REPAIR PRIORITIZATION WITH RESPECT TO INVENTORY REQUIREMENTS

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

LAUREN HIMMELBERG

Dr. Ronald McGarvey, Thesis Co-Supervisor

Dr. James Noble, Thesis Co-Supervisor

MAY 2016
The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

REPAIR PRIORITIZATION WITH RESPECT TO INVENTORY REQUIREMENTS

presented by Lauren Himmelberg
candidate for the degree of master of science
and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Ronald McGarvey

Dr. James Noble

Dr. Srinath Gopalakrishna
DEDICATION

I would like to dedicate this research to my friends and family. With their love and support, I was able to achieve my true potential. They encouraged me when I lost my way and believed in me when I had trouble believing in myself. Thank you to my parents, Dale and Melanie, who instilled in me the importance of hard work and the curiosity to never stop learning. Especially thank you to Adrian, without you, none of this would be possible. You have been my rock throughout this entire process, supporting me every step of the way. Thank you.
ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to everyone who has supported me throughout my years in graduate school. I would like to especially thank my thesis advisors, Dr. Ron McGarvey and Dr. James Noble. Their guidance and support throughout the thesis process has been tremendous. Their insightful questions, constructive feedback, and technical help were greatly appreciated. Without Dr. McGarvey and Dr. Noble, this research would not have been possible.

I would also like to thank the Center of Excellence in Logistics and Distribution (CELDi) research team. I would especially like to thank Jason Robke and Kyle Dorge. Jason tirelessly worked with me to ensure I had everything I might need to complete this research. Kyle was always willing to help and provided much needed assistance in reviewing the mathematical programming model. I am exceedingly grateful for the tremendous support the entire CELDi team provided throughout this research.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ................................................................................................. ii

TABLE OF CONTENTS ................................................................................................. iii

LIST OF TABLES .......................................................................................................... vii

LIST OF ILLUSTRATIONS ............................................................................................ viii

LIST OF ABBREVIATIONS ............................................................................................ x

ABSTRACT .................................................................................................................. xi

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 Reparable Parts Supply Chain</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Problem Statement</td>
<td>8</td>
</tr>
<tr>
<td>1.3 Study Objectives</td>
<td>9</td>
</tr>
<tr>
<td>1.3.1 Issue Effectiveness</td>
<td>9</td>
</tr>
<tr>
<td>1.3.2 Sub-System Availability</td>
<td>10</td>
</tr>
<tr>
<td>1.4 Business Motivation</td>
<td>11</td>
</tr>
<tr>
<td>1.5 Scope of Study</td>
<td>12</td>
</tr>
<tr>
<td>1.6 Summary</td>
<td>12</td>
</tr>
<tr>
<td>2. Literature Review</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Steady-State Models</td>
<td>15</td>
</tr>
</tbody>
</table>
2.3 Non Steady-State Models ................................................................................. 17
2.4 Repair Scheduling and Prioritization ............................................................ 18
2.5 Repair Deferrals .............................................................................................. 20
2.6 Real-Time Decision-Making with Repair Limitations ................................. 21
2.7 Summary .......................................................................................................... 23

3. Model Formulation .......................................................................................... 26
   3.1 Introduction ..................................................................................................... 26
   3.2 Sets .................................................................................................................. 30
   3.3 Parameters ...................................................................................................... 33
   3.4 Decision Variables .......................................................................................... 36
   3.5 Objectives ....................................................................................................... 40
      3.5.1 Issue Effectiveness .................................................................................. 43
      3.5.2 Sub-System Availability ........................................................................... 45
   3.6 Constraints ...................................................................................................... 47
      3.6.1 Budget ...................................................................................................... 47
      3.6.2 Repair Capacity ....................................................................................... 48
      3.6.3 Flow Balance Constraints ....................................................................... 48
      3.6.4 Fixed Variables and Variable Restrictions .............................................. 51
   3.7 Model Customizations ................................................................................... 51
      3.7.1 Model Element Redefinition and Model Reformulation ....................... 51
      3.7.2 Fixed Variables ....................................................................................... 53
5.4 Case Study Model ................................................................. 87
  5.4.1 Base Model Results .......................................................... 87
  5.4.2 Budget Size ...................................................................... 91
  5.4.3 Budget Period Length ....................................................... 94
  5.4.4 Repair Recommendations Output .................................... 96
5.5 Summary ............................................................................. 98

6. Conclusions ............................................................................. 100
  6.1 Contribution ......................................................................... 100
  6.2 Future Work / Room for Expansion ..................................... 101
  6.3 Summary ............................................................................. 102

REFERENCES ............................................................................. 104
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Variable Restrictions</td>
<td>51</td>
</tr>
<tr>
<td>2. Potential Parameter Definitions</td>
<td>52</td>
</tr>
<tr>
<td>3. Potential Fixed Decision Variables</td>
<td>54</td>
</tr>
<tr>
<td>4. Set Data Requirements</td>
<td>57</td>
</tr>
<tr>
<td>5. Parameter Data Requirements &amp; Potential Sources</td>
<td>58</td>
</tr>
<tr>
<td>6. Current State of System Parameter Data Requirements</td>
<td>59</td>
</tr>
<tr>
<td>7. Parameter Modifications</td>
<td>73</td>
</tr>
<tr>
<td>8. Decision Variable Modifications</td>
<td>74</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Repair Cycle Diagram</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>Part-Breakage Decision Point</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>Part Finishes Repair Decision Point</td>
<td>5</td>
</tr>
<tr>
<td>4.</td>
<td>Replacement Part Decision Point</td>
<td>6</td>
</tr>
<tr>
<td>5.</td>
<td>Days per Budget Period Calculations Example</td>
<td>64</td>
</tr>
<tr>
<td>6.</td>
<td>Weeks per Budget Period Calculations Example</td>
<td>68</td>
</tr>
<tr>
<td>7.</td>
<td>Last Day of Week</td>
<td>68</td>
</tr>
<tr>
<td>8.</td>
<td>Case Study Repair Cycle Diagram</td>
<td>72</td>
</tr>
<tr>
<td>9.</td>
<td>Time-Based Issue Effectiveness for Pavg Values</td>
<td>80</td>
</tr>
<tr>
<td>10.</td>
<td>Summary: Time-Based Issue Effectiveness for Pavg Values</td>
<td>80</td>
</tr>
<tr>
<td>11.</td>
<td>FSL Issue Effectiveness for Pavg Values</td>
<td>81</td>
</tr>
<tr>
<td>12.</td>
<td>Summary: FSL Issue Effectiveness for Pavg Values</td>
<td>81</td>
</tr>
<tr>
<td>13.</td>
<td>Time-Based Issue Effectiveness for Pmin Values</td>
<td>82</td>
</tr>
<tr>
<td>14.</td>
<td>Summary: Time-Based Issue Effectiveness for Pmin Values</td>
<td>83</td>
</tr>
<tr>
<td>15.</td>
<td>FSL Issue Effectiveness for Pmin Values</td>
<td>83</td>
</tr>
<tr>
<td>16.</td>
<td>Summary: FSL Issue Effectiveness for Pmin Values</td>
<td>84</td>
</tr>
<tr>
<td>17.</td>
<td>Time-Based Issue Effectiveness for Spltmin Values</td>
<td>85</td>
</tr>
<tr>
<td>18.</td>
<td>FSL Issue Effectiveness for Spltmin Values</td>
<td>85</td>
</tr>
</tbody>
</table>
19. Issue Effectiveness for Varying Objective Function

Parameter Values ...................................................................................................................86

20. Summary: Selected Objective Function Component Weighting ..................................86

21. Average Size of Part Flows by Month .........................................................................88

22. Average Part Fill Rate Over Time .................................................................................88

23. Average Issue Effectiveness by Month .........................................................................89

24. Average Issue Effectiveness over Time ........................................................................90

25. Potential Unavailable Sub-Systems Over Time ...........................................................90

26. Budget Size: Average Part Fill Rate Over Time ..........................................................92

27. Budget Size: Average Issue Effectiveness by Month .....................................................92

28. Budget Size: Potential Unavailable Sub-Systems Over Time .......................................93

29. Budget Size: Fiscal Year Repair Spend .........................................................................94

30. Budget Period Length: Average Size of Repairs by Month ........................................95

31. Budget Period Length: Average Part Fill Rate Over Time ..........................................96

32. Example Weekly Repair Recommendations ..................................................................97

33. Example Repair Recommendation Priority List ..............................................................98
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FSL</td>
<td>Field Stocking Location</td>
</tr>
<tr>
<td>GAMS</td>
<td>General Algebraic Modeling System</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed Integer Program</td>
</tr>
<tr>
<td>MTBDR</td>
<td>Mean Time Between Depot Repairs</td>
</tr>
<tr>
<td>NRTS</td>
<td>Not Repairable This Station</td>
</tr>
<tr>
<td>SKU</td>
<td>Stock Keeping Unit</td>
</tr>
<tr>
<td>TSL</td>
<td>Target Stock Level</td>
</tr>
</tbody>
</table>
REPAIR PRIORITIZATION WITH RESPECT TO INVENTORY REQUIREMENTS

Lauren Himmelberg

Dr. Ronald McGarvey, Thesis Co-Supervisor

Dr. James Noble, Thesis Co-Supervisor

ABSTRACT

In a multi-echelon service parts supply chain with limited repair resources, real-time repair decisions have significant impacts on supply chain performance. Inducting into repair the appropriate breadth and depth of components is essential to meeting short-term and long-term customer service level constraints. Previous research in this area primarily focuses on steady-state supply chains assuming infinite repair capabilities and the repair induction of all non-serviceable parts. This thesis develops a Mixed Integer Linear Program (MIP) which considers the current state of the entire repair system as well as forecasted part-breakages to produce strategic real-time repair recommendations. Repair inductions are prioritized to maximize stock location service levels over multiple budget periods through comparative fill rates encompassing both issue effectiveness and sub-system availability. Multiple model runs were completed to determine objective function parameter specifications that best align with these overall system goals. Model output includes daily or weekly repair recommendations, a prioritized list of repair inductions, and projected supply chain performance for issue effectiveness and sub-system availability.
CHAPTER 1. INTRODUCTION

1.1 Background

The multi-echelon service parts supply chain has long been an area of interest for researchers. Much of the literature focuses on steady-state supply chains and the determination of stock levels to minimize stock-outs and backorders within the system. A less popular area is how to make strategic, real-time decisions in the repair supply chain. Often simplifying assumptions are made such as infinite repair capacity and unlimited repair budget. If these limitations cannot be ignored, a company must prioritize repairs so as to meet any customer service level constraints.

A common assumption is that non-serviceable parts that can be economically repaired will be repaired. This is not always the case. Sometimes parts should not or cannot be repaired. The focus of this thesis is how to handle repair deferrals when limitations exist on the repair cycle. Budgets and limited repair depot capabilities can directly impact repair scheduling. In that case, a company’s decision of what to repair today so as to account appropriately for an uncertain future is especially important. In this thesis, an integrated decision model is developed to make dynamic repair recommendations based upon the current state of the system, future demand, and service level constraints.

1.1.1 Reparable Parts Supply Chain

The general repair cycle process is similar across industries. In industries where purchasing new components can be extremely expensive, it becomes economically
advantageous to consider repairs. Many factors are considered when determining whether
a part is categorized as consumable or reparable.

A consumable part is a part that is not repaired if it breaks. The decision not to
repair could be due to one or more factors including: the part is designed for one-time use
(use until breaks and not repair), part cannot be economically repaired, and part cannot be
repaired to desired safety level or tolerance.

A reparable part is a part that will be considered for repair when it breaks. Though
considered for repair, a reparable part is not required to be repaired. Limited resources
may cause reparable parts that break to wait indefinitely to be scheduled for repair.
Throughout this thesis, non-serviceable parts that are not inducted into repair are
considered deferred.

Some parts are repairable at point of use. This research focuses on repairs which
must be sent to a separate location. These repairs must be scheduled in order to secure
limited repair resources. Multiple part types can break and are repaired using the same
resources. Physical limitations bound the number of repairs that can be made at one time.
These limitations force choices to be made of in what order and when to induct various
components into repair at a repair depot. Decisions need to be made of which component
type and what quantity should be repaired first.

Throughout this thesis, a component refers to a unique stock keeping unit (SKU).
A part refers to the actual physical number of reparable items of each component type.
For example, in a reference to 20 of SKU_1, SKU_1 is the component and 20 is the
number of parts of component SKU_1.
The reparable parts system examined in this thesis is composed of repair facilities, centralized stocking warehouses, and field stocking locations (FSLs) assigned to specific part-breakage locations. The general repair cycle for a reparable part and important decision points are shown in Figure 1.1 below. Field stocking locations are located at each part-breakage locations within the system and service customers. For this research, the facility to which a part is sent for repair is referred to as a repair depot.

![Figure 1.1 Repair Cycle Diagram](image)

When a part fails at point of use at a customer location, the non-serviceable part will be removed from service, and a request is placed for a replacement part of the same component type. The broken part will then join the queue of all other non-serviceable parts to be considered for repair. If selected for repair, the part will be sent to a repair depot to be made serviceable again. Parts not selected may defer repair indefinitely remaining non-serviceable at a FSL until such a point that is economically feasible or strategically beneficial to repair the part.

The repair decision point for the system modeled within this thesis is shown in Figure 1.2. Broken parts may be shipped to any available repair depot to be repaired or
deferred to remain at the customer location. If the part is selected for repair, a decision must be made as to whether to repair normally or expedite the repair in order to reduce repair time at an increased cost. Deferring repair of non-serviceable parts may impact field stocking locations both positively and negatively. The system benefits from delaying repair expenses until such time as the part is selected for repair induction. If not done strategically with a focus on system objectives, these repair delays can worsen supply chain performance creating serviceable part shortages.

Figures 1.2 Part-Breakage Decision Point

The repair depot has many different aspects which may directly impact repair cycle operation. The repair depot could be located within the same facility or location as where the part-breakage occurs or in a separate location. The repairs could be completed
“in-house” with no outside vendors performing or scheduling repairs for reparable SKUs. Repair work may also be contracted out in which case the company relinquishes a large amount of control over repair scheduling relying on contract or due dates to indicate priority or need. Assignment of repairs to depots may depend upon distance between a customer and the depot, repair cost or time of the component at a specific repair depot, and repair capabilities of a specific repair depot such as capacity, current workload, or which components the repair depot has the capability to repair.

After a part completes repair at a repair depot, a decision must be made of where to send the newly serviceable part. The part could be sent to a centralized warehouse location. There the part would await shipment to a field stocking location. The part could also be direct shipped from the repair depot to a customer’s field stocking location. The options available when a part finishes repair are shown in Figure 1.3 below.

![Diagram of part finishes repair decision point](image-url)

Figure 1.3 Part Finishes Repair Decision Point
When a part failure occurs, a replacement part of the same component type is desired. Replacement parts could come from several different sources including the customer’s own field stocking location, another field stocking location, a centralized warehouse, or a direct shipment of a newly serviceable part from a repair depot. This decision point is shown in Figure 1.4 below.

![Replacement Part Decision Point Diagram]

**Figure 1.4** Replacement Part Decision Point

In the repair system considered in this research, each customer has an assigned field stocking location. If the customer’s field stocking location has serviceable parts of the same component type in stock, a replacement part will come from this supply. This is ideal as there is minimal downtime since a serviceable part is available on location to immediately replace the broken part. If the part is not held in stock or is currently unavailable at the location due to stock-out, a part can be sent from a centralized warehouse. In some cases, there may be no serviceable stock at either the part-breakage...
location or any warehouse. When this occurs, a part of the same component type can be sent laterally from another field stocking location or a part finishing repair at a repair depot could be direct shipped to the initial part-breakage location.

Many factors impact how these decisions are made. Part availability at field stocking locations, serviceable parts warehouses, and repair depots influences repair supply chain decisions. The urgency of the need for a replacement part is one of the largest deciding factors when making reparable service parts supply chain decisions. For example, a part failure may negatively impact the customer service level constraints such as issue effectiveness and sub-system availability. This creates a need to repair parts strategically in order to align with these overall system goals.

In order to minimize negative impact on the repair supply chain, service levels indicating the recommended number of parts to keep in stock may be used. In this thesis, this recommended number of parts is called a target stock level (TSL). Any stocking location may have a TSL which indicates the number of serviceable parts for each component type to keep “on the shelf” at that specific location. Field stocking location and warehouse TSLs are determined based on various system specifications. Aspects of the component type, such as holding costs and the physical space required to store the component can impact the selected number of serviceable parts to be held at stocking locations.

What drives the repair cycle is similar throughout industries. Companies may operate reparable service parts supply chains differently, but the same basic building blocks are present. Parts break, parts are sent to repair, and replacement parts take the place of failed parts. No matter how a company operates its repair supply chain, some
form of repair scheduling must take place. A successful repair policy will prioritize repairs in order to meet overall system objectives.

1.2 Problem Statement

Much of the literature on reparable service parts supply chains assumes that all broken reparable parts that can be repaired will be repaired. The question is what to do if this is not the case. What if a part breaks and delaying repair is an option? Sometimes deferring the repair until a later date, or possibly forever, is the best decision for the system. Given the option to defer repair, how can a company determine which parts to repair and when? The purpose of this study is to look at this issue of how to make logical, strategic real-time repair decisions.

Considering the possibility of repair deferral allows for more flexible repair scheduling. Deferring repair may be the only option when a part breaks due to limited budget or repair depot limitations. By recognizing this, a company can send to repair components that will keep the entire supply chain running as smoothly as possible. Through strategic repairs and deferrals, a company can make the most use out of limited resources whether that is a repair budget or a repair depot capacity or workload limit.

Repair decisions in the present have long-term impacts throughout the repair supply chain. Repair decisions need to consider not only the short-term impact of which parts are sent to repair, but also the long-term state of the system. A strategic repair schedule considers the current state of the system, forecasted part-breakages, and other system specifications when making real-time decisions. This thesis focuses on how to make these real-time repair decisions based upon the overall impact to the reparable service parts supply chain.
1.3 Study Objectives

The objective of this study is to create a repair schedule which takes the current state of the system as well as future breakages into consideration while trying to optimize system performance. There are multiple objectives that the repair schedule can be optimized towards depending on a company’s overall mission or contract requirements but most revolve around minimizing backorders.

