK(2,2,2); JUNK(2,2,2) = JUNK(3,2,2);
JUNK(4,2,2) = JUNK(5,2,2); JUNK(5,2,2) = JUNK(6,2,2);
JUNK(7,2,2) = JUNK(8,2,2); JUNK(8,2,2) = JUNK(9,2,2);
JUNK(10,2,2) = JUNK(11,2,2); JUNK(11,2,2) = JUNK(12,2,2);
JUNK(13,2,2) = JUNK(14,2,2); JUNK(14,2,2) = JUNK(15,2,2);
JUNK(16,2,2) = JUNK(17,2,2); JUNK(17,2,2) = JUNK(18,2,2);
JUNK(19,2,2) = JUNK(20,2,2); JUNK(20,2,2) = JUNK(21,2,2);
JUNK(22,2,2) = JUNK(23,2,2); OUT(23,2,2) = JUNK(24,2,2);
JUNK(25,2,2) = JUNK(26,2,2); JUNK(26,2,2) = JUNK(27,2,2);

JUNK(1,1,2) = JUNK(2,1,2); JUNK(2,1,2) = JUNK(3,1,2); JUNK(3,1,2) = JUNK(4,1,2);
JUNK(4,1,2) = JUNK(5,1,2); JUNK(5,1,2) = JUNK(6,1,2); JUNK(6,1,2) = JUNK(7,1,2);
JUNK(7,1,2) = JUNK(8,1,2); JUNK(8,1,2) = JUNK(9,1,2);
JUNK(10,1,2) = JUNK(11,1,2); JUNK(11,1,2) = JUNK(12,1,2);
JUNK(12,1,2) = JUNK(13,1,2); JUNK(13,1,2) = JUNK(14,1,2);
JUNK(14,1,2) = JUNK(15,1,2); JUNK(15,1,2) = JUNK(16,1,2);
JUNK(16,1,2) = JUNK(17,1,2); JUNK(17,1,2) = JUNK(18,1,2);
JUNK(19,1,2) = JUNK(20,1,2); JUNK(20,1,2) = JUNK(21,1,2);
JUNK(21,1,2) = JUNK(22,1,2); JUNK(22,1,2) = JUNK(23,1,2);
JUNK(23,1,2) = JUNK(24,1,2); JUNK(24,1,2) = JUNK(25,1,2);
JUNK(25,1,2) = JUNK(26,1,2); JUNK(26,1,2) = JUNK(27,1,2);

JUNK(1,2,3) = JUNK(2,2,3); JUNK(2,2,3) = JUNK(3,2,3);
JUNK(4,2,3) = JUNK(5,2,3); JUNK(5,2,3) = JUNK(6,2,3);
JUNK(7,2,3) = JUNK(8,2,3); JUNK(8,2,3) = JUNK(9,2,3);
JUNK(10,2,3) = JUNK(11,2,3); JUNK(11,2,3) = JUNK(12,2,3);
JUNK(13,2,3) = JUNK(14,2,3); JUNK(14,2,3) = JUNK(15,2,3);
JUNK(16,2,3) = JUNK(17,2,3); JUNK(17,2,3) = JUNK(18,2,3);
JUNK(19,2,3) = JUNK(20,2,3); JUNK(20,2,3) = JUNK(21,2,3);
JUNK(22,2,3) = JUNK(23,2,3); JUNK(23,2,3) = JUNK(24,2,3);
JUNK(25,2,3) = JUNK(26,2,3); JUNK(26,2,3) = JUNK(27,2,3);

JUNK(1,1,3) = JUNK(2,1,3); JUNK(2,1,3) = JUNK(3,1,3); JUNK(3,1,3) = JUNK(4,1,3);
JUNK(4,1,3) = JUNK(5,1,3); JUNK(5,1,3) = JUNK(6,1,3); JUNK(6,1,3) = JUNK(7,1,3);
JUNK(7,1,3) = JUNK(8,1,3); JUNK(8,1,3) = JUNK(9,1,3);
JUNK(10,1,3) = JUNK(11,1,3); JUNK(11,1,3) = JUNK(12,1,3);
JUNK(12,1,3) = JUNK(13,1,3); JUNK(13,1,3) = JUNK(14,1,3);
PEN(1,1,3)=PEN(2,1,3); PEN(2,1,3)=PEN(3,1,3); PEN(3,1,3)=PEN(4,1,3);
PEN(4,1,3)=PEN(5,1,3); PEN(5,1,3)=PEN(6,1,3); PEN(6,1,3)=PEN(7,1,3);
PEN(7,1,3)=PEN(8,1,3); PEN(8,1,3)=PEN(9,1,3);
PEN(10,1,3)=PEN(11,1,3); PEN(11,1,3)=PEN(12,1,3); PEN(12,1,3)=PEN(13,1,3);
PEN(13,1,3)=PEN(14,1,3); PEN(14,1,3)=PEN(15,1,3); PEN(15,1,3)=PEN(16,1,3);
PEN(16,1,3)=PEN(17,1,3); PEN(17,1,3)=PEN(18,1,3);
PEN(19,1,3)=PEN(20,1,3); PEN(20,1,3)=PEN(21,1,3); PEN(21,1,3)=PEN(22,1,3);
PEN(22,1,3)=PEN(23,1,3); PEN(23,1,3)=PEN(24,1,3); PEN(24,1,3)=PEN(25,1,3);
PEN(25,1,3)=PEN(26,1,3); PEN(26,1,3)=PEN(27,1,3);