The overall goal of the repair schedule is to maximize supply chain performance. A common approach to this goal is to focus on minimizing component backorders in the repair system. Some companies use customer service level metrics such as sub-system availability and issue effectiveness to measure reparable system performance. No matter the goal, much of the repair cycle performance is driven by the designated target stock levels.

1.3.1 Issue Effectiveness

A company may calculate issue effectiveness to measure its supply chain performance. Issue effectiveness is concerned with the percentage of part failures that occur where a replacement part is immediately available. No-delay replacement occurs if a part is available on site at a field stocking location when a part failure occurs. This is often referred to as aggregate fill rate. (Basten and Houtum 44) This metric captures the proportion of time part failures create backorder situations.

An issue effectiveness focus drives a reparable supply chain to repair parts such that a serviceable part is available and on-hand whenever a part fails. This metric alone does not consider that backorders of components may not have equal impact on supply
chain performance. Additionally, issue effectiveness by itself ignores the effect of having multiple concurrent backorders of a component. This metric does capture important elements of repairable supply chain performance, but other metrics may better capture other factors of interest.

1.3.2 Sub-System Availability

Sub-system availability is another performance metric which tries to capture supply chain performance. When components are combined to form sub-systems within the repair supply chain, sub-system availability is a potential strategic focus for repair scheduling. Each sub-system’s functionality depends upon all its components being operational. The company is negatively impacted when any component of the sub-system is unavailable. When a component is missing the sub-system is unable to operate and is forced into downtime. Sub-system downtime may have a direct cost impact due to loss of production or contracted monetary penalties. Sub-system unavailability has an adverse effect upon the repair supply chain whether through direct monetary costs or through decreased overall performance.

Sub-system availability captures whether all required parts for an operating sub-system are available and serviceable at any given time. This performance metric also aims to minimize the number of times a part breaks and no serviceable parts are immediately available to replace the broken part. In addition to this, it also captures the impact of having multiple backorders of the same component. If a required part for the sub-system breaks, the sub-system cannot operate until a serviceable part is obtained to replace the broken part. This approach concentrates on improving the “weakest link” with the system only being as good as its “worst” part. Hence, a sub-system is unserviceable if
any of its components are unserviceable. The focus is on replacement parts to maximize sub-system availability and only after addressing that issue looking to repair other parts to prevent future downtime of sub-systems (Basten and Houtum 44).

1.4 Business Motivation

The case study presented in this research is based upon a company which schedules repairs for a large-scale global program with many sub-systems with multiple required components of repairable parts. These parts must be serviceable for the larger sub-system they are a part of to operate.

Repair limitations exist impacting which part repairs are made and when the repairs occur. These limitations model reality with budget and repair capability constraints. If a company is unable to always induct parts into repair after a failure occurs, then how can the company ensure that the right breadth and depth of parts are repaired? No matter what the overall objective of the repair system, there is a question of how to decide which components to repair, how many parts of each to repair, and when should these repairs occur. Many factors impact these repair decisions.

When making repair decisions, the impact on both the short-term and long-term performance of the supply chain should be considered. With a large number of parts, multiple part-breakage locations, warehouses, and repair depots, the repair system quickly becomes complex making it difficult to understand all of the interactions occurring. The human mind cannot consider all parts, locations, and repair supply chain interactions simultaneously. This is especially true when future demand (part-breakage) is unpredictable. This creates a need for a mathematical model that can concurrently consider all of these decisions as well as account for an unpredictable future.
1.5 Scope of Study

The scope of this study is to provide a tactical tool which provides short-term repair recommendations while considering future impact on issue effectiveness and sub-system availability. The tool translates the reparable parts supply chain into a mathematical model which considers both the present state of the system as well as future part failures. This study’s direct output is repair induction recommendations based on logical, mathematical optimization. This study focuses on real-time recommendations. All other decisions made in the model are meant only to drive those repair recommendations. This model is not intended to create a robust demand forecast or to make repair or shipping recommendations outside of the near future.

This study is limited by both time and data. A tactical model which can be run often is required, so the model size cannot be too large or take too long to run for its frequent use to be practical. The model only makes recommendations for the near future. It assumes target stock levels are predetermined before the start of the model. The study is also limited by the level of detail of available data. The data required for the model directly impacted the final model formulation used for the case study.

1.6 Summary

Defining characteristics of the model presented in this thesis include the following:

- Strategically determine the order in which to repair broken parts given that deferring repair is a viable option.
• Scheduling repairs across multiple repair budget periods

• Balancing repair depot workload

• Using current state of system, desired service levels (from TSLs), and forecasted demand (part-breakage) to make strategic repair decisions/inductions

• Develop a general framework that can be customized to match available detail level of data as well as specific supply chain system configuration

The remainder of this thesis is formatted as follows. Chapter 2 includes a review of literature related to this repair scheduling problem. Chapter 3 presents a general mathematical formulation as well as descriptions of the various mathematical components. Chapter 4 details the case study data and involved transformations required to model a specific real-world case based on data availability. Chapter 5 discusses the results of various model configurations as well as their implications on the model output. Lastly, Chapter 6 summarizes the results and recommendations from the model and makes suggestions for future research.
CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

The reparable parts supply chain has been the topic of much research. The literature ranges from steady-state models such as METRIC to real-time inventory allocation models. The problem considered in this thesis is most similar to the latter. Real-time repair decision-making tools are sparse in the literature. Repair scheduling and repair prioritization is approached differently in the literature examining the problem at a repair depot level. The model presented herein uses an integrated approach to produce repair recommendations in the present time in a multi-echelon reparable supply chain.

The multi-echelon reparable system structure is a recurrent theme in the literature. Much of this literature is concerned with steady-state systems and determining stocking levels. Through the years, the research has expanded upon this subject considering various modifications and additions to more accurately model the supply chain. (Basten and Houtum). These constraints include repair budgets, repair capacity, expediting, and batching. One of these extensions first optimized target stock levels for the system as a whole and then used a heuristic approach to allocate the stock to individual field stocking locations (Wong, Cattrysse, and van Oudheusden). These models did not consider a dynamic repair environment.

Real-time multi-echelon models are less common. They were developed to manage inventory allocation and dynamic repair scheduling rules. Research evolved simultaneously considering repair prioritization and spare part inventory levels. Extensions of this research look at repair capacity and cost. The literature includes
optimization models, but when trying to solve larger real-world problems a heuristic method or algorithm is used.

Part shortages can greatly impact the performance of a supply chain. A 2001 report discusses the tremendous influence a part being out of stock could have on the performance of both operations and the repair supply chain. Unexpected part failures and longer than expected time to repair parts were the main driving factors of these shortages (GAO). Based upon this knowledge, the model presented in this thesis aims to prevent these repairable part shortages when generating a repair schedule.

Still today an integrated approach considering all field stocking locations and parts simultaneously when making repair decision is not common. Without considering all repairable parts in the system and their current state, repair inductions are extremely difficult to align with long-term strategic goals (Caggiano, Muckstadt, and Rappold 293). A mathematical model can consider the implications of each repair decision for many parts and locations at once. This provides an opportunity to determine which part to repair to have the most positive impact on strategic system goals such as issue effectiveness and sub-system availability.

2.2 Steady-State Models

A large amount of the literature examines the multi-echelon repairable system structure. Many papers examine steady-state service part systems to optimize inventory levels. Various extensions have been made to the initial METRIC model. The impact of repair budget, repair capacity, expediting, and batching are some areas covered in the literature. This research does not overlap much with the focus of this thesis, but it is still
discussed as it plays such a large role in the overall reparable service parts supply chain literature.

Developed by Sherbrooke in 1968, the METRIC model derives a process for optimizing the distribution of FSL stock to minimize expected backorders for a two-echelon system, FSL and repair depot ("Metric"). Based on this research, the literature examines modifications of METRIC looking at batching, cannibalization of parts, lateral shipments, and expediting. Another extension of METRIC includes minimizing costs while maintaining customer service levels with each individual customer (Caglar, Li, and Simchi-Levi). These models focus on steady-state demand and setting TSLs. Neither of these objectives are the goal of this project, but they are the basis for the service parts repair supply chain.

Sherbrooke used the VARI-METRIC model which expanded upon the METRIC model to improve estimated backorder accuracy ("VARIMETRIC"). VARI-METRIC has also had extensions considering repair prioritization when determining target stock levels and minimizing cost (Sleptchenko, van der Heijden, and van Harten, "Using Repair Priorities"). Another VARI-METRIC based model extension optimized both repair capacity and target stock levels (Sleptchenko, van der Heijden, and van Harten, "Trade-off").

Others have considered time-based issue effectiveness in a multi-echelon reparable system when optimizing FSL stock levels (Caggiano et al., "Efficient Computation"). Another model allowed lateral and expedited shipments when determining target stock levels using heuristic approaches to minimize cost (Wong et al., "Simple, Efficient Heuristics").
These steady-state models form the basis for much of the research into multi-echelon reparable supply chains. These models are not particularly useful for the tactical repair decisions that are the focus of this thesis. METRIC-type models optimize inventory levels based upon minimizing expected backorders. Though the model presented in this research is concerned with minimizing part backorders, inventory levels are inputs to the model rather than outputs of interest as in METRIC.

2.3 Non Steady-State Models

Non-steady state models are less common. These models dynamically consider a multi-echelon system and are fundamentally challenging to develop. This line of research focuses on the daily decisions made within the repair system. Steady-state models are not as useful for this purpose.

Dynamic models may consider time-based performance based upon a multi-echelon reparable system similar to METRIC (Basten and Houtum). DYNA-METRIC approaches this line of research in a slightly different way. In DYNA-METRIC, simulation is used to look at the impact of limited repair capabilities and thus the importance of prioritizing repairs. Time-dependent stock and repair performance measures including sub-system availability are of interest. (Hillestad).

Not all dynamic models capture the entire reparable supply chain. One approach is adjusting the repair schedule whenever new jobs enter a repair depot, but does not capture part allocation decisions (Bajestani and Beck). Another approach develops dynamic, heuristic scheduling rules for a small system composed of one repair depot and a single field stocking location (Tiemessen and van Houtum). Nonlinear programming
has been used to link inventory allocation, target stock level determination, and repair depot location (Rappold and van Roo).

The model presented in this thesis builds on these ideas of time-dependent performance measures to make operational decisions in real-time rather than focusing only on minimizing cost.

### 2.4 Repair Scheduling and Prioritization

Much of the literature on repair scheduling and prioritization considers only the impact on the repair depot. Repair inductions are based solely upon what is occurring at the repair depot. In a multi-echelon system, repair inductions directly impact field stocking locations’ stock levels. Therefore, an integrated approach to making decisions in the reparable supply chain may be appropriate. Real-time models examine the impact of these daily decisions upon the entire supply chain’s performance.

The research for repair-depot level scheduling considers both static and dynamic approaches. Static repair prioritization for a single repair depot and field stocking location has been found to be an improvement over the simple policy of first in first out (Adan, Sleptchenko, and van Houtum). Repair depots must handle a large amount of variability, and more realistic results have been produced by accounting for this variability (Guide Jr, Srivastava, and Kraus).

Repair scheduling in a service parts supply chain typically concentrates on maximizing sub-system availability. The problem has also been approached from a third-party repair depot perspective. In this case, a multi-echelon system with uncertain demand prioritized repairs based upon minimizing cost and meeting due dates (Zanjani and Nourelfath). EXPRESS was developed by the RAND Corporation to prioritize
repairs and inventory allocation based upon field stocking location need determined by target stock levels (Clarke).

At a depot-level, dynamically reacting to uncertain demand may be done through repair rescheduling whenever new jobs enter a repair depot (Bajestani and Beck). Dynamic multi-echelon models have also been used to analyze service parts supply chains with time-based service level constraints looking at issue effectiveness-type performance metrics (Caggiano et al. “Optimizing Service Parts Inventory”). These approaches do not consider the impact the repair schedule can have on the entire service parts supply chain.

Real-time reparable system models approach repair scheduling and prioritization differently. The research examines how to make day-to-day repair and allocation decisions. METRIC-type models are concerned with setting overall supply chain parameters, but this line of research is concerned with operational decisions. Decisions are made based upon rules that range from the simple first in first out rule to dynamic repair rules that consider the current state of the system. Repair prioritization using static rules was also a proposed extension of VARI-METRIC and was considered when setting target stock levels so as to minimize cost (Sleptchenko, van der Heijden, and van Harten). These models focus on determining real-time system decisions which align with this thesis’s interest in repair scheduling and prioritization.

Real-time models focus on daily supply chain decisions such as repair and part allocation. Unlike METRIC-type literature, optimizing target stock levels is not a concern. Instead, these models approach operational decision-making in a variety of ways ranging from very simple scheduling rules such as first in first out (FIFO) to dynamic
programming of the supply chain. Previous research disregards dynamic programming for practical operational decision-making. This is due to the massive size and difficulty in solving such models. Much of the research in this area falls in the middle, developing dynamic allocation rules proven to perform better than static rules.

Using dynamic rules, the flow of jobs and thus the repair schedule can be managed in a repair facility. Researches have examined dynamic, heuristic scheduling rules based upon component dependent repair times, backorders, and demand uncertainty. These research considered only a small system composed of one repair depot and a single field stocking location (Tiemessen and van Houtum).

Overall, the repair scheduling and prioritization research varies in both scope and approach. Repair-depot level models are concerned with repair prioritization to meet contracted due dates. Real-time multi-echelon literature ranges from simple to extremely complex and looks to make operational decisions based on the actual current state of the system. This thesis is based upon these real-time models, but focuses mainly on repair decisions rather than part allocation rules.

### 2.5 Repair Deferrals

Another category to consider is deferring repairs. Much of the past research on the reparable service parts supply chain not only ignores repair capacity, but assumes that all parts that fail will be repaired. In the case of our case-study and many other real-life instances, that might not be the case. Repairs may be delayed any length of time from short-term delays due to system limitations to indefinite deferrals as historically used components are phased out and replaced.
Budget constraints are not the only reason for deferral. Improper part balancing at the start of the model and repair limitations such as budget and capacity are reasons to defer repair indefinitely. Additionally, repair deferrals by managers have been found to align with expected earnings (Perry and Grinaker).

2.6 Real-Time Decision-Making with Repair Limitations

Operational decision-making in reparable supply chains is not a main area of focus in the existing research. Literature which considers practical repair limitations such as repair capacity is even scarcer. To do this, the current state of the system is considered when making these daily repair and allocation decisions. This area is the focus of this research and the model presented herein.

Pyke in 1990 was one of the first to consider both repair capacity and allocating serviceable parts in the system in a real-time model. A mathematical programming model formulation proved to be complex and difficult to solve. A Dyna-METRIC extension considering dynamic priority rules was used instead and outperformed the simple first in first out rule. (Pyke)

Caggiano, Muckstadt, and Rappold discuss the importance of this line of research in “Integrated Real-Time Capacity and Inventory Allocation for Reparable Service Parts in a Two-Echelon Supply System.” The authors state in this paper published in 2006, “To our knowledge, no other published research has provided and substantiated such a comprehensive integrated real-time model.” (Caggiano, Muckstadt, and Rappold 298) Research in integrated real-time reparable supply chain decision-making is still sparse today. Little of the literature considers practical applications for making strategic operational decisions based upon the current state of the system.
Caggiano, Muckstadt, and Rappold’s work is based on answering very similar questions to those proposed in this thesis. As the authors stated,

Two critical decisions must be made daily when managing multiechelon repair and distribution systems for service parts: (1) allocating available repair capacity among different items and (2) allocating available inventories to field stocking locations to support service operations. (Caggiano, Muckstadt, and Rappold 292)

The authors considered limited repair capacity, uncertain demand, and the current state of the reparable supply chain when making these allocation decisions. The main focus though is on inventory allocation of serviceable parts with repair decisions playing a supporting role to optimize part allocation. Other research uses an integrated approach to develop an integer program, local search algorithm, and greedy heuristics to make decisions of inventory allocation while considering the system’s performance on sub-system availability (Barahona et. al).

Similar to the case-study example later in this thesis, Caggiano, Muckstadt, and Rappold looked at a single repair facility, centralized warehouse, and multiple field stocking locations (293). The authors developed a “finite-horizon, periodic-review mathematical program” to make repair and part allocation decisions concurrently (292). They built three models looking at stock allocation, stock allocation with option for expedited shipping, and stock allocation and repair decisions with limited repair capacity. All of these models use the current state of the system along with future demand information when making these decisions (294-295). The authors state, “Our problem is to figure out how to best react to the current situation with the resources and information that we have.” (293)
Caggiano, Muckstadt, and Rappold’s model focus is on minimizing service parts supply chain related costs assuming all parts will at some point be repaired. Their model formulation outputs an optimized repair sequence (Caggiano, Muckstadt, and Rappold). In contrast, the model presented herein tries to maximize the customer service level performance metrics of issue effectiveness and sub-system availability while allowing for parts that fail to defer repair indefinitely. This allows for the model to account for and try to correct inventory imbalances throughout the system. This thesis produces a similar “batting order” sequence of parts to repair, but it is not necessarily an optimized list.

2.7 Summary

Much of the existing service parts supply chain literature focuses on steady-state supply chains. Through the years, the research has evolved to include non-steady state and non-stationary models. Little of the research though has focused on creating practical tools to strategically make operational decisions. These decisions can be critical to overall supply chain performance and are the subject of this research.

Steady-state models form the basis for much of the research into multi-echelon reparable supply chains. These models are not particularly useful for the tactical repair decisions that are the focus of this thesis. METRIC-type models optimize field stocking location inventory levels based upon minimizing expected backorders. Though the model presented in this thesis is concerned with minimizing part backorders, target inventory levels are inputs to the model rather than outputs of interest as in METRIC.

Much of the past research has allowed for infinite repair capacity whether the focus is on strategic long-term decisions such as optimizing target stock levels or the day-to-day impact of repair capabilities. The model presented in this thesis generates real-
time repair recommendations based on the inputs of current state of the system, forecasted part-breakages, and system specifications. By taking into consideration these factors as well as repair system limitations, the model is able to prioritize the order of part repair induction.

This thesis presents a method and mathematical model for making daily repair decisions. Modeling the entire service parts supply chain operation allows daily repair recommendations to be made based on model-determined part allocation. Determining the correct depth and breadth of parts to be repaired is the main focus, and inventory allocation is a by-product of the repair decisions rather than a main output.

Using day-to-day decision tools that regularly update the current state of the system, allows for simplifying assumptions to be made. The model assumes repair and shipping times are exponentially distributed. Making use of the memoryless property of the exponential distribution, average historical times are used in the model formulation. No matter the distribution of the times, the model’s use of current state of the system continually updates completion and arrival dates accounting for the true variability in the system.

Overall, the model presented in this thesis considers day-to-day reparable parts decision-making based upon budget, repair capacity, and the ability to defer repair indefinitely. This model is a tool to make tactical repair recommendations and prioritize those repairs. It considers both the short-term and long-term impacts of the repairs, while eliminating some large assumptions from previous research. Past research in this area aims to minimize cost including the cost of backorders and holding costs. The model presented herein differs in that it concentrates on maximizing comparative fill rate
calculations which capture various aspects of customer service performance. It indirectly prevents backorders by stocking to previously optimized target levels and forecasted demand.
3.1 Introduction

In a multi-echelon service parts supply chain, real-time repair decisions have significant impacts on supply chain performance. Inducting into repair the appropriate breadth and depth of components is essential to meeting short-term and long-term customer service level constraints. This is particularly true when repair capabilities are limited. This thesis develops a Mixed Integer Linear Program (MIP) which considers the current state of the entire repair system as well as forecasted part-breakages to produce strategic real-time repair recommendations. Repair inductions are prioritized to maximize stock location service levels over multiple budget periods through comparative fill rates encompassing both issue effectiveness and sub-system availability.

Historically, service parts supply chain research primarily focuses on steady-state supply chains. Much of this research assumes infinite repair capabilities and the repair induction of all non-serviceable reparable parts. Though real-time repair decisions can be critical, fewer researchers have focused on creating tools to strategically make these decisions. The MIP presented herein generates real-time repair recommendations based on the inputs of current state of the system, forecasted part-breakages, and system specifications. By taking into consideration these factors as well as repair system limitations, the model is able to prioritize the order of part repair induction.

Model inputs are of two forms: updated every time the model is run and updated at the user’s discretion. Current state of the system parameters must be updated every model run in order for the model to make the best decisions. These parameters capture
the location of all considered parts in the repair system including stocking locations, in repair, and in transit. The remaining parameters such as repair cost, component criticality, and other system specifications may be updated at the user’s discretion. This may be periodically and/or whenever a change occurs. To reduce the burden of updating all of these values manually, the model uses controlling sets to capture only the relevant data from large automated data pulls.

Using the initial state of the system, the model generates real-time repair recommendations. To do this the model determines repair and part allocation decisions throughout the entire model time horizon while considering strategic company goals related to customer service levels. The output of interest is an ordered list of repair recommendations for the immediate future. The model’s other decisions dictate the flow of parts through the model so that current repair recommendations are made based on their impact on long-term strategic goals. The naturally following output is a sequenced order of repair inductions in the present. A priority list of repair inductions allows for repairs to be made so that the short-term decisions are aligned with long-term company objectives.

In order for the model to make the best informed decisions, the flow of parts throughout the repair supply chain are modeled over a time horizon of more than two years. A multiple year time horizon allows for the vast majority of component repair times within the examined case study to fall within the time scope of the model. This length of time also ensures that at least two budget periods are considered in the model. No matter the date the model is run or the length of the budget period, the model is able to consider the impact of multiple budget periods when determining repair inductions.
Multiple budget periods is a key aspect of this model and is important for strategic decision making. For the model to serve as part of a true tactical tool, it needs to consider the impact of budget period length. Companies often set an annual budget for repairs. This budget may be artificially split into smaller buckets to ensure level repair spending throughout the year. The goal of splitting the budget is to prevent spending the entire budget too long before the end of the fiscal year. Ideally, the budget buckets are set so as to ensure that the bucket size corresponds to the percentage of annual demands that occur in that bucket. In reality, this is not always the case.

Issues may occur when the smaller budget buckets are not aligned to the demand profile. For the service parts supply chain, demands occur when parts break. Demand could be constant throughout the year, but fluctuations such as seasonal demands may occur. In the service parts supply chain, this would include fluctuations in usage throughout the year. For example, the number of sub-systems for which the components are used may change or the running time of the sub-systems may fluctuate. Either of these changes could cause a change in demand due to their impact on part-breakage rates. If the demand is uneven throughout an annual budget period, it is especially important to ensure that budget is allocated appropriately.

The model has the capability to consider monthly, quarterly, semi-annual, and annual budget period lengths. An annual budget can be used since the model considers all of the predicted demands occurring in the entire budget year. An annual budget helps prevent artificial fluctuations in repair due to the end of a budget period; for example, if all of a monthly budget is used before the end of the month. Parts that would otherwise be inducted into repair must defer repair until budget is again available. These deferrals
would accumulate until the next budget period when budget is once again available to make repairs. At that time, many of the previously deferred parts are inducted into repair, potentially overloading the repair depot with a surge of broken parts. Using an annual budget allows for the model to accommodate fluctuations in repairs and allocate repair budget throughout the year.

Finite repair capacity is another major consideration in this model. Without any additional limitations on repair inductions, repair depots can easily enter into a feast or famine type operation. Particularly when budget is a consideration, repair depots may be starved for work at the end of a budget period, then overwhelmed at the start of the next budget period. This does not allow the repair depot to operate effectively. Ideally, repair recommendations are made while taking into account balancing the repair depot workload. The model presented captures this in two ways: limiting the size of a single repair induction and setting a repair depot capacity.

The model formulation presented in this chapter is a general all-inclusive model. The formulation is dependent on the availability of low-level detailed data for various parameters as well as the computing capabilities available to run the model. In some cases, that level of detailed data is unavailable; in that case, the model must be modified to match the level of information readily available. This may mean altering the entire formulation in order to make use of data already captured by the company. The general model formulation may also be modified in order to reduce the size of the model. This allows for the model to be run within the available computing capabilities. The model formulation can be configured as necessary to align with the real-world supply chain it represents.
The model presented in this chapter is designed for flexibility in order to represent various service parts supply chain configurations. The general formulation is a good starting point to adjust the model to align to company specific situations. Depending on the modifications made, the size of the model may be directly impacted. Reducing the size of the model by removing options which are not relevant or of interest can improve run time and reduce the necessary space required for the model. Many potential modifications are listed in 3.7 Model Customizations.

3.2 Sets

Several sets are used to designate the scope of the model. These sets dictate which components, locations, etc. will be used in the model and considered for repair. Using controlling sets allows for easier input and control of the data entering the model, particularly in real-life applications using mathematical programming language. The following sets are used in the general model formulation:

- **I** — set of all component types
- **J** — set of all part-breakage/field stocking locations (FSLs)
- **W** — set of all warehouses (central stocking locations)
- **JW** — set of all FSLs and warehouse (union of sets J and W)
- **T** — set of all time units t considered in the model time horizon
- **P** — set of all considered budget periods
- **K** — set of all repair depots
- **R** — set of all repair method options
The sets and the size of these sets control the scope of the model. Sets I, J, W, K, and R control which parts, locations, and repair methods to consider in the model. Sets T and P control time elements in the model which are necessary for the model’s time horizon. Each set serves a different function within the mathematical model. These sets are the basis for the rest of the model formulation.

Set I contains all component types to be considered for repair in the model. If a component type is not in the list, then it will not be considered for repair. This is true even if system specifications are included for the component. Depending on the company, this set might need to account for master parts. A master part number may be used to represent entire families of substitutable parts within the model. Through the use of master parts, the model considers the impact of part interchangeability when making decisions.

Sets J, W, and JW control all stocking locations within the model. Set J includes all part-breakage locations to be considered in the repair supply chain. Set W includes all stocking locations where part-breakages cannot occur. This may be an actual warehouse or not. Set W includes any location that stocks serviceable parts for replacement that can be sent to part-breakage location(s) when demand occurs. For the model, this could be a virtual warehouse that accounts for all serviceable parts not immediately sent to a part-breakage location after completing repair. Set JW is the union of sets J and W for use in the model. This allows for ease of defining and using parameters and variables shared by part-breakage locations and warehouse stocking locations.

Sets T and P control timing within the model. Set T contains discrete integer time units considered within the model’s time horizon. The model formulation is independent
of whether the time unit considered is a day or a week. Set $T$ is dynamic and updated every model run as it depends on the date the model is run. The time horizon considered includes the month of the run date and the 25 months following. This sets an approximately 26 month time horizon. Set $P$ is the set of all budget periods in the model time horizon. This is dependent on the budget period length selected by the user. The model currently has options of monthly, quarterly, semi-annual, and annual budget periods.

Sets $K$ and $R$ control the repair options available within the model. Set $K$ is the set of all considered repair depot locations used in the model. Set $R$ is the set of all repair methods considered by the model. Specifically, we consider normal repair and expedited repair in the general model formulation.

Not all of these sets are required by the model in order to run. Which sets are used depends on the level of data available. Sets $I$, $J$, $W$, $JW$, and $P$ are required for every model run. The model can be reformulated to exclude $K$ and $R$ or contain only a single element if complete data is unavailable.

The model will run without issue given some modification while excluding the added detail sets $K$ and $R$ provide. Set $K$ could be excluded from the model by simplifying the model to consider a single repair depot. There may be only one repair depot used by a company, the repair depot locations have negligible differences in specifications, or repair depot-level information is incomplete or believed inaccurate and so a single value is used for all depots. Set $R$ is a similar case to set $K$. Sometimes the detail level of the differences in repair time and repair cost for expedited repairs is unavailable.
3.3 Parameters

Several parameters are used throughout the model formulation. These parameters are defined across the sets mentioned in section 3.2. The parameters are of three main types: system specifications, current state of system, and model configuration. Note, the model configuration parameters are primarily used to develop the mathematical model formulation and adjust the objective function. Only system specification and current state of system parameters are discussed in this section. The remaining parameters will be discussed in section 3.5.

System specification parameters capture data for the model which is fairly static and requires only occasional updates. The following parameters fall into this category:

- repaircost(R,I,K): the cost of using repair method R to fix component type I at repair depot K
- transitcost(I,JW,K): the cost to send (ship) part type I between FSL/warehouse JW and repair depot K
- transfercost(I,JW,J): the cost to send (ship) part type I from FSL/warehouse JW to FSL J
- budget(P): total repair budget available for period P
- periodend(P): used to determine the last time unit in budget period P
  - periodend(P) ≥ 0 ; ∀P
- demand(I,J,T): forecast number of part type I breaking at FSL J at time T
  - demand(I,J,T) ∈ ℤ ∀I, ∀J, ∀T
- percent(K): allowable percentage of demand for total parts of any component type sent to repair depot K in a single model time unit
- maxwork(I,K) : maximum repair capacity for part type I at repair depot K
- tsl(I,J) : target stock level for part type I at FSL J
  - tsl(I,J) ∈ ℤ ∀I, ∀J
- repairtime(R,I,K) : time units required using repair method R to repair part type I at repair depot K
  - repairtime(R, I, K) ≥ 0 ; repairtime(R, I, K) ∈ ℤ ∀R, ∀I, ∀K
- transittime(JW,K) : time units repaired for a part to be sent(ship) between FSL/warehouse JW and repair depot K
  - transittime(JW, K) ≥ 0 ; transittime(JW, K) ∈ ℤ ∀JW, ∀K
- transfertime(JW,J) : time units required for a part to be sent(ship) between FSL/warehouse JW and FSL J
  - transfertime(JW, J) ≥ 0 ; transfertime(JW, J) ∈ ℤ ∀JW, ∀J
- totaltime : total time units in model time horizon
- weighti(I) : criticality weighting of part type I
- weightj(J) : criticality weighting of FSL J

In order for the mathematical model to work properly, some of these values must meet certain requirements. The parameters which capture demand and target stock levels refer to whole parts so must be integer in value. Additionally, all time-related parameters (repair, transfer, transit, and periodend) must be integer in value and positive. In actuality many of these time-related values may not be integer, but for the sake of the model are rounded to an integer value.

Current state of the system parameters capture data for the model which must be updated every model run in order to produce relevant results. The model needs to know
where all parts are in the system initially. These parameters capture decision variable values prior to the start of the model. All of these initial values for variables must be integer values as the model only tracks whole parts. Only initial stock values may be less than zero to account for backorders prior to the start of the model. The following parameters fall into this category:

- $z_{\text{init}}(R,I,K,T)$: initial number parts in repair method $R$ of component type $I$ from FSL $J$ to depot $K$ completing repair at time $T$
  
  $z_{\text{init}}(R,I,K,T) \geq 0$ ; $z_{\text{init}}(R,I,K,T) \in \mathbb{Z}$  \hspace{1em} $\forall R, \forall I, \forall K, \forall T$

- $d_{\text{init}}(I,J)$: initial number deferred (broken parts) of component type $I$ at FSL $J$ at start of model
  
  $d_{\text{init}}(I,J) \geq 0$ ; $d_{\text{init}}(I,J) \in \mathbb{Z}$  \hspace{1em} $\forall I, \forall J$

- $a_{\text{init}}(I,JW,K,T)$: initial number parts of component type $I$ moving to FSL/warehouse $JW$ from depot $K$ arriving at time $T$

- $l_{\text{init}}(I,JW,J,T)$: initial number parts of component type $I$ moving from FSL/warehouse $JW$ to FSL $J$ arriving at time $T$
  
  $l_{\text{init}}(I,JW,J,T) \geq 0$ ; $l_{\text{init}}(I,JW,J,T) \in \mathbb{Z}$  \hspace{1em} $\forall I, \forall JW, \forall J, \forall T$

- $stock_{\text{init}}(I,JW)$: initial number parts of component type $I$ in stock (serviceable on the shelf) at FSL/warehouse $JW$
  
  $stock_{\text{init}}(I,JW) \in \mathbb{Z}$  \hspace{1em} $\forall I, \forall JW$

All of the previous parameters serve a function in the model. Some parameters provide the information necessary to model correctly the flow of parts through the system. Other parameters inform the model of the current state of the system in order to allow the model to make strategic decisions based on where the service parts system is
currently and where it should be with regards to service level constraints. These values will be essential in section 3.6.

3.4 Decision Variables

Decision variables serve many purposes in the model formulation. These variables function in several capacities including: repair scheduling, part allocation, bookkeeping, and objective function calculations. These variables are defined across the sets mentioned in section 3.2. Therefore, as more components and locations are considered, the number of decision variables can grow considerably. Objective function calculated variables will be discussed in section 3.5.

The number of decision variables impacts the overall size of the mathematical model. Model size can be altered tremendously depending on how the different decision variables are indexed. This is discussed further in section 3.7. Other decision variables are also used for the objective constraint formulation. These variables will be discussed in section 3.5 in conjunction with the corresponding equation(s).

Repair schedule variables answer the questions of which components should we repair, how many parts of that component should be repaired, and when should these parts be sent to repair. The following variables capture these decisions in the model:

- \( z(R, I, J, K, T) \) : number of (broken) parts of component type I from FSL J to be inducted into repair using repair method R at repair depot K at time T
  - \( z(R, I, J, K, T) \geq 0 \); \( z(R, I, J, K, T) \in \mathbb{Z} \) \( \forall R, \forall I, \forall J, \forall K, \forall T \)

- \( d(I, J, T) \) : number of (broken) parts of component type I deferred at FSL J at time T
  - \( d(I, J, T) \geq 0 \) \( \forall I, \forall J, \forall T \)
Part allocation variables answer the questions of where and when to send serviceable parts to a part-breakage location (FSL) or central stocking location (warehouse). In this model, the repair depot stores no serviceable parts; parts finishing repair are immediately shipped to a stocking location. These allocation decisions determine where a part goes after it completes repair as well as where replacement parts are sourced from when a part breaks at a part-breakage location. The following variables capture these decisions in the model:

- \( a(I,JW,K,T) \): number of repaired parts (now serviceable) of component type I sent to FSL/warehouse JW from depot K at time T
  - \( a(I,JW,K,T) \geq 0 \); \( a(I,JW,K,T) \in \mathbb{Z} \) \( \forall I, \forall JW, \forall K, \forall T \)

- \( l(I,JW,J,T) \): number of serviceable parts of component type I sent from FSL/warehouse JW to FSL J at time T
  - \( l(I,JW,J,T) \geq 0 \); \( l(I,JW,J,T) \in \mathbb{Z} \) \( \forall I, \forall JW, \forall J, \forall T \)

Bookkeeping variables track values in the model which are necessary for various constraints. These variables are used to maintain proper flow of parts through the model. Proper flow refers to both required flow balancing equations as well as user-imposed flow limits. Flow balancing equations ensure logical flow of parts through the repair cycle while flow limits are in place to account for repair depot workload limitations. The following variables are considered bookkeeping variables within the mathematical model:

- \( stock(I,JW,T) \): number of serviceable parts of component type I in stock “on the shelf” at FSL/warehouse JW at time T
  - \( stock(I,JW,T) \geq 0 \) \( \forall I, \forall JW, \forall T \)
The bookkeeping variables of rep(I,K,T) may be unnecessary in a company specific final model formulation due to their use only in non-essential constraints.

The decision variables listed previously are of various types. This is a MIP model so no variable is nonlinear in form. Note, as more variables are included and forced to take integer values, the more difficult it becomes to solve the model in a reasonable amount of time. Though the model considers only discrete numbers of parts, model solution time may be improved by strategically relaxing integer constraints on variables. The model can be simplified by the careful selection of variable type.

All of the previously listed decision variables should ultimately resolve to integer values in any feasible solution. This is due to the use of only discrete parts and time units in the model. This is why the parameters which capture initial variable values must be integer in value. The decision variables that are restricted to positive integer values are:

- z(R,I,J,K,T)
- a(I,JW,K,T)
- l(I,JW,J,T)

This restricts the number of parts sent to repair and the number of parts shipped both laterally and directly from the repair depot to be positive integer values. This follows logically as we consider neither moving partial parts nor continuous time units throughout the model and its inputs.