DATA:
PROB = @OLE('F:\PROB.xls');
RATE2_A = @OLE('F:\RATE Prod.xls');
RATE3_A = @OLE('F:\RATE Prod.xls');
RATE4_A = @OLE('F:\RATE Prod.xls');
RATE2_B = @OLE('F:\RATE Prod.xls');
RATE3_B = @OLE('F:\RATE Prod.xls');
RATE4_B = @OLE('F:\RATE Prod.xls');
DEM2 = 40 40 40;
DEM3 = 40 40 40;
DEM4 = 40 40 40;
Cin = 3 2 2;
Crec = 6 5 7;
Cout = 10 8 11;
Closs = 3 2 2;
Cjunk = 16 14 17;
Cpenalty = 20 18 22;
CAP1 = 50 50 50;
CAP2 = 10 10 10;
ENDDATA

END
Appendix B – Pseudo-code of Monte Carlo Simulation Model

For k = 0 to Replication - 1

'Generate an array of random numbers for the returned products with the specified mean and standard deviation

For j = 0 to Period - 1

TotReturn(j) = ReturnProduct(j) + ExtReturn(j)
If TotReturn(j) <= CapReturnInv Then
    ExtReturn(j + 1) = 0
    WIPReturn(j) = TotReturn(j)
    UnusedReturnInv(j) = CapReturnInv - TotReturn(j)
    OutReturn(j) = 0
    TotCostReturnInv(j) = TotReturn(j) * CostReturnInv
    TotCostReturnOut(j) = 0
Else
    ExtReturn(j + 1) = L * (TotReturn(j) - CapReturnInv)
    UnusedReturnInv(j) = 0
    OutReturn(j) = TotReturn(j) - L * OutReturn(j)
    TotCostReturnInv(j) = CapReturnInv * CostReturnInv
    TotCostReturnOut(j) = OutReturn(j) * CostExtReturnInv
End If

For i = 0 to PartType - 1

'Usable Parts
'Generate an array with random numbers (RandArrayRet) which has the size of value WIPReturn(j)

For m = 0 To (RandArrayRet.Length - 1)
    If RandArrayRet(m) <= UsableYield(i) / 100 Then
        ValUsable = 1
        SumUsable += ValUsable
    Else
        ValUsable = 0
    End If
Next m
Usable(i) = SumUsable
TotUsable(i) = Usable(i) + FinalExtUsable(i, j)
If TotUsable(i) <= CapUsableInv(i) Then
    ExtUsable(i) = 0
    UnusedUsableInv(i) = CapUsableInv(i) - TotUsable(i)
    TotCostUsableInv(i) = TotUsable(i) * CostUsableInv(i)
Else
    ExtUsable(i) = TotUsable(i) - CapUsableInv(i)
    UnusedUsableInv(i) = 0
    TotCostUsableInv(i) = CapUsableInv(i) * CostUsableInv(i)
End If

SumUnusedUsableInv = SumUnusedUsableInv + UnusedUsableInv(i)
Next

'Reconfiguration and external storage decisions for usable parts
For i = 0 to PartType - 1
    If ExtUsable(i) > 0 Then
        If SumUnusedUsableInv > 0 Then

214
If ExtUsable(i) <= SumUnusedUsableInv Then
    TotUsableRecon(i) = ExtUsable(i)
    TotUsableReconCost(i) = TotUsableRecon(i) * CostUsableRecon(i)
    SumUnusedUsableInv -= ExtUsable(i)
    TotUsableOut(i) = 0
    TotUsableOutCost(i) = TotUsableOut(i) * CostUsableOut(i)
    WIPUsable(i) = TotUsable(i)
Else
    TotUsableRecon(i) = SumUnusedUsableInv
    TotUsableReconCost(i) = TotUsableRecon(i) * CostUsableRecon(i)
    SumUnusedUsableInv = 0
    TotUsableOut(i) = ExtUsable(i) - TotUsableRecon(i)
    TotUsableOutCost(i) = TotUsableOut(i) * CostUsableOut(i)
    WIPUsable(i) = TotUsable(i) - L * TotUsableOut(i)
End If
Else
    TotUsableRecon(i) = 0
    TotUsableReconCost(i) = TotUsableRecon(i) * CostUsableRecon(i)
    TotUsableOut(i) = ExtUsable(i)
    TotUsableOutCost(i) = TotUsableOut(i) * CostUsableOut(i)
    WIPUsable(i) = TotUsable(i) - L * ExtUsable(i)
End If
Else
    WIPUsable(i) = TotUsable(i)
    TotUsableRecon(i) = 0
    TotUsableReconCost(i) = TotUsableRecon(i) * CostUsableRecon(i)
    TotUsableOut(i) = 0
    TotUsableOutCost(i) = TotUsableOut(i) * CostUsableOut(i)
End If