By restricting the variables z(R,I,J,K,T), a(I,JW,K,T) and l(I,JW,J,T) to integer values along with the previously mentioned integer initial parameter values for variables,
this forces the remaining variables to maintain integer values. Thus, we can define the following parameters as positive variables and expect their values to also be integer in any feasible solution. The decision variables determined to be restricted to positive values are:

- \( d(I,J,T) \)
- \( rep(I,K,T) \)

The remaining model decision variable is \( stock(I,JW,T) \). Stock is unrestricted in value for all part-breakage locations. Similar to \( d(I,J,T) \) and \( rep(I,K,T) \), \( stock(I,JW,T) \) will also ultimately resolve to integer values without being explicitly defined as integer. Stock is allowed to take negative values at part-breakage locations. This is allowed for robustness. Stock is a bookkeeping value that indicates the number of parts on the shelf if it is positive and the number of backorders if it is negative. Allowing negative values allows the model to avoid errors due to part-breakages when a replacement part cannot be obtained immediately. This is especially important if any components have target stock levels of zero. In those instances, a part-breakage is especially likely to cause a stock-out (backorder) situation as a replacement part takes time to arrive from another location.

These are the base variables for the proper operation of the model. Through model design, all of these variables will have integer (or near integer) values in feasible solutions to the MIP. These design consideration are explained further in 3.6.3 Flow Balance Constraints. By strategically reducing the number of variables restricted to integer values, the repair supply chain can be modeled in such a way as to minimize unnecessary additional restrictions on decision variable values.
3.5 Objectives

The objective of the mathematical model drives the ultimate repair schedule determined in the model solution. The general intent of the produced repair schedule is to make repair decisions strategically considering issue effectiveness and sub-system availability. To do this, the model should repair parts so as to avoid backorders or stock-outs of any form. This idea is translated into the model as a comparative fill rate which considers both the number of parts in stock (or backordered) and the target stock level. The model’s objective is to have parts in stock at the desired TSL.

In order to model the objective function, objective specific decision variables are created. These variables are used only for calculating values needed for the objective function; they serve as bookkeeping variables rather than decision points. Therefore, the following variables are all defined as free variables unrestricted in value:

- **fill(I,J,T)**: number of component type I in stock at FSL J to count towards objective function fill rate at time T
- **fillrate**: objective function calculated comparative weighted fill rate
- **minavg(J)**: minimum component time-averaged fill rate at FSL J calculated for objective function
- **avgmin(J,T)**: minimum component fill rate at FSL J at time T calculated for objective function

The decision variable fill(I,J,T) computes an adjusted stock value for use in the objective function calculation. This variable takes the minimum value of stock(I,J,T) and tsl(I,J). This can be achieved through the use of a constraint and an upper bound on the
variable so that it cannot exceed the target stock level. Equation 3.1 displays the constraint which forces the fill value to be less than stock(I,J,T).

\[ fill_{IJT} \leq stock_{IJT} \quad \forall I, \forall J, \forall T \quad (3.1) \]

Equation 3.2 displays the constraint which forces the fill value to be less than tsl(I,J) or predicted demand.

\[ fill_{IJT} \leq tsl_{IJ} + \begin{cases} demand_{IJT} & tsl_{IJ} = 0 \\ 0 & tsl_{IJ} \neq 0 \end{cases} \quad \forall I, \forall J, \forall T \quad (3.2) \]

Similar to stock(I,J,T), fill(I,J,T) will also result in integer variables which may be positive if parts are in stock or negative if parts are backordered.

Two overarching goals for the general repair supply chain drive the model-determined repair schedule. Motivating service level performance metrics include issue effectiveness and sub-system availability. These goals are similar, but drive towards different end results. Attempting to directly optimize the MIP model to either of these would require exceptionally accurate input data including a part-breakage demand forecast. This is often not realistic, so the model presented herein approaches these customer service level constraints in a more round-about way.

Sub-system availability and issue effectiveness are both concerned with ensuring backorders or stock-outs do not occur. This model approaches this issue by optimizing repairs based upon predetermined target stock levels as well as predicted demand occurrences. This is done through weighted comparative fill rates calculated as shown in Equation 3.3.

\[ \frac{weight_{IJ} \cdot fill_{IJT}}{\begin{cases} demand_{IJT} & tsl_{IJ} = 0 \\ 0 & tsl_{IJ} \neq 0 \end{cases}} \quad \forall I, \forall J, \forall T \quad (3.3) \]
The objective function uses four distinct components to capture sub-system availability and issue effectiveness impacts in both the short-term and long-term when making repair decisions. The model approaches these aims through a combination of maximizing average fill rates and maximizing minimum fill rate at each FSL. Each of these objective components can be considered for both the short-term and long-term. The short term is considered through averaging for each time unit. The long-term is considered by averaging time-averaged fill rate values for each component.

In order to account for these multiple considerations, the following objective parameters are included.

- **spltmin**: objective parameter to switch between any combination of maximize minimum fill rate and maximize average fill rate
  - \( spltmin \geq 0 \); \( spltmin \leq 1 \)
- **spltavg**: objective parameter equal to \( 1.00 \) – \( spltmin \)
- **pmin**: percent split between maximize minimum fill rate objective function components
  - \( pmin \geq 0 \); \( pmin \leq 1 \)
- **pavg**: percent split between maximize average fill rate objective function components
  - \( pavg \geq 0 \); \( pavg \leq 1 \)

The full objective function is shown below in Equations 3.4-3.7 with explanations for each component to follow. Each component is calculated separately for each field-stocking location so as to consider each customer service level and avoid issues due to component stock and target stock level misalignments at the start of the model.
The objective function makes use of many conditional statements based upon the values of the parameters tsl(I,J) and demand(I,J,T). When TSL is zero, demand(I,J,T) is used in place of TSL in the fill rate calculation. By considering all forecasted part failures, the model indirectly captures service level performance in the objective function. The objective calculates these demand-based fill rates only for times T in which a demand occurs.

### 3.5.1 Issue Effectiveness

The model captures issue effectiveness when making repair recommendations by maximizing the average part fill rate at a field stocking location (FSL). Maximizing average part fill rate approximates issue effectiveness as it improves the overall system performance driving components to be stocked to the target stock level at all part-
breakage locations. The model presented herein uses a target stock level, calculated or
determined in an external system. This model does not determine target stock levels.

Maximizing the average part fill rate at a field stocking location leads the model
to repair components which produce the “biggest bang for the buck.” In other words, the
model will repair the components which have the largest impact on the objective
function. This is highly dependent upon components’ target stock levels with priority
given to those parts with the lowest target levels with comparative fill rate less than
100%. This can cause large variance between fill rates for different components as the
model is only concerned with the average rather than being concerned with extremely
low fill rates. This is why the model also considers sub-system availability to account for
those components that issue effectiveness would ignore.

Given target stock levels for all components in the model, average comparative
fill rates can be calculated. Through the use of two separate components in the objective,
the model considers both short-term and long-term impacts of repair inductions. The
long-term approach calculates an average comparative fill rate as an average across all I
components and J locations of weighted component time-averaged comparative part fill
rates. Shown below in Equation 3.8, the comparative average fill rate is calculated as
follows where card(I) is the total unique components in set I and card(T) is the total time
units considered in the model. Note, \(f(I,J,T), g(I,J,T), \) and \(h(I,J)\) are calculated previously
in Equations 3.5-3.7.

\[
\text{Average} \quad \frac{\sum_{t \in [h_{IJ} > 0]} \left( \sum_{T \in g_{IJ} > 0} \frac{\text{weight}_{I} f_{IJT} \cdot t_{s_{IJT}}}{\text{card}(T) \in [g_{IJ} > 0]} \right)}{\text{card}(t) \in [h_{IJ} > 0]} \quad (3.8)
\]
The short-term average fill rate calculates an average component fill rate for each time unit then calculates the average of that average fill rate over the entire model time horizon. This calculation is shown in Equation 3.9 below.

\[
\text{Time- Averaged Average Fill Rate} = \sum_j \text{weight}_j \cdot \frac{\sum_{\text{card}(T) \in [h_{ij} > 0]} \left( \sum_{\text{card}(f) \in [g_{ijT} > 0]} \frac{\text{weights}_{ijT} \cdot \text{fill}_{ijT}}{\text{ts}_{ij} + \text{f}_{ijT}} \right)}{\sum_{\text{card}(T) \in [h_{ij} > 0]} \text{card}(f) \cdot \text{card}(T)}
\]  

(3.9)

By maximizing the above calculated average comparative fill rates, issue effectiveness can be approximately maximized. This works best if the input target values are correctly aligned with demand (part-breakage). If that is not the case, the model may fail to truly maximize issue effectiveness. This is because it is stocking that part to an incorrect level that does not properly account for any inherent risk associated with demand variability. If this level is too low, stock-outs and backorders will occur. In general, the use of this objective is highly dependent on the confidence of the user that the target stock levels (or an equivalent) are correctly balanced for the repair supply chain. The model does account for this by considering each demand that occurs regardless of target stock level.

3.5.2 Sub-System Availability

The model optimizes repair recommendations by maximizing the minimum part fill rate at a field stocking location (FSL) to capture sub-system availability. A goal of maximizing sub-system availability forces the model to focus on minimizing the part stock-outs or backorders from forcing entire sub-systems into downtime. The model does this by stocking to improve the worst or lowest comparative fill rate at each part-breakage locations. This means prioritizing stocking based upon which component at a specific
location has the largest deficit between model-calculated stock level and its target stock level. This is achieved through maximizing minimum fill rates.

Given target stock levels for all considered components, minimum comparative fill rates can be calculated. Minimum fill rates are considered for both the short-term and the long-term similar to the average fill rates calculated in the previous section. To consider long-term system performance, the minimum time-average fill rate is calculated for each field-stocking location. Shown below in Equation 3.10, the comparative minimum fill rate is calculated as follows where card(T) is the total time units considered in the model. Note, f(I,J,T), g(I,J,T), and h(I,J) are calculated previously in Equations 3.5-3.7.

\[
\text{minavg}_{f} \leq \frac{\sum_{T} [h_{IJ} > 0]}{\text{card}(T) \in [h_{IJ} > 0]} \sum_{I, J} \frac{\text{weight}_{I} \times \text{fill}_{IJ}}{\text{ts}_{I} + f_{IJ}} \quad \forall I, \forall J \tag{3.10}
\]

The short-term impact of minimum fill rate is also considered in the model. This is captured by maximizing time-averaged minimum fill rate. In order to model this objective component, Equation 3.11 calculates the minimum fill rate at each field-stocking location for each time unit in the model. Building upon this, Equation 3.12 calculates the time-averaged minimum fill rate at each FSL.

\[
\text{avgmin}_{JT} \leq \frac{\text{weight}_{I} \times \text{fill}_{IJ}}{\text{ts}_{I} + f_{IJ}} \quad \forall I, \forall J, \forall T \tag{3.11}
\]

\[
\text{Time - Averaged Minimum Fill Rate} = \sum_{J} \text{weight}_{J} * \frac{\sum_{T} [h_{IJ} > 0]}{\text{card}(T) \in [h_{IJ} > 0]} \tag{3.12}
\]

By maximizing the previously calculated minimum comparative fill rates, subsystem availability can be approximately maximized. This works best if the input TSL values are correctly aligned with demand (part-breakage). Even if the target stock levels
are not ideally aligned, maximizing the minimum time-averaged comparative part fill rate will drive the repair schedule to improve the worst case scenario so as to reduce the number of backorder occurrences.

### 3.6 Constraints

Many constraints are required in order to mathematically model the repair supply chain. These constraints are a combination of user-determined limitations and flow balance equations. Budget and supplier workload constraints directly impose limitations on the model generated repair schedule based on user-determined parameter values. These constraints are not required for the model to operate and are included at the user’s discretion. The flow balance equations are required for every model run as they dictate how parts flow through the service parts supply chain.

#### 3.6.1 Budget

The budget constraint controls repair spending and therefore repair scheduling within the model. This constraint limits the total amount that can be spent on repairs in each budget period within the model time horizon. Note, the budget period length is not fixed, but rather is a user-determined value. This does not impact the use of the budget constraint though as the budget parameter is already adjusted to the chosen budget period length.

In the general model formulation, shipping costs (cost of moving parts between locations) as well as the true cost to repair are included in total repair cycle costs. This budget constraint limits total repair cycle costs in a budget period and is calculated as follows:
\[ \sum_{t \in \{t > \text{periodend}(P-1) \text{ and } t \leq \text{periodend}(P)\}} \left[ \left( \sum_R \sum_I \sum_J \sum_K z_{RIJKT} \cdot \text{repaircost}_{RIK} \right) \right. \]
\[ \left. + \left( \sum_I \sum_J \sum_W \left[ \left( \sum_K a_{IJWK} \cdot \text{transitcost}_{IJWK} \right) \right. \right. \]
\[ \left. \left. + \sum_I l_{IJWJ} \cdot \text{transfercost}_{IJWJ} \right] \right) \] \forall P \tag{3.13}

### 3.6.2 Repair Capacity

Repair depot capabilities might also be a cause of restrictions for the model repair schedule. The limitation on repairs could be enforced via a direct capacity on number of parts in repair at one time or by setting an upper bound on the number of parts that can be inducted into repair in one time unit. Equations 3.14 and 3.15 model these limitations respectively. Equation 3.14 sets a capacity limit on the number of parts of component type I that can be in repair at depot K concurrently.

\[ rep_{IKT} \leq \text{maxwork}_{IK} \quad \forall I, \forall K, \forall T \tag{3.14} \]

Equation 3.15 forces the total number parts of component type I sent to be repaired at repair depot K at any time T to be less than the maximum of a percentage of annual demand, a percentage of total target stock level for the component.

\[ z_{RIJKT} \leq \max \left[ \left[ \sum_T \sum_I \text{demand}_{IT} \right] \cdot \text{percent}_K \right] \forall R, \forall I, \forall J, \forall K, \forall T \tag{3.15} \]

### 3.6.3 Flow Balance Constraints

Several constraints are included in the model formulation in order to manage the flow of parts through the repair cycle. Equations 3.16-3.21 detail how parts move through the repair supply chain. These equations tell the model where parts go when they break, finish repair, or are needed as replacement parts. The constraints also add logical
limitations on variable values. Below are brief descriptions of what is happening in each constraint.

Equation 3.16 ensures that all parts completing repair are assigned a stock destination. It forces all parts finishing repair at a repair depot to be immediately sent to a FSL or warehouse. This is the point at which a part goes from being considered broken in the model to being an available serviceable part.

\[
\sum_{R} \left[ \sum_{J} z_{RJK(T-repair\_time_{RJ})} \right] + z_{\text{init}_{RIKT}} = \sum_{JW} a_{IJWKT} \quad \forall I, \forall K, \forall T \quad (3.16)
\]

Equation 3.17 updates the number of parts in stock at a part-breakage location based on what occurred in the model the previous time period. At a FSL, the next time unit’s stock is calculated as the stock value less parts that broke and parts laterally shipped from the FSL plus serviceable parts arriving from any location (FSL, warehouse, or repair depot).

\[
stock_{IJ(T+1)} = \begin{bmatrix}
stock_{IJT} - \text{demand}_{IJT} - \sum_{J} l_{IJ(J)T} \\
+ \sum_{K} \left[ a_{IK(T-transit\_time_{JK})} + a_{\text{init}_{IJKT}} \right] \\
+ \sum_{JW} \left[ l_{IJW(T-transfer\_time_{PJ})} + l_{\text{init}_{IJWJT}} \right]
\end{bmatrix} \quad \forall I, \forall J, \forall T \quad (3.17)
\]

Equation 3.18 updates the number of parts in stock at a warehouse stocking location based on what occurred in the model the previous time period. At a warehouse, the next time unit’s stock is calculated as the stock value less parts sent to FSLs plus serviceable parts arriving from any repair depot.

\[
stock_{IW(T+1)} = \begin{bmatrix}
stock_{IWT} - \sum_{J} l_{IWJT} \\
+ \sum_{K} \left[ a_{IK(T-transit\_time_{WK})} + a_{\text{init}_{IWKKT}} \right]
\end{bmatrix} \quad \forall I, \forall W, \forall T \quad (3.18)
\]
Equation 3.19 updates the number of parts for which repair was deferred based on what occurred in the model the previous time period. The next time unit’s number of broken parts not sent for repair is calculated as the parts previously deferred plus parts that broke and less all parts sent for repair.

$$d_{IJ(T+1)} = d_{IJT} + demand_{IJT} - \sum_{R} \sum_{K} z_{RIJKT} \quad \forall I, \forall J, \forall T$$ (3.19)

Equation 3.20 updates the number of parts in repair based on what occurred in the model the previous time period. The total parts in repair in the next time unit equals the previous time unit’s number in repair plus parts inducted into repair less parts completing repair. This constraint is unnecessary if Equation 3.14 is not enforced in the model as rep(I,K,T) is only used for repair depot capacity limits.

$$rep_{IK(T+1)} = rep_{IKT} + \sum_{R} \left[ \sum_{J} [z_{RIJKT} - z_{RIJK(T-repairtime_{RIK})} - z_{init_{RIKT}}] \right] \forall I, \forall K, \forall T$$ (3.20)

Equation 3.21 restricts the number of parts that can be sent from a warehouse stocking location at a single time. The total number of parts sent from a warehouse to all part-breakage locations cannot be greater than the number of serviceable in stock at the warehouse. This constraint is necessary for an accurate, realistic representation of the service parts supply chain.

$$\sum_{J} I_{IWJT} \leq stock_{iWT} \quad \forall I, \forall W, \forall T$$ (3.21)

The previous equations determine the flow of parts through the model’s repair cycle. Though these equations are not the focus of this thesis, they are the driving force which allow the repair supply chain to perform realistically without violating the laws of reality.
3.6.4 Fixed Variables and Variable Restrictions

Several variable values are fixed or restricted within the mathematical model. R  
logical restrictions, model simplification, and setting initial variable values. The modelled  
restrictions are as shown in Table 3.1.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Lower Bound</th>
<th>Fixed</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillrate</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Stock(I,W,T)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D(I,J,'1')</td>
<td></td>
<td>D_init(I,J)</td>
<td></td>
</tr>
<tr>
<td>Stock(I,JW,'1')</td>
<td></td>
<td>Stock_init(I,JW)</td>
<td></td>
</tr>
<tr>
<td>L(I,J,J,T)</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These variable restrictions are required for proper operation of the model. Other potential  
fixing of variables is discussed in the context of model customization located at 3.7.3  
Fixed Variables.