FinalExtUsable(i, j + 1) = L * TotUsableOut(i)
Next

For i = 0 to PartType - 1
    'Acceptable parts
    'Generate an array with random numbers (RandArrayUsable) which has the size of value WIPUsable(i)
    For w = 0 To (RandArrayUsable.Length - 1)
        If RandArrayUsable(w) <= AcceptableYield(i) / 100 Then
            ValAcceptable = 1
            SumAcceptable = SumAcceptable + ValAcceptable
        Else
            ValAcceptable = 0
        End If
    Next w

    Acceptable(i) = SumAcceptable
    XAcceptable(i) = WIPUsable(i) - Acceptable(i)

    'Repairable parts
    'Generate an array with random numbers (RandArrayXAcceptable) which has the size of value XAcceptable(i)
    For z = 0 To (RandArrayXAcceptable.Length - 1)
        If RandArrayXAcceptable(z) <= RepairableYield(i) / 100 Then
            ValRepairable = 1
            SumRepairable = SumRepairable + ValRepairable
        Else
            ValRepairable = 0
        End If
    Next z

    Repairable(i) = SumRepairable
    TotRepairable(i) = Repairable(i) + FinalExtRepairable(i, j)

    If TotRepairable(i) <= CapRepairableInv(i) Then
        ExtRepairable(i) = 0
        UnusedRepairableInv(i) = CapRepairableInv(i) - TotRepairable(i)
        TotCostRepairableInv(i) = TotRepairable(i) * CostRepairableInv(i)
    Else
        TotCostRepairableInv(i) = TotRepairable(i) * CostRepairableInv(i)
    End If
Next
Else
    ExtRepairable(i) = TotRepairable(i) - CapRepairableInv(i)
    UnusedRepairableInv(i) = 0
    TotCostRepairableInv(i) = CapRepairableInv(i) * CostRepairableInv(i)
End If

InspectionDisposable(i) = XAcceptable(i) - Repairable(i)
SumUnusedRepairableInv = SumUnusedRepairableInv + UnusedRepairableInv(i)

Next

'Reconfiguration and external storage decisions for repairable parts
For i = 0 to PartType - 1
    If ExtRepairable(i) > 0 Then
        If SumUnusedRepairableInv > 0 Then
            If ExtRepairable(i) <= SumUnusedRepairableInv Then
                TotRepairableRecon(i) = ExtRepairable(i)
                TotRepairableReconCost(i) = TotRepairableRecon(i) * CostRepairableReconA(i)
                SumUnusedRepairableInv -= ExtRepairable(i)
                WIPRepairable(i) = TotRepairable(i)
                TotRepairableReconUsable(i) = 0
                TotRepairableReconCostUsable(i) = TotRepairableReconUsable(i) * CostRepairableReconB(i)
                TotRepairableOut(i) = 0
                TotRepairableOutCost(i) = TotRepairableOut(i) * CostRepairableOut(i)
            Else
                ExtRepairable(i) -= SumUnusedRepairableInv
                SumUnusedRepairableInv = 0
                TotRepairableRecon(i) = SumUnusedRepairableInv
                TotRepairableReconCost(i) = TotRepairableRecon(i) * CostRepairableReconA(i)
                If SumUnusedUsableInv > 0 Then
                    If ExtRepairable(i) <= SumUnusedUsableInv Then
                        TotRepairableReconUsable(i) = ExtRepairable(i)
                        TotRepairableReconCostUsable(i) = TotRepairableReconUsable(i) * CostRepairableReconB(i)
                        SumUnusedUsableInv -= ExtRepairable(i)
                        WIPRepairable(i) = TotRepairable(i) - P * TotRepairableReconUsable(i)
                        TotRepairableOut(i) = 0
                        TotRepairableOutCost(i) = TotRepairableOut(i) * CostRepairableOut(i)
                    Else
                        TotRepairableReconUsable(i) = SumUnusedUsableInv
                        TotRepairableReconCostUsable(i) = TotRepairableReconUsable(i) * CostRepairableReconB(i)
                        SumUnusedUsableInv = 0
                        TotRepairableOut(i) = ExtRepairable(i) - TotRepairableReconUsable(i)
                        TotRepairableOutCost(i) = TotRepairableOut(i) * CostRepairableOut(i)
                        WIPRepairable(i) = TotRepairable(i) - P * TotRepairableReconUsable(i) - L * TotRepairableOut(i)
                    End If
                Else
                    TotRepairableReconUsable(i) = 0
                    TotRepairableReconCostUsable(i) = TotRepairableReconUsable(i) * CostRepairableReconB(i)
                    TotRepairableOut(i) = ExtRepairable(i)
                    WIPRepairable(i) = TotRepairable(i) - L * TotRepairableOut(i)
                    TotRepairableOutCost(i) = TotRepairableOut(i) * CostRepairableOut(i)
                End If
            End If
        End If
    End If
End If
Appendix B

\[
\begin{align*}
\text{TotRepairableReconUsable}(i) &= \text{ExtRepairable}(i) \\
\text{TotRepairableReconCostUsable}(i) &= \text{TotRepairableReconUsable}(i) \times \text{CostRepairableReconB}(i) \\
\text{WIPRepairable}(i) &= \text{TotRepairable}(i) - P \times \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOut}(i) &= 0 \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOut}(i) \times \text{CostRepairableOut}(i) \\
\text{TotRepairableOutCostUsable}(i) &= \text{TotRepairableOutCostUsable}(i) \times \text{CostRepairableOut}(i) \\
\text{ExtRepairable}(i) &= \text{ExtRepairable}(i) \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOutCost}(i) \times \text{CostRepairableOut}(i) \\
\text{WIPRepairable}(i) &= \text{TotRepairable}(i) - P \times \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOut}(i) &= \text{ExtRepairable}(i) - \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOutCost}(i) \times \text{CostRepairableOut}(i) \\
\text{WIPRepairable}(i) &= \text{TotRepairable}(i) - P \times \text{TotRepairableReconUsable}(i) - L \times \text{TotRepairableOut}(i) \\
\text{FinalExtRepairable}(i, j + 1) &= P \times \text{TotRepairableReconUsable}(i) + L \times \text{TotRepairableOut}(i) \\
\text{Next} \\
\text{For } i = 0 \text{ to PartType} - 1 \\
\text{'Repaired parts} \\
\text{'Generate an array with random numbers (RandArrayRepairable) which has the size of value WIPRepairable(i)'} \\
\text{For } g = 0 \text{ To (RandArrayRepairable.Length} - 1) \\
\text{If } \text{RandArrayRepairable}(g) \leq \text{RepairedYield}(i) / 100 \text{ Then} \\
\text{ValRepaired} &= 1 \\
\text{SumRepaired} &= \text{SumRepaired} + \text{ValRepaired} \\
\text{Else} \\
\text{ValRepaired} &= 0 \\
\text{End If} \\
\text{Next } g \\
\text{Repaired}(i) &= \text{SumRepaired} \\
\text{RepairDisposable}(i) &= \text{WIPRepairable}(i) - \text{Repaired}(i) \\
\text{'Disposable parts} \\
\text{Disposable}(i) &= \text{WIPReturn}(j) - \text{Usable}(i) \\
\text{TotDisposable}(i) &= \text{Disposable}(i) + \text{FinalExtDisposable}(i, j) + \text{InspectionDisposable}(i) + \text{RepairDisposable}(i) \\
\text{If } \text{TotDisposable}(i) \leq \text{CapDisposableInv}(i) \text{ Then} \\
\text{ExtDisposable}(i) &= 0 \\
\text{UnusedDisposableInv}(i) &= \text{CapDisposableInv}(i) - \text{TotDisposable}(i) \\
\text{TotCostDisposableInv}(i) &= \text{TotDisposable}(i) \times \text{CostDisposableInv}(i) \\
\text{End If} \\
\text{Else} \\
\text{WIPRepairable}(i) &= 0 \\
\text{TotRepairableReconUsable}(i) &= 0 \\
\text{TotRepairableReconCostUsable}(i) &= \text{TotRepairableReconUsable}(i) \times \text{CostRepairableReconB}(i) \\
\text{TotRepairableOut}(i) &= \text{ExtRepairable}(i) - \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOut}(i) \times \text{CostRepairableOut}(i) \\
\text{WIPRepairable}(i) &= \text{TotRepairable}(i) - L \times \text{TotRepairableOut}(i) \\
\text{End If} \\
\text{Else} \\
\text{WIPRepairable}(i) &= 0 \\
\text{TotRepairableReconUsable}(i) &= 0 \\
\text{TotRepairableReconCostUsable}(i) &= \text{TotRepairableReconUsable}(i) \times \text{CostRepairableReconB}(i) \\
\text{TotRepairableOut}(i) &= \text{ExtRepairable}(i) - \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOut}(i) \times \text{CostRepairableOut}(i) \\
\text{WIPRepairable}(i) &= \text{TotRepairable}(i) - L \times \text{TotRepairableOut}(i) \\
\text{End If} \\
\text{Else} \\
\text{WIPRepairable}(i) &= 0 \\
\text{TotRepairableReconUsable}(i) &= 0 \\
\text{TotRepairableReconCostUsable}(i) &= \text{TotRepairableReconUsable}(i) \times \text{CostRepairableReconB}(i) \\
\text{TotRepairableOut}(i) &= \text{ExtRepairable}(i) - \text{TotRepairableReconUsable}(i) \\
\text{TotRepairableOutCost}(i) &= \text{TotRepairableOut}(i) \times \text{CostRepairableOut}(i) \\
\text{End If} \\
\text{End If} \\
\text{End If} \\
\text{End If} \\
\text{End If} \\
\text{End If} \\
\text{Next} \\
\text{For } i = 0 \text{ to PartType} - 1 \\
\text{Next } i \\
\text{FinalExtRepairable}(i, j + 1) &= P \times \text{TotRepairableReconUsable}(i) + L \times \text{TotRepairableOut}(i)
Else
    ExtDisposable(i) = TotDisposable(i) - CapDisposableInv(i)
    UnusedDisposableInv(i) = 0
    TotCostDisposableInv(i) = CapDisposableInv(i) * CostDisposableInv(i)
End If
SumUnusedDisposableInv = SumUnusedDisposableInv + UnusedDisposableInv(i)
Next