3.7 Model Customizations

The mathematical model presented is a general framework for the repair cycle. It  
can be customized to better match a specific existing repair cycle and/or adjusted to  
available input data. Fixing variable values is another customization option available.

Through the use of these methods, the mathematical model can be tailored to suit a  
variety of service parts supply chain configurations.

3.7.1 Model Element Redefinition and Model Reformulation

Many of the previously defined parameters, variables, and equations are defined  
across multiple indices. The big question is what to do if the input data available does not  
match these predefined indices. This could be due to data being collected at a higher level
than in the general model formulation. Or this could be due to a difference in repair supply chain operation. Either way, by redefining parameters and variables the overall model configuration can be adjusted to match a variety of repair supply chain operations.

Table 3.2 below shows a variety of potential parameter redefinitions for model customization. Affected elements that may need to be redefined are also indicated.

<table>
<thead>
<tr>
<th>Original</th>
<th>Potential</th>
<th>Impacted Elements</th>
<th>Relevant Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repaircost(R,I,K)</td>
<td>Repaircost(I,K)</td>
<td></td>
<td>3.13</td>
</tr>
<tr>
<td></td>
<td>Repaircost(R,I)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Repaircost(I)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transitcost(I,JW,K)</td>
<td>Shipping lumped with repair cost</td>
<td>Repaircost</td>
<td>3.13</td>
</tr>
<tr>
<td>Transfercost(I,JW,J)</td>
<td>Shipping lumped with repair cost</td>
<td>Repaircost</td>
<td>3.13</td>
</tr>
<tr>
<td>Percent(K)</td>
<td>Percent(I,K)</td>
<td></td>
<td>3.15</td>
</tr>
<tr>
<td></td>
<td>Percent(R,K)</td>
<td></td>
<td>3.15</td>
</tr>
<tr>
<td></td>
<td>Percent(R,I,K)</td>
<td></td>
<td>3.15</td>
</tr>
<tr>
<td>Maxwork(I,K)</td>
<td>Maxwork(R,I,K)</td>
<td>Rep</td>
<td>3.14</td>
</tr>
<tr>
<td>Repairtime(R,I,K)</td>
<td>Repairtime(I,K)</td>
<td></td>
<td>3.16, 3.20</td>
</tr>
<tr>
<td></td>
<td>Repairtime(R,I)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sets R, J, and K are all included in the general mathematical model. But what do we do if we do not have data for multiple repair method options, part-breakage locations, and/or repair depot locations? This could be due to a difference in repair cycle operation or due to a lack of low level data. The model can be reformulated and simplified by reducing the number of variables to account for this.

Eliminating sets is a situation specific case. Removing set R restricts the model to a single form of repair. Removing set J aggregates data from all part-breakage locations. This can be useful if we are most concerned with component types’ total comparative fill rates without concern for fill at specific part-breakage locations. Set K can be removed if only one centralized repair depot will be used for the model. Note, a set can be defined...
with a single element instead of eliminated from the model formulation. Through eliminating sets or reducing sets to contain single elements, the number of variables and constraints considered in the model can be reduced and thus the model simplified.

### 3.7.2 Fixed Variables

The model can benefit from fixing additional variables in addition to those previously mentioned in 3.6.4 Fixed Variables and Variable Restrictions. Through fixing variables, the general mathematical model can be configured to model a specific repair supply chain. This is an alternate method to altering the general mathematical model formulation.

Fixing variables can be used to simplify the model. It reduces the number of variables which the model must find optimal values for without having to make any changes to the general model elements indices. Fixing variable values can be a quick fix to adjust the flow of parts through the repair cycle. Table 3.3 lists some possible reasons for fixing various decision variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Potential Fixed Variable Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z(R,I,J,K,T)</td>
<td>=0 if FSL J does not send its broken parts to repair depot K</td>
</tr>
<tr>
<td>Z(R,I,J,K,T)</td>
<td>=0 if repair method R is not available at repair depot K or for part I</td>
</tr>
<tr>
<td>Z(R,I,J,K,T)</td>
<td>=0 if repair depot K does not repair part I</td>
</tr>
<tr>
<td>L(I,JW,J,T)</td>
<td>=0 if FSL/warehouse JW does not send parts laterally to FSL J</td>
</tr>
<tr>
<td>A(I,JW,K,T)</td>
<td>=0 if FSL/warehouse JW cannot receive parts from repair depot K</td>
</tr>
</tbody>
</table>
3.8 Summary

Overall, the model is highly adaptable to various service parts supply chain configurations and available input data. Through various user-dictated and flow balance constraints, the model aims to improve repair cycle issue effectiveness and sub-system availability. Through various methods, the model may be altered to better represent an existing repair system while considering model size and efficient running of the model.
4.1 Introduction

The model’s overall goal is to provide real-time repair recommendations. Repair inductions are optimized based upon the current state of the system and forecasted part-breakages input into the model. Therefore, the data plays an important role in all stages of the model: formulation, running, solution, and implementation.

Both the level of detail of data available as well as the quality of the data (accuracy and properly leveled target stock levels) impact the model formulation as well as the quality of its solution. The implications of accommodating the available data in the model formulation was mentioned in section 3.3 as well as section 3.7. The impact on solution quality was previously discussed in section 3.5 as it relates to the quality of the target stock levels used for the model.

As in many mathematical models, high quality data will produce the best, most relevant results. This is not surprising considering the influence the data has on not only the solution quality, but also the formulation of the model itself. Since data plays such an important role throughout the model, it is fitting that an entire chapter is spent discussing where the data comes from, how to format it for use in the model, and the impact it has on the actual real-life use of the model.

4.2 Data Collection / Sources

To make repair recommendations, the model requires data to populate its sets and assign parameter values. The model makes use of data commonly collected or tracked by
enterprise resource planning (ERP) systems. The parameter values determining the current state of the system must be updated every model run in order to produce the best model solutions. Due to the large amount of data updated every model run which may be already collected by companies, a company can potentially benefit from automated uploading of data to the model using information from a database.

The data required could be collected by a single company ERP system. Inopportunely, there could potentially be multiple systems tracking various portions of the service parts supply chain with none capturing the entirety of the required data. This could be the case in many large companies due to multiple legacy ERP-type systems. This could be due to mergers, acquisitions, or simply came about as a company adopted new technological and digital computing systems as they were developed. No matter the reason, a hodge-podge of systems rather than one universal system for the entire company may be collecting the required data leading to additional data formatting and reconciliation before use in the model.

All sets in the model are user-determined. Determining which elements to include in the sets is an important decision as only those elements included will be considered by the model when it is making repair decisions. Table 4.1 details the data required for the model’s sets. Some sets require additional formatting or considerations. These sets are discussed further in section 4.3.
<table>
<thead>
<tr>
<th>Set</th>
<th>Data Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>List of all SKUs/components to consider in model – discussed further in 4.3.1 Master Parts</td>
</tr>
<tr>
<td>J</td>
<td>List of all part-breakage locations to consider in model</td>
</tr>
<tr>
<td>W</td>
<td>List of all warehouses stocking locations to consider in model</td>
</tr>
<tr>
<td>JW</td>
<td>Union of sets J and W – Contains all elements from J and W</td>
</tr>
<tr>
<td>K</td>
<td>List of all repair depots to consider in model</td>
</tr>
<tr>
<td>R</td>
<td>List of all repair methods to consider in model</td>
</tr>
<tr>
<td>T</td>
<td>Set of all time units considered in model – externally calculated based on the number of days that fall within the model time horizon – discussed further in 4.3.3 Time</td>
</tr>
<tr>
<td>P</td>
<td>Set of all budget periods – externally calculated based on user set budget period length – discussed further in 4.3.2 Budget Period</td>
</tr>
</tbody>
</table>

Some parameters are general specifications and provide information about the repair supply chain. These parameters are either updated periodically from a system, manually adjusted when changes are known to have occurred, or are automatically calculated for use in the model. Table 4.2 details these parameters and their associated data requirements or potential sources.
Table 4.2  Parameter Data Requirements & Potential Sources

<table>
<thead>
<tr>
<th>Set</th>
<th>Data Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>repaircost(R,I,K)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>transitcost(I,JW,K)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>transfercost(I,JW,J)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>periodend(P)</td>
<td>Calculated based on the number of time units included in the set T and user-determined budget period length</td>
</tr>
<tr>
<td>budget(P)</td>
<td>User-determined and/or prorated for period length based on annual budget – discussed further in 4.3.2 Budget Period</td>
</tr>
<tr>
<td>demand(I,J,T)</td>
<td>Part-breakage forecast, external calculation of demand rates and program to generate a demand forecast – discussed further in 4.4 Demand Forecast</td>
</tr>
<tr>
<td>percent(K)</td>
<td>User-determined value</td>
</tr>
<tr>
<td>maxwork(I,K)</td>
<td>User-determined value or system specification</td>
</tr>
<tr>
<td>tsl(I,J)</td>
<td>System specification ideally determined from an external optimization based on various factors</td>
</tr>
<tr>
<td>repairtime(R,I,K)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>transittime(JW,K)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>transfertime(JW,J)</td>
<td>system specification: Contract, historical data, estimate, exclude if negligible, etc.</td>
</tr>
<tr>
<td>totaltime</td>
<td>Calculated based on the number of time units included in the set T</td>
</tr>
<tr>
<td>weighti(I)</td>
<td>User-determined or automatically calculated based on system specification of criticality.</td>
</tr>
<tr>
<td>weightj(J)</td>
<td>User-determined or automatically calculated based on system specification of part-breakage priority.</td>
</tr>
</tbody>
</table>

The model requires a snapshot of the current state of the system in order to perform correctly. The current number of each part type at all locations or in transit at start of model are required. These parameters are used as the initial values for several decision variables. Therefore, these parameters are particularly important to capture as accurately as possible. These are the parameters of most interest for automatically
updating before each model run and also potentially tracked by the company ERP system.

Table 4.3 details these parameters and their associated data requirements.

<table>
<thead>
<tr>
<th>Set</th>
<th>Data Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>( z_{\text{init}}(R,I,K,T) )</td>
<td>Number of parts at repair depot contracted/scheduled to complete repair and when they will complete repair – ( T = ) number of days between start of model and contract/scheduled completion date</td>
</tr>
<tr>
<td>( d_{\text{init}}(I,J) )</td>
<td>Number of parts that broke prior to start of model for which repair was deferred – number of unserviceable at part-breakage locations</td>
</tr>
<tr>
<td>( a_{\text{init}}(I,JW,K,T) )</td>
<td>Number of serviceable parts being sent from the repair depot after completing repair - ( T = ) number of days between start of model and contract/scheduled arrival date or transit time minus days since part shipped/sent</td>
</tr>
<tr>
<td>( l_{\text{init}}(I,JW,J,T) )</td>
<td>Number of serviceable parts being sent from a FSL or warehouse - ( T = ) number of days between start of model and contract/scheduled arrival date or transfertime minus days since part shipped/sent</td>
</tr>
<tr>
<td>( \text{stock}_{\text{init}}(I,JW) )</td>
<td>Number of serviceable parts on the shelf (not in service) in stock at the start of the model</td>
</tr>
</tbody>
</table>

4.3 Formatting

A large amount of data is required for the model, and the hope is that this data is already being collected in some form at the company. Though this may possibly reduce the data collection requirements, the data itself is likely in some form other than that required by the model. Besides the effects of data availability on model formulation, other data issues to consider include part interchangeability, master parts, need for components to calculate a required data element (rather than the data element itself), and formatting time into the appropriate time unit buckets. Additionally, since data may come from multiple systems, a large amount of data matching may be required.
4.3.1 Master Parts and Part Interchangeability

Sometimes components exist in the service parts system which may have substitutability for other components. These parts could be completely interchangeable or one-way substitutable. Due to the use of master parts and/or differences in nomenclature between data collection systems, data input may require formatting to ensure all desired components are considered within the model.

For the purposes of this model, the components included were master parts. Non-master components were aggregated to a master part level if interchangeable or considered their own master otherwise. Excel’s index matching was used to set up and find the master part number for each component indexed input parameter. Then, Sum If was used to total all values for rows matching the same indices’ values. Finally, Remove Duplicates was used to avoid assigning multiple values to the same parameter.

For large-scale implementation, the above process would ideally be automated to pass all incoming data through to aggregate to a master part level. As data is pulled for the model, a list matching all components to consider and their individual master parts should be used. Only master part parameter values would be used. For scalability, a similar process of summing parameters with matching indices and then retaining only the unique records would be suggested.

4.3.2 Budget Period Length and Time

The modeled repair cycle is largely impacted by both budget period length as well as the date the model in run. The model formulation allows for a choice of monthly, quarterly, semi-annual, and annual budget periods. This impacts the model’s set $T$ which
is dynamically calculated based upon the model start date. Time horizon calculations are made prior to the start of the model.

This section will discuss the implications that the model run date and the selected budget period length have upon the model time units in set $T$. The number of time units in the model time horizon is dependent on both the date the model is run as well as the size of the time unit bucket. Time unit bucket size will be examined in section 4.3.3.

The model runs for a total of the remainder of the month of the model run date plus the 13 following months. This length was determined to ensure the model considers at least 2 distinct budget periods no matter when the model is run if an annual budget period length is used. More than 95% of component mean repair times fall within this time horizon. The number of days in the time horizon is calculated automatically for the model as follows:

$$Days \text{ in Time Horizon} = \left[ Days \text{ Until End of Current Month} + Total Days in Next 13 Months \right]$$ (4.1)

To allow for flexibility in budget period length, calculations are made to assign a budget limit for each budget period. The model allows the user to manually input budget amounts for each budget period. This is especially important for the current budget period, as the remaining budget will directly impact what real-time repair recommendations can be made. Otherwise if no information besides an annual budget is indicated, calculations will be made to automatically prorate the budget based on the relative time units in each budget period in the model. This is particularly true for the first and last budget periods in the model.
The budget period length impacts not only the budget\( (P) \) parameter, but also dictates how time units are assigned to a budget period. The general steps for calculating the end dates of each budget period of all budget period lengths are as follows:

1. Calculate the total number of days in model time horizon
2. Determine last day of monthly budget periods
   a. Determine number of days in each month
      i. First monthly budget period day = the number of days remaining in the month at the start day of the model
      ii. Remaining 13 monthly budget period days = calendar days in the month
   b. Calculate last day of each monthly budget period
      i. First monthly budget period = previously calculated (a) number of days in first month
      ii. Last day in monthly budget period = previous month’s budget period last day + number of days in current month
         1. Note: the last day of the last budget period should equal the total number of days in the considered time horizon
3. Calculate last day of quarterly budget periods
   a. Determine the month number (in the year out of 12) for each considered monthly budget period
   b. Calculate number of days in first budget period
      i. Sum previously calculated number of days per month for all months from start of model to the end of first quarterly period
1. In the model, quarterly budget periods end in March, June, September, and December

c. Calculate last day of remaining budget periods
   i. Sum remaining budget period months’ calculated number days in month in tuples
   ii. Last period may not be complete tuple
      1. Note: the last day of the last budget period should equal the total number of days in the considered time horizon

4. Calculate last day of semi-annual budget periods
   a. Calculate number of days in first budget period
      i. Sum previously calculated number of days per month for all months from start of model to the end of first semi-annual period
      1. In the model, semi-annual budget periods end in June and December
   b. Calculate last day of remaining budget periods
      i. Sum remaining budget period months’ calculated number days in month in sextets
      ii. Last period may not be complete sextet
         1. Note: the last day of the last budget period should equal the total number of days in the considered time horizon

5. Calculate last day annual budget periods
   a. Calculate number of days in first budget period
i. Sum previously calculated number of days per month for all months from start of model to the end of first annual period

1. In the model, annual budget periods end in December

b. Calculate last day of last budget periods

i. The total number of days in the considered time horizon

An example of these calculations for use in the model are shown in Figure 4.1.

<table>
<thead>
<tr>
<th>Period</th>
<th>Month</th>
<th>Days in Month</th>
<th>1 month</th>
<th>3 Months</th>
<th>6 Months</th>
<th>1 year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>118</td>
<td>302</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>30</td>
<td>57</td>
<td>118</td>
<td>302</td>
<td>422</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>31</td>
<td>88</td>
<td>210</td>
<td>422</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>30</td>
<td></td>
<td>118</td>
<td>302</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>31</td>
<td>149</td>
<td>392</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>31</td>
<td>180</td>
<td></td>
<td>422</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>30</td>
<td></td>
<td>210</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>31</td>
<td></td>
<td>241</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>30</td>
<td></td>
<td>271</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>31</td>
<td></td>
<td>302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>31</td>
<td></td>
<td>333</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>28</td>
<td></td>
<td>361</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>31</td>
<td></td>
<td>392</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>30</td>
<td></td>
<td>422</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td>422</td>
<td>422</td>
<td>422</td>
<td>422</td>
<td>422</td>
</tr>
</tbody>
</table>

Figure 4.1   Days per Budget Period Calculations Example

4.3.3 Time: Days to Weeks

Depending on the number of elements in the model sets, the model size and the size of its output can grow quite large. This is due to the large number of indices on the decision variables. One method to reduce the size of the model is to run the model with time units of weeks instead of days.