'Reconfiguration and external storage decisions for disposable parts
For i = 0 to PartType - 1
    If ExtDisposable(i) > 0 Then
        If SumUnusedDisposableInv > 0 Then
            If ExtDisposable(i) <= SumUnusedDisposableInv Then
                TotDisposableRecon(i) = ExtDisposable(i)
                TotDisposableReconCost(i) = TotDisposableRecon(i) * CostDisposableReconA(i)
                SumUnusedDisposableInv -= ExtDisposable(i)
                TotDisposableOut(i) = 0
                TotDisposableOutCost(i) = TotDisposableOut(i) * CostDisposableOut(i)
            Else
                TotDisposableRecon(i) = SumUnusedDisposableInv
                TotDisposableReconCost(i) = TotDisposableRecon(i) * CostDisposableReconA(i)
                SumUnusedDisposableInv = 0
                TotDisposableOut(i) = ExtDisposable(i) - TotDisposableRecon(i)
                TotDisposableOutCost(i) = TotDisposableOut(i) * CostDisposableOut(i)
            End If
        Else
            TotDisposableRecon(i) = 0
            TotDisposableReconCost(i) = TotDisposableRecon(i) * CostDisposableReconA(i)
            TotDisposableOut(i) = ExtDisposable(i)
            TotDisposableOutCost(i) = TotDisposableOut(i) * CostDisposableOut(i)
        End If
    Else
        TotDisposableRecon(i) = 0
        TotDisposableReconCost(i) = TotDisposableRecon(i) * CostDisposableReconA(i)
        TotDisposableOut(i) = 0
        TotDisposableOutCost(i) = TotDisposableOut(i) * CostDisposableOut(i)
    End If
    FinalExtDisposable(i, j + 1) = L * TotDisposableOut(i)
Next

'Finished parts
For i = 0 to PartType - 1
    Finished(i) = Acceptable(i) + Repaired(i)
    TotFinished(i) = Finished(i) + FinalExtFinished(i, j) + XExtFinished(i, j)
    PartDemand(i) = Demand(j) * PartNumber(i)
    If TotFinished(i) <= CapFinishedInv(i) Then
        ExtFinished(i) = 0
        UnusedFinishedInv(i) = CapFinishedInv(i) - TotFinished(i)
        TotCostFinishedInv(i) = TotFinished(i) * CostFinishedInv(i)
    Else
        ExtFinished(i) = TotFinished(i) - CapFinishedInv(i)
        UnusedFinishedInv(i) = 0
        TotCostFinishedInv(i) = CapFinishedInv(i) * CostFinishedInv(i)
    End If
    SumUnusedFinishedInv = SumUnusedFinishedInv + UnusedFinishedInv(i)
Next

'Reconfiguration and external storage decisions for finished parts
For i = 0 to PartType - 1
    If ExtFinished(i) > 0 Then
        If SumUnusedFinishedInv > 0 Then
            If ExtFinished(i) <= SumUnusedFinishedInv Then
                FinalExtFinished(i, j + 1) = L * ExtFinished(i)
            Else
                FinalExtFinished(i, j + 1) = L * SumUnusedFinishedInv
            End If
        End If
Next
TotFinishedRecon(i) = ExtFinished(i)
TotFinishedReconCost(i) = TotFinishedRecon(i) * CostFinishedRecon(i)
SumUnusedFinishedInv -= ExtFinished(i)
TotFinishedOut(i) = 0
TotFinishedOutCost(i) = TotFinishedOut(i) * CostFinishedOut(i)
WIPFinished(i) = TotFinished(i)

Else
TotFinishedRecon(i) = SumUnusedFinishedInv
TotFinishedReconCost(i) = TotFinishedRecon(i) * CostFinishedRecon(i)
SumUnusedFinishedInv = 0
TotFinishedOut(i) = ExtFinished(i) - TotFinishedRecon(i)
TotFinishedOutCost(i) = TotFinishedOut(i) * CostFinishedOut(i)
WIPFinished(i) = TotFinished(i) - L * TotFinishedOut(i)
End If
Else
TotFinishedRecon(i) = 0
TotFinishedReconCost(i) = TotFinishedRecon(i) * CostFinishedRecon(i)
TotFinishedOut(i) = ExtFinished(i)
TotFinishedOutCost(i) = TotFinishedOut(i) * CostFinishedOut(i)
WIPFinished(i) = TotFinished(i) - L * ExtFinished(i)
End If
Else
WIPFinished(i) = TotFinished(i)
TotFinishedRecon(i) = 0
TotFinishedReconCost(i) = TotFinishedRecon(i) * CostFinishedRecon(i)
TotFinishedOut(i) = 0
TotFinishedOutCost(i) = TotFinishedOut(i) * CostFinishedOut(i)
End If

FinalExtFinished(i, j + 1) = L * TotFinishedOut(i)

If WIPFinished(i) < PartDemand(i) Then
NewPartDemand(i) = PartDemand(i) - WIPFinished(i)
XExtFinished(i, j + 1) = 0
Else
NewPartDemand(i) = 0
XExtFinished(i, j + 1) = WIPFinished(i) - PartDemand(i)
End If

' New parts
If R = 1 Then
If NewPartDemand(i) > FinalExtNewPart(i, j) Then
PartDemandToTotFinishedRatio(i) = Math.Ceiling((NewPartDemand(i) -
FinalExtNewPart(i, j)) / NewPartBatchSize(i))
NewPartOrder(i) = NewPartBatchSize(i) * PartDemandToTotFinishedRatio(i)
Else
NewPartOrder(i) = 0
End If
Else
NewPartOrder(i) = 0
End If
If NewPartOrder(i) <= CapNewPartInv(i) Then
UnusedNewPartInv(i) = CapNewPartInv(i) - NewPartOrder(i)
ExtNewPart(i) = 0
TotCostNewPartInv(i) = NewPartOrder(i) * CostNewPartInv(i)
Else
UnusedNewPartInv(i) = 0
ExtNewPart(i) = NewPartOrder(i) - CapNewPartInv(i)
TotCostNewPartInv(i) = CapNewPartInv(i) * CostNewPartInv(i)
End If

SumUnusedNewPartInv = SumUnusedNewPartInv + UnusedNewPartInv(i)