In order to transform the daily data into weekly data, several calculations are required. In addition to the values previously calculated in section 4.3.2, the number of
weeks in each monthly budget period as well as the number of days in the first and last weeks of each period must be calculated. The general steps for transforming the data into weekly format is as follows:

1. Calculate last week time unit for each monthly budget period
   a. Determine number of weeks in each month
      i. Total weeks (or partial weeks) in month based on a Sunday-Saturday week
      1. Number of weekly rows on a monthly calendar
      ii. For first budget period this is the number of weeks remaining in the month after model start date
   b. Calculate last week of each monthly budget period
      i. First monthly budget period = previously calculated number of weeks in first month
      ii. Last week in monthly budget period = previous month’s budget period last week + number of weeks in current month

2. Calculate last week of quarterly budget periods
   a. Determine the month number (in the year out of 12) for each considered monthly budget period
   b. Calculate number of weeks in first budget period
      i. Sum previously calculated number of weeks per month for all months from start of model to the end of first quarterly period
   c. Calculate last week of remaining budget periods
i. Sum remaining budget period months’ calculated number
   weeks in month in tuples

ii. Last period may not be complete tuple

3. Calculate last week of semi-annual budget periods
   a. Calculate number of weeks in first budget period
      i. Sum previously calculated number of weeks per month for all
         months from start of model to the end of first semi-annual
         period
   b. Calculate last week of remaining budget periods
      i. Sum remaining budget period months’ calculated number
         weeks in month in sextets
      ii. Last period may not be complete sextet

4. Calculate last week annual budget periods
   a. Calculate number of weeks in first budget period
      i. Sum previously calculated number of days per month for all
         months from start of model to the end of first annual period
   b. Calculate last week of last budget periods
      i. The total number of weeks in the considered time horizon

5. Transform day time unit values to week time unit values for all parameters
   a. Determine number of days in first and last weeks of each month in
      model time horizon
      i. Days in first week of each month = number of days in first
         calendar week of the month

66
ii. Days in last week of each month = number of days in last calendar week of the month

iii. All weeks in month other than the first and last are 7 days in length

b. Calculate last day time unit for each week time unit in model time horizon

i. First week last day = number of days in first week of first monthly budget period

ii. Calculate remaining last day of week
   1. If week is not last week in month,
      a. Previous week’s last day + 7 days
   2. If week is last week in month,
      a. Previously calculated number of days in last week of monthly budget period

c. Transform daily time parameters into weekly time units

i. Current state of system parameters
   1. Calculate last day of each weekly time unit in model
   2. Excel index matching to assign days to a weekly time unit based upon the calculated last day of each week

ii. System specifications (repairtime, transittime, transfertime)
   1. Divide number of days by 7
   2. Round this number up to the next integer
An example of the last week of budget periods for use in the model are shown in Figure 4.2 below. Figure 4.3 below shows an example of the data showing the last day of each week for use in converting parameters from days to weeks for use in index matching.

<table>
<thead>
<tr>
<th>Period</th>
<th>Month</th>
<th>Weeks in Month</th>
<th>1st Week #Days</th>
<th>Last Week #Days</th>
<th>1 month</th>
<th>3 Months</th>
<th>6 Months</th>
<th>1 year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>52</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>10</td>
<td>20</td>
<td>52</td>
<td>73</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>15</td>
<td>35</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>26</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>31</td>
<td></td>
<td></td>
<td>73</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>6</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>47</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>52</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>73</td>
<td>73</td>
<td>73</td>
<td>73</td>
</tr>
</tbody>
</table>

Figure 4.2    Weeks per Budget Period Calculations Example

<table>
<thead>
<tr>
<th>Week</th>
<th>Month</th>
<th>Last Day</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>43</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>57</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>64</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>71</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>78</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>85</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>92</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>99</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>106</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>113</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>118</td>
</tr>
</tbody>
</table>

Figure 4.3    Last Day of Week
4.4 Demand Forecast

The model strategically makes real-time repair recommendations in order to best prepare for future part-breakages. These breakages could be known such as maintenance or unplanned part failures. Unexpected part failures are just that: unexpected. There is no way to know exactly when or where they will occur. Based on this, the model must generate some form of a demand forecast in order to allow the model to consider future part-breakages when making repair decisions. The remainder of this section discusses how the model handles this demand data.

4.4.1 Demand Rate

Oftentimes part-breakage may be defined by a distribution. In the available data, part-breakages are defined in terms of the historic mean time between depot repairs (MTBDR) and the not repairable this station rate (NRTS). In the case of this research, the demand rate was calculated as follows:

\[
AnnualDemandRate = \frac{Annual\ Operation\ Hours}{MTBDR} \times NRTS \tag{4.2}
\]

\[
DailyDemandRate = \frac{AnnualDemandRate}{365} \tag{4.3}
\]

A daily demand rate was used to forecast part-breakages for the entire model time horizon. For the weekly model run, the total demands occurring in each weekly time unit was calculated in the manner proposed in section 4.3.3. The calculated demand rate was used as the mean of an exponential part failure distribution.
4.4.2 Algorithm

The general algorithm used to develop a simple demand forecast is as follows.

- For each component in set I,
  - For each day in set T (model time horizon),
    - Calculate number of parts of component time I breaking at time T using the demand rate for a Poisson distribution
    - Round demand value to integer value
  - If model formulation considers multiple part-breakage locations,
    - Determine percentage of operational hours that are contribute by each included part-breakage location.
      - Calculate a cumulative list of these percentages in equal proportionality
    - For 0 to the demand value:
      - Generate a random value between 0 and 1
      - Assign to part-breakage location based on where random value falls on cumulative percentage table

4.5 Summary

The model decides repair inductions based upon the current state of the system and forecasted part-breakages input into the model. Therefore, the data plays an important role in all stages of the model: formulation, running, solution, and implementation. Both the level of detail of data available as well as the quality of the data impact the model formulation as well as solution quality.
5.1 Introduction

Using the case study model formulation detailed below and data considerations mentioned in Chapter 4, a mixed integer program was developed in General Algebraic Modeling System (GAMS) calling CPLEX as the solver. With inputs including the current state of the system and various reparable supply chain parameters, the program generates real-time repair recommendations that align with customer service level goals. Various constraints may be used to tailor the model to more accurately represent desired part flow through the service parts supply chain.

Repair is prioritized with an aim towards maximizing issue effectiveness and subsystem availability. With that in mind, objective function parameter values were determined for a case study base model formulation. The impact of budget was also investigated. Integrated decision-making provides an opportunity to make the most of limited repair capabilities.

5.2 Case Study Model Modifications

The case study data examined in this research was obtained from a large firm operating an international service parts supply chain. Based upon the case study data, the general mathematical formulation from Chapter 3 was modified to more accurately represent the reparable system. Changes in the case study base model formulation are noted and explained. The main reasons for model adjustments include accounting for
detail-level of data available and accurately modeling actual service parts supply chain operation to provide overall model simplification.

The general model was modified in several ways. Sets were removed, parameters and decision variables re-indexed, and equations were adjusted accordingly. Note, the objective function did not change. These modifications adjusted the model to represent a single repair depot, a centralized warehouse, and multiple field stocking locations. In the model, all parts completing repair at the depot are sent directly to a single centralized warehouse from which they can be shipped to any FSL. The modified repair supply chain is shown in Figure 5.1.

![Case Study Repair Cycle Diagram](image)

Figure 5.1 Case Study Repair Cycle Diagram

Sets K and R were removed from the case study model formulation. Set K contained all repair facilities considered by the model. Set R contained available repair methods: normal and expedited. These sets are unnecessary for the final formulation as the case study data included only average repair times and costs with no available information on repair expediting. Exhaustive repair allocation decisions among repair
depots is not of primary interest and does not provide additional functionality for the system examined in the case study.

Table 5.1 Parameter Modifications

<table>
<thead>
<tr>
<th>General Model Formulation</th>
<th>Case Study Formulation</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>repaircost(R,I,K)</td>
<td>repaircost(I)</td>
<td>Removed sets R and K.</td>
</tr>
<tr>
<td>repairstime(R,I,K)</td>
<td>repairstime(I)</td>
<td>Removed sets R and K.</td>
</tr>
<tr>
<td>transitcost(I,JW,K)</td>
<td>Removed</td>
<td>Model case study system.</td>
</tr>
<tr>
<td>transfercost(I,JW,J)</td>
<td>Removed</td>
<td>Model case study system.</td>
</tr>
<tr>
<td>percent(K)</td>
<td>percent</td>
<td>Removed set K.</td>
</tr>
<tr>
<td>maxwork(I,K)</td>
<td>maxwork(I)</td>
<td>Removed set K.</td>
</tr>
<tr>
<td>transittime(JW,K)</td>
<td>transittime</td>
<td>Removed set K and model case study system.</td>
</tr>
<tr>
<td>transfertime(JW,J)</td>
<td>transfertime(J)</td>
<td>Model case study system.</td>
</tr>
<tr>
<td>z_init(R,I,K,T)</td>
<td>z_init(I,T)</td>
<td>Removed sets R and K.</td>
</tr>
<tr>
<td>l_init(I,JW,J,T)</td>
<td>l_init(I,J,T)</td>
<td>Model case study system.</td>
</tr>
<tr>
<td>a_init(I,JW,K,T)</td>
<td>Removed</td>
<td>Removed set K and model case study system.</td>
</tr>
</tbody>
</table>

Repair parameters were reduced in index in the case study formulation after removing sets R and K. Shipping and repair were combined in the case study system so were not considered separately in the model. This includes both costs and times. Repair, shipping from a FSL J to the repair facility, and shipping to the centralized warehouse after repair is finished is all accounted for in repair times and costs. Repair facility constraints were reduced in index in the case study model formulation to repair supply chain actual operation or acceptable simplification. Additionally, lateral shipments were disallowed between FSLs matching the case study repair supply chain operation.
Table 5.2  Decision Variable Modifications

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>General Formulation</th>
<th>Decision Case Study Formulation</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>z (R, I, J, K, T)</td>
<td>z (I, J, T)</td>
<td>Removed sets R and K.</td>
<td></td>
</tr>
<tr>
<td>a (I, J, W, K, T)</td>
<td>Removed</td>
<td>Removed set K and model case study system.</td>
<td></td>
</tr>
<tr>
<td>rep(I, K, T)</td>
<td>rep(I, T)</td>
<td>Removed set K.</td>
<td></td>
</tr>
</tbody>
</table>

5.2.1 Case Study Base Model Formulation

Objective Function

\[
\text{fillrate} = \sum_{J} \text{weight}_{j} \times \left[ \begin{array}{c}
\text{spltavg} \\
\text{pavg} \\
(1 - \text{pavg}) \\
\text{spltmin}
\end{array} \right] \times \left[ \begin{array}{c}
\left( \frac{\Sigma_{T \in |h_{IJ}>0} \text{fill}_{IJ} \cdot \text{weight}_{IJ}}{\text{card}(T) \cdot |g_{IJ}>0|} \right) \\
\left( \frac{\Sigma_{T \in |g_{IJ}>0} \text{fill}_{IJ} \cdot \text{weight}_{IJ}}{\text{card}(T) \cdot |g_{IJ}>0|} \right) \\
\left( \frac{\Sigma_{T \in |g_{IJ}>0} \text{fill}_{IJ} \cdot \text{weight}_{IJ}}{\text{card}(T) \cdot |g_{IJ}>0|} \right) \\
\left( \frac{\Sigma_{T \in |h_{IJ}>0} \text{minavg}_{J}}{\text{card}(T) \cdot |h_{IJ}>0|} \right)
\end{array} \right]
\]

\[f_{ijt} = \begin{cases} 
\text{demand}_{ijt} & tsl_{ij} = 0 \\
0 & tsl_{ij} \neq 0 
\end{cases} \quad \forall i, \forall j, \forall T \tag{3.5}
\]

\[g_{ijt} = tsl_{ij} + \text{demand}_{ijt} \quad \forall i, \forall j, \forall T \tag{3.6}
\]

\[h_{ij} = tsl_{ij} + \sum_{T} \text{demand}_{ijt} \quad \forall i, \forall j, \forall T \tag{3.7}
\]

\[\text{minavg}_{J} \leq \frac{\Sigma_{T \in |g_{IJ}>0} \text{fill}_{IJ} \cdot \text{weight}_{IJ}}{\text{card}(T) \cdot |g_{IJ}>0|} \quad \forall i, \forall j \tag{3.10}
\]
\[ \text{avgmin}_{JT} \leq \frac{\text{weight}_{IJ} + \text{fill}_{IJT}}{\text{tsl}_{IJ} + f_{IJT}} \quad \forall I, \forall J, \forall T \]  
(3.11)

\[ \text{fill}_{IJT} \leq \text{stock}_{IJT} \quad \forall I, \forall J, \forall T \]  
(3.1)

\[ \text{fill}_{IJT} \leq \text{tsl}_{IJ} + f_{IJT} \quad \forall I, \forall J, \forall T \]  
(3.2)

Constraints

\[ \sum_{I} l_{IJT} \leq \text{stock}_{IWT} \quad \forall I, \forall W, \forall T \]  
(5.1)

\[ \text{stock}_{IJ(T+1)} = \left[ \text{stock}_{IJT} - \text{demand}_{IJT} + \text{l}_{IJ(T-\text{transfertime}_{J})} + \text{l}_{\text{init}_{IJT}} \right] \quad \forall I, \forall J, \forall T \]  
(5.2)

\[ \text{stock}_{I(W+1)} = \text{stock}_{IWT} - \sum_{I} \left[ -z_{IJ[T-\text{repairtime}_{I}]} + \frac{l_{IJT}}{\text{transsittime}_{I}} \right] \quad \forall I, \forall W, \forall T \]  
(5.3)

\[ d_{IJ(T+1)} = d_{IJT} + \text{demand}_{IJT} - z_{IJ} \quad \forall I, \forall J, \forall T \]  
(5.4)

\[ d_{IJ(T=1)} = d_{\text{init}_{IJ}} \quad \forall I, \forall J \]  
(5.5)

\[ \text{rep}_{I(T=1)} = \sum_{T} z_{\text{init}_{IT}} \quad \forall I, \forall J, \forall T \]  
(5.6)

\[ \text{stock}_{IJW(T=1)} = \text{stock}_{\text{init}_{IJW}} \quad \forall I, \forall J, \forall W \]  
(5.7)

\[ \text{fillrate} \leq 1 \]  
(5.8)

\[ z(I, J, T) \in \mathbb{Z}, l(I, J, T) \in \mathbb{Z} \quad \forall I, \forall J, \forall T \]  
(5.9)

\[ z(I, J, T), l(I, J, T), d(I, J, T), \text{rep}(I, J, T) \geq 0 \quad \forall I, \forall J, \forall T \]  
(5.10)

\[ \text{stock}(I, W, T) \geq 0 \quad \forall I, \forall W, \forall T \]  
(5.11)

\[ \text{stock}(I, J, T) \quad \forall I, \forall J, \forall T \quad \text{unrestricted} \]  
(5.12)

\[ \text{fill}(I, J, T) \quad \forall I, \forall J, \forall T \quad \text{unrestricted} \]  
(5.13)

\[ \text{fillrate} \quad \text{unrestricted} \]  
(5.14)

\[ \text{minavg}(J) \quad \forall J \quad \text{unrestricted} \]  
(5.15)

\[ \text{avgmin}(J, T) \quad \forall J, \forall T \quad \text{unrestricted} \]  
(5.16)
Optional Constraints

\[ \text{budget}_P \geq \sum_{T \in \text{periodend}(P-1)} \sum_{t \subseteq \text{periodend}(P)} \sum_{I} \sum_{J} \left[ z_{IJT} \times \text{repaircost}_I \right] \quad \forall P \] (5.17)

\[ z_{IJT} \leq \max \left[ 1, \left\lceil \frac{\left[ \sum_{T} \sum_{I} \text{demand}_{IJT} \times \text{percent} \right]}{\sum_{T} \text{ts}_{IJ}} \times \text{percent} \right\rceil \right] \quad \forall I, \forall J, \forall T \] (5.18)

\[ \text{rep}_{IT} \leq \text{maxwork}_I \quad \forall I, \forall T \] (5.19)

\[ \text{rep}_{I(T+1)} = \text{rep}_{IT} + \sum_{J} \left[ z_{IJT} - z_{IJ(T-\text{repaitrttime}_{RIK})} \right] - z_{\text{ini}_{IT}} \quad \forall I, \forall T \] (5.20)

5.2.2 Output Calculations

Sets

- **M** – set of all month periods considered in model
- **RN** – set of number of repair induced in a single time unit T

Parameters

- **MetDemand(I,J,M)** – number of times component type I at FSL J broke and was immediately replaced in month M
- **BackOrders(I,J,M)** – number of times component type I at FSL J in month M broke and was not immediately replaced
- **IssueEff(J,M)** – calculate percentage of part-breakages at FSL J in month M in which component was in stock
- **SystemUnavail(J,M)** – number of potential sub-systems unavailable at FSL J in month M due to a serviceable part being unavailable
- **Repairs(J,M)** – number of parts from FSL J sent to repair in month M
- **Deferrals (J,M)** – number of parts that fail at FSL J in month M and are not immediately induced into repair
• Shipped(J,M) – number of serviceable parts sent to (shipped) FSL J in month M from the warehouse

• CostPerPeriod(P) – total cost of all repair related expenses in budget period P

• AvgFillByT(T) – average part fill rate in system at time T

• wt – weighting by part type and FSL for impact calculation

• enddate – last day to consider for repair prioritization/sequencing

• numrepairs(I,T) – total number of component type I inducted into repair from all FSLs at time T to sequence for repair

• impact(I,RN,T) – total number of component type I inducted into repair from all FSLs at time T to sequence for repair