' Reconfiguration and external storage decisions for new parts
For i = 0 to PartType - 1
If ExtNewPart(i) > 0 Then
If SumUnusedNewPartInv > 0 Then
If ExtNewPart(i) <= SumUnusedNewPartInv Then
    TotNewPartRecon(i) = ExtNewPart(i)
    TotNewPartReconCost(i) = TotNewPartRecon(i) * CostNewPartReconA(i)
    WIPNewPart(i) = NewPartOrder(i)
    TotNewPartReconFinished(i) = 0
    TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
    TotNewPartOut(i) = 0
    TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i)
Else
    ExtNewPart(i) -= SumUnusedNewPartInv
    SumUnusedNewPartInv = 0
    TotNewPartRecon(i) = SumUnusedNewPartInv
    TotNewPartReconCost(i) = TotNewPartRecon(i) * CostNewPartReconA(i)
    If SumUnusedFinishedInv > 0 Then
        If ExtNewPart(i) <= SumUnusedFinishedInv Then
            TotNewPartReconFinished(i) = ExtNewPart(i)
            TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
            SumUnusedFinishedInv -= ExtNewPart(i)
            WIPNewPart(i) = NewPartOrder(i) - P * TotNewPartReconFinished(i)
            TotNewPartOut(i) = 0
            TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i)
        Else
            TotNewPartReconFinished(i) = SumUnusedFinishedInv
            TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
            SumUnusedFinishedInv = 0
            TotNewPartOut(i) = ExtNewPart(i) - TotNewPartReconFinished(i)
            TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i)
            WIPNewPart(i) = NewPartOrder(i) - P * TotNewPartReconFinished(i) - L * TotNewPartOut(i)
        End If
    Else
        TotNewPartReconFinished(i) = 0
        TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
        TotNewPartOut(i) = ExtNewPart(i)
        TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i)
        WIPNewPart(i) = NewPartOrder(i) - L * TotNewPartOut(i)
    End If
End If
Else
    TotNewPartRecon(i) = 0
    TotNewPartReconCost(i) = TotNewPartRecon(i) * CostNewPartReconA(i)
    If SumUnusedFinishedInv > 0 Then
        If ExtNewPart(i) <= SumUnusedFinishedInv Then
            TotNewPartReconFinished(i) = ExtNewPart(i)
            TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
            SumUnusedFinishedInv -= ExtNewPart(i)
            WIPNewPart(i) = NewPartOrder(i) - P * TotNewPartReconFinished(i)
        Else
            TotNewPartReconFinished(i) = SumUnusedFinishedInv
            TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
            SumUnusedFinishedInv = 0
            TotNewPartOut(i) = ExtNewPart(i) - TotNewPartReconFinished(i)
            TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i)
            WIPNewPart(i) = NewPartOrder(i) - P * TotNewPartReconFinished(i) - L * TotNewPartOut(i)
        End If
    Else
        TotNewPartReconFinished(i) = 0
        TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i)
    End If
Appendix B

```plaintext
CostNewPartReconB(i) 
TotNewPartOut(i) = ExtNewPart(i) 
TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i) 
WIPNewPart(i) = NewPartOrder(i) - L * TotNewPartOut(i) 
End If 
End If 
Else 
TotNewPartRecon(i) = 0 
TotNewPartReconCost(i) = TotNewPartRecon(i) * CostNewPartReconA(i) 
TotNewPartReconFinished(i) = 0 
TotNewPartReconCostFinished(i) = TotNewPartReconFinished(i) * CostNewPartReconB(i) 
TotNewPartOut(i) = 0 
TotNewPartOutCost(i) = TotNewPartOut(i) * CostNewPartOut(i) 
WIPNewPart(i) = NewPartOrder(i) 
End If 

FinalExtNewPart(i, j + 1) = NewPartOrder(i) - P * TotNewPartReconFinished(i) - L * 
TotNewPartOut(i) - NewPartDemand(i) 
Next 

'Remanufactured products 
RemanufacturedProducts(j) = Demand(j) 
If RemanufacturedProducts(j) < CapRemanufacturedInv Then 
TotRemanufacturedInvCost(j) = RemanufacturedProducts(j) * CostRemanufacturedInv 
TotRemanufacturedOut(j) = 0 
TotRemanufacturedOutCost(j) = 0 
UnusedRemanufacturedInv(j) = CapRemanufacturedInv - RemanufacturedProducts(j) 
Else 
TotRemanufacturedInvCost(j) = CapRemanufacturedInv * CostRemanufacturedInv 
TotRemanufacturedOut(j) = RemanufacturedProducts(j) - CapRemanufacturedInv 
TotRemanufacturedOutCost(j) = TotRemanufacturedOut(j) * RemanufacturedOutCost 
UnusedRemanufacturedInv(j) = 0 
End If 
For i = 0 to PartType - 1 

TotInCostPart(i) = TotCostUsableInv(i) + TotCostDisposableInv(i) + 
TotCostRepairableInv(i) + TotCostFinishedInv(i) + TotCostNewPartInv(i) + 
TotCostExtNewPartInv(i) 
TotRecCostPart(i) = TotCostUsableRecon(i) + TotDisposableReconCost(i) + 
TotDisposableReconCostReturn(i) + TotRepairableReconCost(i) + 
TotRepairableReconCostUsable(i) + TotFinishedReconCost(i) + 
TotNewPartReconCost(i) + 
TotNewPartReconCostFinished(i) 
TotOutCostPart(i) = TotUsableOutCost(i) + TotDisposableOutCost(i) + 
TotRepairableOutCost(i) + TotFinishedOutCost(i) + 
TotNewPartOutCost(i) 

TotUnusedPart = SumUnusedUsableInv + SumUnusedDisposableInv + SumUnusedRepairableInv + 
SumUnusedFinishedInv + SumUnusedNewPartInv 
Next 

TotInCost(j,k) = FindTotPart(TotInCostPart) + TotReturnInvCost(j) + 
TotRemanufacturedInvCost(j) 
TotOutCost(j,k) = FindTotPart(TotOutCostPart) + TotReturnOutCost(j) + 
TotRemanufacturedOutCost(j) 
TotReconCost(j,k) = FindTotPart(TotReconCostPart) 
TotUnused(j,k) = TotUnusedPart + UnusedReturnInv[j] + 
UnusedRemanufacturedInv[j] 
TotCost(j,k) = TotInCost(j,k) + TotOutCost(j,k) + TotReconCost(j,k) + 
OpportunityCost * TotUnused(j,k) 
Next (period) 
Next (replication) 
```
## Appendix C – Numerical Results from DOE Analysis

### Table C.1 Estimated effects and coefficients for total cost - Low difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>7308.43</td>
<td>10.05</td>
<td>726.99</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>202.47</td>
<td>101.24</td>
<td>10.05</td>
<td>10.07</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>-173.37</td>
<td>-86.69</td>
<td>10.05</td>
<td>-8.62</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-88.70</td>
<td>-44.35</td>
<td>10.05</td>
<td>-4.41</td>
<td>0.002</td>
</tr>
<tr>
<td><strong>Repairable Cap</strong></td>
<td><strong>-6.25</strong></td>
<td><strong>-3.12</strong></td>
<td><strong>10.05</strong></td>
<td><strong>-0.31</strong></td>
<td><strong>0.764</strong></td>
</tr>
<tr>
<td>New Cap</td>
<td>407.77</td>
<td>203.89</td>
<td>10.05</td>
<td>20.28</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>1334.52</td>
<td>667.26</td>
<td>10.05</td>
<td>66.37</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Remanufactured Cap</strong></td>
<td><strong>11.10</strong></td>
<td><strong>5.55</strong></td>
<td><strong>10.05</strong></td>
<td><strong>0.55</strong></td>
<td><strong>0.596</strong></td>
</tr>
</tbody>
</table>

### Table C.2 Analysis of variance for total cost - Low difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>7</td>
<td>8105290</td>
<td>1157899</td>
<td>716.07</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>8</td>
<td>12936</td>
<td>1617</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>8118226</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S = 40.2121  
PRESS = 51744.5  
R-Sq = 99.84%  
R-Sq(pred) = 99.36%  
R-Sq(adj) = 99.70%

### Table C.3 Estimated effects and coefficients for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>7303.26</td>
<td>24.06</td>
<td>303.55</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>463.21</td>
<td>231.61</td>
<td>9.63</td>
<td>46.20</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>392.24</td>
<td>196.12</td>
<td>8.15</td>
<td>46.20</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-156.26</td>
<td>-78.13</td>
<td>-3.25</td>
<td>0.012</td>
<td>0.761</td>
</tr>
<tr>
<td><strong>Repairable Cap</strong></td>
<td><strong>-15.16</strong></td>
<td><strong>-7.58</strong></td>
<td><strong>24.06</strong></td>
<td><strong>-0.32</strong></td>
<td><strong>0.761</strong></td>
</tr>
<tr>
<td>New Cap</td>
<td>683.34</td>
<td>341.67</td>
<td>14.20</td>
<td>46.20</td>
<td>0.000</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>1413.71</td>
<td>706.86</td>
<td>29.38</td>
<td>46.20</td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Remanufactured Cap</strong></td>
<td><strong>-27.39</strong></td>
<td><strong>-13.69</strong></td>
<td><strong>24.06</strong></td>
<td><strong>-0.57</strong></td>
<td><strong>0.585</strong></td>
</tr>
</tbody>
</table>
### Table C.4 Analysis of variance for total cost - Moderate difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>7</td>
<td>11437389</td>
<td>1633913</td>
<td>716.42</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>8</td>
<td>74094</td>
<td>9262</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>11511483</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
S = 96.2379 \quad \text{PRESS} = 296375
\]

\[
\text{R-Sq} = 99.36\% \quad \text{R-Sq(pred)} = 97.43\% \quad \text{R-Sq(adj)} = 98.79\%
\]

### Table C.5 Estimated effects and coefficients for total cost - High difference in yield rates

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>Coefficient</th>
<th>SE Coefficient</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td></td>
<td>8269.0</td>
<td>14.08</td>
<td>587.17</td>
<td>0.000</td>
</tr>
<tr>
<td>Return Cap</td>
<td>513.7</td>
<td>256.8</td>
<td>14.08</td>
<td>18.24</td>
<td>0.000</td>
</tr>
<tr>
<td>Usable Cap</td>
<td>344.0</td>
<td>172.0</td>
<td>14.08</td>
<td>12.21</td>
<td>0.000</td>
</tr>
<tr>
<td>Disposable Cap</td>
<td>-694.7</td>
<td>-347.3</td>
<td>14.08</td>
<td>-24.66</td>
<td>0.000</td>
</tr>
<tr>
<td>Repairable Cap</td>
<td>-56.5</td>
<td>-28.2</td>
<td>14.08</td>
<td>-2.00</td>
<td>0.080</td>
</tr>
<tr>
<td>New Cap</td>
<td>80.6</td>
<td>40.3</td>
<td>14.08</td>
<td>2.86</td>
<td>0.021</td>
</tr>
<tr>
<td>Finished Cap</td>
<td>2723.4</td>
<td>1361.7</td>
<td>14.08</td>
<td>96.69</td>
<td>0.000</td>
</tr>
<tr>
<td>Remanufactured Cap</td>
<td>-58.9</td>
<td>-29.4</td>
<td>14.08</td>
<td>-2.09</td>
<td>0.070</td>
</tr>
</tbody>
</table>

### Table C.6 Analysis of variance for total cost - High difference in yield rates

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Effects</td>
<td>7</td>
<td>33178417</td>
<td>4739774</td>
<td>1493.68</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>8</td>
<td>25386</td>
<td>3173</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
<td>33203803</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
S = 56.3312 \quad \text{PRESS} = 101543
\]

\[
\text{R-Sq} = 99.92\% \quad \text{R-Sq(pred)} = 99.69\% \quad \text{R-Sq(adj)} = 99.86\%
\]