Post-Optimization Calculations

\[
\text{MetDemand}_{IJM} = \sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P)\}} \left\{ \begin{array}{ll}
\text{demand}_{IJT} & \text{stock}_{I,J,T} \geq \text{demand}_{IJT} \\
\text{stock}_{IJT} - \text{demand}_{IJT} & 0 < \text{stock}_{IJT} < \text{demand}_{IJT} \\
0 & \text{stock}(I,J,T) \leq 0 \\
\end{array} \right. \\
\forall I, \forall J, \forall M (5.21)
\]

\[
\text{BackOrders}_{IJM} = \sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P)\}} \left\{ \begin{array}{ll}
\text{demand}_{IJT} & \text{stock}_{IJT} \leq 0 \\
\text{demand}_{IJT} - \text{stock}_{IJT} & 0 < \text{stock}_{IJT} < \text{demand}_{IJT} \\
0 & \text{stock}_{IJT} \geq \text{demand}_{IJT} \\
\end{array} \right. \\
\forall I, \forall J, \forall M (5.22)
\]

\[
\text{IssueEff}_{JM} = \frac{\sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P)\}} \text{MetDemand}(I,J,M)}{\sum_{I,T} \text{demand}(I,J,T)} \forall J, \forall M (5.23)
\]

\[
\text{SystemUnavail}_{JM} = \max_{I, T} \left\{ \begin{array}{ll}
\text{demand}_{IJT} & \text{stock}_{IJT} \leq 0 \\
\text{demand}_{IJT} - \text{stock}_{IJT} & 0 < \text{stock}_{IJT} < \text{demand}_{IJT} \\
0 & \text{stock}_{IJT} \geq \text{demand}_{IJT} \\
\end{array} \right. \\
\forall J, \forall M (5.24)
\]

\[
\text{Repairs}_{M} = \sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P)\}} \sum_{I} \sum_{J} z_{IJT} \forall M (5.25)
\]
\[ \text{Deferrals}_M = \sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P) \}} \sum_i \sum_j dl_{IJT} \quad \forall M \] (5.26)

\[ dl_{IJT} = \begin{cases} \text{demand}_{IJT} - z_{IJT} & \text{demand}_{IJT} \geq z_{IJT} \\ 0 & \text{demand}_{IJT} < z_{IJT} \end{cases} \quad \forall I, \forall J, \forall T \] (5.27)

\[ \text{Shipped}_M = \sum_{T \in \{t > \text{monthend}(P-1) \text{ and } t \leq \text{monthend}(P) \}} \sum_i \sum_j l_{IJT} \quad \forall M \] (5.28)

\[ \text{CostPerPeriod}_P = \sum_{T \in \{t > \text{periodend}(P-1) \text{ and } t \leq \text{periodend}(P) \}} \sum_i \sum_j \left[ z_{IJT} \ast \text{repaircost}_i \right] \forall P \] (5.29)

\[ \text{AvgFillBy}_{TT} = \frac{\sum_i \sum_j \left[ \text{weight}_I \ast \text{weight}_J \ast \text{fill}_{IJT} \right]}{\text{card}(I,J) \in [g_{IJT} > 0]} \] (5.30)

\[ wt = \text{weight}_I \ast \sum_{j \in [h_{IJ} > 0]} \text{weight}_J \] (5.31)

\[ \text{enddate} = \min[\text{MonthEnd('pd2'), PeriodEnd('pd1')} \] (5.32)

\[ \text{numrepairs}_{IJ} = \sum_j \left[ z_{IJT} \right] \quad \forall T \in [t < \text{enddate}] \quad \forall I, \forall T \] (5.33)

\[ \text{impact}_{IJRT} = 0 \quad \forall T \in [t < \text{enddate}], \forall I, \forall RN \] (5.34)

\[ \text{impact}_{IJRT} = 0 \quad \forall RN \in [rn > \text{numrepairs}_{IJ}], \forall I, \forall T \] (5.35)

\[ \text{impact}_{IJRT} = \frac{wt \ast \sum_{j \in [h_{IJ} > 0]} \sum_{TT \in [h_{IJ} > 0]} \left[ \text{demand}_{IJTT} \ast \text{fill}_{IJTT} \right]}{\sum_{j} \sum_{TT \in [h_{IJ} > 0]} \left[ \text{demand}_{IJTT} \right]} \quad \forall I, \forall RN, \forall T \] (5.36)

### 5.3 Objective Parameter Determination

The model’s objective function was developed keeping in mind customer service level performance. The repair schedule is optimized through indirectly considering issue effectiveness and sub-system availability in the repair supply chain. In order to achieve this goal, the objective function parameter values need to be determined. In this section, pavg, pmin, spltmin, and spltavg parameter values will be set based upon the
performance of the model for the case study data. The model was run for all possible combinations of the considered objective parameter values. A single demand forecast was used for all of these model runs.

### 5.3.1 Maximize Average Components

The parameter $p_{avg}$ weights the two components of the maximizing average equation of the objective function. The higher the value of $p_{avg}$, the more weight is placed upon the average long-term performance. This is shown in the objective function components shown in Equations 5.37 and 5.38 below. By maximizing the below calculated average comparative fill rates, issue effectiveness can be approximately maximized.

\[
\text{Average Time – Averaged Fill Rate} = \sum_j \text{weight}_j \cdot p_{avg} \cdot \frac{\sum_{\text{time}[h_{IJ}>0]} \text{weight}_j \cdot \text{fill}_{IJ}}{\text{card}(\text{time})[h_{IJ}>0]} \tag{5.37}
\]

\[
\text{Time – Averaged Average Fill Rate} = \sum_j \left[ \frac{\text{weight}_j \cdot (1 - p_{avg}) \cdot \sum_{\text{time}[h_{IJ}>0]} \text{weight}_j \cdot \text{fill}_{IJ}}{\text{card}(\text{time})[h_{IJ}>0]} \right] \tag{5.38}
\]

A value of $p_{avg} = 0.8$ performed best overall in terms of average issue effectiveness for all considered combinations of $p_{min}$ and $s_{pltmin}$. Issue effectiveness was considered based upon performance across the model time horizon and across field stocking locations (FSLs). The average monthly performance for each considered $p_{avg}$ value is shown in Figure 5.2. $p_{avg} = 0.8$ performs the best overall across all month periods in the time horizon as shown in Figure 5.3.
Average Monthly Issue Effectiveness

<table>
<thead>
<tr>
<th>Month</th>
<th>pavg 0.0</th>
<th>pavg 0.2</th>
<th>pavg 0.4</th>
<th>pavg 0.6</th>
<th>pavg 0.8</th>
<th>pavg 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>80.16%</td>
<td>80.74%</td>
<td>81.60%</td>
<td>81.57%</td>
<td>81.89%</td>
<td>82.40%</td>
</tr>
<tr>
<td>M2</td>
<td>82.58%</td>
<td>84.41%</td>
<td>84.23%</td>
<td>84.08%</td>
<td>84.44%</td>
<td>84.31%</td>
</tr>
<tr>
<td>M3</td>
<td>94.34%</td>
<td>94.39%</td>
<td>94.39%</td>
<td>94.39%</td>
<td>94.39%</td>
<td>94.39%</td>
</tr>
<tr>
<td>M4</td>
<td>81.78%</td>
<td>82.45%</td>
<td>82.40%</td>
<td>82.45%</td>
<td>82.54%</td>
<td>82.54%</td>
</tr>
<tr>
<td>M5</td>
<td>91.83%</td>
<td>91.61%</td>
<td>91.40%</td>
<td>91.61%</td>
<td>91.07%</td>
<td>91.18%</td>
</tr>
<tr>
<td>M6</td>
<td>83.24%</td>
<td>82.52%</td>
<td>82.56%</td>
<td>82.56%</td>
<td>82.56%</td>
<td>82.56%</td>
</tr>
<tr>
<td>M7</td>
<td>91.26%</td>
<td>90.73%</td>
<td>90.73%</td>
<td>90.73%</td>
<td>90.73%</td>
<td>90.73%</td>
</tr>
<tr>
<td>M8</td>
<td>81.57%</td>
<td>80.91%</td>
<td>80.64%</td>
<td>80.74%</td>
<td>80.84%</td>
<td>80.73%</td>
</tr>
<tr>
<td>M9</td>
<td>86.99%</td>
<td>86.50%</td>
<td>86.07%</td>
<td>86.50%</td>
<td>86.41%</td>
<td>86.43%</td>
</tr>
<tr>
<td>M10</td>
<td>89.93%</td>
<td>89.64%</td>
<td>89.63%</td>
<td>89.69%</td>
<td>89.55%</td>
<td>89.41%</td>
</tr>
<tr>
<td>M11</td>
<td>91.37%</td>
<td>90.75%</td>
<td>90.17%</td>
<td>90.21%</td>
<td>90.09%</td>
<td>90.17%</td>
</tr>
<tr>
<td>M12</td>
<td>89.26%</td>
<td>89.13%</td>
<td>89.14%</td>
<td>89.12%</td>
<td>89.34%</td>
<td>89.27%</td>
</tr>
<tr>
<td>M13</td>
<td>98.53%</td>
<td>97.66%</td>
<td>98.11%</td>
<td>97.86%</td>
<td>98.15%</td>
<td>97.74%</td>
</tr>
<tr>
<td>M14</td>
<td>95.47%</td>
<td>94.51%</td>
<td>94.07%</td>
<td>94.14%</td>
<td>94.27%</td>
<td>94.32%</td>
</tr>
</tbody>
</table>

Figure 5.2 Time-Based Issue Effectiveness for Pavg Values

Monthly Issue Effectiveness

<table>
<thead>
<tr>
<th>pavg</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>88.58%</td>
<td>88.42%</td>
<td>88.35%</td>
<td>88.39%</td>
<td>88.44%</td>
<td>88.42%</td>
</tr>
<tr>
<td>Maximum</td>
<td>98.53%</td>
<td>97.66%</td>
<td>98.11%</td>
<td>97.86%</td>
<td>98.15%</td>
<td>97.74%</td>
</tr>
<tr>
<td>Minimum</td>
<td>80.16%</td>
<td>80.74%</td>
<td>80.64%</td>
<td>80.74%</td>
<td>80.84%</td>
<td>80.73%</td>
</tr>
<tr>
<td>σ</td>
<td>5.61%</td>
<td>5.25%</td>
<td>5.22%</td>
<td>5.18%</td>
<td>5.14%</td>
<td>5.07%</td>
</tr>
<tr>
<td>Range</td>
<td>18.38%</td>
<td>16.92%</td>
<td>17.47%</td>
<td>17.12%</td>
<td>17.31%</td>
<td>17.01%</td>
</tr>
</tbody>
</table>

Figure 5.3 Summary: Time-Based Issue Effectiveness for Pavg Values

The average monthly performance for each considered pavg value is shown in Figure 5.4. Field stocking location issue effectiveness performance widely varied. Pavg = 0.8 performs adequately across FSLs. Figure 5.5.
Average FSL Issue Effectiveness

<table>
<thead>
<tr>
<th>FSL</th>
<th>pavg 0.0</th>
<th>pavg 0.2</th>
<th>pavg 0.4</th>
<th>pavg 0.6</th>
<th>pavg 0.8</th>
<th>pavg 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSL 1</td>
<td>84.75%</td>
<td>83.95%</td>
<td>84.03%</td>
<td>84.03%</td>
<td>84.03%</td>
<td>84.03%</td>
</tr>
<tr>
<td>FSL 2</td>
<td>75.00%</td>
<td>75.00%</td>
<td>74.81%</td>
<td>75.00%</td>
<td>75.00%</td>
<td>75.00%</td>
</tr>
<tr>
<td>FSL 3</td>
<td>87.01%</td>
<td>86.94%</td>
<td>86.97%</td>
<td>87.22%</td>
<td>87.19%</td>
<td>87.41%</td>
</tr>
<tr>
<td>FSL 4</td>
<td>95.36%</td>
<td>94.01%</td>
<td>94.16%</td>
<td>94.04%</td>
<td>94.09%</td>
<td>94.18%</td>
</tr>
<tr>
<td>FSL 5</td>
<td>90.06%</td>
<td>89.96%</td>
<td>89.96%</td>
<td>90.06%</td>
<td>90.06%</td>
<td>90.06%</td>
</tr>
<tr>
<td>FSL 6</td>
<td>96.58%</td>
<td>96.34%</td>
<td>96.41%</td>
<td>96.30%</td>
<td>96.32%</td>
<td>96.18%</td>
</tr>
<tr>
<td>FSL 7</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>FSL 8</td>
<td>90.20%</td>
<td>89.93%</td>
<td>89.65%</td>
<td>89.43%</td>
<td>89.87%</td>
<td>89.84%</td>
</tr>
<tr>
<td>FSL 9</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>FSL 10</td>
<td>93.60%</td>
<td>95.62%</td>
<td>95.45%</td>
<td>95.34%</td>
<td>95.51%</td>
<td>95.40%</td>
</tr>
<tr>
<td>FSL 11</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>FSL 12</td>
<td>61.11%</td>
<td>60.22%</td>
<td>60.22%</td>
<td>60.64%</td>
<td>60.27%</td>
<td>60.43%</td>
</tr>
<tr>
<td>FSL 13</td>
<td>94.95%</td>
<td>94.50%</td>
<td>94.34%</td>
<td>94.42%</td>
<td>94.50%</td>
<td>94.50%</td>
</tr>
<tr>
<td>FSL 14</td>
<td>61.27%</td>
<td>61.37%</td>
<td>60.48%</td>
<td>60.61%</td>
<td>60.52%</td>
<td>60.50%</td>
</tr>
<tr>
<td>FSL 15</td>
<td>51.89%</td>
<td>51.89%</td>
<td>51.89%</td>
<td>51.89%</td>
<td>51.89%</td>
<td>51.89%</td>
</tr>
<tr>
<td>FSL 16</td>
<td>87.50%</td>
<td>87.50%</td>
<td>87.50%</td>
<td>87.50%</td>
<td>87.50%</td>
<td>87.50%</td>
</tr>
<tr>
<td>FSL 17</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>FSL 18</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.91%</td>
</tr>
</tbody>
</table>

Figure 5.4  FSL Issue Effectiveness for Pavg Values

FSL Issue Effectiveness

<table>
<thead>
<tr>
<th>pavg</th>
<th>Average</th>
<th>Maximum</th>
<th>Minimum</th>
<th>σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>88.58%</td>
<td>100.00%</td>
<td>50.00%</td>
<td>16.54%</td>
</tr>
<tr>
<td>0.2</td>
<td>88.42%</td>
<td>100.00%</td>
<td>50.00%</td>
<td>16.60%</td>
</tr>
<tr>
<td>0.4</td>
<td>88.35%</td>
<td>100.00%</td>
<td>50.00%</td>
<td>16.66%</td>
</tr>
<tr>
<td>0.6</td>
<td>88.39%</td>
<td>100.00%</td>
<td>50.00%</td>
<td>16.66%</td>
</tr>
<tr>
<td>0.8</td>
<td>88.44%</td>
<td>100.00%</td>
<td>50.00%</td>
<td>16.66%</td>
</tr>
<tr>
<td>1.0</td>
<td>88.42%</td>
<td>100.00%</td>
<td>50.91%</td>
<td>16.54%</td>
</tr>
</tbody>
</table>

Figure 5.5  Summary: FSL Issue Effectiveness for Pavg Values

5.3.2  Maximize Minimum Components

The parameter pmin weights the two components of the maximizing minimum equation of the objective function. The higher the value of pmin, the more weight is placed upon the average long-term performance. This is shown in the objective function.
components shown in Equations 5.39 and 5.40 below. By maximizing the below calculated minimum comparative fill rates, sub-system availability is considered in the objective function.

\[
\text{Minimum Time} \text{ – Averaged} = \sum_j \left[ \text{weight}_j \cdot \text{pmin} \cdot \text{minavg}_{j}\right]
\] (5.39)

\[
\text{Time} \text{ – Averaged Minimum Fill Rate} = \sum_j \left[ \text{weight}_j \cdot (1 - \text{pmin}) \cdot \frac{\sum_{T \in [h_{ij} > 0]} \text{avgmin}_{T}}{\text{card}(T) \in [h_{ij} > 0]} \right]
\] (5.40)

A value of \( p_{min} = 0.8 \) performed best overall in terms of average issue effectiveness for all values of \( \text{pavg} \) and \( \text{splitmin} \). Issue effectiveness was considered based upon performance across the model time horizon and across field stocking locations (FSLs). The average monthly performance for each considered \( p_{min} \) value is shown in Figure 5.6. \( p_{min} = 0.8 \) performs the best overall across all month periods in the time horizon as shown in Figure 5.6.

<table>
<thead>
<tr>
<th>Month</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>81.42%</td>
<td>81.13%</td>
<td>81.53%</td>
<td>81.31%</td>
<td>81.42%</td>
<td>81.53%</td>
</tr>
<tr>
<td>M2</td>
<td>84.30%</td>
<td>83.86%</td>
<td>84.23%</td>
<td>83.96%</td>
<td>83.81%</td>
<td>83.88%</td>
</tr>
<tr>
<td>M3</td>
<td>94.39%</td>
<td>94.38%</td>
<td>94.38%</td>
<td>94.38%</td>
<td>94.38%</td>
<td>94.38%</td>
</tr>
<tr>
<td>M4</td>
<td>82.64%</td>
<td>82.31%</td>
<td>82.35%</td>
<td>82.26%</td>
<td>82.31%</td>
<td>82.31%</td>
</tr>
<tr>
<td>M5</td>
<td>91.72%</td>
<td>91.61%</td>
<td>91.29%</td>
<td>91.18%</td>
<td>91.18%</td>
<td>91.72%</td>
</tr>
<tr>
<td>M6</td>
<td>82.63%</td>
<td>82.67%</td>
<td>82.67%</td>
<td>82.67%</td>
<td>82.67%</td>
<td>82.67%</td>
</tr>
<tr>
<td>M7</td>
<td>90.72%</td>
<td>90.84%</td>
<td>90.84%</td>
<td>90.84%</td>
<td>90.84%</td>
<td>90.84%</td>
</tr>
<tr>
<td>M8</td>
<td>80.61%</td>
<td>81.22%</td>
<td>80.88%</td>
<td>80.86%</td>
<td>80.98%</td>
<td>80.89%</td>
</tr>
<tr>
<td>M9</td>
<td>86.37%</td>
<td>86.43%</td>
<td>86.32%</td>
<td>86.54%</td>
<td>86.64%</td>
<td>86.61%</td>
</tr>
<tr>
<td>M10</td>
<td>89.49%</td>
<td>89.63%</td>
<td>89.57%</td>
<td>89.56%</td>
<td>89.73%</td>
<td>89.87%</td>
</tr>
<tr>
<td>M11</td>
<td>90.28%</td>
<td>90.50%</td>
<td>90.61%</td>
<td>90.50%</td>
<td>90.50%</td>
<td>90.38%</td>
</tr>
<tr>
<td>M12</td>
<td>88.98%</td>
<td>89.17%</td>
<td>89.25%</td>
<td>89.24%</td>
<td>89.32%</td>
<td>89.30%</td>
</tr>
<tr>
<td>M13</td>
<td>97.88%</td>
<td>98.13%</td>
<td>97.97%</td>
<td>98.11%</td>
<td>97.89%</td>
<td>98.07%</td>
</tr>
<tr>
<td>M14</td>
<td>94.71%</td>
<td>94.42%</td>
<td>94.29%</td>
<td>94.40%</td>
<td>94.46%</td>
<td>94.47%</td>
</tr>
</tbody>
</table>

Figure 5.6 Time-Based Issue Effectiveness for \( p_{min} \) Values

82
Overall p\textsubscript{min}=0.8 performs best across all field stocking locations. The average monthly performance for each considered p\textsubscript{min} value is shown in Figure 5.8. Field stocking location issue effectiveness performance widely varied. P\textsubscript{min} = 0.8 performs adequately across FSLs as shown in Figure 5.9.
<table>
<thead>
<tr>
<th>pmin</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>88.40%</td>
<td>88.43%</td>
<td>88.42%</td>
<td>88.41%</td>
<td>88.44%</td>
<td>88.49%</td>
</tr>
<tr>
<td>Maximum</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Minimum</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
<td>50.00%</td>
</tr>
<tr>
<td>σ</td>
<td>16.74%</td>
<td>16.54%</td>
<td>16.61%</td>
<td>16.60%</td>
<td>16.59%</td>
<td>16.53%</td>
</tr>
</tbody>
</table>

Figure 5.9 Summary: FSL Issue Effectiveness for Pmin Values

5.3.3 Maximize Average vs Maximize Minimum

The parameter `spltmin` weights the objective function between maximizing average fill rate and maximizing the minimum rate at each field stocking location. This weighting allows the model to consider both issue effectiveness and sub-system availability when making repair recommendations.

Issue effectiveness is typically improved the more the objective function is weighted towards maximizing the average. A combination of the maximizing the average and the minimum performs best though. Issue effectiveness performance is the worst if the objective function focuses only on maximizing average fill rate alone. Overall, though FSLs on average did best for lower values of `spltmin` weighting towards maxing average fill rate. If using only the average with no consideration for maximizing the minimum fill rate, the model performs the worst over all categories with the lowest minimum, largest standard deviation, and the lowest average issue effectiveness. Issue effectiveness is shown by month in Figure 5.10 and across FSLs in Figure 5.11.
Ultimately, the objective function parameter values were set based upon issue effectiveness performance. Their values throughout all following model runs are as follows: \( s\text{pltmin} = 0.1, p\text{avg} = 0.8, \text{ and } p\text{min} = 0.8 \). The model performed best with lower, but not zero, values of \( s\text{pltmin} \). This value was in the top 10\% of average issue effectiveness for the parameter values shown in Figure 5.12.
### Average Issue Effectiveness

<table>
<thead>
<tr>
<th>spltmin</th>
<th>pavg</th>
<th>0.0</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0</td>
<td>89.1%</td>
<td>89.2%</td>
<td>89.6%</td>
<td>89.6%</td>
<td>89.5%</td>
<td>89.5%</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>89.5%</td>
<td>89.7%</td>
<td>89.2%</td>
<td>89.5%</td>
<td>89.5%</td>
<td>89.7%</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>89.4%</td>
<td>89.4%</td>
<td>89.4%</td>
<td>89.6%</td>
<td>89.7%</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>89.7%</td>
<td>89.4%</td>
<td>88.9%</td>
<td>89.6%</td>
<td>89.4%</td>
<td>89.7%</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>89.6%</td>
<td>89.4%</td>
<td>89.3%</td>
<td>89.6%</td>
<td>89.7%</td>
<td>89.3%</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>89.7%</td>
<td>89.7%</td>
<td>89.7%</td>
<td>88.7%</td>
<td>89.3%</td>
<td>89.7%</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0</td>
<td>88.7%</td>
<td>89.3%</td>
<td>89.3%</td>
<td>89.3%</td>
<td>89.1%</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>89.7%</td>
<td>89.7%</td>
<td>89.1%</td>
<td>89.6%</td>
<td>89.7%</td>
<td>89.1%</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>89.6%</td>
<td>89.6%</td>
<td>89.6%</td>
<td>89.6%</td>
<td>89.5%</td>
<td>89.5%</td>
</tr>
<tr>
<td></td>
<td>0.6</td>
<td>89.8%</td>
<td>89.2%</td>
<td>89.4%</td>
<td>89.7%</td>
<td>90.0%</td>
<td>89.4%</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>89.5%</td>
<td>89.9%</td>
<td>89.4%</td>
<td>89.5%</td>
<td>89.2%</td>
<td>89.5%</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>89.7%</td>
<td>89.5%</td>
<td>89.7%</td>
<td>89.1%</td>
<td>89.4%</td>
<td>89.6%</td>
</tr>
</tbody>
</table>

**Figure 5.12**  Issue Effectiveness for Varying Objective Function Parameter Values

The objective function parameter values determine the weighting of each objective function component. Spltmin = 0.1 weights the objective 90% towards issue effectiveness and 10% towards sub-system availability. Pavg and pmin both equal to 0.80 weights the objective 80% towards long-term performance and 20% towards the short-term. This is shown in Figure 5.13 below.

![Diagram showing issue effectiveness and sub-system availability weights](image)

**Figure 5.13**  Summary: Selected Objective Function Component Weighting
5.4 Case Study Model

Model performance was analyzed using the same model formulation and case study data previously used to determine objective function parameter values. The flow of parts through the system was analyzed using the average number of parts of each component sent to repair, deferred, and shipped for each month in the time horizon. Average part fill rate, issue effectiveness, and sub-system availability were all used to determine model performance. The impacts of budget size and budget period length were also examined. Lastly, examples of model outputs including recommended weekly repair list and a prioritized repair sequence are presented.

The case study model included the following: 1619 components, 22 field stocking locations, 1 warehouse, and 1 repair depot. Equal monthly budgets were used for all models unless mentioned otherwise. Weekly time units were used for set T.

5.4.1 Base Model Results

Parts flowing through the case study mathematical model include repair, deferrals, and shipments of serviceable parts. The average size of these part flows fluctuates with time. In the early months of the model, the average number of parts of each component type sent to repair starts low and increases until halfway through the time horizon. As initial part imbalances in the system are addressed, the breadth of components repaired decreases while the number of parts of each component increases. As time passes, shipment size increases as more parts finish repair and are sent to field stocking locations. Average size of repair deferrals remains level throughout the model time horizon. This is shown in Figure 5.14.
The model generated results which improved the average part fill rate in the system over the model time horizon. Part allocations at the start of the model create an initial spike in average fill rate. As time passes and repairs are completed, average fill rate steadily improves. The percentage increase in AvgFillByT(T) from the start of the model is shown in Figure 5.15.
Average monthly issue effectiveness fluctuated at the start of the model before trending upward about halfway through the time horizon. As time passes, inventory is rebalanced as parts unserviceable at the start of the model are able to be repaired and rejoin the pool of available serviceable parts. This is the cause for the improvement in issue effectiveness in later months shown in Figure 5.16.

![Average FSL Issue Effectiveness by Month](image)

Figure 5.16  Average Issue Effectiveness by Month

Issue effectiveness decreased at the start of the model until halfway through when it steadily improved for the remainder of the time horizon. The initial drop in issue effectiveness is due to part failures occurring before the model has time to repair and allocate enough serviceable parts to meet these demands. As time passes, repairs are completed and issue effectiveness improves. This is shown in Figure 5.17.
Similar to issue effectiveness performance, sub-system availability improves over time. Potential unavailable sub-systems are determined based on the maximum number of backorders occurring at time T. Early in the time horizon, the model is unable to make repairs in time to meet part failures. As time passes, the number of backorders decreases and sub-system availability improves as shown in Figure 5.18.
The case study model results indicate that parts are flowing through the system as desired. Model performance shows the objective function is succeeding at improving service levels. Average system fill rate improved by more than 5% over the course of the model time horizon. Also model results indicate improvements in issue effectiveness and sub-system availability over the time. Part failures that occur near the start of the model can have a negative impact on initial performance. The model does not have enough time to repair a part to meet the demand, and so a backorder occurs. Once repairs have time to get through the system, service level performance is improved and backorders are reduced.

5.4.2 Budget Size

The model was also run for various budget amounts. The base monthly model, monthly model with half the original budget, monthly model with twice the original budget, and a model run without the budget constraint were all considered. Model performance for average part fill rate, issue effectiveness, and sub-system availability were all examined.

The model generated results which improved the average part fill rate in the system over the model time horizon for all budget sizes. The unlimited and double budgets did not significantly outperform the initial monthly budget in the long-run. Even the monthly half budget had a more than 5% increase in average fill rate over the course of the model. This is shown in Figure 5.19.
For all budget amounts, the average issue effectiveness varied in the early months and then improved in the last few months. Issue effectiveness is identical for the first several months of the model. Larger budget amounts slightly outperform smaller budgets in the later months of the model. This is shown in Figure 5.20.

**Average FSL Issue Effectiveness by Month**

Figure 5.20  Budget Size: Average Issue Effectiveness by Month
All considered budget amounts performed identically in the first half of the model time horizon. Similar to issue effectiveness performance, potential unavailable sub-systems is only slightly improved in model runs with increased budget. In the later months of the model, larger budgets had fewer backorders and these backorders typically occurred earlier. This is shown in Figure 5.21.

![Potential Unavailable Sub-Systems Over Time](image)

Figure 5.21 Budget Size: Potential Unavailable Sub-Systems Over Time

Larger budget amounts resulted in earlier improvements to average part fill rate, but did not significantly outperform the original monthly budget model. When comparing these model runs, issue effectiveness and sub-system availability were identical for the first half of the time horizon. Larger (or unlimited) budgets only slightly outperformed smaller budgets near the end of the time horizon. These results indicate that service level performance early in the model is not due to lack of repair budget. This is further shown by the fiscal year spend for each model. Monthly, monthly double budget, and unlimited budget have similar total repair spends. The monthly budget model actually spends more on repairs and performs worse than the monthly double budget. This indicates that maybe
the issue is not the total budget amount but rather the amount of budget available early in the model time horizon. The total repair spend for the fiscal year is shown in Figure 5.22 below.

**Figure 5.22  Budget Size: Fiscal Year Repair Spend**

### 5.4.3 Budget Period Length

The impact of budget period length was also considered. Service level performance in annual, semi-annual, quarterly, and monthly budget periods was analyzed. The same total budget amount from the initial monthly base model was used for all budget period models. Comparisons of budget period lengths include the average size of repair inductions and improvement in average part fill rate over time.

Budget length impacts the average number of parts of each component sent to repair. For the monthly budget model, early repairs inductions focus on repair breadth; then over the next several months the model transitions to depth, repairing fewer components but more parts of each. The annual budget has the total repair budget available at the start of the model. Therefore, the annual model does not need to choose
between breadth and depth in the early months of the model. An annual budget repairs a large number of parts in the first month to improve initial part imbalances in the system and then maintains smaller repair inductions for the remaining months. This is shown in Figure 5.23.

**Average Size of Component Repair Inductions by Month**

![Bar Chart: Average Size of Repairs by Month](image)

*Figure 5.23  Budget Period Length: Average Size of Repairs by Month*

Longer budget period lengths saw quicker improvement in average part fill rate over time. Though longer budget periods improved the average comparative part fill rate earlier, the ending percentage increase did not differ significantly between models. This is shown in Figure 5.24.
Total budget is less important than the amount of budget available early in the model. More money early in the time horizon allows for more repairs at the start of the model and quicker improvements in average fill rate. Longer budget period lengths repair many parts early in the model potentially correcting initial part imbalances in the system.

### 5.4.4 Repair Recommendations Output

The model produces two types of repair recommendation outputs: a weekly (or daily) repair list and a prioritized “batting order” list of repairs. The weekly repair list provides a list of total number of parts of each component to repair in the current week. The prioritized repair list takes the weekly repair list and orders recommended repairs based on their impact on system performance.
The weekly repair recommendations are captured in the model by the variable $z(I,J,T=1)$. The repair list is simply the total number of parts of component I sent to repair from all fields stocking locations J in the first week of the model time horizon. An example weekly repair list is shown in Figure 5.25.

<table>
<thead>
<tr>
<th>Parts To Repair This Week</th>
<th>Total Repairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 0200</td>
<td>2</td>
</tr>
<tr>
<td>Part 0274</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>5</td>
</tr>
<tr>
<td>Part 0871</td>
<td>14</td>
</tr>
<tr>
<td>Part 0145</td>
<td>1</td>
</tr>
<tr>
<td>Part 0167</td>
<td>4</td>
</tr>
<tr>
<td>Part 1440</td>
<td>3</td>
</tr>
<tr>
<td>Part 1541</td>
<td>4</td>
</tr>
<tr>
<td>Part 1464</td>
<td>2</td>
</tr>
<tr>
<td>Part 0211</td>
<td>3</td>
</tr>
<tr>
<td>Part 1233</td>
<td>3</td>
</tr>
<tr>
<td>Part 0172</td>
<td>1</td>
</tr>
<tr>
<td>Part 0141</td>
<td>7</td>
</tr>
<tr>
<td>Part 0342</td>
<td>1</td>
</tr>
<tr>
<td>Part 0952</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 5.25  Example Weekly Repair Recommendations

The model generates a prioritized list of repair recommendations. To do this, the parameter impact(I,RN,T) is generated with the model decision variable results using post-optimization calculations. These values are then sorted first by time T in increasing order, and then by impact value in non-increasing order. An example of this repair sequencing output is shown in Figure 5.26.
<table>
<thead>
<tr>
<th>Parts In Order to Repair</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 0200</td>
<td>2</td>
</tr>
<tr>
<td>Part 0274</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>1</td>
</tr>
<tr>
<td>Part 0871</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>1</td>
</tr>
<tr>
<td>Part 0145</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>1</td>
</tr>
<tr>
<td>Part 0167</td>
<td>1</td>
</tr>
<tr>
<td>Part 1440</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>1</td>
</tr>
<tr>
<td>Part 1541</td>
<td>1</td>
</tr>
<tr>
<td>Part 0871</td>
<td>1</td>
</tr>
<tr>
<td>Part 1456</td>
<td>1</td>
</tr>
<tr>
<td>Part 1464</td>
<td>1</td>
</tr>
<tr>
<td>Part 1541</td>
<td>1</td>
</tr>
<tr>
<td>Part 0211</td>
<td>1</td>
</tr>
<tr>
<td>Part 1233</td>
<td>1</td>
</tr>
<tr>
<td>Part 0172</td>
<td>1</td>
</tr>
<tr>
<td>Part 0141</td>
<td>1</td>
</tr>
<tr>
<td>Part 1541</td>
<td>2</td>
</tr>
<tr>
<td>Part 1440</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5.26  Example Repair Recommendation Priority List

5.5  Summary

Using the model formulation presented at the start of this chapter, an optimization model was run for a variety of settings. Repair was prioritized with an aim towards maximizing issue effectiveness and sub-system availability. Objective function parameter values were also determined for the case study data. The model showed increasing trends in both average part fill rates and issue effectiveness. Sub-system availability showed improvement over the model time horizon as well.
An end-of-horizon effect does occur with fewer repairs being made later in the model’s time horizon. This occurs because of long repair times which prevent the model from seeing a benefit to the objective function within the time horizon. This happens when a part does not have time to be repaired and return to serviceability to improve comparative part fill rate before the end of the model’s time horizon. The model’s output naturally minimizes the impact of the end-of-horizon on the model’s output since the repair recommendations of interest are at the start of the model time horizon.
CHAPTER 6. CONCLUSIONS

6.1 Contribution

A multi-echelon reparable parts supply chain can benefit from making strategic real-time decisions. Aligning repair and part allocation decisions with system goals can have a significant impact on supply chain performance. To do this, it is essential that the appropriate breadth and depth of parts are repaired. This is particularly important when customer service levels must be met. Real-time repair recommendations are the focus of this thesis.

The service parts supply chain has long been a topic of research. A large portion of the literature focuses on steady-state supply chains. The research includes models such as METRIC which optimize target stock levels by minimizing expected backorders. Through the years, various extensions have been made including batching, expediting, repair capacity, and budget. Overall, this research considering steady-state demand is not particularly applicable to real-time decision making.

Over the years, the literature has grown to include non-steady state and non-stationary models. Less research though has focused on creating applicable tools to make strategic real-time decisions such as repair inductions and part allocations. Much of this research is concerned with directly maximizing issue effectiveness or sub-system availability. Some of the literature included mathematical models, but due to the large state space of such a model, algorithms, heuristics, and simulations were often used instead.
In this thesis, a mixed integer program is used to produce strategic day-to-day repair recommendations based upon the current state of the system, forecasted part failures, and budget. Repair inductions are prioritized over multiple budget periods through the use of weighted comparative fill rates which indirectly capture both issue effectiveness and sub-system availability. Much of the existing research tries to capture these service level metrics directly in the objective. Other research in this area aims to minimize cost including the cost of backorders and holding costs. The model presented herein indirectly prevents backorders by stocking to previously optimized target stock levels and forecasted demand.

6.2 Future Work / Room for Expansion

This thesis explores several different aspects to consider when making real-time repair decisions. Future extensions in this line of research include: further examination of model time horizon and stochastic demand.

One future extension would be to compare the results if the model were run for varying time horizon lengths. Currently, the number of days in the model time horizon is as follows:

\[
\text{Days in Time Horizon} = \left\lfloor \frac{\text{Days Until End of Current Month} + \text{Total Days in Next 13 Months}}{} \right\rfloor \quad (4.1)
\]

This amount of time allows components with longer repair times to have an opportunity to impact the objective function. This time horizon accounts for 95% of the mean repair times in the case study data. A longer time horizon of approximately 26 months was also considered, but this would almost double the elements in set T while only accounting for about 3% more component repair times. This may not be the case for all reparable
systems, so there is potential for future research to determine optimal planning horizon based on system specifics.

Another potential extension includes considering multiple demand forecasts to generate repair recommendations. Currently, the model is run using a simple demand forecast. Since future part failure is not known prior to the start of the model, considering multiple part-breakage schedules in the model could potentially produce more robust repair recommendations. Stochastic demand was considered for this model, but using demand as a random variable will create a nonlinear model if the objective function is not reformulated. The model’s current objective uses conditional statements based upon demand values and conditionals on random variables are not allowed in MIP formulations.

6.3 Summary

In a multi-echelon service parts supply chain with limited repair resources, real-time repair decisions have significant impacts on supply chain performance. Inducting into repair the appropriate breadth and depth of components is essential to meeting short-term and long-term customer service level constraints.

Previous research in this area primarily focuses on steady-state supply chains assuming infinite repair capabilities and the repair induction of all non-serviceable parts. A Mixed Integer Program was developed which considers current state of the entire repair system as well as forecasted part-breakages to produce strategic real-time repair recommendations.

A case study example was used to test real-world application of the model. Multiple model runs were completed to determine objective function parameter
specifications that best aligns with overall system goals of issue effectiveness and sub-system availability. Model output includes daily or weekly repair recommendations as well as a prioritized list of repair inductions. Projected supply chain performance for issue effectiveness and sub-system availability is shown for the next 13 month period.
REFERENCES


